ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

---oo0oo---

NGUYỄN HUY TÌNH

PHÁT TRIỂN VÀ ĐÁNH GIÁ CÁC PHƯƠNG PHÁP ƯỚC LƯỢNG MÔ HÌNH THAY THẾ AXIT AMIN CHO CÁC TẬP DỮ LIỆU CÓ KÍCH THƯỚC LỚN

LUẬN ÁN TIẾN SĨ KHOA HỌC MÁY TÍNH

HÀ NỘI - 2025

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

---oo0oo---

NGUYỄN HUY TÌNH

PHÁT TRIỂN VÀ ĐÁNH GIÁ CÁC PHƯƠNG PHÁP ƯỚC LƯỢNG MÔ HÌNH THAY THẾ AXIT AMIN CHO CÁC TẬP DỮ LIỆU CÓ KÍCH THƯỚC LỚN

Ngành : Khoa học Máy tính

Mã số : 9480101

LUẬN ÁN TIẾN SĨ KHOA HỌC MÁY TÍNH

NGHIÊN CỨU SINH **CÁN BỘ HƯỚNG DẪN**

XÁC NHẬN CỦA ĐƠN VỊ ĐÀO TẠO

Hà Nội - 2025

MỤC LỤC

Lời cam đoan

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các kết quả được viết chung với các tác giả khác đều được sự đồng ý của các đồng tác giả trước khi đưa vào luận án. Các kết quả nêu trong luận án là trung thực và chưa từng được công bố trong các công trình khác.

Tác giả

Nguyễn Huy Tình

Lời cảm ơn

Luận án được thực hiện tại Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội, dưới sự hướng dẫn của GS.TS. Lê Sỹ Vinh và TS. Đặng Cao Cường.

Tôi xin bày tỏ lòng biết ơn sâu sắc tới GS.TS. Lê Sỹ Vinh và TS. Đặng Cao Cường, những người đã có những định hướng đúng đắn giúp tôi thành công trong việc nghiên cứu của mình. Các thầy đã động viên và chỉ bảo giúp tôi vượt qua những khó khăn để tôi hoàn thành được luận án này.

Tôi cũng xin gửi lời cảm ơn tới các Thầy, Cô thuộc Khoa Công nghệ Thông tin, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội đã tạo mọi điều kiện thuận lợi giúp tôi trong quá trình làm nghiên cứu sinh.

Cuối cùng, tôi xin gửi lời cảm ơn sâu sắc tới gia đình và bạn bè, những người đã cho tôi điểm tựa vững chắc để tôi có được thành công như ngày hôm nay.

Danh mục các ký hiệu và chữ viết tắt

l	*Chiều dài của một sắp hàng*
m	*Số lượng trình tự có trong một sắp hàng*
N	*Số lượng sắp hàng trong một tập các sắp hàng*
$\mathbf{S}$	*Tập hợp 20 axit amin*
q_{ij}	*Tốc độ biến đổi tức thời giữa axit amin i và axit amin j*
π_i	*Tần số của axit amin i*
r_{ij}	*Hệ số hoán đổi giữa axit amin i và axit amin j*
α	*Tham số hình dạng của phân phối gamma*
$\mathbf{D}$	*Tập các sắp hàng*
D	*Một sắp hàng đa trình tự*
D^a	*Sắp hàng thứ a trong một tập các sắp hàng*
D_i	*Vị trí thứ i trong sắp hàng D*
Q	*Ma trận tốc độ biến đổi tức thời*
Π	*Véc tơ tần số của 20 axit amin*
R	*Ma trận hệ số hoán đổi*
T	*Cây phân loài tương ứng với sắp hàng D*
Q_k	*Ma trận thứ k của một mô hình đa ma trận*
w_k	*Trọng số của ma trận Q_k*
ρ_k	*Tốc độ của ma trận Q_k*

Danh mục ký hiệu và từ viết tắt

AIC	Akaike information criterion	Độ đo lý thuyết thông tin Akaike
BIC	Bayesian information criterion	Độ đo lý thuyết thông tin Bayesian
DNA	Deoxyribonucleic acid	Axit deoxyribonucleic
MF	ModelFinder	Phương pháp ModelFinder
ML	Maximum likelihood	Cực đại hợp lý
MP	Maximum parsimony	Cực tiểu số lượng biến đổi
MSE	Mean squared error	Trung bình bình phương sai số
nRF	Normalised Robinson-Fould distance	Khoảng cách Robinson-Fould được chuẩn hóa
PCA	Principal component analysis	Phân tích thành phần chính
RF	Robinson-Fould distance	Khoảng cách Robinson-Fould
RHAS	Rate heterogeneity across sites	Tốc độ không đồng nhất qua các vị trí
RNA	Ribonucleic acid	Axit ribonucleic
SE	Squared error	Bình phương sai số

Danh mục các bảng

Danh mục các hình vẽ, đồ thị

Lời mở đầu

1. Đặt vấn đề

Tin sinh học là một lĩnh vực nghiên cứu đa ngành với sự kết hợp của công nghệ thông tin, sinh học phân tử và toán thống kê. Cùng với sự phát triển bùng nổ và ứng dụng sâu rộng của công nghệ thông tin, tin sinh học ngày càng được đầu tư và đem lại những lợi ích lớn cả về khoa học lẫn hiệu quả kinh tế. Các kết quả trong nghiên cứu tin sinh học được ứng dụng rộng rãi trong y tế, nông nghiệp, di truyền, và các lĩnh vực khác.

Dữ liệu chính và phổ biến trong tin sinh học là các trình tự DNA và trình tự axit amin [1-4]. Với sự phát triển của các máy giải trình tự thế hệ mới, số lượng trình tự này đang tăng liên tục cả về số lượng lẫn độ chính xác. Qua đó, một lượng lớn dữ liệu được cung cấp một cách thường xuyên tạo cơ sở giúp cho các nhà khoa học nghiên cứu đưa ra các kết quả mới để giải quyết những bài toán khó hiện tại đang gặp phải.

Các bài toán cơ bản trong tin sinh học tiến hóa liên quan đến trình tự axit amin bao gồm: sắp hàng đa trình tự, tìm kiếm trình tự tương đồng hay xây dựng cây phân loài. Trong đó bài toán xây dựng cây phân loài được coi là bài toán trung tâm trong tin sinh học tiến hóa. Tuy nhiên, dữ liệu sinh học thì rất đa dạng, sự biến đổi, tiến hóa của dữ liệu có thể đã xảy ra nhiều lần nhưng chúng ta chỉ quan sát được tại một thời điểm nhất định. Để giải quyết bài toán này, chúng ta cần xây dựng một mô hình toán học (dạng ma trận) để giải thích một cách gần đúng nhất (có xác suất cao nhất) quá trình thay thế giữa các axit amin trong trình tự, các mô hình này được gọi là mô hình thay thế axit amin, sau đây gọi tắt là mô hình. Tối ưu các mô hình thay thế cũng như ước lượng các mô hình thay thế mới sẽ giúp quá trình nghiên cứu sát với thực tế tiến hóa và mang lại các kết quả tốt hơn.

2. Mục đích nghiên cứu

Các mô hình thay thế axit amin tùy vào thuộc tính về mặt thời gian mà chúng ta cần phải ước lượng 208 tham số tự do (thời gian thuận nghịch: tốc độ biến đổi giống nhau theo cả hai hướng tiến và lùi) hay 379 tham số tự do (thời gian không thuận nghịch: tốc độ biến đổi khác nhau theo hai hướng tiến và lùi). Các mô hình có thể tồn tại ở dạng đơn ma trận hay kết hợp của nhiều ma trận khác nhau. Có nhiều mô hình đơn ma trận được ước lượng từ các tập dữ liệu trình tự axit amin đã được công bố như PAM [5], JTT [6], WAG [7] hay LG [8], đây được coi là những mô hình chung (general model) và thường được dùng như mô hình khởi tạo cho quá trình ước lượng các mô hình khác. Ngoài ra còn có các mô hình được ước lượng từ các tập dữ liệu cụ thể của từng loài riêng biệt, ví dụ mô hình Q.plant ước lượng từ bộ dữ liệu các loài thực vật hay Q.bird cho các loài chim. Ước lượng mô hình thay thế axit amin là bài toán khó, thường phải trải qua nhiều bước phức tạp và tốn kém cả về mặt thời gian lẫn chi phí tính toán. Các phương pháp truyền thống để giải quyết bài toán này bao gồm:

- Phương pháp đếm: đây là phương pháp cổ điển được dùng để ước lượng các mô hình như PAM, Dayhoff. Phương pháp này đơn giản nhưng chỉ phù hợp với các sắp hàng đa trình tự có độ tương đồng cao, vì vậy, với lượng dữ liệu khổng lồ và khác biệt lớn ngày nay thì phương pháp này không còn được áp dụng trong ước lượng mô hình thay thế axit amin nữa.
- Phương pháp cực đại hợp lý (maximum likelihood): đây là phương pháp phổ biến và mang lại kết quả tốt hơn. Với phương pháp này, trước hết chúng ta cần xây dựng cây phân loài dựa trên các mô hình chung như LG, sau đó dựa trên cây phân loài đã có, mô hình mới sẽ được ước lượng nhằm cực đại hóa giá trị hợp lý của cây.

Việc ước lượng dựa trên phương pháp cực đại hợp lý có thể dựa trên các đặc điểm sinh học khác nhau và tạo ra các mô hình đơn hay đa ma trận khác nhau. Rất nhiều công

trình nghiên cứu đã đề xuất các phương pháp tối ưu quá trình ước lượng dựa trên phương pháp cực đại hợp lý như XRate [9], RAxML [10]. Gần đây, nhóm tác giả Minh Bùi và cộng sự [11] đề xuất QMaker, Cường Đặng và cộng sự [12] đã đề xuất nQMaker, đây là hai phương pháp mới được dùng để tự động ước lượng mô hình thay thế của trình tự axit amin. Với các phương pháp này, nhiều mô hình đã được ước lượng từ các bộ dữ liệu sắp hàng khác nhau. Các mô hình như Q.plant, Q.bird hay Q.yeast [11] được ước lượng từ 1000 sắp hàng, tốn nhiều thời gian và năng lực tính toán trên các hệ thống hiệu năng cao mà không phải ai cũng có thể tiếp cận được. Vậy thì, với năng lực tính toán thông thường, chúng ta có thể ước lượng được một mô hình chấp nhận được về độ hiệu quả mà trong khoảng thời gian hợp lý hay không? Từ đó, một số câu hỏi nghiên cứu được đặt ra như:

- Hiệu quả của mô hình ước lượng thay đổi ra sao tương ứng theo dữ liệu hay số lượng dữ liệu cần thiết để ước lượng mô hình đạt yêu cầu về hiệu năng?
- Ngoài ra, cách tiếp cận dựa trên tính chất thời gian thuận nghịch cũng như việc dùng đơn ma trận để mô hình hóa quá trình tiến hóa có phù hợp với thực tế?

Tiếp theo, việc lựa chọn (hay ước lượng) nhanh mô hình thay thế axit amin cho một sắp hàng bất kì cho trước là vấn đề quan trọng. Do số lượng mô hình ngày càng nhiều nên khi nghiên cứu một dữ liệu sắp hàng bất kỳ, thay vì ước lượng mô hình mới mất nhiều thời gian, chúng ta cần tìm ra mô hình phù hợp nhất với dữ liệu đó. Thông thường, để chọn được mô hình hợp lý nhất chúng ta có thể sử dụng phương pháp dựa trên tiêu chuẩn cực đại hợp lý, tiêu biểu là phương pháp ModelFinder [13] hay ModelTest-NG [14]. Ngoài ra, để giải quyết bài toán này, nhiều nhà khoa học cũng sử dụng các mạng học sâu để tối ưu hơn về mặt thời gian. Đây là một hướng đi mới trong tin sinh học tiến hóa. Gần đây, với phương pháp này các nhà khoa học đã đề xuất các mạng học sâu để dự đoán mô hình cho dữ liệu DNA [15], [16]. Do đó, một câu hỏi nghiên cứu nữa được đặt ra đó là: có thể phát triển mạng học sâu để lựa chọn (hay ước lượng) nhanh mô hình thay thế cho dữ liệu trình tự axit amin hay không? Về cơ bản, một

mạng học sâu sẽ học các đặc trưng của sắp hàng đa trình tự, sắp hàng này có thể tuân theo một mô hình thay thế axit amin nào đó. Các đặc trưng này bao gồm các hệ số biến đổi của các cặp axit amin trong các trình tự, tần số của các loại axit amin trong sắp hàng, từ đó, sau khi được đào tạo, mạng học sâu có thể đưa ra dự đoán một sắp hàng đa trình tự bất kì thuộc về một mô hình thay thế axit amin nào đó nằm trong tập các mô hình nó đã được huấn luyện.

3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của luận án là các tập dữ liệu sắp hàng axit amin có kích thước lớn, với dữ liệu thật có thể lên đến hàng chục nghìn sắp hàng, với dữ liệu mô phỏng lên đến hàng triệu sắp hàng.

Luận án tập trung vào việc mô hình quá hóa trình tiến hóa của sắp hàng axit amin dựa trên các đặc tính sinh học như sự không đồng nhất về tốc độ tiến hóa hay tính không thuận nghịch về mặt thời gian trong quá trình tiến hóa của các vị trí trên sắp hàng. Luận án cũng đi sâu vào cách ước lượng mô hình thay thế sử dụng nhiều ma trận tốc độ biến đổi khác nhau. Ngoài ra, luận án nghiên cứu các phương pháp dựa trên học sâu nhằm lựa chọn nhanh các mô hình phù hợp cho sắp hàng với thời gian tối ưu.

4. Phương pháp nghiên cứu

Phương pháp được luận án sử dụng là sự kết hợp giữa nghiên cứu lý thuyết và làm các thực nghiệm, gồm các bước cơ bản như sau:

- Nghiên cứu cơ sở lý thuyết nhằm đạt được những nền tảng căn bản của tin sinh học tiến hóa.
- Khảo sát, nghiên cứu các công trình khoa học có liên quan đến nội dung nghiên cứu đồng thời thu thập các bộ dữ liệu tiêu chuẩn.
- Đề xuất các phương pháp hoặc mô hình mới theo nội dung của luận án.

- Tiến hành các thực nghiệm đánh giá phương pháp và mô hình mới trên các bộ dữ liệu đã thu thập, phân tích kết quả nhằm cải tiến phương pháp và mô hình đã được đề xuất.

5. Những kết quả và đóng góp chính của luận án

Luận án đã đạt được một số kết quả và đóng góp chính như sau:

1. Đề xuất một quy trình đánh giá các phương pháp ước lượng mô hình dựa trên tiêu chuẩn cực đại hợp lý. Việc này giúp các nhà tin sinh học xác định được lượng dữ liệu tối ưu cho việc ước lượng mô hình thay thế axit amin để vừa đảm bảo về mặt hiệu suất mô hình cũng như chi phí thời gian.
   Các kết quả được công bố trong công trình [CT1] và [CT2].
2. Đề xuất các mô hình thay thế axit amin mới gồm nhiều ma trận trong đó có sử dụng các đặc tính về mặt sinh học là tốc độ biến đổi tại các vị trí trên trình tự và thuộc tính thời gian không thuận nghịch. Việc này sẽ giúp nâng cao khả năng và tính chính xác của mô hình thay thế axít amin khi phân tích dữ liệu cũng như xây dựng cây phân loài.
   Các kết quả được công bố trong công trình [CT3], [CT4] và [CT6].
3. Xây dựng một mạng học sâu để lựa chọn nhanh mô hình thay thế axit amin của một sắp hàng đa trình tự bất kì. Việc này giúp người dùng tiết kiệm rất nhiều thời gian do quá trình dự đoán này nhanh hơn nhiều lần so với việc sử dụng phương pháp cực đại hợp lý truyền thống.
   Các kết quả được công bố trong công trình [CT5].

6. Bố cục của luận án

Ngoài phần kết luận, luận án được tổ chức như sau:

Chương 1 giới thiệu tổng quan về trình tự DNA, trình tự axit amin và các phép biến đổi trên trình tự axit amin. Sau đó luận án giới thiệu về bài toán mô hình hoá quá

trình biến đổi axít amin và bài toán ước lượng mô hình thay thế axít amin. Tiếp theo là phần trình bày về hai cách tiếp cận chính để ước lượng mô hình thay thế axít amin là phương pháp đếm và phương pháp cực đại hợp lý. Phần cuối của chương này giới thiệu về phương pháp xây dựng cây phân loài bằng phương pháp cực đại hợp lý và các phương pháp so sánh, đánh giá hai mô hình thay thế axít amin.

Chương 2 đề xuất một quy trình đánh giá các phương pháp ước lượng mô hình thay thế axit amin. Với sự ra đời của QMaker và nQMaker là hai phương pháp mới để ước lượng nhanh mô hình thay thế, rất nhiều mô hình có thể được tạo ra từ dữ liệu thật cũng như dữ liệu mô phỏng. Để đánh giá kĩ hơn hai phương pháp này, luận án đưa ra các tiêu chí đánh giá mô hình cũng như đề xuất các tiêu chuẩn về mặt dữ liệu nhằm giúp các nhà nghiên cứu vừa tiết kiệm thời gian cũng như đạt được hiệu quả mong muốn.

Chương 3 của luận án giới thiệu phương pháp ước lượng mô hình thay thế axít amin sử dụng nhiều ma trận. Phương pháp này được thể hiện dưới dạng một phần mềm và có thể tự động ước lượng một số lượng ma trận tùy ý. Các thí nghiệm trên hai bộ dữ liệu HSSP và TreeBase đã cho thấy độ tin cậy và hiệu quả phương pháp này. Đồng thời luận án cũng ước lượng và đề xuất các mô hình thay thế axit amin sử dụng đa ma trận mới được ước lượng dựa trên các bộ dữ liệu HSSP và bộ dữ liệu các loài thực vật. Các mô hình đa ma trận mới đều bao gồm bốn ma trận con tương ứng với bốn phân lớp tốc độ khác nhau. Các thực nghiệm cho thấy các mô hình mới có hiệu quả cao trong việc nghiên cứu dữ liệu sắp hàng đa trình tự axit amin.

Chương 4 trình bày một mạng học sâu và phương pháp để trích xuất đặc trưng dữ liệu nhằm giải quyết bài toán lựa chọn mô hình thay thế axit amin phù hợp nhất cho một sắp hàng đa trình tự bất kì cho trước. Với mạng học sâu này các nhà khoa học có thể rất nhanh chóng tìm ra mô hình phù hợp cho dữ liệu (nhanh gấp hơn 20 lần so với phương pháp cực đại hợp lý truyền thống) từ đó xây dựng cây phân loài hoặc thực hiện các thí nghiệm khác dựa trên mô hình đã được dự đoán.

Chương 1. Cơ sở lý thuyết

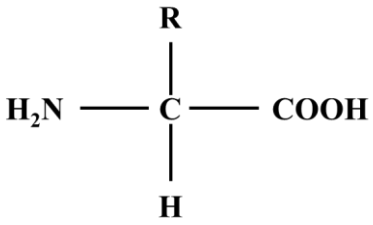
Trong phần này, luận án sẽ trình bày các khái niệm cơ bản về DNA, axit amin, sắp hàng đa trình tự (sắp hàng), cây phân loài, mô hình thay thế axit amin, mô hình tốc độ biến đổi tại các vị trí và một số phương pháp đánh giá mô hình thông dụng. Ngoài ra, chương này cũng đề cập đến các phương pháp lựa chọn mô hình và các bộ dữ liệu được luận án sử dụng.

1.1. Một số khái niệm cơ bản

1.1.1. Trình tự DNA và axit amin

Deoxyribo Nucleic Axit (viết tắt DNA) là phân tử sinh học mang thông tin di truyền được cấu tạo từ các phân tử nhỏ hơn gọi là nucleotit [1, 4]. Phân tử DNA (chuỗi DNA) có cấu trúc dạng xoắn kép được tạo thành từ bốn loại nucleotit là: Adenine (A), Thymine (T), Cytosine (C), và Guanine (G). Các nucleotit đóng vai trò trung tâm trong quá trình trao đổi chất, có tác dụng đặc biệt quan trọng đối với sự tăng trưởng và phát triển của cơ thể. Nucleotit giữa hai trình tự liên kết với nhau theo nguyên tắc: A với T và G với C. Gene là một đoạn cụ thể trên chuỗi DNA mang đầy đủ các thông tin cần thiết để tổng hợp nên protein. Một phân tử DNA có thể chứa hàng nghìn gene có kích thước và đặc điểm khác nhau tạo nên sự đa dạng sinh học trong tự nhiên.

Protein hay còn gọi là chất đạm, là những phân tử sinh học chứa một hay nhiều mạch axit amin liên kết với nhau bởi liên kết peptit. Axit amin là nguyên liệu để tổng hợp nên các protein cần thiết cho cơ thể sống. Mỗi axit amin được tạo thành từ ba nhân tố chính bao gồm: nhóm amin ($-NH_2$), nhóm cacboxyl (-COOH) cùng với một mạch bên (nhóm biến đổi R), cụ thể xem trong Hình 1.1. Chúng ta thường phân biệt cấu trúc của phân tử protein thành bốn bậc dựa trên sự sắp xếp và tương tác không gian giữa các gốc axit amin [2].



Hình 1.1. Công thức cấu tạo của axit amin.

Có 20 loại axit amin khác nhau được mã hóa bởi bộ gene. Chúng ta có thể viết tắt tên của axit amin theo hai cách là dùng một ký tự hay ba ký tự, xem Bảng 1.1.

Bảng 1.1. Danh sách 20 axit amin

STT	Tên axit amin	Tên viết tắt (3 ký tự)	Tên viết tắt (1 ký tự)
1	Alanine	Ala	A
2	Arginine	Arg	R
3	Asparagine	Asn	N
4	Aspartic	Asp	D
5	Cysteine	Cys	C
6	Glutamine	Gln	Q
7	Glutamic	Glu	E
8	Glycine	Gly	G
9	Histidine	His	H
10	Isoleucine	Ile	I
11	Leucine	Leu	L
12	Lysine	Lys	K
13	Methionine	Met	M
14	Phenylalanine	Phe	F
15	Proline	Pro	P
16	Serine	Ser	S
17	Threonine	Thr	T
18	Tryptophan	Trp	W
19	Tyrosine	Tyr	Y
20	Valine	Val	V

Tổng hợp protein là quá trình tế bào tổng hợp những phân tử protein cần thiết cho sự tồn tại và sinh trưởng. Quá trình này bao gồm việc phiên mã, dịch mã và tổng hợp. RiboNucleic Axit (viết tắt RNA) là một trong những phân tử sinh học quan trọng nhất của sự sống, được sử dụng để mã hóa, giải mã, điều hòa và biểu hiện gene. Trong quá trình phiên mã, các thông tin được lưu trong DNA sẽ được sao chép sang RNA thông tin (mRNA), chuỗi RNA gồm bốn loại nucleotit là: Adenine (A), Uracil (U), Cytosine (C) và Guanine (G). Tiếp theo, mỗi một bộ ba nucleotit này gọi là một codon sẽ mã hóa một loại axit amin (xem Bảng 1.2), có tất cả 64 codon để mã hóa cho 20 loại axit amin, do vậy có nhiều codon cùng mã hóa một axit amin. Dựa theo quy tắc bộ ba mã hóa này, quá trình dịch mã sẽ chuyển mã di truyền trong mRNA thành các trình tự axit amin tương ứng. Việc dịch mã thường bắt đầu với codon AUG và kết thúc khi gặp các codon như UAA, UAG hay UGA. Các trình tự axit amin khác nhau sẽ tạo thành các protein khác nhau và phục vụ các chức năng khác nhau.

Bảng 1.2. Danh sách 64 codon mã hóa axit amin

	U		**C**		**A**		**G**		
	Codon	Axit amin	Codon	Axit amin	Codon	Axit amin	Codon	Axit amin	
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	**U**
	UUC	Phe	UCC	Ser	UAC	Tyr	UGC	Cys	**C**
	UUA	Leu	UCA	Ser	UAA	Dừng	UGA	Dừng	**A**
	UUG	Leu	UCG	Ser	UAG	Dừng	UGG	Trp	**G**
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg	**U**
	CUC	Leu	CCC	Pro	CAC	His	CGC	Arg	**C**
	CUA	Leu	CCA	Pro	CAA	Gln	CGA	Arg	**A**
	CUG	Leu	CCG	Pro	CAG	Gln	CGG	Arg	**G**
A	AUT	Ile	ACU	Thr	AAU	Asn	AGU	Ser	**U**
	AUC	Ile	ACC	Thr	AAC	Asn	AGC	Ser	**C**
	AUA	Ile	ACA	Thr	AAA	Lys	AGA	Arg	**A**
	AUG	Met	ACG	Thr	AAG	Lys	AGG	Arg	**G**
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	**U**
	GUC	Val	GCC	Ala	GAC	Asp	GGC	Gly	**C**
	GUA	Val	GCA	Ala	GAA	Glu	GGA	Gly	**A**
	GUG	Val	GCG	Ala	GAG	Glu	GGG	Gly	**G**

1.1.2. Sự biến đổi trên trình tự axit amin

Trình tự axit amin hay protein ảnh hưởng trực tiếp đến cấu trúc và chức năng sinh học, do vậy, nó phản ánh tín hiệu tiến hóa, chọn lọc tự nhiên. Trong quá trình sinh trưởng và phát triển, việc sao chép lượng thông tin di truyền khổng lồ luôn có thể xảy ra lỗi, dẫn đến những biến đổi trong trình tự axit amin. Những biến đổi này một số là có lợi, giúp gia tăng khả năng thích nghi của sinh vật với môi trường tự nhiên. Tuy nhiên, cũng có những biến đổi không phù hợp và gây ra các bệnh tật cho cơ thể sinh vật. Sự biến đổi trên trình tự axit amin có nhiều nguyên nhân, có thể do bản thân vùng mã hóa của trình tự DNA đã có sự biến đổi trước đó hoặc biến đổi xảy ra trong quá trình dịch mã và phiên mã. Có ba loại biến đổi chính trên trình tự axit amin là:

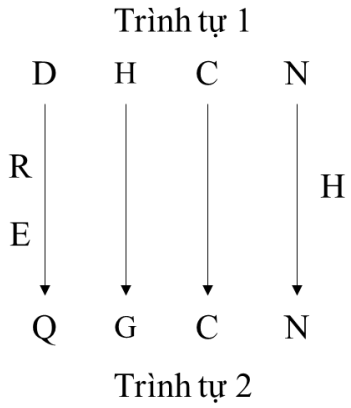
- Xoá: một axit amin bị xoá khỏi trình tự axit amin, độ dài của trình tự giảm đi một.
- Chèn: một axit amin được chèn vào trình tự axit amin, độ dài trình tự tăng lên một.
- Thay thế: một axit amin nào đó bị thay thế bởi một axít amin khác, độ dài của trình tự không thay đổi.

Bảng 1.3 là một ví dụ thể hiện sự biến đổi của hai trình tự axit amin. Trong đó, các cột một và hai chứa các loại axit amin khác nhau thể hiện cho sự thay thế; ký tự (-) ở cột ba và bốn thể hiện biến đổi xóa hoặc biến đổi chèn. Các biến đổi trên trình tự axit amin có thể làm biến đổi cấu trúc của protein, từ đó khiến chức năng protein thay đổi và cuối cùng ảnh hưởng đến sự phát triển của sinh vật.

Bảng 1.3. Minh họa sự biến đổi trên hai trình tự axit amin

Vị trí	1	2	3	4	5	6	7	8	9	10	11	12
Trình tự 1	C	N	-	-	D	H	C	N	-	N	H	T
Trình tự 2	D	Q	L	R	D	H	C	N	N	N	H	T

Nếu chỉ quan sát thông thường, chúng ta có thể thấy đây là các biến đổi đơn và duy nhất. Tuy nhiên, thực tế lại phức tạp hơn nhiều, xem minh họa trên Hình 1.2, do sự đa biến đổi (nhiều biến đổi trung gian tại một vị trí trước khi được quan sát, ví dụ D chuyển thành R và E trước khi trở thành Q), biến đổi song song (cùng biến đổi giống nhau trên hai trình tự trước khi được quan sát) hoặc biến đổi ngược (ví dụ N-H-N) [3]. Do đó, việc phân tích quá trình biến đổi này sẽ thiếu tính khách quan và chính xác nếu chỉ dựa vào dữ liệu được quan sát tại một thời điểm. Vì thế, chúng ta cần một mô hình xác suất để mô hình hóa quá trình biến đổi (thay thế) này gọi là mô hình thay thế axit amin. Các mô hình này sẽ tính toán khả năng (xác suất) mà một axit amin bất kì có thể được thay thế bởi một axit amin khác, Bảng 1.4 là một ví dụ về mô hình thay thế axit amin LG được đề xuất bởi tác giả Lê và Gascuel [8], đây là một mô hình thay thế axit amin dùng chung được sử dụng rộng rãi trong các nghiên cứu tiến hóa.



Hình 1.2. Minh họa sự đa biến đổi trên hai trình tự axit amin.

Bảng 1.4. Mô hình thay thế axit amin LG

	A	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
A																				
R	0.425																			
N	0.277	0.752																		
D	0.395	0.124	5.076																	
C	2.489	0.535	0.529	0.063																
Q	0.970	2.808	1.696	0.523	0.085															
E	1.039	0.364	0.542	5.244	0.003	4.129														
G	2.066	0.390	1.438	0.845	0.569	0.268	0.349													
H	0.359	2.427	4.509	0.927	0.641	4.814	0.424	0.311												
I	0.150	0.127	0.192	0.011	0.321	0.073	0.044	0.009	0.109											
L	0.395	0.302	0.068	0.015	0.594	0.582	0.070	0.044	0.366	4.145										
K	0.537	6.326	2.145	0.283	0.013	3.234	1.807	0.297	0.697	0.159	0.138									
M	1.124	0.484	0.371	0.026	0.894	1.673	0.174	0.140	0.442	4.274	6.312	0.657								
F	0.254	0.053	0.090	0.017	1.105	0.036	0.019	0.090	0.682	1.113	2.593	0.024	1.799							
P	1.178	0.333	0.162	0.394	0.075	0.624	0.419	0.197	0.509	0.078	0.249	0.390	0.100	0.094						
S	4.727	0.858	4.008	1.240	2.784	1.224	0.612	1.740	0.990	0.064	0.182	0.749	0.347	0.362	1.338					
T	2.140	0.579	2.001	0.426	1.143	1.080	0.605	0.130	0.584	1.034	0.303	1.137	2.020	0.165	0.571	6.472				
W	0.181	0.594	0.045	0.030	0.670	0.236	0.078	0.268	0.597	0.112	0.620	0.050	0.696	2.457	0.095	0.249	0.141			
Y	0.219	0.314	0.612	0.135	1.166	0.257	0.120	0.055	5.307	0.233	0.300	0.132	0.481	7.804	0.090	0.401	0.246	3.152		
V	2.548	0.171	0.084	0.038	1.959	0.210	0.245	0.077	0.119	10.649	1.703	0.185	1.899	0.655	0.297	0.098	2.188	0.190	0.249	
Π	0.079	0.056	0.042	0.053	0.013	0.041	0.072	0.057	0.022	0.062	0.099	0.065	0.023	0.042	0.044	0.061	0.053	0.012	0.034	0.069

1.1.3. Sắp hàng các trình tự axit amin tương đồng

Hai trình tự axit amin gọi là tương đồng nếu chúng cùng tiến hóa từ một trình tự axit amin tổ tiên [3]. Sự khác biệt của các trình tự tương đồng chủ yếu do các biến đổi trong quá trình tiến hóa và do vậy làm chúng khác nhau cả về nội dung lẫn độ dài. Để nghiên cứu các phép biến đổi này tốt hơn, việc sắp hàng các trình tự này là cần thiết. Lĩnh vực tin sinh học chủ yếu làm việc với các sắp hàng đa trình tự, mỗi sắp hàng đa trình tự sẽ được thể hiện như một ma trận các axit amin, trong đó mỗi hàng là một trình tự, mỗi cột là một vị trí chứa các axit amin tương đồng. Nhờ có các sắp hàng mà chúng ta có thể dễ dàng hơn trong việc xây dựng cây phân loài (cây tiến hóa) để xác định nguồn gốc tiến hóa của các loài trong sắp hàng.

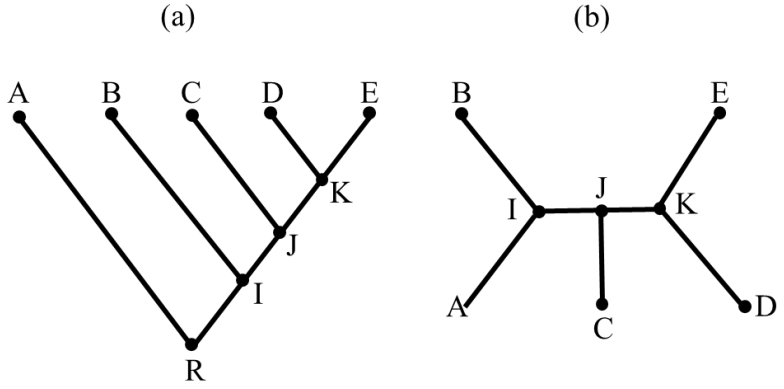
Có một vài phương pháp để sắp hàng nhiều trình tự axit amin, nổi bật là CLUSTALW [17], MAFFT [18], PROBCONS [19] và MUSCLE [20]. Mặc dù nhiều phương pháp đã được đề xuất nhưng đây vẫn là một bài toán quan trọng trong lĩnh vực tin sinh học nhằm tối đa hóa sự chính xác và hiệu quả của dữ liệu sau sắp hàng. Một ví dụ về việc sắp hàng đa trình tự tương đồng trong Bảng 1.5, dấu gạch ngang thể hiện một phép chèn hoặc xóa trong quá trình biến đổi axit amin.

Bảng 1.5. Ví dụ về sắp hàng đa trình tự tương đồng

Đầu vào	Trình tự 1	N	G	N	N	R	R	H	
	Trình tự 2	N	G	N	R	R	H		
	Trình tự 3	R	N	G	N	H			
	Trình tự 4	R	N	G	N	G	G	H	
	Trình tự 5	R	N	G	T	G	G	H	
Đầu ra	Trình tự 1	-	N	G	N	N	R	R	H
	Trình tự 2	-	N	G	N	-	R	R	H
	Trình tự 3	R	N	G	N	-	-	-	H
	Trình tự 4	R	N	G	N	-	G	G	H
	Trình tự 5	R	-	G	N	T	G	G	H

1.1.4 Cây phân loài

Trong nghiên cứu tin sinh học, cây phân loài (cây tiến hóa, cây phả hệ) là một loại biểu đồ dạng cây nhị phân, dùng để mô hình hóa mối quan hệ phát sinh loài hay lịch sử tiến hóa của một nhóm các sinh vật (xem minh họa trên Hình 1.3). Các nút lá (A, B, C, D, E) thể hiện các loài sinh vật, các nút cha (R, I, J, K) đại diện cho tổ tiên gần nhất của hai nút con. Nút R được xem là tổ tiên chung nhất của tất cả các loài. Cạnh hay nhánh của cây thể hiện khoảng cách tiến hóa giữa các nút của cạnh, khoảng cách tiến hóa ở đây có thể hiểu là số phép biến đổi xảy ra giữa các loài sinh vật. Như vậy, khoảng cách tiến hóa cũng thể hiện khoảng thời gian trôi đi từ khi phân tách hai loài khỏi nút cha.



Hình 1.3. Ví dụ về cây phân loài, hình (a) là cây có gốc, hình (b) là cây không gốc.

Cây phân loài tồn tại ở hai dạng là cây có gốc như trong Hình 1.3a và cây không gốc như trong Hình 1.3b. Một số phương pháp để xây dựng cây phân loài có thể kể đến như sau:

- Phương pháp ma trận khoảng cách (Distance Matrix): sử dụng ma trận khoảng cách (được tính toán dựa trên khoảng cách tiến hóa) và thuật toán nhóm không trọng số dùng trung bình số học (UPGMA - Unweighted PairGroup with Method using arthmetic Averages) [21] để tạo ra cây phân loài.

- Phương pháp hàng xóm kết hợp Neighbor-Joining [22]: hai trình tự được coi là hàng xóm nếu giữa chúng chỉ tồn tại một nút. Phương pháp này tìm các trình tự hàng xóm và nối chúng với nhau để tạo thành nút mới, nó chạy rất nhanh để ra được kết quả nhưng có thể bỏ qua cây tốt nhất.
- Phương pháp cực tiểu số lượng biến đổi (Maximum Parsimony) [23, 24]: cây tiến hóa được chọn là cây có số lượng biến đổi ít nhất. Nói cách khác, cây được tạo ra sẽ là cây có sự thay đổi giữa loài là nhỏ nhất.
- Phương pháp cực đại hợp lý (Maximum Likelihood) [25]: phương pháp này tính toán xác suất khả năng một cây được suy diễn từ dữ liệu đa sắp hàng. Cây phân loài sẽ là cây cho xác suất tạo thành cao nhất trong tất cả các cây được xem xét.
- Ngoài ra còn một số phương pháp khác như: phương pháp Bayes [26], phương pháp đồng hồ phân tử (Molecular clock) [27], cũng được dùng để xây dựng cây phân loài.

Trong luận án này, phương pháp cực đại hợp lý được sử dụng để giải quyết các bài toán về ước lượng mô hình thay thế axit amin và xây dựng cây phân loài.

1.2 Bài toán mô hình hóa quá trình thay thế trình tự

1.2.1 Mô hình Markov

Quá trình thay thế tại một vị trí bất kì trên trình tự DNA hay axit amin được xem là ngẫu nhiên và liên tục theo thời gian. Quá trình này có thể được mô hình hóa bằng chuỗi Markov [28, 29], với các thuộc tính căn bản là:

- Tính liên tục: sự biến đổi diễn ra liên tục tại bất kì thời điểm nào.
- Tính độc lập: sự biến đổi chỉ phụ thuộc vào trạng thái hiện tại, không phụ thuộc vào trạng thái trong quá khứ.

Do trước đây năng lực tính toán của các hệ thống phần cứng còn nhiều hạn chế, nên nhằm đơn giản hóa tối đa việc ước lượng thì các nhà khoa học thường sử dụng thêm các thuộc tính khác như sau:

- Tính đồng nhất: tốc độ biến đổi giữa các nucleotit hay axit amin không đổi trong suốt quá trình biến đổi.
- Tính ổn định: tần số của các nucleotit hay axit amin là không đổi trong suốt quá trình biến đổi.
- Tính thuận nghịch: tốc độ biến đổi là đồng nhất theo các hướng tiến và lùi.

Các thuộc tính bổ sung này mặc dù không phù hợp với thực tế nhưng nó giúp phát triển các phương pháp tính toán để ước lượng những mô hình thay thế nucleotit hay axit amin, ví dụ một số mô hình dùng chung như JC69 [30], K80 [31] cho dữ liệu nucleotit hay PAM [5], PMB [32], Dayhoff [33], JTT [6], WAG [7] hay LG [8] cho dữ liệu sắp hàng axit amin. Khi loại bỏ một hay nhiều thuộc tính trong số này, chúng ta sẽ ước lượng được những mô hình phù hợp với thực tế hơn. Trong Chương 3, luận án sẽ trình bày phương pháp ước lượng mô hình sử dụng thuộc tính không thuận nghịch, đồng thời sử dụng nhiều ma trận tốc độ biến đổi thay vì đơn ma trận như các mô hình ở trên.

1.2.2 Mô hình thay thế nucleotit

Trước tiên, do dữ liệu DNA chỉ bao gồm bốn loại ký tự là A, T, G và C nên chúng ta xem xét quá trình thay thế nucleotit trong chuỗi DNA với bốn trạng thái. Để mô tả quá trình thay thế này, chúng ta có thể sử dụng ma trận tốc độ biến đổi tức thì Q như trong Bảng 1.6 dưới đây [34] dựa trên một số điểm lưu ý như sau:

- Q là ma trận 4x4 thể hiện quá trình chuyển đổi giữa bốn trạng thái nucleotit. q_{ij} là số lần xảy ra sự thay thế từ nucleotit *i* thành nucleotit *j* trong một đơn vị thời gian.

- Các hệ số π_A, π_T, π_G, π_C là tần suất xuất hiện của các nucleotit A, T, G và C.

- Các hệ số r_{ij} là hệ số hoán đổi giữa các nucleotit. Ở đây, do chúng ta giả thiết về tính chất thuận nghịch trong quá trình thay thế nên $r_{ij} = r_{ji}$.

- Do tính chất ổn định của mô hình nên tổng số biến đổi từ trạng thái i sang j sau khoảng thời gian t là bằng 0, tức là chúng ta có:

$$q_{ii} = -\sum_{j\#i} q_{ij} \tag{1.1}$$

Bảng 1.6. Ma trận biến đổi tức thì Q cho dữ liệu sắp hàng nucleotit

Q =		A	T	G	C
	A	$-\sum_{j\#A} q_{Aj}$	$r_{AT}\pi_T$	$r_{AG}\pi_G$	$r_{AC}\pi_C$
	T	$r_{TA}\pi_A$	$-\sum_{j\#T} q_{Tj}$	$r_{TG}\pi_G$	$r_{TC}\pi_C$
	G	$r_{GA}\pi_A$	$r_{GT}\pi_T$	$-\sum_{j\#G} q_{Gj}$	$r_{GC}\pi_C$
	C	$r_{CA}\pi_A$	$r_{CT}\pi_T$	$r_{CG}\pi_G$	$-\sum_{j\#C} q_{Cj}$

Chúng ta gọi $\mathbf{P}(t) = \{p_{ij}(t),\ i \in \mathbf{S},\ j \in \mathbf{S}\}$ là ma trận xác suất chuyển đổi, với $p_{ij}(t)$ là xác suất chuyển đổi từ trạng thái i sang trạng thái j trong một đơn vị thời gian t, ta có thể tính $\mathbf{P}(t)$ từ Q và t bằng công thức sau:

$$\mathbf{P}(t) = e^{Qt} \tag{1.2}$$

1.2.3 Mô hình thay thế axit amin

Tương tự như trên, xét quá trình Markov với tập trạng thái **S** = {A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V}, đây là tập gồm 20 loại axit amin. Gọi $\Pi = \{\pi_i\}$ với i = 1, …, 20 là véc tơ tần số xuất hiện của 20 axit amin, như vậy $\sum_{i=1}^{20} \pi_i = 1$ và các

π_i không đổi theo thời gian. Ta cũng kí hiệu $R = \{r_{ij}\}$ là ma trận hệ số hoán đổi trong đó r_{ij} là hệ số hoán đổi giữa hai axit amin i và j.

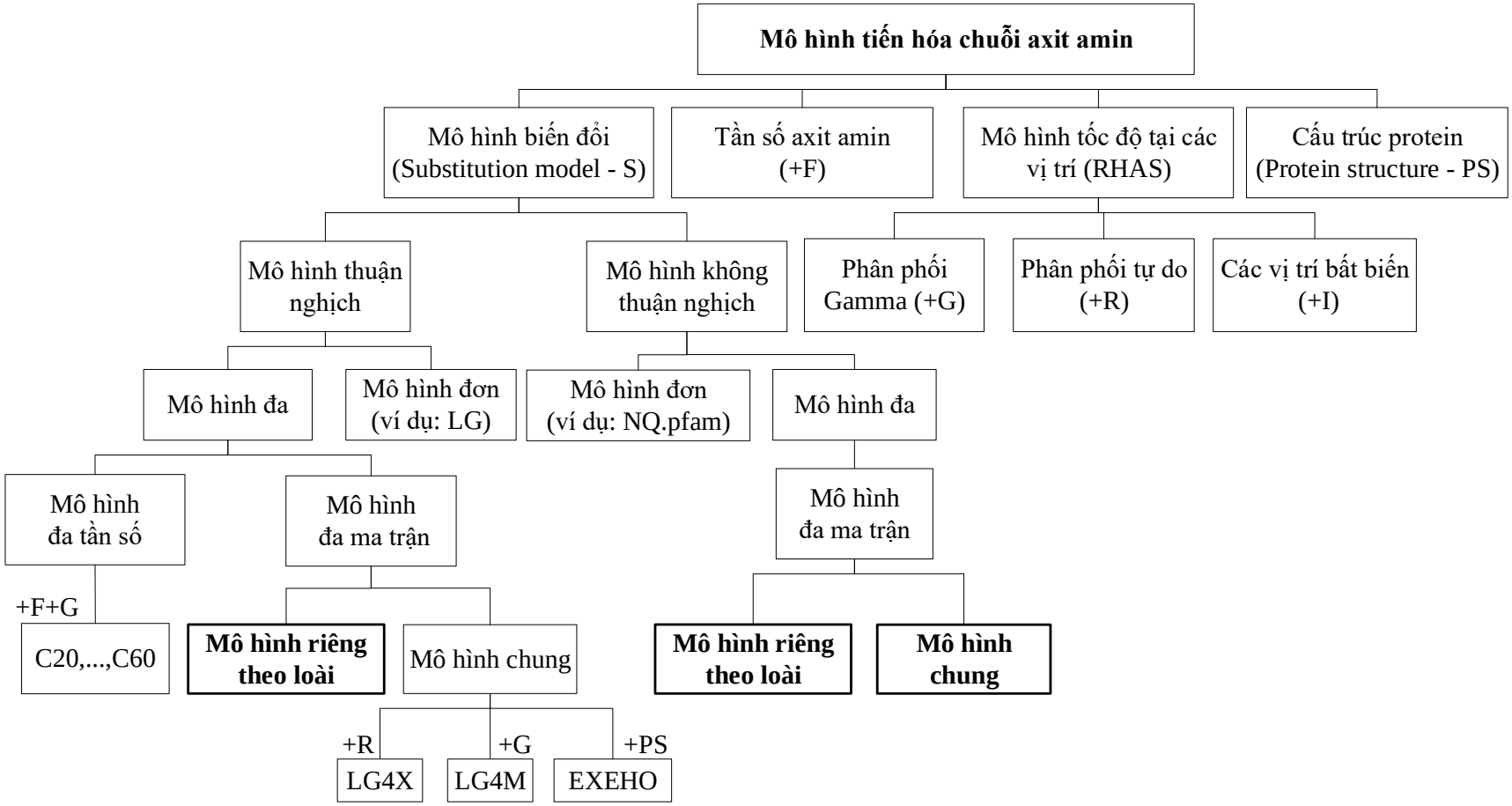
Chúng ta ký hiệu $Q = \{q_{ij},\ i \in \boldsymbol{S},\ j \in \boldsymbol{S}\}$ là ma trận tốc độ biến đổi tức thì có kích thước 20×20, trong đó q_{ij} là tốc độ biến đổi tức thì từ axit amin i sang axit amin j. Ma trận Q có thể được biểu diễn bởi ma trận $R = \{r_{ij}\}$ và vectơ $\Pi = \{\pi_i\}$ như sau:

$$q_{ij} = \begin{cases} \pi_j r_{ij} & nếu\ i \neq j \\ -\sum_{x \neq i} q_{ix} & nếu\ i = j \end{cases} \quad (1.3)$$

hoặc có thể được viết gọn dưới dạng: $Q = \Pi \times R$ trong đó, Π là vector tần suất còn R được gọi là ma trận hệ số hoán đổi. Do tính chất thuận nghịch ta có $r_{ij} = r_{ji}$ hay ma trận hệ số hoán đổi R đối xứng qua đường chéo chính. Hệ số hoán đổi giữa hai trạng thái i và j càng lớn thể hiện sự biến đổi giữa hai trạng thái i và j xảy ra càng nhiều và ngược lại. Như vậy chúng ta có thể ước lượng Π và R thay cho ước lượng Q. Ma trận biểu diễn tốc độ biến đổi tức thì Q còn được gọi là mô hình Q, việc ước lượng, lựa chọn và đánh giá các phương pháp ước lượng mô hình Q chính là mục tiêu của luận án sẽ được trình bày ở những phần tiếp theo. Số tham số cần ước lượng của Π là 19 do véc tơ Π có 20 thành phần nhưng tổng của 20 thành phần bằng 1. Ngoài ra, với mô hình tuân theo thuộc tính thời gian thuận nghịch thì R là ma trận đối xứng và được chuẩn hoá bởi hệ số $\mu = -\sum_i \pi_i q_{ii}$ để tổng số lượng biến đổi trong một đơn vị thời gian bằng 1, do đó số tham số cần ước lượng của R là $19 \times 20/2 - 1 = 189$. Như vậy, để ước lượng Q chúng ta cần phải ước lượng tổng cộng $189 + 19 = 208$ tham số. Trong trường hợp mô hình tuân theo thuộc tính thời gian không thuận nghịch thì do ma trận R không còn đối xứng nên số lượng tham số cần ước lượng cho Q sẽ là 379 tham số. Với lượng tham số lớn nên chúng ta cần phải dùng các tập dữ liệu có kích thước lớn để ước lượng mô hình đạt hiệu quả tốt nhất.

Sau khi có được mô hình Q, chúng ta có thể xây dựng lại cây cực đại hợp lý tối ưu hơn. Có nhiều cách phân chia các loại mô hình như: theo mục đích dùng (riêng-chung), theo số lượng ma trận (đơn ma trận hay đa ma trận) hoặc theo thuộc tính sử dụng để ước lượng (thời gian thuận nghịch hay không thuận nghịch). Khi chia theo mục đích dùng, chúng ta có: mô hình dùng chung là những mô hình được ước lượng từ tập cơ sở dữ liệu lớn và đa dạng các loài, từ đó mô hình sinh ra sẽ phù hợp khi nghiên cứu nhiều loài khác nhau; còn mô hình riêng theo loài là những mô hình được ước lượng từ tập dữ liệu theo các loài cụ thể, ví dụ dữ liệu về các loài thực vật hay các loài chim, do vậy nó phù hợp để nghiên cứu về các loài thực vật hay các loài chim hơn là các loài khác. Với mô hình thuận nghịch, bởi vì tốc độ biến đổi từ axit amin *i* sang axit amin *j* bằng với tốc độ biến đổi từ axit amin *j* sang axit amin *i* nên mô hình dạng này không thể hiện được hướng tiến hóa của các trình tự, do đó cây phân loài được sinh ra là cây không có gốc; ngược lại, mô hình không thuận nghịch sẽ xây dựng được cây phân loài có gốc.

Lược đồ trên Hình 1.4 thể hiện sự phân chia và kết hợp về mục đích sử dụng cũng như các thuộc tính khác nhau để tạo thành những mô hình cụ thể. Ví dụ, từ bộ dữ liệu Pfam [35] kết hợp thuộc tính thời gian thuận nghịch, mô hình đơn ma trận Q.pfam [11] đã được ước lượng; khi loại bỏ thuộc tính này các tác giả đã ước lượng được mô hình đơn ma trận (single model) không thuận nghịch như NQ.pfam [12]; hoặc dựa trên bộ dữ liệu HSSP [36] khi loại bỏ thuộc tính đồng nhất các nhà khoa học đã giới thiệu mô hình hỗn hợp đa ma trận (multi-matrix mixture model) như LG4X (tốc độ tuân theo phân bố tự do – free rate) hay LG4M (tốc độ tuân theo phân bố gamma) [37]. Luận án sẽ đi sâu vào việc ước lượng mô hình chung đa ma trận sử dụng thuộc tính thời gian không thuận nghịch và đồng thời cũng ước lượng mô hình đa ma trận dành riêng cho một nhóm loài cụ thể, xem phần in đậm trong Hình 1.4.



Hình 1.4. Lược đồ phân cấp các mô hình thay thế axit amin.

1.2.4 Bài toán ước lượng mô hình thay thế axit amin

1.2.4.1 Phát biểu bài toán

Dữ liệu đầu vào: Cho một tập N các sắp hàng $\mathbf{D} = \{D_1, \dots, D_N\}$. Mỗi sắp hàng bao gồm nhiều trình tự tương đồng.

Bài toán: Đề xuất phương pháp ước lượng mô hình thay thế axit amin cho tập N sắp hàng trên sao cho đảm bảo về độ chính xác của mô hình cũng như tối ưu về mặt thời gian thực thi.

Kết quả đầu ra: Mô hình thay thế axit amin Q tương ứng giải thích một cách tốt nhất sự biến đổi của các trình tự axit amin trong các sắp hàng đa trình tự của tập **D**.

Ước lượng Q là một bài toán khó và phức tạp do có nhiều tham số cần phải được tối ưu cùng lúc trong khi tài nguyên tính toán và thời gian là có giới hạn. Các phương pháp khác nhau để ước lượng mô hình Q có thể kể đến như: phương pháp đếm và phương pháp cực đại hợp lý. Với phương pháp đếm, các tham số cần ước lượng sẽ được tính toán trực tiếp từ tập các dữ liệu sắp hàng. Hai ma trận phổ biến sinh ra từ phương pháp này là PAM [5] và Blosum62 [38]. Tuy nhiên, phương pháp này chỉ phù hợp với những bộ dữ liệu có sự tương đồng cao hay các trình tự có liên hệ gần nhau [8]. Với sự đa dạng về dữ liệu như hiện nay thì phương pháp này không còn phù hợp nữa. Do vậy, luận án sẽ tập trung đi sâu vào phương pháp thứ hai sử dụng tiêu chuẩn cực đại hợp lý (Maximum Likelihood – ML) [8], [39] như trong phần trình bày tiếp theo dưới đây.

1.2.4.2 Phương pháp cực đại hợp lý ước lượng mô hình thay thế axit amin

Cho tập dữ liệu đầu vào gồm N sắp hàng $\mathbf{D} = (D_1, \dots, D_N)$ và $\mathbf{T} = (T_1, \dots T_N)$ là tập các cây phân loài tương ứng với các sắp hàng trong **D**. Theo đó, giá trị hợp lý của mô hình Q và tập cây phân loài **T** đối với tập sắp hàng **D** được tính theo công thức như sau:

$$L(Q, \mathbf{T}|\mathbf{D}) = \prod_{i=1}^{N} L(Q, T_i|D_i) \tag{1.4}$$

ở đây, $L(Q, T_i|D_i)$ là giá trị hợp lý của Q và T_i đối với sắp hàng D_i. Mô hình Q được ước lượng bằng cách tìm cực đại của giá trị hợp lý $L(Q, \mathbf{T}|\mathbf{D})$ theo công thức:

$$\begin{aligned} Q &= argmax_Q(L(Q, \mathbf{T}|\mathbf{D}) \\ &= argmax_Q\{\prod_{i=1}^{N} L(Q, T_\mathrm{i}|D_i)\} \end{aligned} \tag{1.5}$$

Theo Công thức 1.5, để tìm cực đại của giá trị hợp lý, chúng ta cần tối ưu đồng thời các tham số của Q và T_i, trong đó có thể là 379 tham số cho mô hình Q không thuận nghịch, đây là bài toán rất khó và phức tạp. Hiện tại, với việc nhiều mô hình chung đã được giới thiệu như LG, WAG, JTT, chúng ta có thể thực hiện việc ước lượng theo cách đơn giản hơn theo hướng xấp xỉ [8] mà vẫn đảm bảo độ tốt của mô hình như sau:

- Trước tiên, cố định mô hình Q, bằng cách sử dụng một mô hình chung, ví dụ mô hình LG [8]. Từ đó, chúng ta thực hiện tối ưu cây T_i theo mô hình Q đã cho.
- Sau khi có được cây T_i đã tối ưu, chúng ta cố định T_i và ước lượng lại mô hình Q để đạt được giá trị cực đại hợp lý.

Tối ưu T_i là bài toán NP-khó [29, 40], do số lượng cây tăng rất nhanh theo số lượng trình tự, nếu sắp hàng có m ($m \geq 4$) trình tự thì tổng số lượng cây không gốc sẽ là: $\prod_{\mathrm{i}=3}^{m}(2\mathrm{i} - 5)$, xem chi tiết số lượng cây tương ứng với số trình tự trong Bảng 1.7. Như vậy, với các sắp hàng có nhiều hơn 15 trình tự (thực tế có thể tới hàng trăm trình tự) thì để tìm được cây phân loài tối ưu nhất, chúng ta cần lượng tính toán rất lớn.

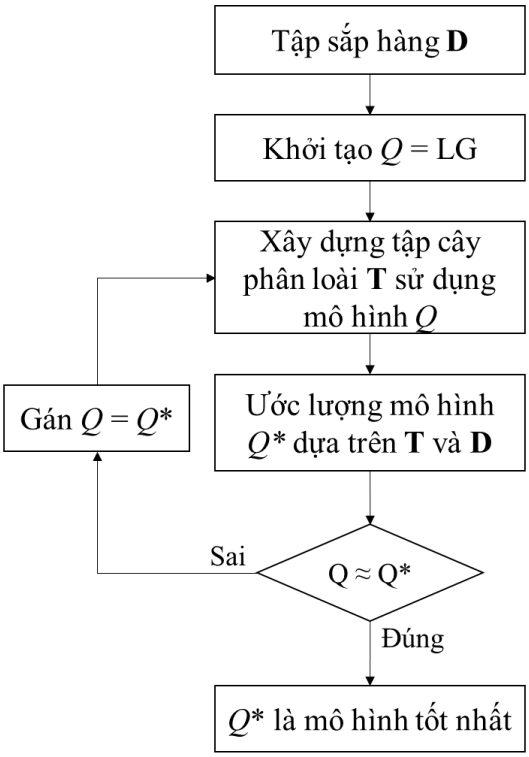
Bảng 1.7. Số lượng cây không gốc tương ứng với số trình tự trong sắp hàng

Số lượng trình tự	Số lượng cây không gốc
3	1
4	3
5	15
6	105
7	945
10	2,027,025
15	7,905,853,580,625

Để giải quyết bài toán này, tác giả Cường Đặng và cộng sự đã đề xuất phương pháp gần đúng FastMG [41]. Tư tưởng chính của phương pháp này là chia các sắp hàng đa trình tự có kích thước lớn ra thành các sắp hàng đa trình tự có kích thước nhỏ hơn, nhưng vẫn đảm bảo giữ đủ các thông tin về mối quan hệ giữa các trình tự để ước lượng tập các cây **T** và ma trận Q.

Tác giả Minh Bùi và cộng sự đã giới thiệu phương pháp IQ-TREE [42] tìm cây phân loài với tư tưởng cốt lõi phát triển từ thuật toán leo đồi và tiêu chuẩn cực đại hợp lý. Bên cạnh đó, chúng ta có thể thực hiện tìm mô hình phù hợp nhất cho mỗi đầu vào bằng phương pháp ModelFinder [13], nó cho phép nhận dữ liệu đầu vào với nhiều định dạng khác nhau như: phylip, fasta, nexus. Phương pháp này hiện đang được sử dụng rất rộng rãi với thời gian chạy nhanh và tốn ít bộ nhớ.

Gần đây, các nhà khoa học đã tiếp tục giới thiệu hai phương pháp mới để ước lượng nhanh mô hình Q gọi là QMaker [11] giành cho ước lượng mô hình thời gian thuận nghịch và nQMaker [12] cho ước lượng mô hình thời gian không thuận nghịch. Hai phương pháp này dùng tiêu chuẩn cực đại hợp lý để ước lượng mô hình thay thế axit amin theo sơ đồ tóm tắt như trong Hình 1.5.



Hình 1.5. Sơ đồ các bước ước lượng mô hình thay thế axit amin dựa trên phương pháp cực đại hợp lý.

Với hai phương pháp này, các tác giả đã ước lượng một số mô hình cho từng loài riêng biệt ví dụ như Q.plant và NQ.plant cho bộ dữ liệu thực vật (Plant) [43], Q.bird và NQ.bird cho bộ dữ liệu các loài chim (Bird) [44], ngoài ra còn có mô hình cho các bộ dữ liệu Yeast, Mammal, Insect, Pfam [11, 12] hay Metazoan [45].

Nhiều nghiên cứu đã chỉ ra rằng mặc dù một số mô hình thay thế axit amin khác nhau đã được giới thiệu nhưng các mô hình này đều đã có sự tác động lên cấu trúc cây phân loài [11, 12, 40-42, 46]. Việc này chứng tỏ rằng mô hình mới sẽ ngày càng tối ưu cấu trúc cây và việc ước lượng mô hình thay thế mới là rất quan trọng, cần thiết trong nghiên cứu tiến hóa. Vì vậy, việc đánh giá các phương pháp ước lượng mô hình là cần thiết. Nội dung Chương 2 của luận án sẽ tập trung vào phát triển quy trình đánh giá phương pháp ước lượng mô hình thay thế axit amin sử dụng tiêu chuẩn cực đại hợp lý,

đồng thời đưa ra các tiêu chuẩn về độ đo, khối lượng dữ liệu cần thiết cho việc ước lượng mô hình.

Tiếp theo, khi số lượng các mô hình ngày càng nhiều và đa dạng, để nghiên cứu một sắp hàng đa trình tự axit amin nào đó, chúng ta cần lựa chọn mô hình tối ưu và phù hợp nhất với sắp hàng đó. Thông thường, phương pháp cực đại hợp lý được sử dụng, tiêu biểu là phương pháp ModelFinder [13], với mỗi mô hình cần đánh giá, phương pháp này sẽ thực hiện tính giá trị hợp lý của cây và tìm ra mô hình cho giá trị hợp lý tốt nhất. Tuy nhiên, phương pháp này tốn nhiều thời gian và không gian tính toán. Do vậy, chúng ta cần phát triển một phương pháp tính toán mới sao cho có thể ước lượng hay lựa chọn nhanh chóng mô hình thay thế cho một sắp hàng đa trình tự axit amin bất kì. Đây là nội dung của Chương 4, giới thiệu phương pháp lựa chọn mô hình dựa trên mạng học sâu.

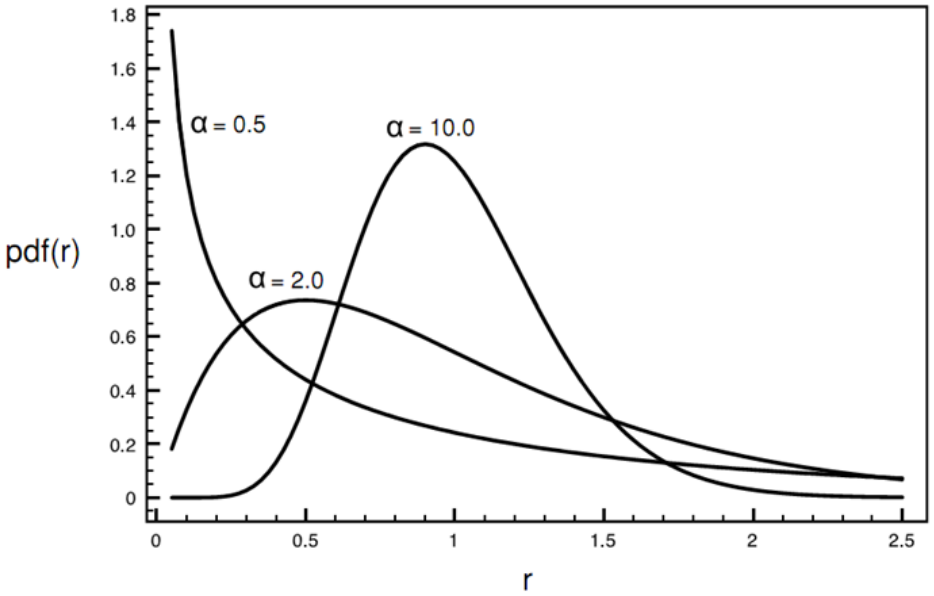
1.2.4.3 Mô hình tốc độ biến đổi tại các vị trí trên trình tự

Việc giả thiết rằng tốc độ biến đổi tại tất cả các vị trí đều như nhau là không phù hợp với thực tế, nhiều nghiên cứu đã cho thấy tốc độ biến đổi là không đồng nhất [40, 41, 46-48]. Do đó, việc sử dụng một mô hình tốc độ biến đổi là cần thiết để giải thích hiện tượng này. Thông thường, chúng ta có thể mô hình hóa bằng phân phối gamma (Γ) với giá trị kỳ vọng 1 và phương sai $1/\alpha$ ($\alpha > 0$) [48]. Trong đó, công thức của hàm mật độ như sau:

$$Pdf(r) = \frac{\alpha^{\alpha} r^{\alpha-1}}{e^{\alpha r}\Gamma(\alpha)} \qquad (1.6)$$

ở đây, $\Gamma(\alpha) = \int_0^{\infty} e^{-t} t^{\alpha-1} dt$. Hình dạng của hàm mật độ thay đổi với các giá trị α khác nhau được minh họa trên Hình 1.6, vì thế α còn được gọi là tham số hình dạng. Chúng ta có thể thấy hình dạng của đồ thị rất khác biệt tùy theo giá trị của α, ví dụ với

$\alpha = 0.5$ đồ thị có dạng chữ L và hầu hết các vị trí có tốc độ biến đổi chậm, chỉ nhanh ở một vài vị trí. Ngược lại, với $\alpha = 10$ thì đồ thị hình chuông với đỉnh tại $r = 1$, nghĩa là hầu hết các vị trí có tốc độ gần bằng nhau.



Hình 1.6. Hàm mật độ xác suất của phân phối gamma.

Áp dụng phân phối gamma vào Công thức 1.2 ta có công thức tính giá trị hợp lý của mô hình Q và tập cây **T** với tập sắp hàng **D** giờ đây trở thành như sau:

$$L(Q, \mathbf{T}|\mathbf{D}) = \prod_{i=1}^{N} L(\Gamma(\alpha)Q, T_i|D_i) \qquad (1.7)$$

Các nghiên cứu thường sử dụng phân phối gamma rời rạc với số K hữu hạn các phân lớp tốc độ, ví dụ bốn hoặc tám phân lớp tùy theo nhu cầu mỗi bài toán, thông thường ta lấy bốn phân lớp tốc độ tương ứng là "rất chậm" (very slow), "chậm" (slow), "bình thường" (medium) và "nhanh" (fast) để mô tả tốc độ biến đổi trên các vị trí. Sử dụng bốn hay tám phân lớp này là vừa đủ để xấp xỉ phân phối gamma, cũng như hợp lý cho việc mô hình hóa mô hình tốc độ mà lại không quá phức tạp về mặt tính toán.

Ước lượng các mô hình thay thế axit amin Q đi kèm với mô hình tốc độ tại các vị trí là nội dung sẽ được trình bày trong Chương 3 của luận án.

1.3 Các phương pháp so sánh hai mô hình thay thế axit amin

1.3.1 So sánh dựa trên giá trị các hệ số của hai mô hình

Một trong những tiêu chuẩn phổ biến để so sánh hai mô hình là độ tương quan Pearson. Theo đó, giá trị này sẽ cho ta thấy mối quan hệ tuyến tính giữa các hệ số tương ứng của hai mô hình. Cho hai biến ngẫu nhiên *X* và *Y*, công thức tính độ tương quan Pearson, $p(X,Y)$, của *X* và *Y* như sau:

$$p(X,Y) = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \quad (1.8)$$

trong đó, σ_X, σ_Y là độ lệch chuẩn (standard deviation) của *X* và *Y*, μ_X, μ_Y tương ứng là giá trị trung bình (mean) và *E* là giá trị kì vọng. Độ tương quan Pearson có giá trị trong khoảng từ -1 đến 1, trong đó nếu độ tương quan bằng 0 thể hiện hai ma trận không tương quan với nhau, giá trị 1 (-1) thể hiện hai ma trận có tương quan cùng tăng (giảm).

Áp dụng cho hai mô hình thay thế axit amin, ta có thể coi đây là hai vector gồm N phần tử tương ứng với nhau. Do vậy, công thức tính độ tương quan Pearson trở thành:

$$p(X,Y) = \frac{\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i-\bar{x})^2}\sqrt{\sum_{1}^{N}(y_i-\bar{y})^2}} \quad (1.9)$$

ở đây, N là số phần tử của mỗi vector; x_i, y_i là các điểm dữ liệu tại vị trí *i* tương ứng; $\bar{x} = \frac{1}{N}\sum_{1}^{N} x_i$, $\bar{y} = \frac{1}{N}\sum_{1}^{N} y_i$ là các giá trị trung bình của hai vector.

Bên cạnh độ đo Pearson, chúng ta cũng có thể so sánh về độ lớn của giá trị các hệ số trong mô hình. Các mốc để đánh giá thường là "2x" hay "5x" nghĩa là giá trị hệ số có sự chênh lệch khác nhau hai lần hoặc năm lần. Việc này sẽ giúp đánh giá trực quan mối

liên hệ giữa mô hình và thực tế về mặt sinh học xem liệu đã có sự hợp lý về tốc độ biến đổi giữa các cặp axit amin hay chưa. Ngoài ra chúng ta có thể dùng các giá trị như trung bình bình phương sai số (mean squared error – MSE) để đánh giá độ lệch giữa các hệ số mô hình. Giả sử ta có hai vector *X, Y* là các giá trị hệ số của hai mô hình thay thế, vậy độ lệch MSE được tính như sau:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \qquad (1.10)$$

với x_i, y_i là các giá trị tại vị trí i tương ứng của hai mô hình.

Các giá trị này đặc biệt hữu ích khi làm việc với các dữ liệu mô phỏng, do dữ liệu được sinh ra từ một mô hình đã có, nên chúng ta cần đánh giá xem mô hình sinh ra từ dữ liệu mô phỏng có liên hệ như thế nào với mô hình đúng được dùng để tạo dữ liệu. Khi ấy, các hệ số của mô hình đúng được coi là giá trị tham chiếu (hay giá trị chuẩn), còn hệ số của mô hình mô phỏng được xem như giá trị quan sát được.

1.3.2 So sánh dựa trên giá trị hợp lý

Dựa trên giá trị hợp lý là cách phổ biến trong so sánh hai mô hình được thực hiện bằng cách xây dựng cây phân loài dựa trên cùng một tập các sắp hàng cho trước. Mô hình nào tạo được cây có giá trị hợp lý tốt hơn được xem là tốt hơn. Do giá trị hợp lý rất nhỏ nên để thuận tiện trong tính toán và tối ưu ta thường sử dụng giá trị logarit của giá trị hợp lý, giá trị này là một số âm.

Bên cạnh giá trị hợp lý, hai tiêu chuẩn khác cũng thường được sử dụng là tiêu chuẩn thông tin Bayesian (Bayesian information criterion - BIC) [49] và tiêu chuẩn thông tin Akaike (Akaike information criterion - AIC) [50]. Trong đó, công thức tính của hai tiêu chuẩn tương ứng như sau:

$$BIC(Q|D) \;=\; 2 \times k \times ln(n) \;-\; 2 \times ln\big(L(T|Q,D)\big) \tag{1.11}$$

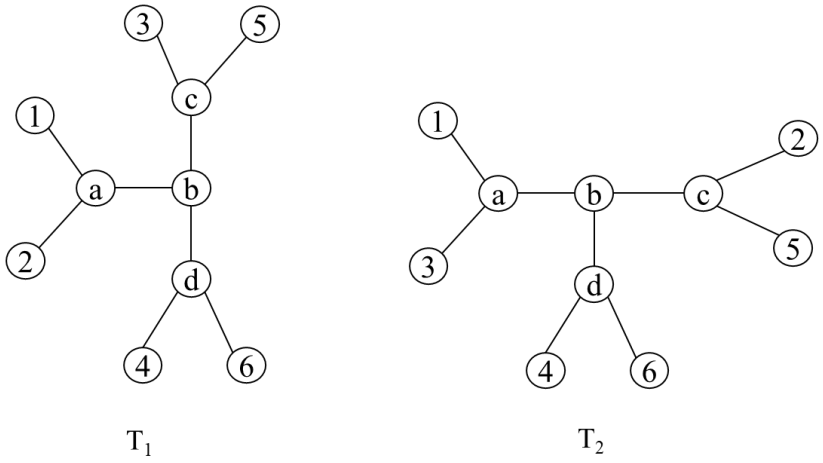
$$AIC(Q|D) = 2 \times k - 2 \times ln(L(Q,T|D)) \tag{1.12}$$

ở đây, k là số tham số tự do của mô hình, n là kích thước của sắp hàng. Như vậy, việc bổ sung giá trị tham số của mô hình vào công thức sẽ làm cho việc so sánh công bằng hơn. Do giá trị $ln(L(Q,T|D))$ là số âm nên giá trị BIC hay AIC càng nhỏ thì mô hình càng tốt và ngược lại.

1.3.3 So sánh dựa trên cấu trúc cây phân loài

Trên cùng một tập dữ liệu đa sắp hàng, các mô hình khác nhau có thể tạo ra các cây phân loài có cấu trúc khác nhau. Phép so sánh cấu trúc cây không thể cho biết mô hình nào tốt hơn nhưng chúng ta có thể đánh giá được sự khác biệt của chúng trong việc xây dựng cây. Cây có cấu trúc càng giống nhau thì chứng tỏ hai mô hình càng gần nhau.

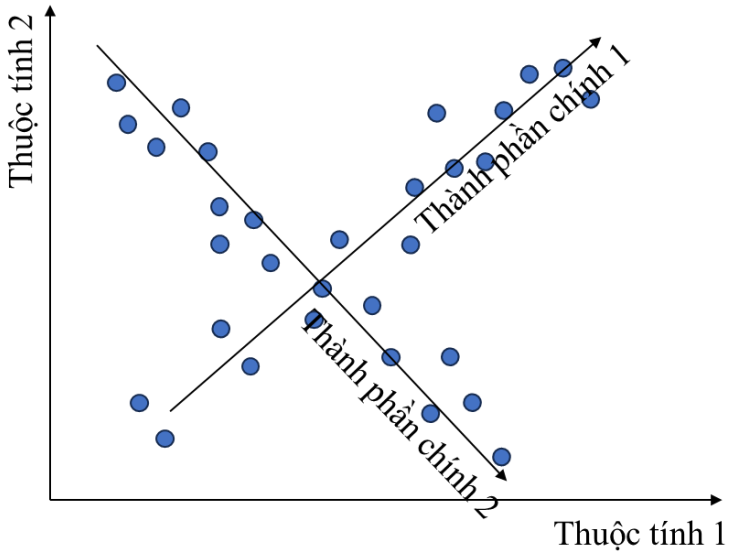
Trong phần này, luận án sử dụng khoảng cách Robinson-Foulds (RF) [51]. Giá trị RF được tính bằng cách lấy tỉ lệ giữa số nhánh (phân hoạch) chỉ tồn tại ở một trong hai cây trên tổng số nhánh của cả hai cây. Giá trị RF được chuẩn hóa và có phạm vi từ 0 đến 1, trong đó giá trị 1 thể hiện hai cấu trúc cây hoàn toàn khác nhau, giá trị càng thấp thì hai cây càng giống nhau, giá trị 0 thể hiện hai cấu trúc cây hoàn toàn giống nhau. Xét hai cây nhị phân không gốc T_1 và T_2 trong Hình 1.7, ta có các phân hoạch tương ứng của cây T_1 là PH_1 = {(1,2)|(3,4,5,6)}, PH_2 = {(4,6)|(1,2,3,5)} và PH_3 = {(3,5)|(1,2,4,6)}; phân hoạch tương ứng của cây T_2 là PH_4 = {(1,3)|(2,4,5,6)}, PH_5 = {(2,5)|(1,3,4,6)} và PH_6 = {(4,6)|(1,2,3,5)}. Như vậy tổng cộng ta có sáu phân hoạch PH_1, …, PH_6, trong đó hai phân hoạch PH_1, PH_3 chỉ tồn tại trên cây T_1 và hai phân hoạch PH_4, PH_5 chỉ tồn tại trên T_2; phân hoạch PH_2 và PH_6 giống nhau tồn tại trên cả hai cây. Như vậy khoảng cách RF giữa hai cây T_1 và T_2 trong trường hợp này là 4/6.



Hình 1.7. Khoảng cách Robinson-Foulds giữa hai cây T_1 và T_2.

1.3.4 So sánh sử dụng phân tích thành phần chính

Phân tích thành phần chính (principal component analysis - PCA) là phương pháp giảm số chiều của một bộ dữ liệu nhằm mục đích trích xuất ra mô hình và xu hướng quan trọng của dữ liệu. Mỗi mô hình thay thế axit amin bao gồm 400 phần tử, chúng ta có thể coi nó như một vector gồm có 400 chiều. PCA sẽ thực hiện giảm số chiều từ 400 về hai chiều thành phần chính, mỗi ma trận sẽ được đặc trưng bởi một điểm trên không gian hai chiều đó. Khoảng cách và phân bố của các điểm dữ liệu sẽ cho chúng ta thấy xu hướng và đặc trưng của các mô hình ban đầu. Về mặt kỹ thuật, PCA sẽ thực hiện xoay trục tọa độ sao cho trong hệ trục mới ta có thể loại bỏ hầu hết các chiều có phương sai (khoảng cách từ điểm dữ liệu gốc đến trục thành phần chính) rất nhỏ, chúng ta chỉ cần giữ lại các thành phần quan trọng hơn mà thôi, xem minh họa trong Hình 1.8. Thực hiện các bước xoay trục nhiều lần ta có thể xác định hai hoặc nhiều hơn số thành phần chính trong hệ trục mới.



Hình 1.8. Minh họa phương pháp phân tích thành phần chính.

1.4 Các phương pháp lựa chọn mô hình phù hợp nhất với dữ liệu

1.4.1 Phương pháp dựa trên cực đại hợp lý

Đây là phương pháp truyền thống và được áp dụng phổ biến trong việc lựa chọn mô hình phù hợp nhất với dữ liệu. Một số phương pháp tiêu biểu có thể kể đến là PhyML [52], RAxML [10], ModelTest-NG [14] và ModelFinder [13]. Về mặt kĩ thuật, các phương pháp này cho phép chúng ta tính toán giá trị hợp lý (LikeLihood) hoặc các giá trị dựa trên giá trị hợp lý như BIC [49] hay AIC [50] của cây dựa trên một mô hình nào đó cho trước. Các mô hình có thể là tổ hợp của mô hình thay thế axit amin kết hợp với mô hình tốc độ biến đổi tại các vị trí. Tổ hợp nào cho giá trị hợp lý tốt nhất sẽ là kết quả cuối cùng. Phương pháp dựa trên tiêu chuẩn cực đại hợp lý tuy rằng có độ chính xác cao nhưng lại yêu cầu lượng tính toán rất lớn, nhất là khi số lượng mô hình thay thế tăng lên.

1.4.2 Phương pháp dựa trên học máy

Đây là phương pháp hiện đại và được áp dụng gần đây trong lĩnh vực tin sinh học tiến hóa. Một số công bố có thể kể đến như ModelTeller [16] sử dụng RandomForest

hay ModelRevelator [15] sử dụng mạng học sâu ResNet-18 [53]. Để sử dụng các phương pháp này, đầu tiên, chúng ta cần trích chọn các đặc trưng từ dữ liệu như: các biến đổi trình tự, entropy, độ dài cạnh. Các phương pháp học máy sẽ học các đặc trưng dữ liệu từ đó có thể tính toán và đưa ra mô hình thay thế có xác suất cao nhất. Phương pháp sử dụng trí tuệ nhân tạo như học sâu thì có ưu điểm chính là hoạt động nhanh, độ chính xác cao. Tuy nhiên, nếu lượng dữ liệu lớn đến hàng triệu sắp hàng và số lượng thuộc tính đào tạo mô hình lên tới hàng chục nghìn sẽ yêu cầu khả năng tính toán rất lớn. Điều này đòi hỏi các phần cứng chuyên dụng như GPU hay các hệ thống tính toán hiệu năng cao, đây là những phần cứng đắt tiền, đôi khi vượt quá khả năng của nhiều nhà khoa học. Vì thế, chúng ta cần phải phát triển các phương pháp trích chọn được đặc trưng tối ưu nhất từ dữ liệu, để tối thiểu hóa số lượng thuộc tính trong khi tối đa hóa hiệu quả mạng học sâu. Chương 4 của luận án sẽ làm rõ những vấn đề này và đề xuất một mạng học sâu hiệu quả cho lựa chọn mô hình thay thế tối ưu.

1.5 Các bộ dữ liệu

Các bộ dữ liệu thật được sử dụng trong luận án bao gồm:

- Các bộ dữ liệu chung: HSSP [54], TreeBASE [55] và Pfam [35]. Đây là các bộ dữ liệu được dùng để ước lượng và đánh giá các mô hình dùng chung như LG [8], LG4X, LG4M [37] hay Q.pfam [11]. Đặc điểm chung của các bộ dữ liệu này là sự đa dạng về số trình tự trong các sắp hàng và độ dài các sắp hàng. Mô hình chung thì có thể được sử dụng để phân tích nhiều loại sắp hàng thuộc nhiều loài khác nhau.

- Các bộ dữ liệu riêng theo loài: plants [43], birds [44], yeasts [56], mammals [57] và insects [58]. Đây là các bộ dữ liệu của từng nhóm loài vì vậy mỗi sắp hàng của từng bộ dữ liệu có số trình tự bằng nhau. Từ các bộ dữ liệu này, nhiều mô hình riêng theo loài như Q.plant, Q.bird [11] đã được đề xuất, tuy nhiên, các mô hình riêng theo nhóm loài

thì chỉ phù hợp để phân tích các dữ liệu của các loài đó mà thôi. Ví dụ, Q.plant thì chỉ phù hợp phân tích các dữ liệu về các loài thực vật.

Các bộ dữ liệu thật thường có chứa các kí tự đặc biệt như:

- Ký tự trống (gap, kí hiệu '-'): đây là các kí tự xuất hiện trong quá trình sắp hàng các trình tự, thể hiện sự chèn hay xóa dữ liệu.

- Ký tự bị mất (missing, kí hiệu 'x' hoặc '?'): đây là một dạng không được chỉ rõ là chèn hay xóa, thường do lỗi kỹ thuật hoặc thiếu thông tin trong quá trình xử lý giải trình tự chuỗi.

Các kí tự đặc biệt này xuất hiện nhiều sẽ gây nhiễu loạn thông tin và là thách thức lớn trong các bài toán ước lượng và lựa chọn mô hình.

1.6 Tổng kết chương

Dữ liệu về các trình tự axit amin chứa thông tin quan trọng mà con người vẫn chưa thể khám phá hết được. Với sự tiến bộ của các máy giải trình tự thế hệ mới, ngày càng nhiều dữ liệu được công bố đi kèm với thách thức không nhỏ để xử lý và giải thích chúng. Quá trình tiến hóa luôn xảy ra những biến đổi trong trình tự axit amin, với đặc điểm môi trường sống ngày nay thì nó diễn biến lại càng phức tạp hơn khi mà hóa chất, thuốc men đang lan tràn và gây ảnh hưởng mạnh mẽ đến cơ thể sống của mọi sinh vật. Nghiên cứu về sự tiến hóa và sự khác biệt giữa các loài là nhiệm vụ trọng tâm trong tin sinh học, trong đó có bài toán về mô hình thay thế axit amin. Đây là bài toán quan trọng và căn bản bậc nhất trong tin sinh học tiến hóa.

Chuỗi Markov thường được sử dụng để mô hình hóa quá trình biến đổi này, nó được thể hiện dạng ma trận hay mô hình Q. Ước lượng Q là bài toán khó và phức tạp, đòi hỏi kết hợp nhiều phương pháp khác nhau, trong đó phương pháp truyền thống và phổ biến nhất hiện nay là phương pháp cực đại hợp lý. Với phương pháp này, các nhà

khoa học đã ước lượng được một số mô hình đơn ma trận dùng chung như LG, WAG hay JTT. Ngoài ra những mô hình thay thế riêng cho các loài cũng đã được đề xuất nhằm thuận tiện hơn trong việc nghiên cứu chuyên sâu về các loài đó, ví dụ mô hình Q.plant cho dữ liệu các loài thực vật và Q.bird cho dữ liệu các loài chim. Những mô hình này (chung hay riêng) là nền tảng để tiếp tục ước lượng các mô hình mới nhằm tối ưu hóa cây phân loài.

Chương 2. Quy trình đánh giá các phương pháp ước lượng mô hình thay thế axit amin

Chương 2 giới thiệu về quy trình đánh giá các phương pháp ước lượng mô hình thay thế axit amin, thực nghiệm trên hai phương pháp mới được đề xuất là QMaker và nQMaker. Quy trình bao gồm các bước như: sinh dữ liệu mô phỏng, ước lượng mô hình và đánh giá mô hình dựa trên các tiêu chí như chênh lệch hệ số mô hình, giá trị hợp lý của cây và khoảng cách Robinson-Foulds.

2.1 Giới thiệu chung

Như đã nêu ở Chương 1, dữ liệu thật ngày càng nhiều lên và bao phủ trên một phạm vi rộng các chủ đề nghiên cứu. Tuy nhiên, dữ liệu còn mang tính rời rạc, trong nhiều trường hợp như là các bộ dữ liệu riêng biệt theo loài thì chúng ta chưa có đủ dữ liệu thật nhằm đánh giá một cách tổng thể, bao quát các phương pháp tính toán trong tin sinh học tiến hóa. Trong tình huống đó, dữ liệu mô phỏng có thể bù đắp lại phần thiếu hụt này một cách hiệu quả để đánh giá các phương pháp ước lượng mô hình thay thế axit amin.

Trong lĩnh vực tin sinh học, nhiều công trình nghiên cứu đã sử dụng dữ liệu mô phỏng để đánh giá hiệu quả mô hình, thời gian thực thi cũng như khả năng đáp ứng của phương pháp được đề xuất. Ví dụ trong [59], các tác giả sử dụng dữ liệu mô phỏng để đánh giá mối liên hệ giữa các loài gần nhau, hay dữ liệu mô phỏng được dùng để sinh dữ liệu đào tạo và dữ liệu kiểm thử trong các nghiên cứu như ModelTeller [16] và ModelRevelator [15] nhằm ước lượng nhanh mô hình thay thế của trình tự DNA; hoặc dữ liệu mô phỏng còn được dùng để đào tạo mô hình trí tuệ nhân tạo với mục đích dự đoán độ dài cạnh của cây phân loài [60].

Một số phương pháp sinh dữ liệu mô phỏng đã được giới thiệu và sử dụng rộng rãi như SeqGen [61], INDELible [62] hay gần đây là AliSim [63, 64]. Các phương pháp này

đều cho phép người dùng tạo dữ liệu theo một mô hình thay thế axit amin và mô hình tốc độ nào đó một cách linh hoạt với hiệu quả đã được chứng minh trong thực nghiệm.

Quay trở lại với vấn đề mô hình thay thế axit amin, như đã nói ở chương trước, ngoài phương pháp truyền thống như XRate [9] hay RAxML [10], hai phương pháp mới hơn và hiệu quả hơn được giới thiệu gồm QMaker [11] và nQMaker [12] dùng để ước lượng mô hình thay thế axit amin từ một bộ dữ liệu bất kì. Với hai phương pháp này, các tác giả đã ước lượng một số mô hình theo các loài cụ thể như Plant, Bird, Yeast, Mammal, Insect với các thuộc tính thuận nghịch hay không thuận nghịch về thời gian. Các mô hình theo loài này đã thể hiện rõ ưu thế vượt trội so với các mô hình chung như LG [8], WAG [7] hay JTT [6] trên các tập dữ liệu theo loài tương ứng [11, 12]. Điều đó chứng tỏ việc ước lượng các mô hình riêng cho từng loài là hợp lý và hai phương pháp QMaker, nQMaker đã thể hiện được vai trò cũng như tiềm năng trong việc giải quyết bài toán này.

Ước lượng mô hình theo phương pháp cực đại hợp lý cần rất nhiều thời gian và tài nguyên tính toán đặc biệt khi số lượng dữ liệu tăng lên hàng trăm nghìn hay hàng triệu sắp hàng. Tuy nhiên, để tiết kiệm thời gian, đôi khi chúng ta cần có được mô hình đạt hiệu năng chấp nhận được trong thời gian nhanh nhất. Bài toán cần giải quyết sẽ có dạng như như sau đây:

Đầu vào: tập dữ liệu sắp hàng **D** gồm N sắp hàng.

Bài toán: dựa trên phương pháp cực đại hợp lý, xác định số lượng tối thiểu dữ liệu để ước lượng mô hình thay thế axit amin mà giải thích tốt nhất sự biến đổi của các sắp hàng trong tập dữ liệu **D**.

Đầu ra: số lượng tối thiểu sắp hàng.

Để giải quyết bài toán này, chúng ta cần đánh giá các phương pháp ước lượng mô hình, ví dụ QMaker và nQMaker. Tuy vậy, việc đánh giá các phương pháp ước lượng này vẫn còn chưa đầy đủ. Trong đó, có những câu hỏi cần khám phá như:

- Độ tốt của mô hình thay đổi như thế nào theo kích thước dữ liệu? Độ tốt ở đây được hiểu là khả năng xây dựng cây theo tiêu chuẩn cực đại hợp lý và ảnh hưởng của mô hình đến cấu trúc cây phân loài khi kích thước của dữ liệu thay đổi.
- Chúng ta cần bao nhiều dữ liệu để đạt được mô hình đủ tốt? Dữ liệu axit amin có nhiều bộ với nhiều kích thước khác nhau có thể lên tới hàng chục nghìn hoặc trăm nghìn sắp hàng khác nhau. Dữ liệu càng lớn thì mô hình càng tối ưu nhưng cũng đồng nghĩa với việc thời gian và công sức tính toán tăng lên.
- Trên thực tế, chúng ta không có mô hình đúng và cây đúng cho các dữ liệu thật. Vậy làm thế nào để đánh giá các phương pháp ước lượng mô hình thay thế axit amin?

Với câu hỏi số 1 và số 2, chúng ta có thể áp dụng các phương pháp mô phỏng theo nhiều kích thước sắp hàng khác nhau để đánh giá. Để giải quyết câu hỏi số 3, chúng ta có thể sử dụng dữ liệu mô phỏng. Dữ liệu mô phỏng có một số thuận lợi như sau đây:

- Thứ nhất, chúng ta có thể tạo ra một số lượng dữ liệu mô phỏng tùy ý để đánh giá phương pháp.
- Thứ hai, để sinh dữ liệu mô phỏng, chúng ta cần phải dựa trên một mô hình thay thế axit amin đã có từ trước, do vậy, mô hình này sẽ được coi là "mô hình đúng" của dữ liệu mô phỏng.
- Cuối cùng, cây phân loài dùng trong quá trình sinh dữ liệu mô phỏng cũng được coi là "cây đúng" đối với dữ liệu được sinh ra. Khái niệm "cây đúng" (true tree) này khác biệt với "cây thực tế" (real tree hay biological tree), chúng ta có thể thu được "cây thực tế" dựa trên các phân tích mang tính sinh học.

Từ các luận điểm trên, luận án đề xuất quy trình đánh giá phương pháp ước lượng dựa trên dữ liệu mô phỏng gồm các bước như sau:

- Bước 1: Thu thập các dữ liệu sắp hàng thật, ước lượng mô hình thay thế Q, cây phân loài T và các tham số tốc độ biến đổi tại các vị trí.
- Bước 2: Sinh dữ liệu mô phỏng dựa trên mô hình Q, cây phân loài T và các tham số liên quan như tốc độ biến đổi tại các vị trí, tham số hình dạng α của phân phối gamma tìm được ở Bước 1.
- Bước 3: Ước lượng mô hình Q^* mới bằng các phương pháp ước lượng mô hình dựa trên dữ liệu sinh ra ở Bước 2.
- Bước 4: Đánh giá mô hình Q^* so với Q sử dụng các tiêu chuẩn như: giá trị hệ số biến đổi của mô hình, tiêu chuẩn cực đại hợp lý và khoảng cách RF.

Mục tiêu phần này của luận án là dựa trên việc sử dụng dữ liệu mô phỏng để ước lượng mô hình, rồi đánh giá xem độ tốt của mô hình được ước lượng thay đổi như thế nào đối với từng kích thước dữ liệu đào tạo.

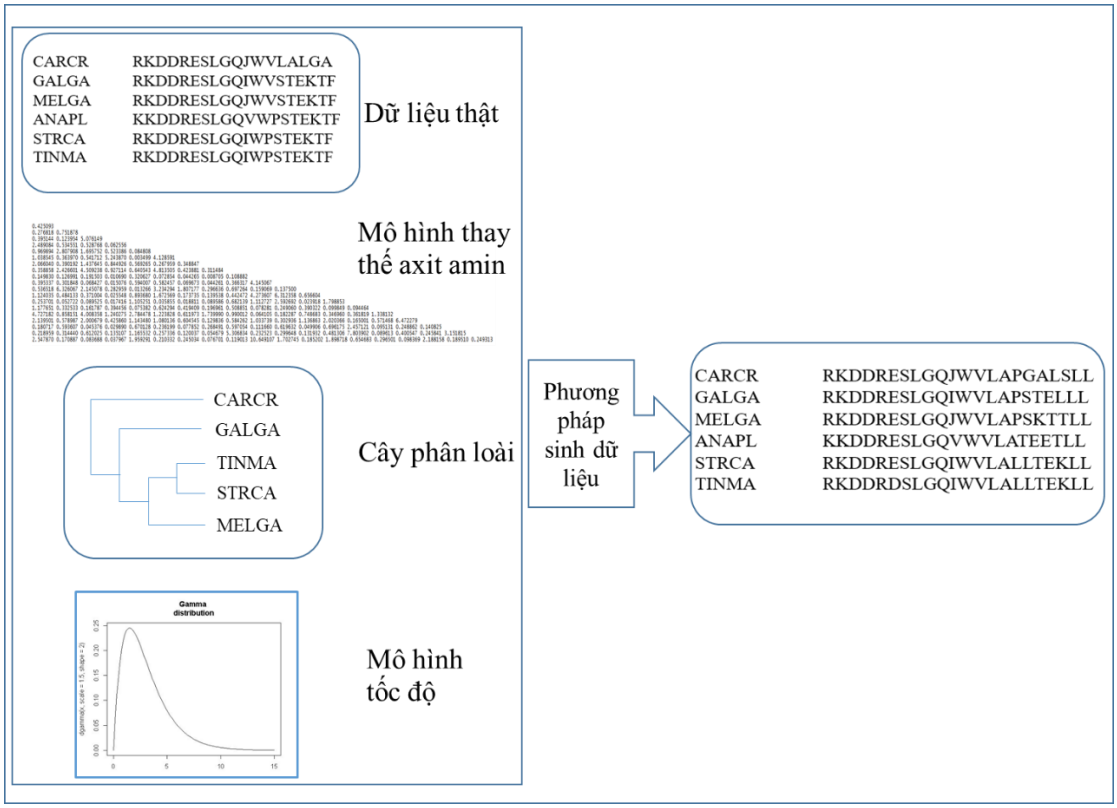
2.2 Phương pháp

2.2.1 Phương pháp sử dụng dữ liệu đa sắp hàng mô phỏng

Với mỗi sắp hàng thật R_i quá trình để sinh một sắp hàng $D_i \in \mathbf{D}$ bao gồm hai bước chính như trong Hình 2.1, cụ thể như sau:

- Ước lượng tham số: với mỗi sắp hàng thật R_i gồm m_i trình tự với độ dài l. Bằng cách chạy IQ-TREE [42] cho mỗi sắp hàng, luận án sẽ xác định được các tham số như mô hình tốc độ biến đổi trên mỗi vị trí ρ_i, giá trị tham số hình dạng α. Ngoài ra, chạy IQ-TREE cho cả tập dữ liệu chúng ta thu được mô hình $Q^{defined}$, cây $T^{defined}$ dùng để sinh dữ liệu.
- Sinh dữ liệu: từ mô hình $Q^{defined}$, cây $T^{defined}$ đã có kết hợp mô hình tốc độ ρ_i và α tìm được ở trên, luận án thực hiện sinh dữ liệu đa sắp hàng D_i với m_i sắp hàng có độ dài l_i ký tự sử dụng chương trình sinh dữ liệu mô phỏng.

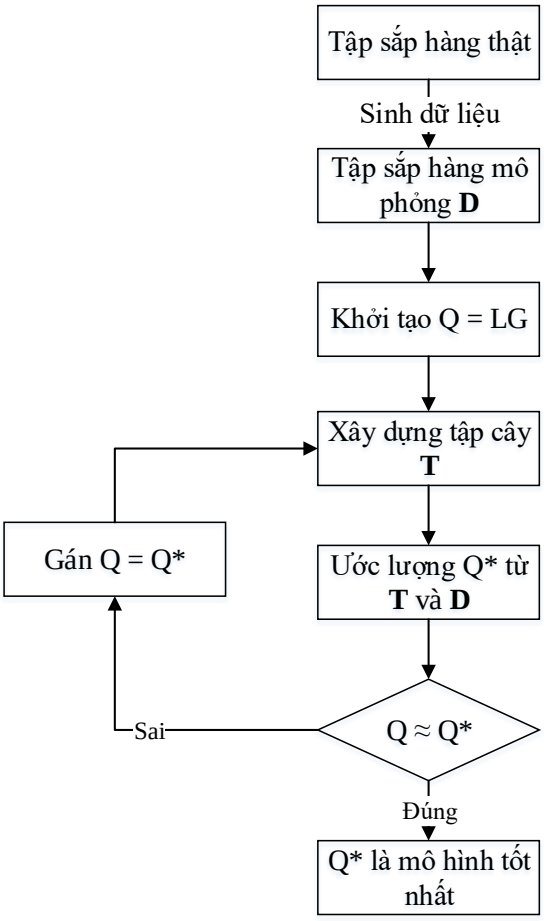
Để tạo một bộ dữ liệu **D** gồm *N* sắp hàng, luận án thực thi quá trình trên *N* lần với *N* sắp hàng thực khác nhau.



Hình 2.1. Quá trình sinh dữ liệu mô phỏng.

2.2.2 Phương pháp ước lượng và đánh giá các mô hình từ dữ liệu mô phỏng

Sau khi có được dữ liệu mô phỏng, luận án tiến hành ước lượng mô hình theo phương pháp cực đại hợp lý. Quá trình này được thể hiện qua lược đồ trong Hình 2.2 với điều kiện dừng đạt được khi giá trị tương quan Pearson giữa hai mô hình tốt nhất hiện tại và mô hình được ước lượng > 0.999, đồng nghĩa với hai mô hình rất gần nhau. Theo đó, với mỗi bộ dữ liệu được sinh theo một kích thước N sắp hàng, luận án sẽ ước lượng được một mô hình tương ứng.



Hình 2.2. Sơ đồ ước lượng mô hình từ dữ liệu mô phỏng.

Gọi Q^{simN} là mô hình được ước lượng từ bộ dữ liệu mô phỏng gồm N sắp hàng, ví dụ Q^{sim100} là mô hình ước lượng từ 100 sắp hàng mô phỏng. Luận án thực hiện đánh giá dựa trên ba tiêu chuẩn sau:

- Tiêu chuẩn 1: đánh giá về mặt giá trị hệ số tốc độ biến đổi thông qua tỉ lệ độ lớn giữa chúng. Ngoài ra, luận án sẽ dùng hệ số tương quan Pearson để so sánh hai mô hình. Việc dùng giá trị tương quan Pearson sẽ cho chúng ta biết liệu hai mô hình có cùng xu hướng thay đổi hay không. Do các ma trận đã được chuẩn hóa nên giá trị tương quan có độ tin cậy cao.

- Tiêu chuẩn 2: so sánh hiệu suất của mô hình mô phỏng Q^{sim} với mô hình đúng $Q^{defined}$ trong việc xây dựng cây phân loài theo tiêu chuẩn cực đại hợp lý. Với một sắp hàng dùng cho kiểm tra (testing alignment), mỗi mô hình sẽ xây dựng được một cây phân loài tương ứng là T^{sim} và $T^{defined}$. Nếu giá trị hợp lý của cây T^{sim} tốt hơn cây $T^{defined}$, ta kết luận rằng mô hình Q^{sim} tốt hơn mô hình $Q^{defined}$ và ngược lại.
- Tiêu chuẩn 3: phân tích đến khía cạnh xây dựng cây phân loài bằng việc so sánh cấu trúc cây sử dụng tiêu chuẩn khoảng cách Robinson-Foulds [51]. Như đã trình bày ở Chương 1, khoảng cách Robinson-Foulds (nRF) thể hiện số lượng nhánh chỉ tồn tại trong một cây trên tổng số nhánh của cả hai cây, khoảng cách nRF có giá trị từ 0 (hai cây đồng nhất) đến 1 (hai cây hoàn toàn khác biệt). Do đã có cây đúng ($T^{defined}$) nên chúng ta có thể đánh giá cấu trúc của cây được xây dựng từ dữ liệu khác biệt thế nào với cây đúng.

2.3 Kết quả

2.3.1 Kết quả sử dụng dữ liệu mô phỏng

Luận án sử dụng phương pháp AliSim [63] để sinh dữ liệu dựa trên các tham số từ dữ liệu sắp hàng thật và thực nghiệm trên hai bộ dữ liệu Plants [43] và Birds [44]. Đây là hai bộ phổ biến, trong đó bộ Plants gồm 1308 sắp hàng, mỗi sắp hàng có 35 trình tự, còn bộ Birds có hơn 6000 sắp hàng và mỗi sắp hàng có 48 trình tự. Luận án dùng các sắp hàng trong các bộ dữ liệu trên để làm cơ sở tạo ra dữ liệu mô phỏng đánh giá phương pháp. Mô hình được sử dụng để tạo dữ liệu là Q.plant và NQ.plant, Q.bird và NQ.bird tương ứng với hai bộ dữ liệu này. Cây phân loài được sử dụng trong phần này được gọi là cây đúng lấy từ hai nghiên cứu tương ứng.

Luận án thực hiện sinh dữ liệu mô phỏng với số lượng sắp hàng lần lượt là {1, 2, 5, 10, 20, 50, 100, 200, 250} sắp hàng. Sự đa dạng kích thước này đảm bảo được việc đánh giá sẽ khách quan và đầy đủ.

2.3.2 Kết quả ước lượng và đánh giá mô hình từ dữ liệu mô phỏng

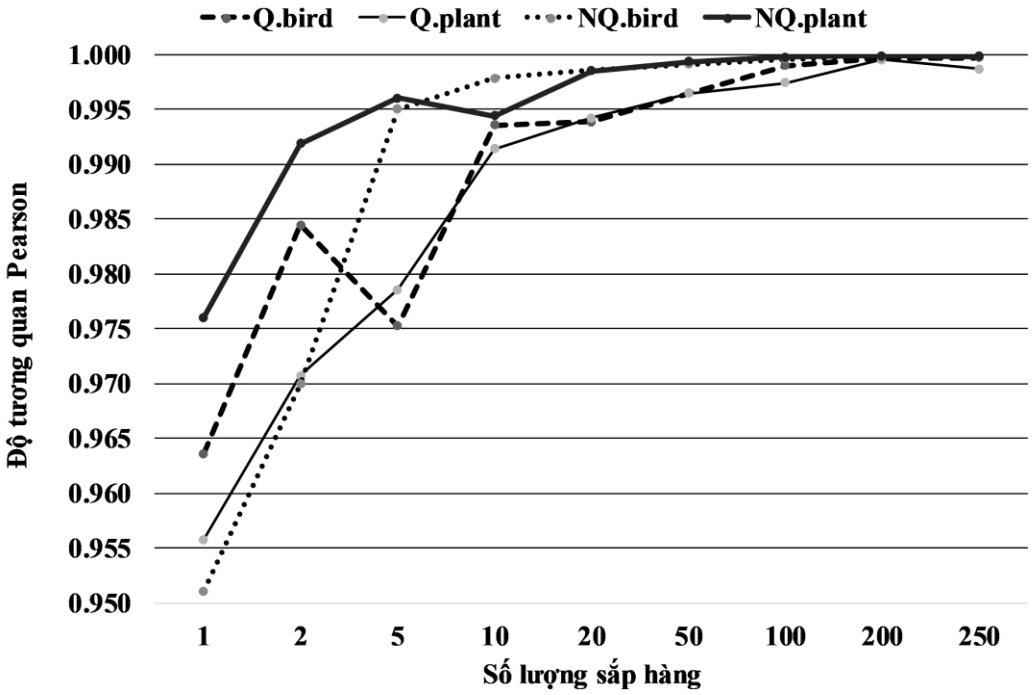
2.3.2.1 Kết quả ước lượng mô hình

Với dữ liệu sau khi sinh ra, các phương pháp QMaker [11] và nQMaker [12] được dùng để ước lượng mô hình thay thế axit amin theo thuộc tính thuận nghịch và không thuận nghịch thời gian. Với mỗi kích thước sắp hàng, năm bộ dữ liệu mô phỏng đã được tạo ra, mỗi bộ sẽ được dùng để ước lượng một mô hình thay thế. Để thuận tiện, những mô hình này được gọi là "mô hình mô phỏng" nghĩa là mô hình được ước lượng từ dữ liệu mô phỏng. Do vậy, luận án sẽ nhận được năm mô hình mô phỏng cho mỗi kích thước dữ liệu đầu vào là một sắp hàng, 10 sắp hàng hay 100 sắp hàng tương ứng. Sau đó, với các mô hình được ước lượng hay mô hình mô phỏng, luận án tiến hành đánh giá hiệu suất của nó với bộ dữ liệu kiểm tra gồm 308 sắp hàng của bộ plant và 500 sắp hàng của bộ bird.

Với hai bộ dữ liệu plant và bird cùng với hai thuộc tính thuận nghịch và không thuận nghịch về thời gian, 9 nhóm kích thước sắp hàng từ 1 sắp hàng đến 250 sắp hàng như đã nêu, mỗi kích thước sinh ra năm bộ dữ liệu khác nhau, tổng cộng chúng ta có 180 bộ dữ liệu để ước lượng 180 mô hình để cho việc phân tích và so sánh. Số lượng mô hình này đủ lớn và đa dạng cho việc phân tích hiệu suất của hai phương pháp trên, các mô hình ước lượng bởi 250 sắp hàng cần khá nhiều thời gian để hoàn thành (hai ngày cho một mô hình theo dữ liệu Plant và ba ngày cho một mô hình theo dữ liệu Bird).

2.3.2.2 Kết quả phân tích mô hình

Với mỗi kịch bản để tạo dữ liệu mô phỏng, giá trị trung bình của độ đo tương quan giữa mô hình đúng và năm mô hình mô phỏng được tính toán và lấy trung bình, kết quả xem tại Hình 2.3. Theo đó, xu hướng của độ tương quan tăng dần theo kích thước của số lượng sắp hàng dùng để ước lượng mô hình, tuy nhiên tất cả đều có độ tương quan mạnh với nhau (>0.95). Ví dụ, với các mô hình ước lượng từ một sắp hàng, các hệ số tương quan giữa NQ^{sim}.bird với NQ.bird thấp nhất đạt 0.951, với các dữ liệu gồm 250 sắp hàng, các mô hình ước lượng từ dữ liệu mô phỏng đã rất gần với mô hình đúng, độ tương quan đều lớn hơn 0.998. Từ 100 sắp hàng trở lên, các mô hình mô phỏng có sự tương quan rất mạnh với mô hình đúng (lớn hơn 0.996).



Hình 2.3. Độ tương quan Pearson giữa các hệ số tốc độ thay thế của mô hình đúng với mô hình mô phỏng theo các kích thước sắp hàng khác nhau.

Kết quả tương quan trên bộ plant và bird đều có xu hướng giống nhau, tăng dần theo số lượng sắp hàng. Chúng ta biết rằng tính toán độ tương quan Pearson là cách đơn giản để biết được sự tương quan tuyến tính của hai ma trận. Tuy nhiên, lưu ý rằng hệ số này phụ thuộc vào khoảng giá trị của các hệ số tính toán. Các giá trị tốc độ lớn thì có ảnh hưởng mạnh đến giá trị cuối cùng. Do vậy, luận án cũng tính độ lệch chuẩn của giá trị tương quan cho từng điều kiện sinh dữ liệu, xem chi tiết trong Bảng 2.1. Kết quả cho thấy rằng với các mô hình ước lượng từ trên 100 sắp hàng thì độ lệch chuẩn rất thấp, gần với 0. Điều này xác nhận sự ổn định của phương pháp Qmaker và nQMaker trong việc ước lượng mô hình với nhiều dữ liệu đầu vào. Càng nhiều dữ liệu thì mô hình lại càng gần với mô hình đúng cả về hiệu suất lẫn các hệ số biến đổi.

Bảng 2.1. Độ lệch chuẩn của hệ số tương quan giữa mô hình mô phỏng và mô hình đúng theo các số sắp hàng khác nhau

Số sắp hàng / Mô hình	1	2	5	10	20	50	100	200	250
Q.plant	0.0110	0.0136	0.0043	0.0032	0.0032	0.0009	0.0011	0.0009	0.0006
NQ.plant	0.0048	0.0082	0.0007	0.0016	0.0004	0.0001	0.0001	0.0001	0.0001
Q.bird	0.0291	0.0105	0.0063	0.0060	0.0024	0.0014	0.0006	0.0003	0.0002
NQ.bird	0.0143	0.0147	0.0023	0.0020	0.0007	0.0002	0.0001	0.0001	0.0001

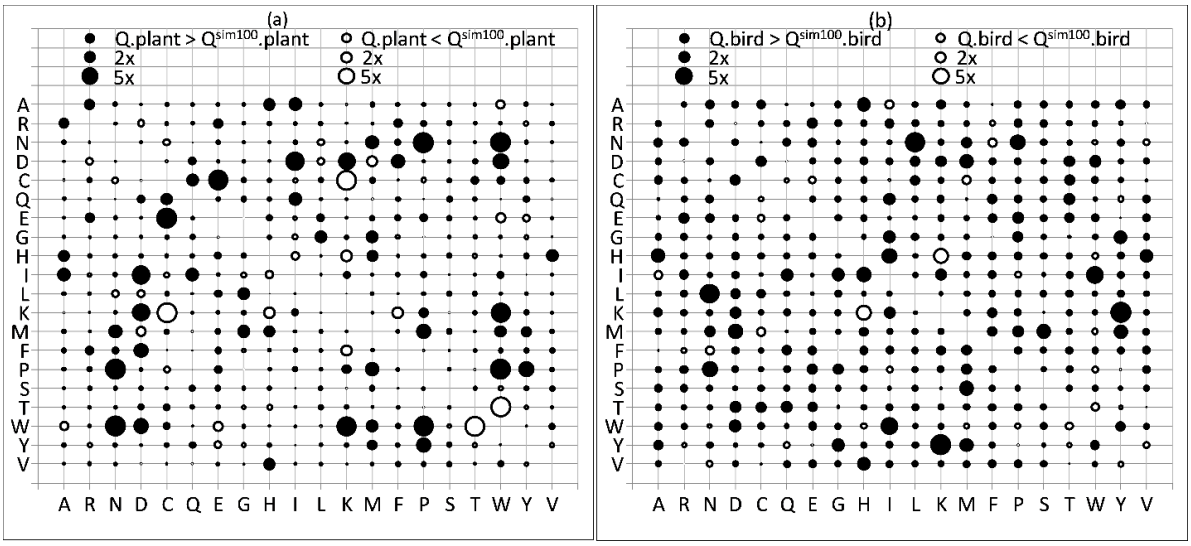
Chi tiết hơn, luận án tiếp tục phân tích trực tiếp các hệ số tốc độ biến đổi xem giữa hai ma trận có sự liên hệ như thế nào. Trước tiên, luận án thực hiện đếm số lượng các hệ số mà chênh lệch tối thiểu hai lần hoặc năm lần (có thể nhỏ hơn hoặc lớn hơn) của mô hình mô phỏng và mô hình đúng. Bảng 2.2 cho ta thấy số lượng chênh lệch này theo các điều kiện sinh dữ liệu khác nhau. Ví dụ, có 29 trên 190 hệ số trong mô hình Q^{sim100}.plant mà lớn hơn ít nhất hai lần so với hệ số của mô hình đúng, và có chín hệ số nhỏ hơn ít nhất hai lần. Với các mô hình tạo từ các bộ dữ liệu ít hơn 100 sắp hàng, luận án nhận thấy số lượng hệ số mà chênh lệch ít nhất hai lần là đáng kể (từ 10 cho tới hơn

210 trăm hệ số), trong khi đó, với 200 hay 250 sắp hàng, chỉ một vài hệ số (dưới 20 hệ số) là khác biệt hai lần hoặc năm lần.

Bảng 2.2. Số lượng trung bình các hệ số tốc độ chuyển đổi chênh lệch hai lần hay năm lần giữa các mô hình mô phỏng với mô hình đúng với thời gian thuận nghịch

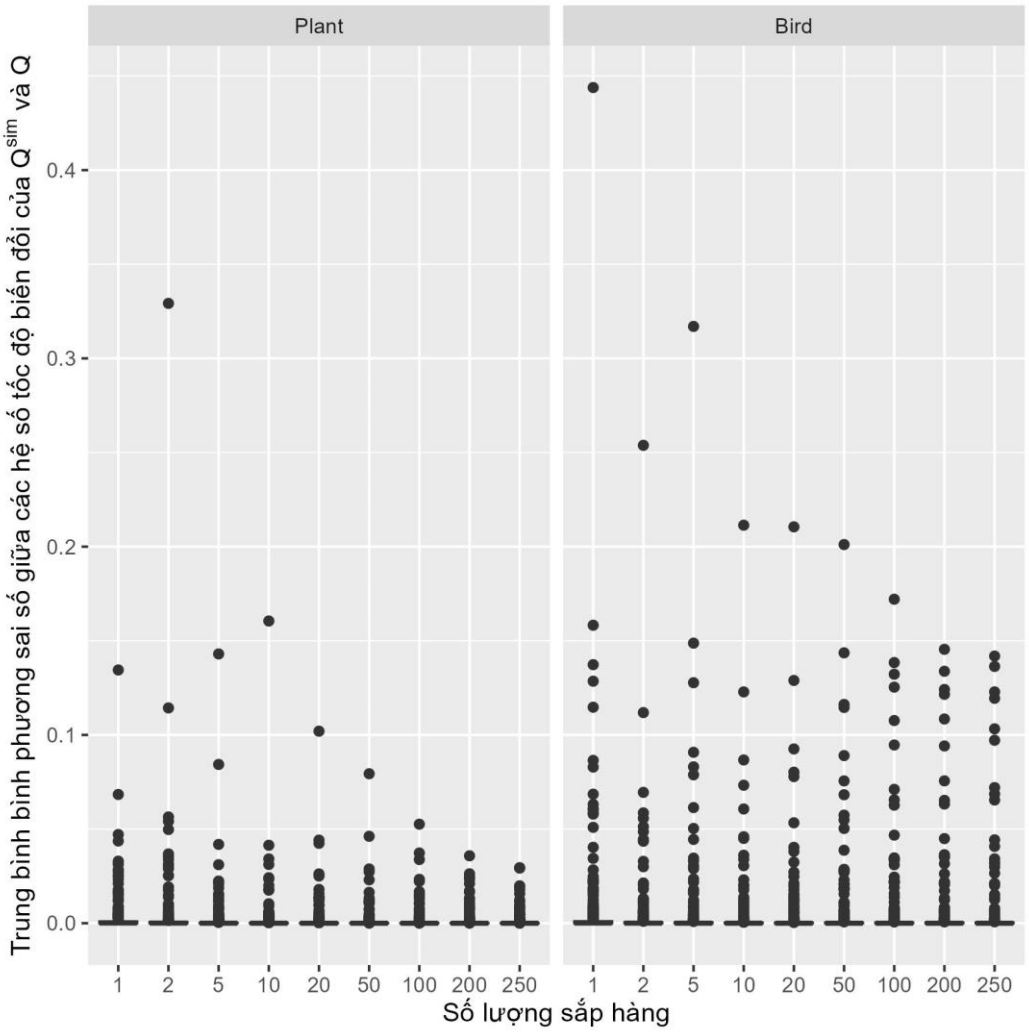
		Số lượng sắp hàng								
Cặp so sánh	**Lớn hơn/ Nhỏ hơn**	**1**	**2**	**5**	**10**	**20**	**50**	**100**	**200**	**250**
Q^{sim}.plant và Q.plant	2x	171.8	154.6	108.6	85.2	56.6	38.8	29.8	18.6	13.2
	5x	150	126	80.4	59.6	34.4	19.8	13	8.4	6.8
	-2x	48	48	32	33	18.8	11.4	9.8	8.8	6.4
	-5x	17.6	14.2	9.8	8.8	4.4	4.6	4.4	4.2	4
Q^{sim}.bird và Q.bird	2x	218.8	148.2	136.6	93.6	69.6	38.8	36.2	18.4	14.4
	5x	199	132.8	109.4	71.6	34.4	12.4	8	3.6	2
	-2x	39.4	49	34.6	27.2	12.8	5.8	0.6	1.8	1.2
	-5x	12.6	9.6	5.6	3.6	0.4	0.4	0	0.4	0

Luận án thể hiện trực quan sự khác biệt này qua biểu đồ bong bóng như trong Hình 2.4. Đây là trường hợp các mô hình mô phỏng được ước lượng từ 100 sắp hàng khi so sánh với mô hình đúng $Q^{defined}$. Độ chênh lệch được thể hiện qua độ lớn của bong bóng, bong bóng to nhất thể hiện sự khác biệt năm lần giữa các hệ số biến đổi tốc độ. Chỉ một vài bong bóng có kích thước 5x xuất hiện trong hình Hình 2.4b của mô hình cho dữ liệu bird.



Hình 2.4. Độ lệch các hệ số tốc độ thay thế giữa mô hình đúng và mô hình mô phỏng tạo từ dữ liệu 100 sắp hàng. Chú thích: 2x (5x) có ý nghĩa các hệ số tốc độ biến đổi khác biệt ít nhất hai lần hoặc năm lần.

Chi tiết hơn nữa về sự khác biệt này, luận án sử dụng giá trị trung bình bình phương sai số (mean squared error – MSE) để xem xét độ lệch giữa các cặp hệ số biến đổi tốc độ giữa các mô hình mô phỏng và mô hình đúng. Kết quả trong Hình 2.5 cho thấy rằng giá trị MSE cao nhất là 0.443 cho các mô hình bird và 0.329 cho các mô hình plant. Giá trị trung bình của tất cả các giá trị MSE với 100 sắp hàng là 0.00076 cho các mô hình bird và 0.00056 cho các mô hình plant.



Hình 2.5. Trung bình bình phương sai số giữa các hệ số tốc độ biến đổi giữa mô hình đúng Q và mô hình mô phỏng Q^{sim}.

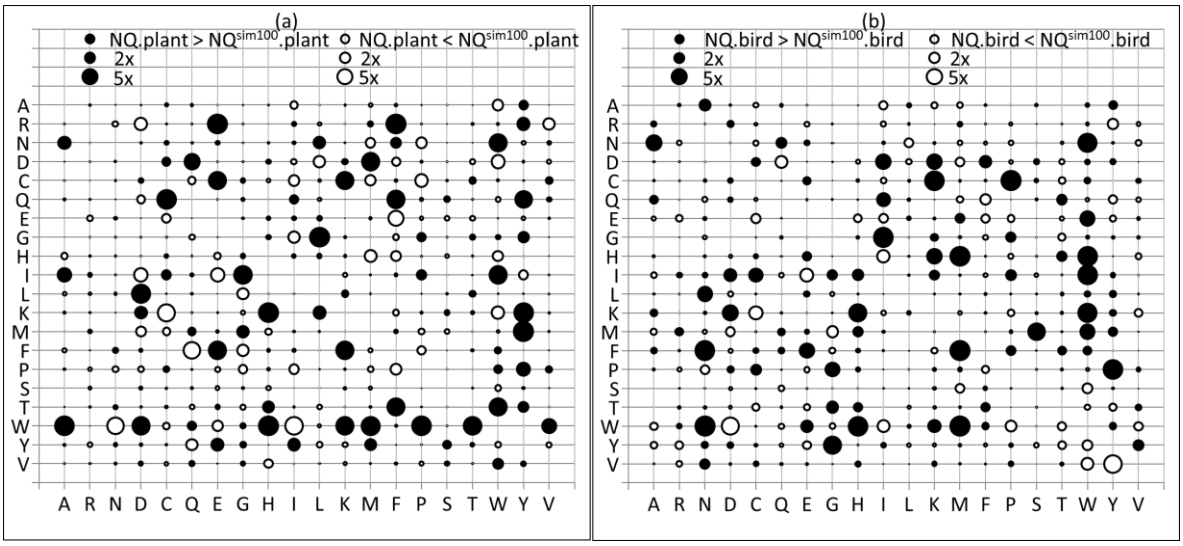
Theo cách tương tự, luận án thực hiện quá trình phân tích với các mô hình có thuộc tính không thuận nghịch theo thời gian NQ. Luận án so sánh các mô hình NQ^{sim}.plant và NQ^{sim}.bird với các mô hình đúng tương ứng là NQ.plant và NQ.bird, xem Bảng 2.3. Do mô hình NQ có nhiều gấp hai lần số lượng hệ số so với mô hình Q, nên chúng ta thấy kết quả cũng tăng lên rõ rệt. Ví dụ, mô hình NQ^{sim100}.plant có 39 hệ số mà khác biệt hai lần so với NQ.plant, mô hình NQ^{sim100}.bird có 30 hệ số mà khác biệt hai lần so với mô

hình NQ.bird. Rõ ràng, với mô hình không thuận nghịch chứa nhiều tham số hơn, chúng ta cần phải ước lượng từ lượng dữ liệu lớn hơn so với mô hình thuận nghịch.

Bảng 2.3. Số lượng trung bình các hệ số tốc độ biến đổi chênh lệch hai lần hay năm lần giữa các mô hình mô phỏng với mô hình đúng với thuộc tính thời gian không thuận nghịch

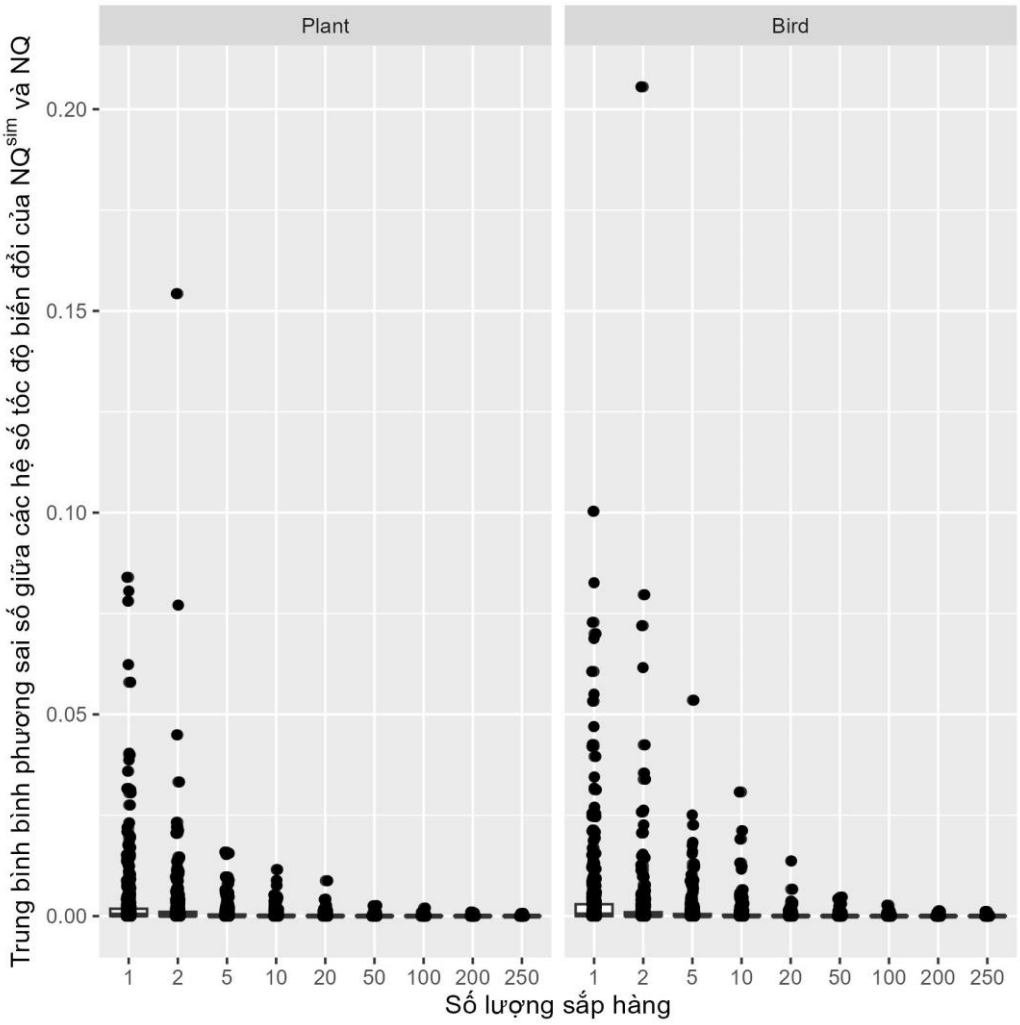
		Số lượng sắp hàng								
Cặp so sánh	**Lớn hơn /Nhỏ hơn**	**1**	**2**	**5**	**10**	**20**	**50**	**100**	**200**	**250**
NQ^{sim}.plant và NQ.plant	2x	210	178.6	126	106.8	74.8	49.6	39.2	25	18.8
	5x	196.2	164.8	110.2	88.2	60.2	37.2	25.4	18.8	14
	-2x	51.4	54.4	52.4	49.4	43	28.4	23.2	17.6	17.2
	-5x	22.6	22.6	14	12.6	12.2	9.6	4	4	3.6
NQ^{sim}.bird và NQ.bird	2x	244.2	182.4	146.8	106	81.6	48.2	30.6	15	15.6
	5x	236.2	171	134.4	92.4	61.6	28.4	15.4	7	8.2
	-2x	41.8	53.6	57.6	56.8	41	18.6	11.2	7.2	7.6
	-5x	19.6	18.8	16.2	13.6	5.6	2	2	1.8	1.2

Hình 2.6 diễn tả sự khác biệt giữa mô hình đúng NQ.plant với mô hình ước lượng từ 100 sắp hàng NQ^{sim100}.plant (2.6a) và mô hình đúng NQ.bird với mô hình ước lượng từ 100 sắp hàng NQ^{sim100}.bird (2.6b). Trong đó, sự biến đổi từ axit amin W (Tryptophan) sang các axit amin khác có độ lệch hệ số lớn hơn các cặp biến đổi khác thuộc bộ plant (8 hệ số lệch 5x, ví dụ: W (Tryptophan) biến đổi thành A (Alanine), N (Asparagine), D (Aspartic)), ngược lại với bộ bird thì dường như biến đổi đến W lại có nhiều hệ số độ lệch 5x hơn cả (bốn hệ số 5x, ví dụ: N (Alanine), H (Histidine), I (Isoleucine) biến đổi thành W (Tryptophan)).



Hình 2.6. Độ lệch các hệ số tốc độ biến đổi giữa mô hình đúng và mô hình mô phỏng tạo từ dữ liệu 100 sắp hàng. Chú thích: 2x (5x) có ý nghĩa các hệ số tốc độ biến đổi khác biệt ít nhất hai lần hoặc năm lần.

Luận án cũng khảo sát phân bố MSE của các hệ số tốc độ biến đổi, các giá trị được tính từ cả năm mô hình ước lượng từ năm lần mô phỏng cho mỗi kích thước dữ liệu, xem Hình 2.7. MSE của các mô hình không thuận nghịch nhỏ hơn so với các mô hình thuận nghịch ở trên. Toàn bộ giá trị MSE đều nhỏ hơn 0.205 và gần với 0 khi kích thước sắp hàng từ 100 trở lên. Với các mô hình không thuận nghịch, phân bố các giá trị giảm đều theo kích thước. Chỉ một giá trị ngoại lệ lớn hơn 0.2 xảy ra với bộ bird khi mô hình được ước lượng từ hai sắp hàng.

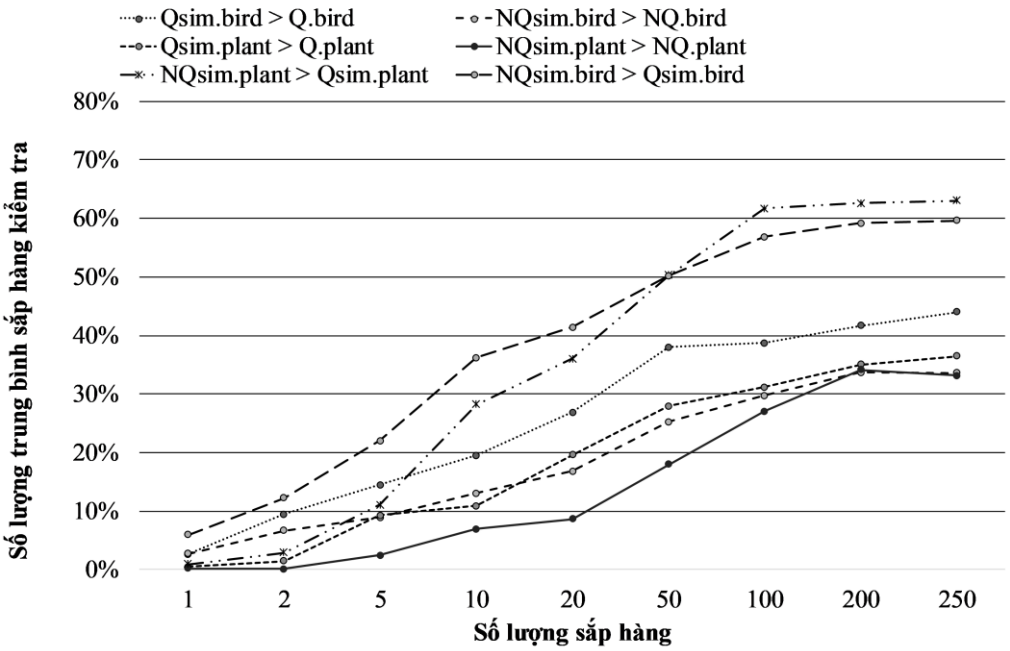


Hình 2.7. Trung bình bình phương sai số (MSE) giữa các hệ số tốc độ biến đổi giữa mô hình đúng NQ và mô hình mô phỏng NQ^{sim}.

2.3.2.3 Đánh giá khả năng xây dựng cây cực đại hợp lý

Trong phần này, luận án sử dụng giá trị hợp lý (được lấy logarit) để kiểm tra hiệu quả của các mô hình trong việc xây dựng cây theo tiêu chuẩn cực đại hợp lý. Với mỗi sắp hàng, mô hình A tốt hơn mô hình B nếu cây phân loài được xây dựng bởi mô hình A có giá trị hợp lý cao hơn cây phân loài được xây dựng bởi mô hình B. Ở đây, luận án đã so sánh các mô hình mô phỏng với mô hình đúng để xem độ tốt của mô hình mô

phỏng thay đổi ra sao theo các kích thước sắp hàng, và cũng đánh giá hiệu suất của các mô hình mô phỏng với nhau, chi tiết xem Hình 2.8.



Hình 2.8. Hiệu suất của các mô hình trong việc xây dựng cây theo tiêu chuẩn cực đại hợp lý với số lượng sắp hàng mô phỏng khác nhau.

Kết quả trên Hình 2.8 cho thấy rằng hầu hết cây phân loài xây dựng với mô hình đúng cho giá trị hợp lý cao hơn cây được xây dựng với mô hình mô phỏng. Giá trị hợp lý của cây tạo với mô hình mô phỏng tốt dần theo số lượng sắp hàng được dùng để ước lượng mô hình. Ví dụ, cây được xây dựng với mô hình đúng Q.plant có giá trị hợp lý cao hơn giá trị hợp lý của cây xây dựng với mô hình Q^{sim10}.plant, Q^{sim50}.plant và Q^{sim100}.plant tương ứng là 89.9%, 72.1% và 68.8%. Xu hướng của mô hình ước lượng từ bộ plant và bird là như nhau, độ tốt tăng dần theo số lượng sắp hàng. Tuy nhiên, các mô hình đúng vẫn cho ra các cây tốt hơn trên 50% số đa sắp hàng trong trường hợp mô hình mô phỏng tốt nhất (được ước lượng từ 250 sắp hàng). Ngoài ra, khi so sánh giữa mô hình thuận nghịch với mô hình không thuận nghịch, luận án nhận thấy với kích thước dữ liệu nhỏ

(<50 sắp hàng) thì mô hình thuận nghịch tốt hơn mô hình không thuận nghịch. Nhưng mô hình không thuận nghịch tốt hơn khi dữ liệu tăng dần, ví dụ với 100 sắp hàng, mô hình NQ^{sim100}.bird tốt hơn Q^{sim100}.bird ở 56.8% số sắp hàng kiểm tra còn NQ^{sim100}.plant tốt hơn Q^{sim100}.plant ở 61.69% số sắp hàng.

2.3.2.4 Đánh giá ảnh hưởng của mô hình đến cấu trúc cây

Đến đây, sau khi đã so sánh các mô hình tạo ra từ hai phương pháp QMaker và nQMaker thông qua các hệ số biến đổi tốc độ cũng như khả năng xây dựng cây cực đại hợp lý, luận án tiếp tục đánh giá chi tiết hơn về hình dạng của cây được xây dựng bởi các mô hình này. Luận án tiến hành tính khoảng cách Robinson-Foulds (RF) giữa các cây tạo bởi mô hình đúng, mô hình mô phỏng và cây thực (real tree). Kết quả chi tiết được thể hiện trong Bảng 2.4 thể hiện rằng hầu hết các cây được xây dựng bởi mô hình ước lượng từ 20 sắp hàng là đồng nhất với cây được tạo bởi mô hình đúng (true model). Trong cả hai bộ dữ liệu plant và bird, với kích thước từ 100 sắp hàng, các giá trị RF tiến sát về 0 hoặc bằng 0.

Bảng 2.4. Khoảng cách Robinson-Foulds giữa các cây tạo ra bởi mô hình đúng và mô hình mô phỏng theo các kích thước sắp hàng khác nhau. Mô hình với thuộc tính thuận nghịch theo thời gian

Số sắp hàng / nRF	**1**	**2**	**5**	**10**	**20**	**50**	**100**	**200**	**250**
Q.plant vs. Q^{sim}.plant	0	0.012	0.006	0	0	0	0	0	0
NQ.plant vs. NQ^{sim}.plant	0.018	0.012	0.012	0.006	0	0	0	0	0
Q.bird vs. Q^{sim}.bird	0.020	0	0.106	0.004	0.004	0	0	0	0
NQ.bird vs. NQ^{sim}.bird	0.120	0.04	0.238	0.048	0.088	0	0.035	0.09	0.039
Q^{sim}.plant vs. plant tree	0.031	0.018	0.025	0.031	0.031	0.031	0.031	0.031	0.031
NQ^{sim}.plant vs. plant tree	0.018	0.018	0.025	0.031	0.031	0.031	0.031	0.031	0.031
Q^{sim}.bird vs. bird tree	0.556	0.529	0.556	0.556	0.556	0.556	0.556	0.556	0.556
NQ^{sim}.bird vs. bird tree	0.507	0.511	0.409	0.538	0.528	0.538	0.542	0.451	0.416

Khi so sánh với cây thật (real tree), luận án cũng ghi nhận sự trái ngược giữa dữ liệu Plant và Bird. Trong khi kết quả RF của các mô hình ước lượng từ dữ liệu Plant vẫn tốt (trong khoảng 0.01-0.03), gần với 0 thì kết quả RF của các mô hình từ dữ liệu Bird lại khá lớn (trong khoảng 0.4-0.5). Hiện tượng này là do dữ liệu của bộ bird với 48 trình tự lớn hơn và phức tạp hơn so với bộ plant chỉ gồm 35 trình tự, ngoài ra dữ liệu bird có tới 11% là ký tự trống khiến cho nó có độ ổn định thấp hơn.

Luận án cũng khảo sát trung bình độ dài cạnh của các cây được xây dựng bởi các mô hình mô phỏng, kết quả chi tiết tại Bảng 2.5. Trung bình cạnh của cây thực (real tree) là 0.051 trong khi độ dài cạnh trung bình của cây được xây dựng bởi mô hình mô phỏng nằm trong khoảng 0.061 đến 0.065. Cụ thể, trung bình độ dài cạnh của cây tạo từ Q^{sim100}.plant và NQ^{sim100}.plant là 0.062 và 0.061. Với bộ bird, trung bình cạnh của cây thực là 0.017, còn trung bình cạnh của cây tạo với mô hình Q^{sim100}.bird và NQ^{sim100}.bird đều bằng 0.027. Nhìn chung, dữ liệu ước lượng càng lớn (nhiều sắp hàng) thì mô hình càng tốt trong việc ước lượng độ dài cạnh. Với các bộ dữ liệu từ 100 sắp hàng trở lên, mô hình mô phỏng xây dựng các cây có độ dài cạnh tương đương nhau.

Bảng 2.5. Trung bình độ dài cạnh của các cây xây dựng với mô hình mô phỏng theo các kích thước khác nhau

Số sắp hàng	**1**	**2**	**5**	**10**	**20**	**50**	**100**	**200**	**250**
Q^{sim}.plant	0.065	0.064	0.063	0.062	0.062	0.062	0.062	0.062	0.062
NQ^{sim}.plant	0.064	0.063	0.062	0.061	0.061	0.061	0.061	0.061	0.061
Q^{sim}.bird	0.030	0.028	0.028	0.027	0.027	0.027	0.027	0.027	0.027
NQ^{sim}.bird	0.032	0.029	0.027	0.027	0.027	0.027	0.027	0.026	0.027

Ngoài ra, luận án cũng khảo sát các giá trị MSE và SD của độ dài cạnh của các cây được xây dựng bởi mô hình mô phỏng khi so sánh với cây đúng. Kết quả trong Bảng 2.6 cho thấy các giá trị MSE tương đối nhỏ và giảm dần theo kích thước sắp hàng. Xu

hướng chung này đúng cho cả hai bộ plant và bird. Ví dụ, trung bình của MSE cho Q^{sim1}.plant là 0.001 trong khi của Q^{sim100}.plant là 0.0008.

Bảng 2.6. Trung bình bình phương sai số - MSE（× 1000) và độ lệch chuẩn - SD（× 1000) của các giá trị độ dài cạnh của các cây xây dựng từ mô hình mô phỏng với cây đúng

Số lượng sắp hàng	**Q^{sim}.plant**		**Q^{sim}.bird**		**NQ^{sim}.plant**		**NQ^{sim}.bird**	
	MSE	**SD**	**MSE**	**SD**	**MSE**	**SD**	**MSE**	**SD**
1	1.003	0.120	2.218	0.310	1.033	0.124	2.044	0.542
2	0.974	0.133	1.989	0.167	0.877	0.056	1.913	0.338
5	0.893	0.032	2.219	0.312	0.827	0.036	1.836	0.357
10	0.845	0.009	1.901	0.130	0.811	0.012	1.789	0.115
20	0.856	0.018	1.773	0.103	0.806	0.004	1.916	0.568
50	0.851	0.014	1.774	0.130	0.795	0.005	1.727	0.053
100	0.848	0.009	1.844	0.114	0.795	0.004	1.840	0.135
200	0.853	0.005	1.775	0.133	0.795	0.004	1.945	0.234
250	0.848	0.003	1.735	0.106	0.793	0.004	1.829	0.209

2.4 Tổng kết chương

Tổng kết lại, trong chương này, luận án đã trình bày một quy trình từ sinh dữ liệu, ước lượng và đánh giá mô hình với dữ liệu mô phỏng. Như đã thảo luận, mô hình đúng cho dữ liệu thật là chưa biết, do vậy, chúng ta không thể đánh giá chất lượng của mô hình mô phỏng với mô hình đúng dựa trên dữ liệu thật. Do đó, luận án thực hiện đánh giá bằng dữ liệu mô phỏng với các kích thước khác nhau.

Các thực nghiệm đã cho thấy, dữ liệu càng nhiều thì mô hình càng tốt trên cả khía cạnh xây dựng cây cực đại hợp lý lẫn cấu trúc của cây phân loài. Việc ước lượng mô hình luôn tiêu thụ nhiều tài nguyên vật lý và cần nhiều thời gian để hoàn thành. Để đảm bảo chất lượng mô hình, các bộ dữ liệu cần có tối thiểu từ 100 sắp hàng. Với kết

quả về MSE và SD, phương pháp ước lượng như QMaker và nQMaker đã cho thấy độ ổn định và đáng tin cậy, do đó, chúng ta chỉ cần sinh một bộ dữ liệu duy nhất để cho việc ước lượng mô hình được nhanh chóng và hiệu quả.

Chương 3. Ước lượng mô hình thay thế axit amin sử dụng đa ma trận

Chương này giới thiệu một phương pháp mới, gọi là QMix, ước lượng mô hình thay thế axit amin sử dụng nhiều ma trận tốc độ biến đổi khác nhau sử dụng tiêu chuẩn cực đại hợp lý với tính chất thời gian không thuận nghịch. Luận án thực nghiệm phương pháp mới trên hai bộ dữ liệu là HSSP và Plant và đề xuất các mô hình thay thế mới sử dụng bốn ma trận tốc độ khác nhau. Luận án đánh giá và so sánh mô hình mới với các mô hình đang tồn tại trên các bộ dữ liệu thật.

3.1 Giới thiệu chung

Mô hình đơn ma trận như WAG, JTT hay LG có lợi thế là đơn giản và có thể nhanh chóng xây dựng cây phân loài. Nhưng nhược điểm của những mô hình này là coi tốc độ biến đổi tại các vị trí như nhau, điều này là không phù hợp với thực tế. Để giải quyết vấn đề này, nhiều nhà nghiên cứu đã ước lượng các mô hình hỗn hợp như: sử dụng nhiều ma trận biến đổi khác nhau [37, 65] hay sử dụng nhiều bộ tần số khác nhau [66, 67] để mô hình hóa sự biến đổi trên chuỗi cũng như xây dựng cây phân loài sát với thực tế hơn.

Cho đến hiện tại, nhiều mô hình đa ma trận đã được giới thiệu như EX2, CF4, UL3 hay gần đây là LG4X, LG4M [37, 66, 67]. Các mô hình đa ma trận như LG4X, LG4M được sử dụng rộng rãi do đạt hiệu suất tốt. Tuy nhiên, các mô hình này đều được ước lượng dựa trên thuộc tính thuận nghịch về mặt thời gian nên không thể xây dựng cây phân loài có gốc. Ngoài ra, các mô hình đa ma trận đang tồn tại được ước lượng dựa trên quy trình phức tạp và chưa có phần mềm nào hỗ trợ. Vì vậy, bài toán cần giải quyết trong chương này được phát biểu như sau đây:

Đầu vào: tập dữ liệu sắp hàng **D** gồm N sắp hàng.

Bài toán: phát triển phương pháp ước lượng mô hình thay thế sử dụng nhiều ma trận tốc độ biến đổi, kết hợp sử dụng các tính chất về mặt sinh học như tốc độ biến đổi khác nhau tại các vị trí và thuộc tính không thuận nghịch, nhằm giải thích tốt nhất tập dữ liệu **D**.

Đầu ra: Phương pháp ước lượng và các mô hình thay thế đa ma trận tốc độ biến đổi.

Từ đó, luận án sẽ trình bày phương pháp ước lượng các mô hình đa ma trận, được gọi là QMix. Để minh chứng cho sự hiệu quả của phương pháp, luận án áp dụng QMix trên các bộ dữ liệu khác nhau để ước lượng các mô hình mới. Đầu tiên, luận án sử dụng bộ dữ liệu HSSP và ước lượng hai mô hình dùng chung mới, gọi là nT4X, nT4M, dựa trên thuộc tính không thuận nghịch về thời gian. Tiếp đó, luận án cũng thực nghiệm QMix trên bộ dữ liệu riêng cho các loài thực vật và giới thiệu các mô hình đa ma trận riêng là QPlant.mix và nQPlant.mix. Các mô hình mới đều có bốn ma trận đại diện cho bốn phân lớp tốc độ là "rất chậm", "chậm", "bình thường" và "nhanh".

3.2 Phương pháp

3.2.1 Mô hình thay thế axit amin sử dụng đa ma trận

Như luận án đã giới thiệu ở Chương 1, tốc độ biến đổi tại các vị trí là không đồng nhất và thường được mô hình hóa bởi một phân phối, thông thường là phân phối gamma rời rạc với C lớp tốc độ khác nhau [48]. Theo đó, công thức tính giá trị hợp lý của mô hình Q và cây T đối với đa sắp hàng D có l vị trí như sau:

$$L(Q,T|\mathrm{D}) = \prod_{i=1}^{l} L(Q,T|D_i) \tag{3.1}$$

Yang [48] đã giới thiệu mô hình hỗn hợp dựa trên một ma trận thay thế nhưng tốc độ biến đổi tại các vị trí tuân theo phân phối gamma rời rạc với C phân lớp có trọng số bằng nhau. Công thức tính giá trị hợp lý theo mô hình của Yang có dạng sau:

$$L(Q,T,\alpha|D) = \prod_{i=1}^{l}\left(\frac{1}{C}\sum_{k=1}^{C} L(\Gamma(\alpha,k)Q,T|D_i)\right) \quad (3.2)$$

trong đó, $\Gamma(\alpha,k)$ là tốc độ thứ *k* của phân phối gamma với tham số hình dạng α. Trọng số của các phân lớp tốc độ đều bằng 1/C.

Tổng quát hóa, luận án áp dụng một phân phối tốc độ với C nhóm tốc độ $\mathbf{P} = \{\rho_1, \rho_2, \ldots, \rho_C\}$ và có trọng số tương ứng $\mathbf{R} = \{r_1, r_2, r_3, .., r_C\}$ vào Công thức 3.2 để thu được công thức mới chứa các thành phần tốc độ như sau:

$$L(\mathrm{Q},T,\mathbf{P},\mathbf{R}|D) = \prod_{i=1}^{l}\left(\sum_{k=1}^{C} r_k L(\rho_k \mathrm{Q},T|D_i)\right) \quad (3.3)$$

trong đó, ρ_k là tốc độ thứ *k* của phân phối tốc độ với trọng số tương ứng là r_k.

Một số nghiên cứu đã đề xuất mô hình đa ma trận [68-70] với *M* ma trận $\mathbf{Q} = \{Q_1, Q_2, \ldots, Q_M\}$, và các trọng số tương ứng ký hiệu là $\boldsymbol{W} = \{w_1, w_2, \ldots, w_M\}$ với điều kiện cho trước $\sum_m w_m = 1$.

$$\begin{aligned} L(\mathbf{Q} = \{Q_1, Q_2, \ldots, Q_M\}, T, \boldsymbol{W} = \{w_1, w_2, \ldots, w_M\}|D) \\ = \prod_{i=1}^{l}\left(\sum_{m=1}^{M} w_m L(Q_m, T|D_i)\right) \end{aligned} \quad (3.4)$$

Kết hợp Công thức 3.3 và 3.4, chúng ta có mô hình đa ma trận kết hợp với mô hình tốc độ biến đổi tổng quát như sau:

$$L\left(\begin{matrix}\mathbf{Q}=\{Q_1,Q_2,..,Q_M\},\mathbf{W}=\{w_1,w_2,..,w_M\},\\ \mathbf{P}=\{\rho_1,\rho_2,\ldots,\rho_C\},\mathbf{R}=\{r_1,r_2,..,r_C\},T\end{matrix}\middle|D\right) = \prod_{i=1}^{l}\left(\sum_{m=1}^{M} w_m \sum_{k=1}^{C} r_k L(\rho_k Q_m, T|D_i)\right) \quad (3.5)$$

ở đây, **Q** là tập các ma trận, **W** là các trọng số của các ma trận, **P** là tập các phân lớp tốc độ còn **R** là trọng số tương ứng của các phân lớp tốc độ.

Từ công thức tổng quát này, áp dụng cho các mô hình tốc độ khác nhau các nhà khoa học đã ước lượng một số mô hình thay thế axit amin. Cụ thể, khi sử dụng phân phối gamma gồm C=4 phân lớp tốc độ có trọng số bằng nhau (=1/4), các tác giả Lê và Gascuel [65, 66] đã ước lượng một số mô hình đa ma trận như EX2 (gồm hai ma trận), UL3 (gồm ba ma trận) dựa trên công thức như sau:

$$L(\mathbf{Q}=\{Q_1,Q_2,..,Q_M\},\mathbf{W}=\{w_1,w_2,..,w_M\},T,\alpha|D) = \prod_{i=1}^{l}\left(\sum_{m=1}^{M} \frac{w_m}{C} \sum_{k=1}^{C} L(\Gamma(\alpha,k)Q_m, T|D_i)\right) \quad (3.6)$$

Với $C = M = 4$ đồng thời mỗi phân lớp tốc độ chỉ áp dụng cho một ma trận, các nhà khoa học đã ước lượng mô hình LG4M bao gồm bốn ma trận tương ứng với bốn phân lớp tốc độ có trọng số bằng nhau [37] dựa trên công thức như sau:

$$L(\mathbf{Q}=\{Q_1,Q_2,Q_3,Q_4\},T,\alpha|D) = \prod_{i=1}^{l}\left(\frac{1}{4}\sum_{k=1}^{4} L(\Gamma(\alpha,k)Q_k, T|D_i)\right) \quad (3.7)$$

trong đó, các phân lớp tốc độ của phân phối gamma đều có trọng số bằng nhau (=1/4).

Áp dụng công thức tổng quát 3.5 với $C = M = 4$, và cũng như trên, mỗi phân lớp tốc độ được áp dụng cho một ma trận, chúng ta thu được công thức cho mô hình đa ma trận với tốc độ tuân theo phân phối tự do như sau:

$$L(\mathbf{Q}, T, \mathbf{P} = \{\rho_1, \rho_2, \rho_3, \rho_4\}, \mathbf{W} = \{w_1, w_2, w_3, w_4\}|D) = \prod_{i=1}^{l}\left(w_k \sum_{k=1}^{4} L(\rho_k Q_k, T|D_i)\right) \quad (3.8)$$

ở đây, mỗi ma trận được gán trọng số w_k và tốc độ ρ_k; các tham số w_k, ρ_k được ước lượng từ dữ liệu và thỏa mãn hai điều kiện $\sum_{k=1}^{M} w_k = 1$ và $\sum_{k=1}^{M} w_k \rho_k = 1$. Dựa trên Công thức 3.8, các nhà khoa học đã ước lượng mô hình LG4X [37] gồm bốn ma trận. Trong chương này, dựa trên các Công thức 3.7 và 3.8, luận án cũng ước lượng các mô hình thay thế axit amin sử dụng bốn ma trận với các phân lớp tốc độ tương ứng là "rất chậm", "chậm", "bình thường" và "nhanh". Phương pháp ước lượng được trình bày trong phần tiếp theo sau đây.

3.2.2 Phương pháp QMix ước lượng mô hình thay thế axit amin sử dụng đa ma trận

Giả sử chúng ta có một tập *N* các sắp hàng ký hiệu là $\mathbf{D} = \{D^1, D^2, \ldots, D^N\}$ và chúng ta cần ước lượng mô hình gồm *M* ma trận $\mathbf{Q}^* = \{Q_1^*, Q_2^*, .., Q_M^*\}$ sao cho giá trị hợp lý đạt cực đại:

$$\mathbf{Q}^* = \underset{\mathbf{Q}=\{Q_1,Q_2,..,Q_M\},T,P,W}{\operatorname{argmax}} \left\{\prod_{n=1}^{N} L(\mathbf{Q}, T^n, \rho^n, w^n|D^n)\right\} \quad (3.9)$$

ở đây, D^n là sắp hàng thứ *n* trong tập dữ liệu **D**, còn T^n, ρ^n, w^n tương ứng là cây phân loài, mô hình tốc độ biến đổi tại các vị trí và trọng số của sắp hàng D^n, còn $L(\mathbf{Q}, T^n, \rho^n, w^n|D^n)$ là giá trị hợp lý của mô hình **Q**, cây T^n, mô hình tốc độ ρ^n, trọng số w^n đối với sắp hàng D^n.

Để ước lượng được ma trận $\mathbf{Q}^*$ chúng ta phải tối ưu cùng lúc **Q**, **T**, **P**, và **W**, đây là một thách thức tính toán rất lớn. Tuy nhiên, theo các nghiên cứu đã công bố [7, 11, 12, 37, 41], chúng ta có thể tối ưu từng phần một cách tuần tự để thu được mô hình $\mathbf{Q}^*$. Cụ thể, quá trình tối ưu sẽ gồm hai bước chính như sau:

- Bước 1: Cố định các ma trận Q_k, tối ưu T^n, ρ^n, w^n bằng phương pháp cực đại hợp lý theo Công thức 3.7 cho tốc độ theo phân phối gamma hoặc Công thức 3.8 cho mô hình tốc độ tự do. Giả sử thu được T^*, ρ^*, w^*.
- Bước 2: Sau khi có T^*, ρ^*, w^*, thực hiện tối ưu **Q** theo Công thức 3.9 để thu được $\mathbf{Q}^*$.

Lặp lại hai bước này cho tới khi $\mathbf{Q}^*$ không tối ưu thêm được nữa thì dừng.

Do tính độc lập của các thành phần tốc độ, trọng số và cây phân loài nên chúng ta hoàn toàn có thể tối ưu T^n, ρ^n, w^n cho từng sắp hàng D^n riêng biệt rồi tổng hợp để có được kết quả cuối cùng là mô hình $\mathbf{Q}^*$. Ước lượng $\mathbf{Q}^*$ gồm M ma trận cần tối ưu một lượng lớn tham số ($M \times 208$ tham số cho thuộc tính thuận nghịch theo thời gian hoặc $M \times 379$ tham số cho thuộc tính không thuận nghịch theo thời gian). Nhóm tác giả Lê và cộng sự [37] đã đề xuất phương pháp xấp xỉ để tối ưu hóa quá trình ước lượng $\mathbf{Q}^*$ bằng cách sử dụng phân lớp tốc độ có xác suất lớn nhất. Như vậy $\mathbf{Q}^*$ sẽ được ước lượng theo công thức:

$$\mathbf{Q}^* = (Q_1^*, Q_2^*, \dots, Q_M^*) = \underset{\mathbf{Q}=\{Q_1,Q_2,\dots,Q_M\},T,P,W}{\operatorname{argmax}} \left\{ \prod_{n=1}^{N} \prod_{i=1}^{l} L(T^n, \rho_{c_i}^n Q_{c_i} | D_i^n) \right\} \tag{3.10}$$

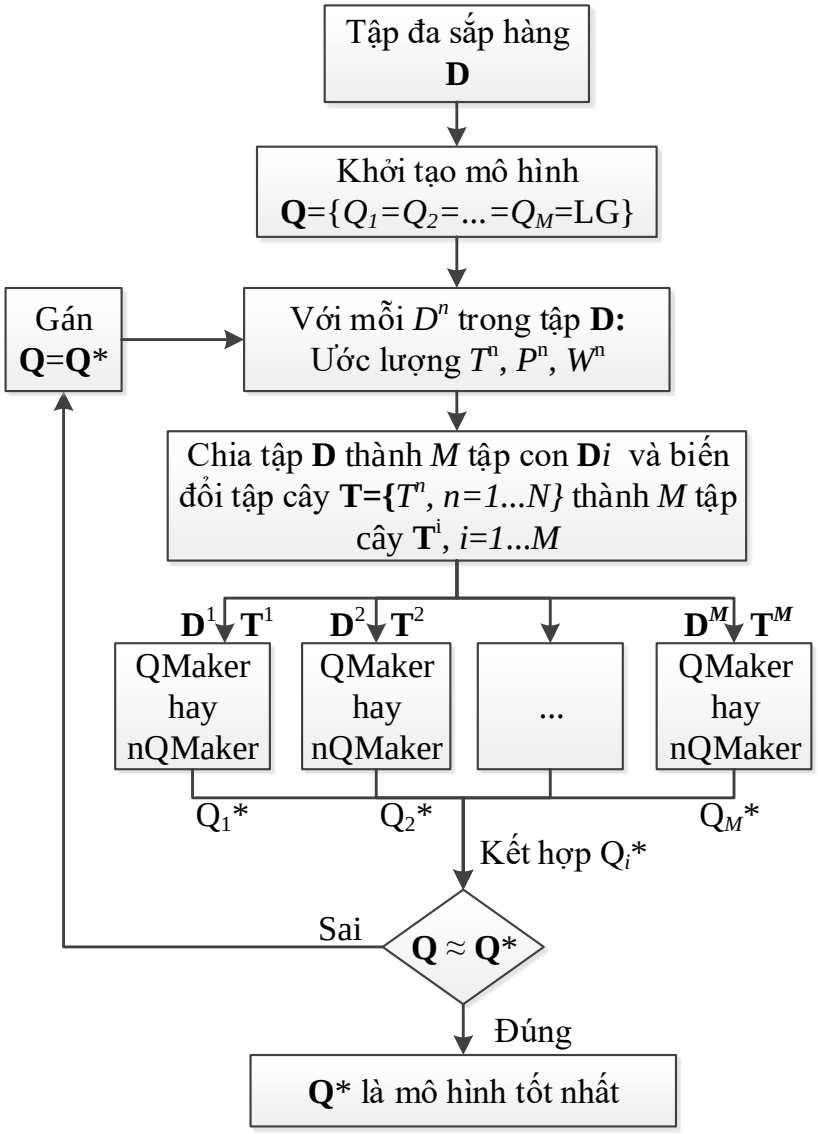
ở đây, D_i^n là vị trí thứ i của sắp hàng D^n, c_i là phân lớp tốc độ biến đổi có xác suất cao nhất cho vị trí D_i^n còn $\rho_{c_i}^n$ là tốc độ của phân lớp c_i tương ứng với ma trận Q_{c_i}. Mỗi ma trận có thể được ước lượng riêng biệt thông qua công thức:

$$\forall k = 1 \dots M, Q_k^* = \underset{Q_k}{\operatorname{argmax}} \left\{ \prod_{n=1}^{N} \prod_{i=1, c_i=k}^{l} L(T^n, \rho_k^n Q_k | D_i^n) \right\} \quad (3.11)$$

Dựa vào Công thức 3.10 và 3.11, luận án đề xuất phương pháp QMix có thể ước lượng hai mô hình mới dựa trên thuộc tính thuận nghịch hoặc không thuận nghịch về thời gian đồng thời tốc độ biến đổi trên các vị trí tuân theo phân bố gamma hoặc phân bố tự do, xem lưu đồ thuật toán trong Hình 3.1. Theo đó, luận án sẽ thực hiện quá trình xử lý trên từng sắp hàng của tập dữ liệu. Sau khi ước lượng được T^*, ρ^*, w^*, luận án thực hiện việc chia các vị trí thành *M* nhóm tương ứng với *M* phân lớp tốc độ. Đồng thời, với việc thay đổi độ dài cạnh bằng cách nhân tỉ lệ với tốc độ của các phân lớp, luận án cũng thu được các cây phân loài mới tương ứng với các nhóm vị trí. Từ đó, tập dữ liệu ban đầu sẽ được phân ra thành *M* tập con tương ứng *M* tập cây phân loài. Mỗi tập con gồm các sắp hàng đã được chia tách và các cây phân loài đã được điều chỉnh độ dài nhánh sẽ được sử dụng để ước lượng một ma trận tốc độ biến đổi. Cuối cùng, luận án sẽ nhận được *M* ma trận biến đổi tương ứng với các phân lớp tốc độ khác nhau.

Các mô hình đa ma trận hiện có đều dựa trên thuộc tính thuận nghịch theo thời gian và chưa có mô hình nào dựa trên thuộc tính không thuận nghịch theo thời gian được giới thiệu. Hai mô hình thay thế axit amin đa ma trận LG4X và LG4M [37] được ước lượng sử dụng hai chương trình XRate [9] và PhyML [52], tuy nhiên, quá trình này tốn nhiều thời gian và công sức tính toán, LG4X mất một tuần để ước lượng. Luận án sử dụng IQ-TREE [42] là phương pháp mới hơn thay cho PhyML để thực hiện tối ưu cây, cùng với QMaker [11] hoặc nQMaker [12] thay cho XRate để ước lượng các mô hình thuận nghịch hay không thuận nghịch thời gian từ các tập con và thiết lập mô hình $\boldsymbol{Q}^* = (Q_1^*, Q_2^*, \dots, Q_M^*)$, quá trình ước lượng dừng khi $\boldsymbol{Q}^*$ không tối ưu thêm được nữa. Việc này được hiểu là độ tương quan Pearson giữa các mô hình cũ và mới tối thiểu đạt giá trị kì vọng 0.999 hay $Pearson(Q_i, Q_i^*) \geq 0.999, i = 1, \dots, M$. Nói cách khác, các hệ số biến

đổi của các ma trận đã rất sát nhau, có sự chênh lệch rất nhỏ và mô hình đạt được trạng thái tối ưu.



Hình 3.1. Lưu đồ thuật toán ước lượng mô hình đa ma trận.

3.3 Kết quả

Đầu tiên, luận án tiến hành thử nghiệm phương pháp trên bộ dữ liệu HSSP để so sánh với hai mô hình LG4X, LG4M và đánh giá độ ổn định của phương pháp. Tiếp theo,

luận án trình bày hai mô hình mới cũng được ước lượng từ bộ dữ liệu HSSP nhưng theo thuộc tính không thuận nghịch về thời gian và hai mô hình mới được ước lượng từ bộ dữ liệu thực vật Plant. Luận án đánh giá hiệu suất các mô hình mới bằng việc so sánh với các mô hình khác trên các tập sắp hàng thực.

3.3.1 Thực nghiệm đánh giá độ ổn định phương pháp Qmix

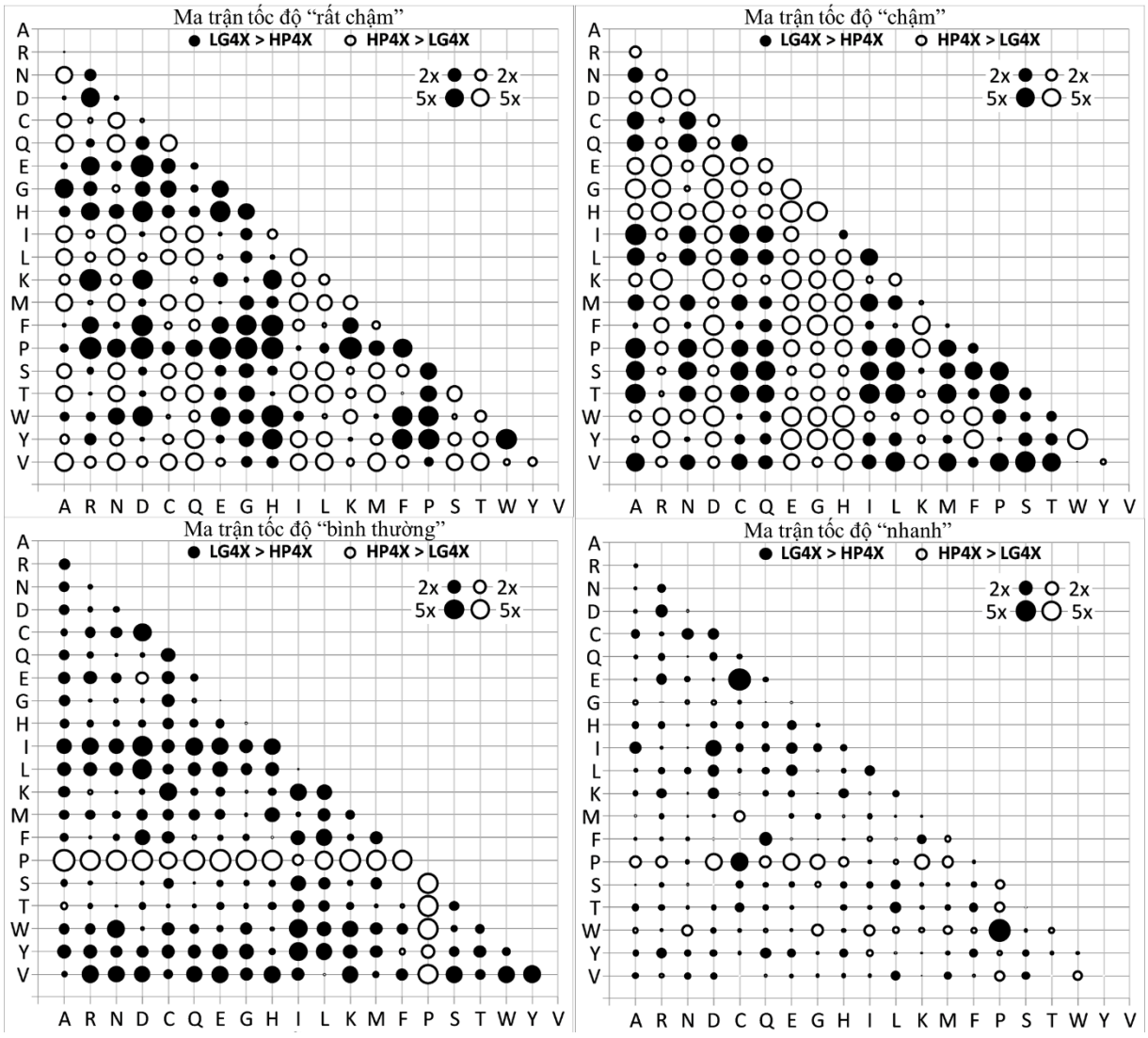
Đầu tiên, luận án thực hiện ước lượng lại hai mô hình tương tự với LG4X, LG4M từ 1471 sắp hàng thuộc tập dữ liệu HSSP, đây chính là tập dữ liệu dùng để ước lượng hai mô hình LG4X, LG4M. Mỗi mô hình gồm có bốn ma trận với bốn phân lớp tốc độ khác nhau gồm "rất chậm", "chậm", "bình thường", và "nhanh". Mục đích của việc này là đánh giá hiệu suất của mô hình xem có tương đương với hai mô hình gốc hay không. Hai mô hình mới được gọi là HP4X và HP4M trong đó HP4X tuân theo phân phối tốc độ tự do giống như LG4X, còn HP4M theo phân phối gamma như LG4M. Ngoài ra, luận án cũng ước lượng thêm hai mô hình HP4X.200, HP4M.200 có thuộc tính tương tự nhưng chỉ dùng 200 sắp hàng HSSP thay vì 1471 sắp hàng, mục đích là đánh giá xem tác động của số lượng dữ liệu đầu vào đến mô hình đầu ra.

Bảng 3.1 là kết quả tính độ tương quan Pearson giữa các ma trận của các mô hình LG4X, LG4M, HP4X và HP4M. Theo đó, ma trận tốc độ "bình thường" và "nhanh" có độ tương quan cao hơn tương quan của các ma trận "rất chậm" và "chậm". Ma trận "rất chậm" và "chậm" của mô hình LG4X rất khác biệt với hai ma trận dạng này nhưng của các mô hình khác, điều này do độ nhạy của phần mềm XRate với các giá trị khởi tạo [37]. Mô hình sinh ra bởi QMix với 1471 sắp hàng hay 200 sắp hàng thì đều có độ tương quan tốt, ví dụ giữa HP4X và HP4X.200 đều lớn hơn 0.9, với HP4M và HP4M.200 thì độ tương quan cũng đạt lớn hơn 0.75. Các ma trận tốc độ "nhanh" đều có độ tương quan cao trên 0.96, trong khi các ma trận "rất chậm" cho tương quan thấp hơn nhưng cũng đạt trên 0.7. Điều này cho thấy độ ổn định cao của QMix trong việc ước lượng mô hình.

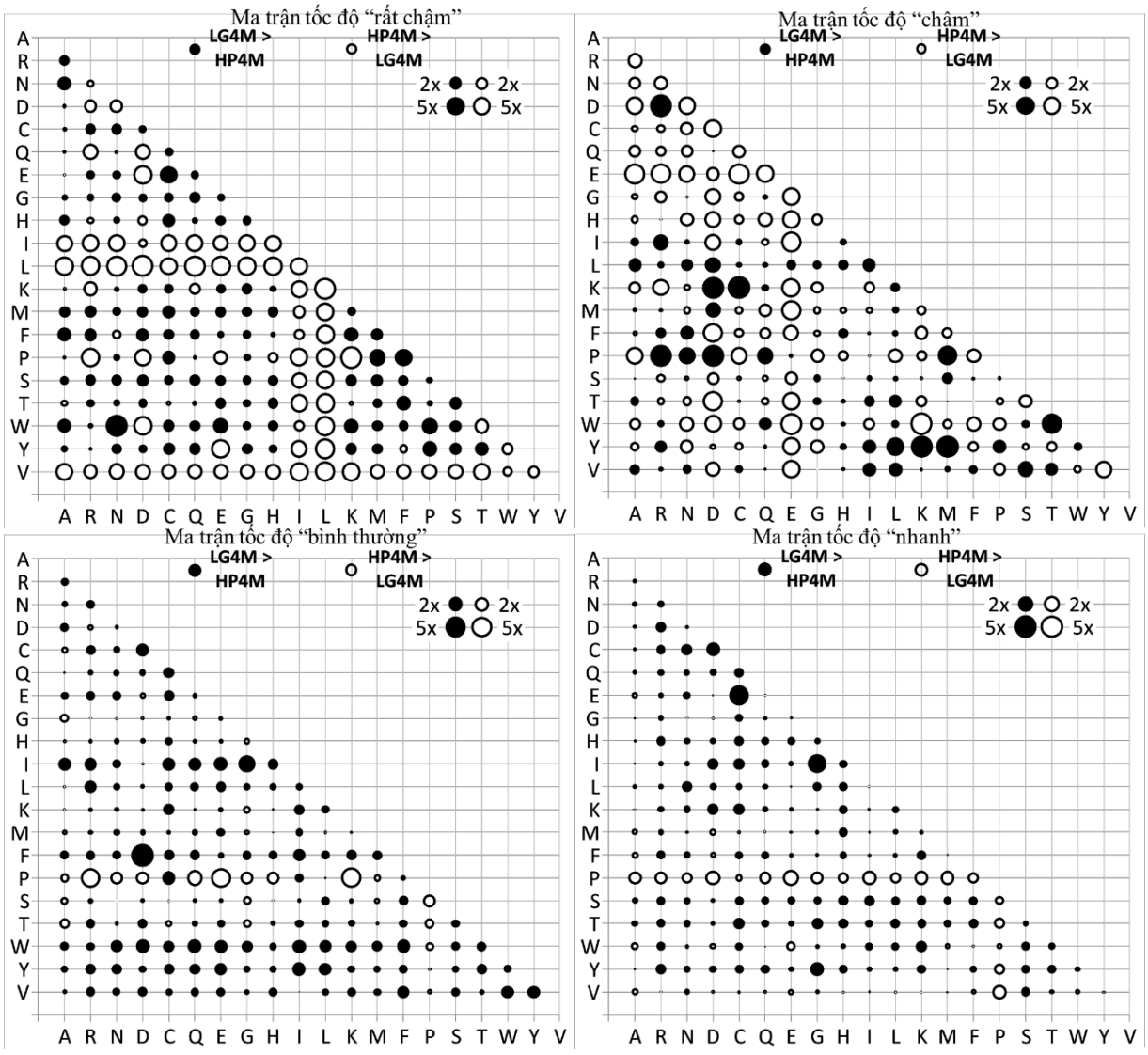
Bảng 3.1. Độ tương quan Pearson giữa các mô hình LG4X, LG4M, HP4X, HP4M

	Rất chậm	**Chậm**	**Bình thường**	**Nhanh**
LG4X vs. HP4X	0.237	0.442	0.931	0.989
LG4X vs. LG4M	0.055	0.653	0.898	0.991
HP4X vs. HP4X.200	0.944	0.967	0.955	0.962
LG4M vs. HP4M	0.773	0.843	0.982	0.995
HP4M vs. HP4M.200	0.752	0.881	0.944	0.964
HP4X vs. HP4M	0.825	0.943	0.961	0.996

Sự chênh lệch giữa các hệ số biến đổi của các ma trận được thể hiện theo Hình 3.2 và Hình 3.3 dưới dạng các bóng với kích thước khác nhau, bóng nhỏ thể hiện độ lệch nhỏ và ngược lại độ lệch lớn thể hiện bằng bóng lớn, kết quả cho thấy các ma trận "bình thường" (medium) và "nhanh" (fast) của mô hình LG4X và HP4x rất gần nhau với chỉ có 9 bóng có kích thước 2x (hơn kém nhau hai lần), 5x (hơn kém nhau năm lần). Các ma trận "rất chậm" và "chậm" của LG4X và HP4X có tổng gần 100 điểm dữ liệu có độ lệch 2x, 5x. Kết quả giữa các mô hình tốc độ gamma hay tốc độ tự do đều có xu hướng giống nhau.



Hình 3.2. Mối liên hệ giữa các hệ số biến đổi của các ma trận HP4X và LG4X. Trong đó: 2x (5x) thể hiện kích thước của sự khác biệt hai lần hay năm lần.



Hình 3.3. Mối liên hệ giữa các hệ số biến đổi của các ma trận HP4M và LG4M. Trong đó: 2x (5x) thể hiện kích thước của sự khác biệt hai lần hay năm lần.

Một đánh giá quan trọng không thể thiếu với các mô hình được ước lượng từ QMix là hiệu suất trong việc xây dựng cây cực đại hợp lý. Luận án tiến hành xây dựng cây với bộ dữ liệu kiểm tra bao gồm 300 sắp hàng HSSP [36] và 84 sắp hàng TreeBASE [55]. TreeBASE là một bộ dữ liệu độc lập đã được sử dụng trong rất nhiều nghiên cứu trước đây [19, 65, 71]. Kết quả trong Bảng 3.2 cho thấy hiệu suất của HP4X (HP4M)

tương đương với LG4X (LG4M) trong xây dựng cây cực đại hợp lý và vượt trội hơn hẳn các mô hình đơn khác.

Bảng 3.2. So sánh hiệu suất của HP4X, HP4M với các mô hình khác dựa trên 300 sắp hàng HSSP và 84 sắp hàng TreeBASE. #M1>M2: số lượng sắp hàng mà mô hình M1 tốt hơn mô hình M2

M_1	M_2	$\#M_1>M_2$ trên 300 sắp hàng HSSP	$\#M_1>M_2$ trên 84 sắp hàng TreeBASE
HP4X	LG4X	152	48
HP4M	LG4M	160	63
HP4X	LG	299	84
HP4M	LG	299	84
HP4X	Q.pfam	291	80
HP4M	Q.pfam	275	65

Với 200 sắp hàng, QMix cần khoảng 10 giờ để hoàn thành việc ước lượng mô hình HP4X.200. Dựa trên độ tin cậy, hiệu suất và sự đơn giản, luận án đề xuất QMix là công cụ cho các nhà khoa học sử dụng để ước lượng theo các tập dữ liệu cho từng bài toán riêng.

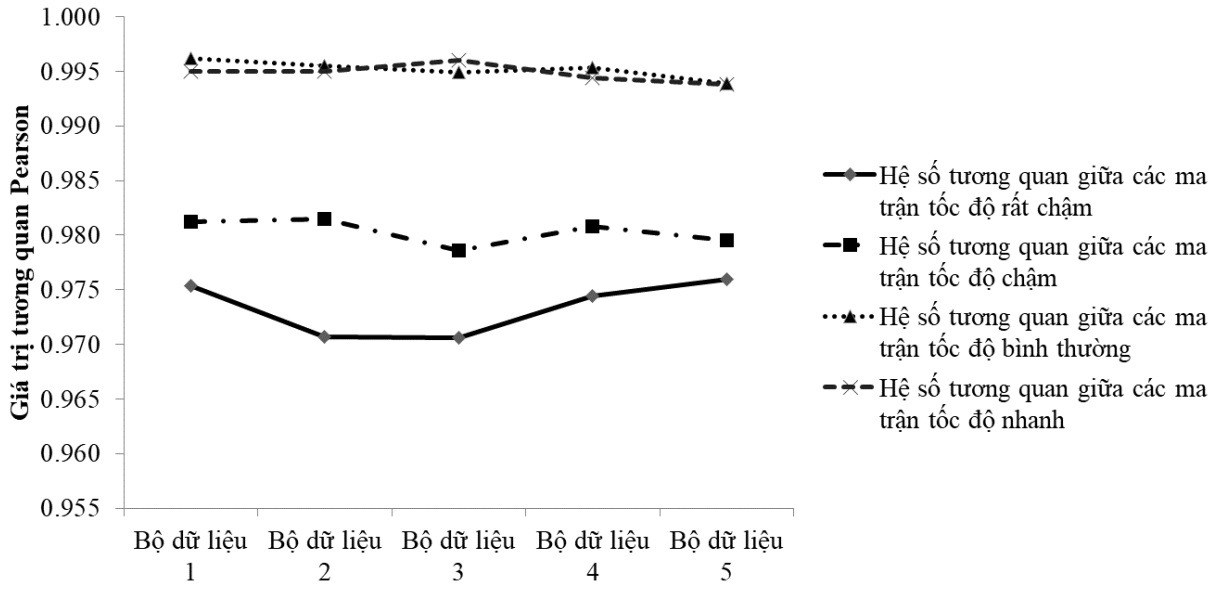
3.3.2 Ước lượng hai mô hình chung sử dụng đa ma trận mới

Các mô hình chung như JTT [6], WAG [7] hay LG [8] đóng vai trò quan trọng trong quá trình ước lượng các mô hình mới. Đây là những mô hình khởi tạo cho quá trình ước lượng, ngoài ra, do các mô hình này được ước lượng từ các bộ dữ liệu dùng chung rất đa dạng cả về số lượng trình tự, độ dài và cây phân loài nên chúng cũng thường được lựa chọn cho các phân tích ban đầu về sự tiến hóa của trình tự axit amin bất kì. Tuy nhiên, các mô hình này vẫn đang tồn tại ở dạng đơn ma trận. Do vậy, một số mô hình gồm nhiều ma trận đã được giới thiệu như LG4X, LG4M [37] gồm bốn ma trận, EXEHO [65] gồm sáu ma trận hay các mô hình đa tần số như C20, C60 [71] gồm có 20 hay 60 tần số khác nhau.

Tuy nhiên, tất cả các ma trận này đều được ước lượng dựa trên thuộc tính thời gian thuận nghịch, nghĩa là tốc độ biến đổi là cố định theo hướng tiến hay lùi, do vậy các mô hình này không có khả năng xây dựng cây có gốc. Vì vậy, mục tiêu phần này của luận án là sử dụng QMix kết hợp với nQMaker ước lượng hai mô hình chung mới, nT4X và nT4M, tương ứng tuân theo mô hình tốc độ tự do và theo phân phối gamma. Các mô hình này đều có bốn ma trận và dựa trên thuộc tính thời gian không thuận nghịch. Luận án sử dụng bộ dữ liệu dùng chung HSSP [54] để ước lượng mô hình và bộ dữ liệu TreeBASE [55] để đánh giá mô hình.

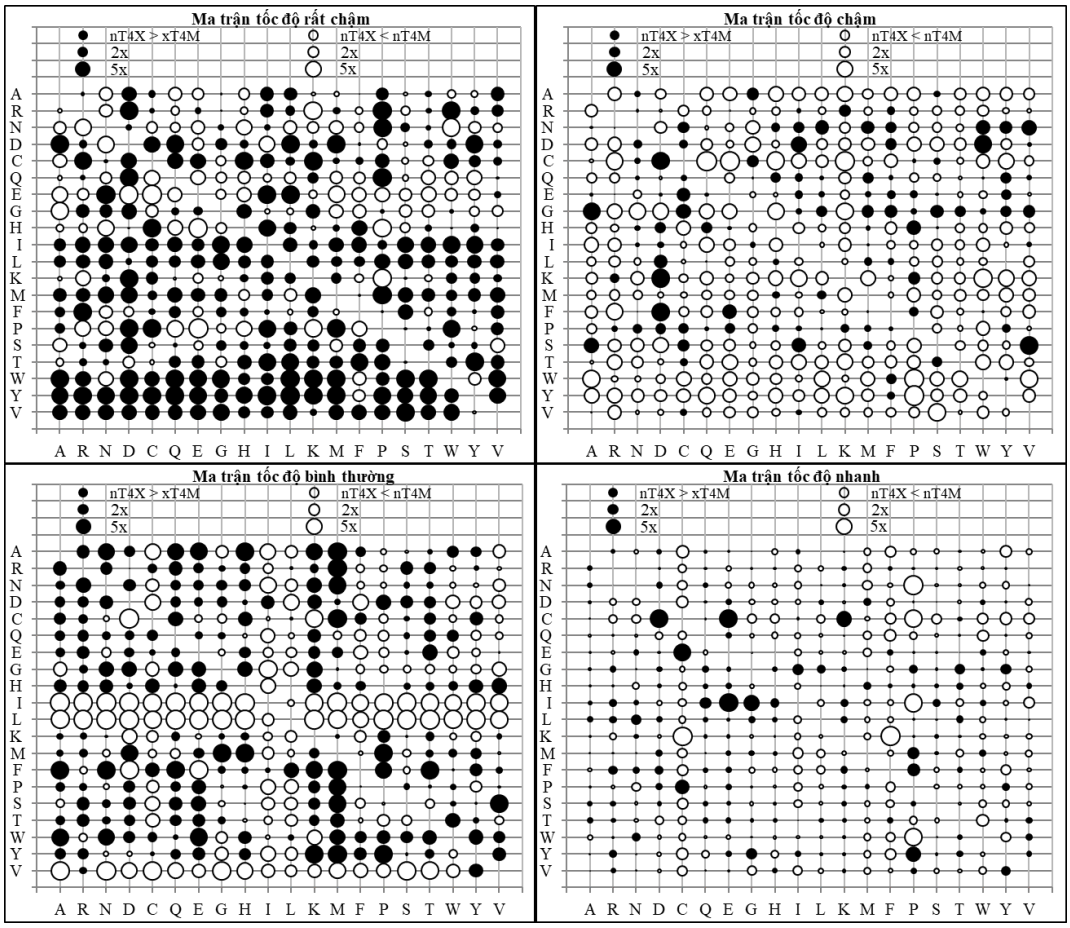
3.3.2.1 Phân tích các hệ số của mô hình

Trước tiên, dựa trên phương pháp đánh giá tại Chương 2, luận án một lần nữa đánh giá độ ổn định của phương pháp ước lượng bằng việc tạo ra năm bộ dữ liệu mô phỏng khác nhau, mỗi bộ gồm 200 sắp hàng, rồi thực hiện ước lượng năm mô hình mới từ dữ liệu mô phỏng đó. Các bộ dữ liệu này được tạo ra bằng việc sử dụng chương trình AliSim [64] cùng với mô hình mới nT4X. Hình 3.4 thể hiện hệ số tương quan Pearson giữa các ma trận của mô hình nT4X với các ma trận tương ứng của năm mô hình mới. Kết quả cho thấy tất cả các hệ số tương quan đều lớn hơn 0.97 và chứng tỏ sự ổn định của quá trình ước lượng cũng như phương pháp ước lượng mô hình.



Hình 3.4. Hệ số tương quan Pearson giữa nT4X với năm mô hình ước lượng từ năm bộ dữ liệu mô phỏng.

Ở đây, do chưa tồn tại một mô hình đa ma trận dùng chung nào dựa trên thuộc tính không thuận nghịch về thời gian, nên ta sẽ thực hiện so sánh các hệ số của hai mô hình nT4X và nT4M với nhau để xem xét sự khác biệt giữa chúng, xem Hình 3.5. Các ma trận "nhanh" của hai mô hình rất gần nhau, với chỉ 17 hệ số tốc độ là khác biệt năm lần trong khi ma trận "rất chậm" có 117 hệ số khác biệt ít nhất năm lần. Như vậy có thể thấy, mô hình tốc độ tự do và tốc độ theo phân phối gamma có sự khác biệt nhiều ở dải tốc độ thấp.



Hình 3.5. So sánh hệ số của nT4X và nT4M. 2x(5x) thể hiện sự khác biệt tối thiểu hai lần hay năm lần giữa các hệ số của các ma trận.

3.3.2.2 Đánh giá khả năng xây dựng cây cực đại hợp lý

Luận án thực nghiệm trên 300 sắp hàng HSSP và 84 sắp hàng TreeBASE để đánh giá khả năng của mô hình trong việc xây dựng cây phân loài. Về mặt kĩ thuật, luận án sử dụng chương trình IQ-TREE [42] để xây dựng các cây phân loài cho từng sắp hàng kiểm tra. Sau đó, tiêu chuẩn thông tin AIC [50] được sử dụng để đánh giá hiệu suất các mô hình. Các mô hình có số lượng tham số tự do khác nhau, thông tin cụ thể về các mô hình được đánh giá và số lượng tham số chi tiết trong Bảng 3.3, các mô hình đơn ma trận được sử dụng kèm với phân phối gamma với bốn phân lớp tốc độ (+Γ4). Luận án đã lựa

chọn đánh giá so sánh với nhiều kiểu mô hình khác nhau như mô hình đa ma trận, ví dụ UL3 [66] gồm ba ma trận, LG4X [37] gồm bốn ma trận, EXEHO [65] gồm sáu ma trận; mô hình đa tần số như C20, C60 [71] và các mô hình đơn ma trận [12].

Với mô hình *M* và sắp hàng *D*, nhắc lại công thức tính AIC như sau:

$$AIC(M|D) = 2 \times k - 2 \times ln(L(Q,T|D))$$

ở đây, $ln(L(Q,T|D))$ là giá trị hợp lý được lấy logarit, *k* là số lượng tham số tự do của mô hình *M*. Cho hai mô hình M_1 và M_2, nếu $\Delta AIC = AIC(M_1,D) - AIC(M_2,D) < 0$ thì M_1 được xem như tốt hơn M_2 và ngược lại.

Bảng 3.3. Số lượng ma trận và tham số tự do của cac mô hình

Mô hình	**Số tham số tự do**	**Số ma trận**	**Kiểu mô hình**
nT4X	6	4	Đa ma trận
nT4M	1	4	Đa ma trận
LG4X	6	4	Đa ma trận
LG4M	1	4	Đa ma trận
UL3	3	3	Đa ma trận
EXEHO	6	6	Đa ma trận
C20+Γ4	20	1	Đa tần số
C60+Γ4	60	1	Đa tần số
CF4+Γ4	24	1	Đa tần số
NQ.bird+Γ4	1	1	Đơn ma trận
NQ.plant+Γ4	1	1	Đơn ma trận
NQ.insect+Γ4	1	1	Đơn ma trận
NQ.mammal+Γ4	1	1	Đơn ma trận
NQ.yeast+Γ4	1	1	Đơn ma trận
NQ.pfam+Γ4	1	1	Đơn ma trận

Để đánh giá sâu hơn khi so sánh độ tốt giữa các mô hình, luận án sử dụng kiểm định KH-Test [25] với *p*-value là 0.01. Về cách thực hiện, luận án sử dụng kĩ thuật lấy mẫu có hoàn lại RELL bootstrap [72] với 10,000 lần lặp để đánh giá phân phối của giá

trị ΔAIC. Nếu giá trị p-value < 0.01 thì M_1 thực sự tốt hơn M_2 còn ngược lại chúng ta không thể khẳng định M_1 tốt hơn M_2.

Bảng 3.4. So sánh dựa trên AIC giữa nT4X, nT4M với 13 mô hình khác dựa trên các bộ dữ liệu HSSP và TreeBASE. Chú thích: #M_1>M_2: số lượng sắp hàng mà AIC của M_1 tốt hơn của M_2. #M_1>M_2(p<0.01): số lượng sắp hàng mà AIC của M_1 tốt hơn thực sự AIC của M_2. Tương tự cho #M_1>M_2 và #M_1>M_2(p<0.01)

M_1	M_2	300 sắp hàng HSSP			84 sắp hàng TreeBASE		
		#M_1>M_2	#M_1>M_2 (p<0.01)	#M_2>M_1 (p<0.01)	#M_1>M_2	#M_1>M_2 (p<0.01)	#M_2>M_1 (p<0.01)
nT4M	LG4M	217	68	18	72	16	6
nT4X	EXEHO	137	23	37	38	8	4
nT4X	LG4X	170	27	26	49	11	5
nT4M	EXEHO	103	12	37	11	3	8
nT4X	UL3	208	56	37	55	6	3
nT4M	UL3	183	33	42	23	5	14
nT4M	CF4	297	111	2	83	18	0
nT4X	CF4	297	111	2	83	25	0
nT4X	NQ.pfam	294	66	0	81	13	1
nT4M	C20	260	53	13	52	10	3
nT4X	C20	271	57	7	65	12	1
nT4X	NQ.insect	284	112	2	80	8	1
nT4M	C60	247	55	15	33	13	7
nT4X	NQ.yeast	284	101	1	79	17	0
nT4X	C60	258	61	15	48	18	3
nT4M	NQ.pfam	277	60	5	65	10	6
nT4M	NQ.insect	273	97	4	56	8	2
nT4M	NQ.yeast	276	82	3	66	10	3
nT4M	NQ.plant	272	105	8	71	17	4
nT4X	NQ.plant	277	125	2	76	21	1
nT4M	NQ.mammal	281	99	11	76	11	3
nT4X	NQ.mammal	283	110	10	77	16	1
nT4M	NQ.bird	283	102	7	77	14	3
nT4X	NQ.bird	286	123	7	77	24	2

Kết quả trong Bảng 3.4 cho thấy nT4X và nT4M tốt hơn các mô hình đơn ma trận ở hầu hết các sắp hàng của hai bộ HSSP và TreeBASE. Cụ thể, nT4X có AIC tốt hơn NQ.pfam ở 294 (98%) trên tổng 300 sắp hàng HSSP, với bộ TreeBASE, nT4X tốt hơn NQ.pfam ở 81 (96%) và trong đó có 13 (15%) sắp hàng là tốt hơn thực sự (*p*-value < 0.01).

3.3.2.3 Đánh giá ảnh hưởng của mô hình đến cấu trúc cây

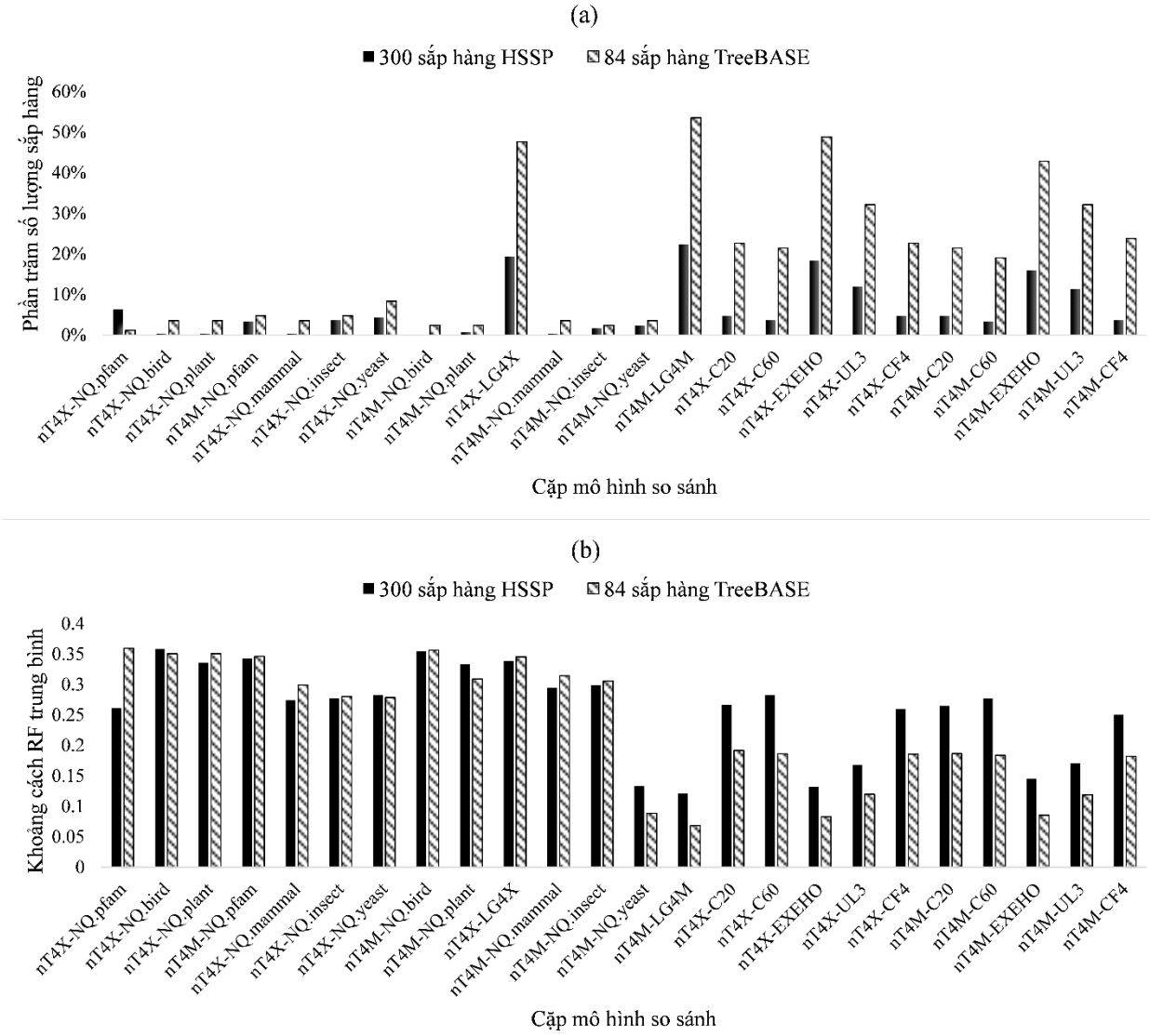
Ở phần này, luận án so sánh cấu trúc cây được xây dựng bởi nT4X, nT4M với các mô hình khác sử dụng khoảng cách nRF, xem kết quả chi tiết trong Bảng 3.5. Do không biết cây đúng (true tree) nên trong bảng này, luận án chỉ thống kê các sắp hàng mà các mô hình cho cấu trúc cây khác nhau (giá trị nRF>0), điều này sẽ thể hiện tác động của mô hình mới lên các sắp hàng. Kết quả cho chúng ta thấy rằng rất nhiều cây được xây dựng bởi hai mô hình mới và các mô hình cũ có cấu trúc khác nhau. Ví dụ, trong 300 sắp hàng kiểm tra của bộ HSSP thì nT4X có AIC tốt hơn NQ.pfam ở 294 sắp hàng (xem Bảng 3.4), trong đó có 276 sắp hàng hai mô hình cho cây khác nhau (xem Bảng 3.5). Tương tự thế, nT4M có AIC tốt hơn LG4M ở 217 sắp hàng HSSP trong đó có 188 sắp hàng cho cây khác nhau.

Bảng 3.5. So sánh cấu trúc cây trên các sắp hàng HSSP và TreeBASE. #T1>T2: số lượng sắp hàng mà cây T1 được xây dựng bởi mô hình M1 có giá trị AIC tốt hơn cây T2 được xây dựng bởi M2 và T1, T2 có cấu trúc khác nhau. #T1>T2($p<0.01$): cùng ý nghĩa như T1>T2 nhưng T1 thực sự tốt hơn T2. Chú thích tương tự cho các ký hiệu còn lại

M_1	M_2	300 sắp hàng HSSP				84 sắp hàng TreeBASE			
		$\#T_1>T_2$	$\#T_2>T_1$	$\#T_1>T_2$ ($p<0.01$)	$\#T_2>T_1$ ($p<0.01$)	$\#T_1>T_2$	$\#T_2>T_1$	$\#T_1>T_2$ ($p<0.01$)	$\#T_2>T_1$ ($p<0.01$)
nT4M	LG4M	188	45	53	13	33	6	8	4
nT4X	EXEHO	73	146	20	31	18	25	5	3
nT4X	LG4X	136	106	23	21	30	14	9	0
nT4M	EXEHO	80	172	12	34	3	45	1	8
nT4X	UL3	152	91	49	37	34	23	3	3
nT4M	UL3	156	110	28	39	11	46	4	12
nT4M	CF4	287	2	111	2	63	1	16	0
nT4X	CF4	283	3	110	2	64	1	22	0
nT4X	NQ.pfam	276	5	62	0	80	3	13	1
nT4M	C20	246	40	52	13	41	25	7	3
nT4X	C20	246	29	55	7	50	15	10	0
nT4X	NQ.insect	274	15	110	2	76	4	8	1
nT4M	C60	237	53	53	15	26	42	12	6
nT4X	NQ.yeast	273	14	99	1	72	5	17	0
nT4X	C60	236	42	58	15	39	27	17	2
nT4M	NQ.pfam	268	22	58	5	62	18	10	6
nT4M	NQ.insect	269	26	95	4	55	27	8	2
nT4M	NQ.yeast	269	24	80	3	64	17	10	3
nT4M	NQ.plant	270	28	104	8	69	13	10	6
nT4X	NQ.plant	277	22	125	2	73	8	20	1
nT4M	NQ.mammal	280	19	98	11	73	8	11	3
nT4X	NQ.mammal	282	17	110	10	74	7	16	1
nT4M	NQ.bird	283	17	102	7	75	7	14	3
nT4X	NQ.bird	285	14	123	7	74	7	24	2

Với bộ kiểm tra TreeBASE, nT4M tốt hơn LG4M ở 72 sắp hàng, trong đó có 33 sắp hàng cho cây khác nhau và có tám sắp hàng cho cây khác nhau mà nT4M thực sự tốt hơn LG4M. Ngoài ra, luận án thống kê giá trị nRF của các cây xây dựng bởi từng cặp mô hình được so sánh, kết quả chi tiết của 24 cặp mô hình được thể hiện trong biểu đồ Hình 3.6. Đúng như kì vọng, cặp nT4M-LG4M có độ tương đồng của các cây là cao nhất với 53.6% trên bộ TreeBASE và 22.3% trên bộ HSSP. Mô hình nT4X và EXEHO xây dựng cùng một cấu trúc cây trên 48.8% số sắp hàng TreeBASE và 18.3% số sắp hàng HSSP. Mô hình nT4X, nT4M tạo ra cây khác biệt với các mô hình đơn, chưa đến 10% sắp hàng có cùng cấu trúc, xem Hình 3.6a.

Về giá trị trung bình của khoảng cách nRF, chi tiết xem Hình 3.6b, cặp nT4M và LG4M cho kết quả 0.12 trên bộ HSSP và 0.07 trên bộ TreeBASE. Đây là kết quả tốt nhất khi so sánh với trung bình của các cặp khác. Điều này phù hợp với kết quả về độ tương quan cao giữa nT4M và LG4M như đề cập ở các phần trước. Khoảng cách nRF trung bình giữa mô hình nT4X, nT4M với các mô hình đơn dao động trong khoảng 0.26 đến 0.36.

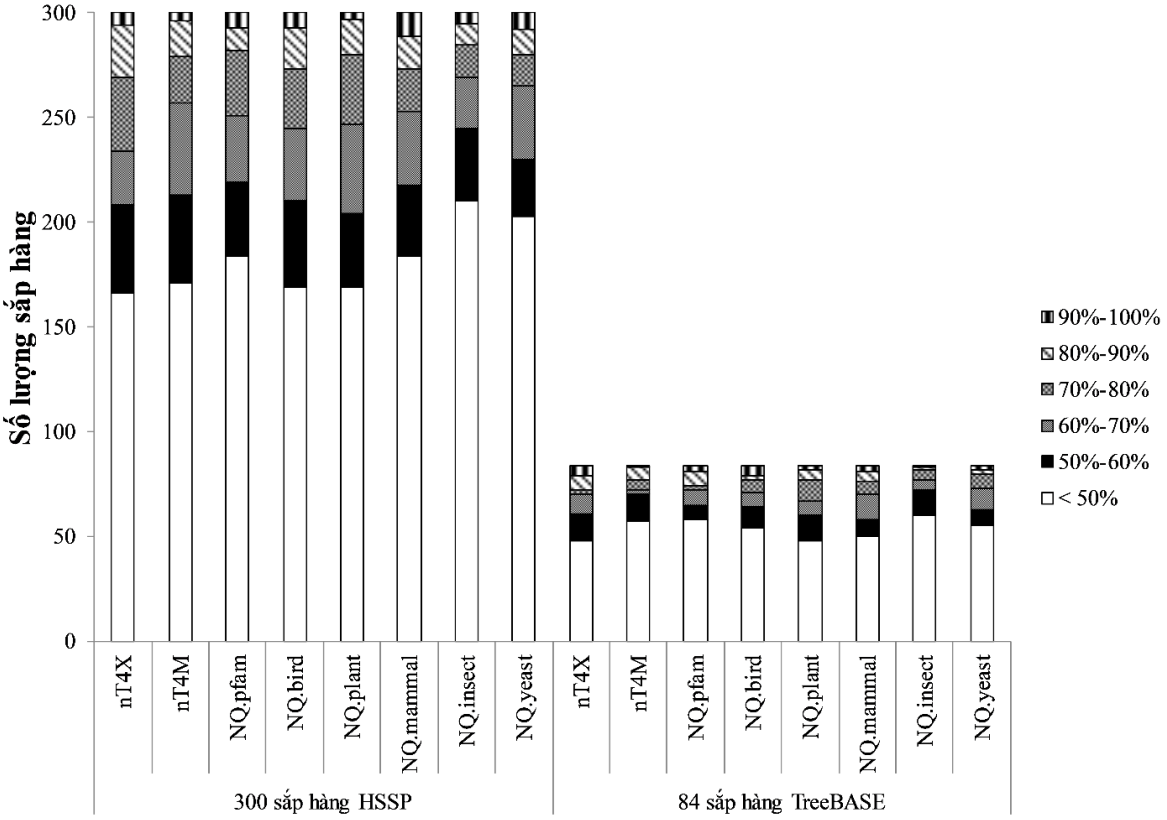


Hình 3.6. Khoảng cách RF trên 24 cặp mô hình so sánh trên các bộ dữ liệu kiểm tra HSSP và TreeBase.

3.3.2.4 Đánh giá khả năng xây dựng cây có gốc

Để kiểm nghiệm khả năng trong việc xây dựng cây có gốc của các mô hình không thuận nghịch về mặt thời gian, luận án sử dụng phương pháp rootstrap [73] với 1000 lần lặp. Về mặt kĩ thuật, với mỗi sắp hàng, phương pháp này sẽ thực hiện xây dựng cây 1000

lần, giá trị hỗ trợ rootstrap của một nhánh được định nghĩa là số lần mà vị trí gốc xuất hiện trên một nhánh đó. Tiếp đó, luận án thực hiện thống kê các sắp hàng mà độ hỗ trợ rootstrap tại vị trí gốc tối thiểu là 50%, nghĩa là ít nhất 500 lần vị trí gốc xuất hiện tại đúng vị trí vốn có của nó, chi tiết xem trong Hình 3.7. Ở đây, luận án chỉ khảo sát các mô hình không thuận nghịch gồm nT4X, nT4M, NQ.plant, NQ.bird, NQ.yeast, NQ.mammal, NQ.insect và NQ.pfam bởi vì những mô hình này mới xây dựng được cây có gốc. Kết quả cho thấy mô hình nT4X có thể xây dựng cây có độ hỗ trợ rootstrap tại vị trí gốc trên 50% tại 134 trên 300 sắp hàng HSSP và 36 trên 84 sắp hàng TreeBASE.



Hình 3.7. Phân phối độ hỗ trợ rootstrap tại vị trí gốc của các cây phân loài xây dựng bởi các mô hình không thuận nghịch.

3.3.3 Ước lượng hai mô hình riêng sử dụng đa ma trận cho các loài thực vật

Nhu cầu ước lượng các mô hình riêng cho từng nhóm loài là cấp thiết để nâng cao hơn nữa hiệu quả quá trình nghiên cứu tiến hóa của các loài đó. Các phương pháp QMaker và nQMaker đã đề xuất một số mô hình cả thời gian thuận nghịch và không thuận nghịch theo các nhóm loài cụ thể như Q.plant, Q.bird [11] hay NQ.plant, NQ.bird [12]. Thực vật (Plants) có vai trò đặc biệt quan trọng trong đời sống con người cũng như hệ sinh thái tự nhiên. Việc nghiên cứu về sự tiến hóa của các loài thực vật là quan trọng và cần thiết. Do đó, trong phần này, luận án tiếp tục sử dụng phương pháp QMix để ước lượng mô hình đa ma trận cho bộ dữ liệu thực vật [43] gọi là QPlant.mix và nQPlant.mix, cả hai mô hình đều có bốn ma trận, trong đó QPlant.mix tuân theo thuộc tính thuận nghịch thời gian, nQPlant.mix theo thuộc tính không thuận nghịch thời gian.

Để đánh giá hiệu suất của hai mô hình mới, luận án thực hiện các thí nghiệm trên 308 sắp hàng kiểm tra của bộ dữ liệu Plant [43].

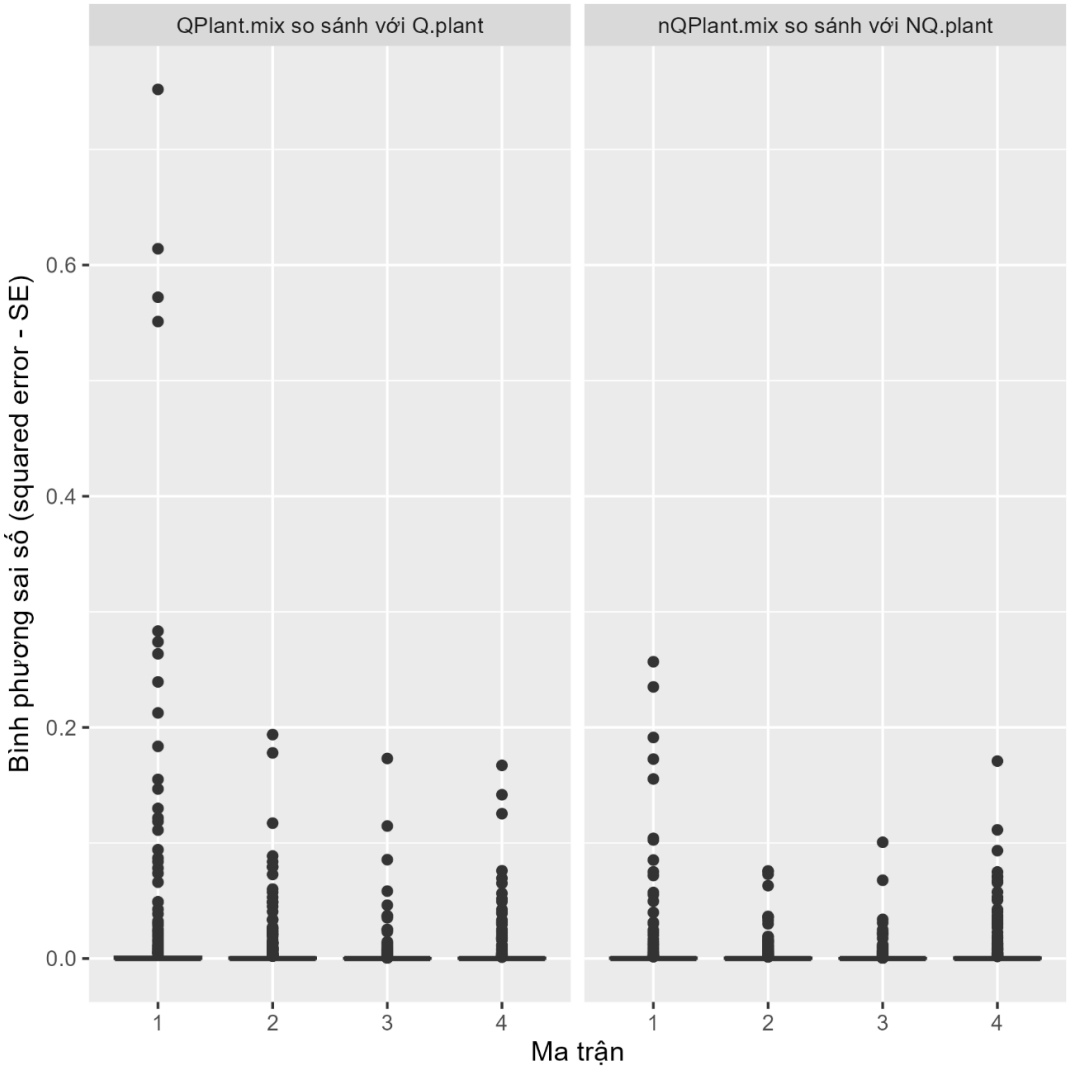
3.3.3.1 Phân tích các hệ số của mô hình

Trước tiên, luận án xem xét mức độ tương đồng giữa cặp mô hình đơn Q.plant với mô hình đa ma trận QPlant.mix và cặp NQ.plant với nQPlant.mix. Dùng độ chênh lệch 2x (5x) tương tự như các phần trên, luận án thống kê số lượng tham số chênh lệch hơn kém nhau hai lần hoặc năm lần cùng với giá trị tương quan, xem chi tiết trong Bảng 3.6. Dễ dàng có thể nhận thấy độ tương quan mạnh giữa những ma trận đơn và các ma trận của mô hình đa ma trận. Tất cả giá trị tương quan đều lớn hơn 0.9 trong đó các ma trận tốc độ "chậm" và "bình thường" là cao hơn cả với 0.98, 0.98 của cặp không thuận nghịch và 0.92, 0.97 của cặp ma trận thuận nghịch. Giá trị tương quan của các ma trận không thuận nghịch cao hơn các ma trận thuận nghịch, điều này phù hợp với số lượng chênh lệch 2x, 5x của các ma trận thuận nghịch có tỉ lệ cao hơn các ma trận không thuận nghịch.

Bảng 3.6. Giá trị tương quan giữa các hệ số của ma trận của nQPlant.mix (Qplant.mix) với hệ số của NQ.plant (Q.plant). 2x (-2x) thể hiện hệ số của nQPlant.mix (hay Qplant.mix) lớn hơn (nhỏ hơn) hai lần so với NQ.plant (hay Q.plant). Chú thích tương tự cho 5x và -5x

		Rất chậm	Chậm	Bình thường	Nhanh
nQPlant.mix & NQ.plant	Độ tương quan	0.97	0.98	0.98	0.97
	2x	17.75%	5.25%	2.75%	14.50%
	5x	5.75%	1.00%	0.00%	3.00%
	-2x	26.75%	40.75%	26.00%	29.75%
	-5x	19.50%	18.50%	16.75%	21.50%
QPlant.mix & Q.plant	Độ tương quan	0.90	0.92	0.97	0.89
	2x	8.42%	4.21%	7.37%	16.84%
	5x	1.05%	1.58%	1.05%	0.53%
	-2x	40.00%	50.00%	20.00%	26.32%
	-5x	18.95%	19.47%	12.63%	13.68%

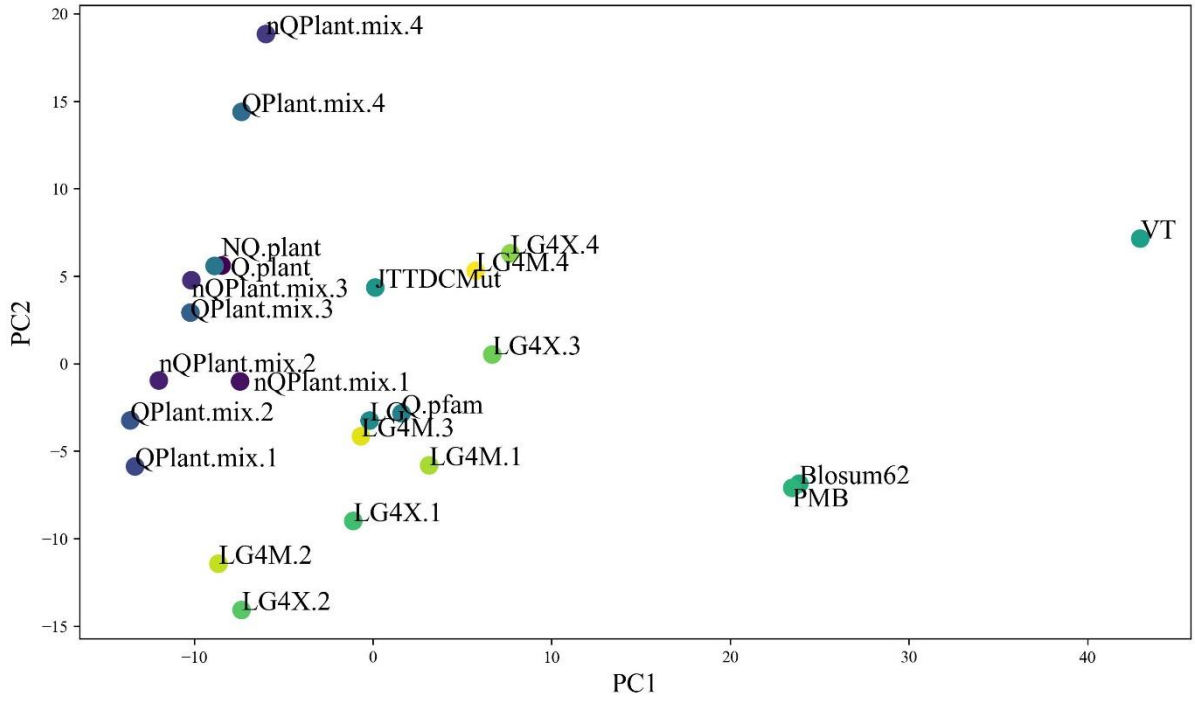
Khảo sát thêm về bình phương sai số (squared error - SE) giữa các giá trị hệ số của mô hình đơn với các ma trận của mô hình đa ma trận, luận án nhận thấy các ma trận "rất chậm" thường có độ lệch cao hơn các ma trận khác. Có bốn điểm dữ liệu của ma trận "rất chậm" theo thuộc tính thuận nghịch mà giá trị SE cao hơn 0.5 còn lại hầu hết các giá trị khác đều nhỏ hơn 0.4, xem thêm trong Hình 3.8. Giá trị MSE của cặp mô hình không thuận nghịch là 0.0053 cho ma trận "rất chậm", 0.0021 cho ma trận "chậm", 0.0015 cho "bình thường" và 0.0041 cho ma trận "nhanh".



Hình 3.8. Phân phối bình phương sai số (SE) của các hệ số tốc độ biến đổi của các ma trận mới khi so sánh với các mô hình Q.plant và NQ.plant. Chú thích: 1, 2, 3, 4 tương ứng với các ma trận "rất chậm", "chậm", "bình thường" và "nhanh".

Tiếp đó, luận án phân tích thêm mối quan hệ giữa hai mô hình mới với các mô hình đã có bằng công cụ phân tích thành phần chính (PCA). Kết quả PCA trong Hình 3.9 cho thấy các ma trận đã hình thành nên một vài cụm riêng biệt. Trong đó, các ma trận rất gần nhau phải kể đến như: Q.plant, NQ.plant và các ma trận "rất chậm", "chậm" và "bình

thường” của hai mô hình QPlant.mix và nQPlant.mix. Hai ma trận “nhanh” của cả hai mô hình đã có sự khác biệt rõ ràng hơn với các ma trận còn lại. Điều này tạo nên sự khác biệt cho hai mô hình mới và mang lại nhiều ý nghĩa trong quá trình thực nghiệm. Ngoài ra, việc các mô hình đa ma trận mới cách xa các mô hình đa ma trận dùng chung như LG4X, LG4M cũng thể hiện sự hợp lý và cần thiết phải dùng mô hình mới để nghiên cứu các dữ liệu cụ thể như các loài thực vật.

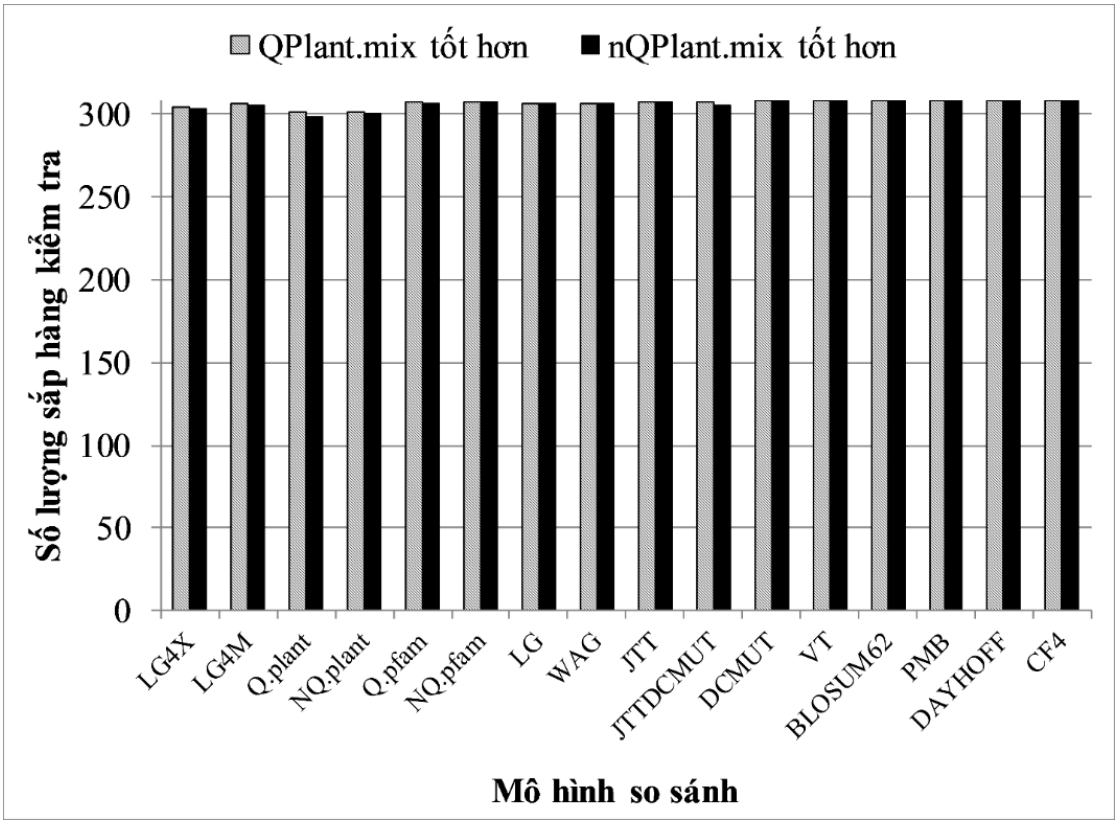


Hình 3.9. Phân tích thành phần chính giữa các ma trận.

3.3.3.2 Đánh giá khả năng xây dựng cây cực đại hợp lý

Luận án tiến hành kiểm tra hiệu suất các mô hình trong việc xây dựng cây cực đại hợp lý dựa trên 308 sắp hàng kiểm tra của bộ dữ liệu Plant và sử dụng tiêu chuẩn BIC bằng cách chạy chương trình IQ-TREE [42] cho toàn bộ các sắp hàng đó. Hai mô hình mới đã thể hiện hiệu năng vượt trội khi tốt hơn gần như tuyệt đối các mô hình còn lại, xem kết quả cụ thể trong Hình 3.10. QPlant.mix tốt hơn Q.plant ở 301 trên 308 sắp hàng,

tương tự nQPlant.mix cũng tốt hơn NQ.plant ở 301 trong tổng số 308 sắp hàng. So sánh với hai mô hình LG4X và LG4M, hai mô hình mới cũng tốt hơn ở hầu hết sắp hàng, cụ thể, QPlant.mix tốt hơn LG4X ở 304 sắp hàng. Với các mô hình khác như DCMCut, VT, Blosum62, PMB và CF4 thì hai mô hình mới tốt hơn ở toàn bộ 308 sắp hàng.



Hình 3.10. Hiệu suất các mô hình trong việc xây dựng cây cực đại hợp lý.

3.3.3.3 Đánh giá ảnh hưởng của mô hình đến cấu trúc cây

Để xem xét mức độ ảnh hưởng của mô hình mới đến hình dạng của cây phân loài. Luận án tính khoảng các nRF giữa các cây được xây dựng bởi hai mô hình mới với các cây được xây dựng bởi các mô hình khác, xem kết quả trong Bảng 3.7. Các mô hình mới đã có tác động đáng chú ý lên cấu trúc của cây. Chỉ có một vài sắp hàng mà cây được xây dựng có cùng cấu trúc. Ví dụ, QPlant.mix và Q.plant xây dựng cùng cấu trúc

cây ở 52 trên tổng 308 sắp hàng, đây là kết quả cao nhất trong số các mô hình. Với mô hình CF4 thì chỉ có duy nhất một sắp hàng mà các cây được xây dựng là cùng cấu trúc. Giá trị khoảng cách nRF trung bình nằm trong phạm vi 0.102 đến 0.227.

Bảng 3.7. Khoảng cách RF giữa các cây được xây dựng bởi mô hình mới và các mô hình đang có trên bộ dữ liệu kiểm tra

Model	QPlant.mix		nQPlant.mix	
	#nRF=0	avg(nRF)	#RF=0	avg(nRF)
LG	30	0.143	15	0.138
WAG	13	0.139	13	0.139
JTT	30	0.127	28	0.123
DCMut	14	0.149	11	0.147
JTTDCMut	31	0.129	27	0.123
VT	12	0.137	19	0.141
Blosum62	7	0.161	10	0.163
PMB	7	0.157	10	0.161
CF4	1	0.227	1	0.227
Q.pfam	19	0.142	15	0.138
Q.plant	52	0.114	40	0.102
NQ.pfam	19	0.142	12	0.136
NQ.plant	44	0.117	42	0.112
LG4M	13	0.143	13	0.142
LG4X	16	0.141	11	0.143

3.4 Tổng kết chương

Mô hình thay thế đa ma trận có nhiều ưu điểm phù hợp với các tính chất về mặt sinh học của trình tự axit amin. Độ phức tạp và yêu cầu cao về hiệu năng phần cứng khiến cho việc ước lượng mô hình đa ma trận trở nên khó khăn. Trong chương này, luận án đã trình bày một phương pháp mới nhằm đơn giản hóa quá trình ước lượng mô hình đa ma trận, đó là phần mềm QMix. Các nhà nghiên cứu có thể thực thi quá trình ước

lượng bằng một câu lệnh duy nhất và rất nhanh chóng thu nhận được mô hình đa ma trận mới.

Tiếp đó, luận án cũng đề xuất hai mô hình mới gọi là nT4X, nT4M ước lượng từ bộ dữ liệu HSSP với thuộc tính không thuận nghịch về thời gian. Đây là hai mô hình dùng chung đầu tiên sử dụng đa ma trận dựa trên thuộc tính thời gian không thuận nghịch. Hai mô hình mới đã thể hiện được những ưu điểm và ảnh hưởng quan trọng trong việc xây dựng cây cực đại hợp lý cũng như tìm vị trí gốc của cây.

Và cuối cùng, trong chương này, luận án cũng đã thử nghiệm ước lượng hai mô hình đa ma trận mới cho bộ dữ liệu thực vật (Plant). Hai mô hình QPlant.mix và nQPlant.mix có bốn ma trận sử dụng trong đó QPlant.mix dựa trên tính chất thời gian thuận nghịch còn nQPlant.mix dựa trên tính chất thời gian không thuận nghịch. Cả hai mô hình cho thực vật đều cho thấy hiệu suất vượt trội hơn hẳn các mô hình riêng biệt theo loại đang tồn tại là Q.plant và NQ.plant.

Chương 4. Phương pháp lựa chọn nhanh mô hình thay thế axit amin

Chương 4 giới thiệu về cách trích chọn đặc trưng từ dữ liệu sắp hàng và đề xuất một kiến trúc mạng học sâu đơn giản nhằm lựa chọn nhanh mô hình thay thế axit amin. Do không biết mô hình đúng của dữ liệu thật nên dữ liệu mô phỏng được sử dụng để đào tạo và đánh giá phương pháp. Luận án so sánh hiệu năng phương pháp mới với phương pháp sử dụng tiêu chuẩn cực đại hợp lý là ModelFinder. Luận án thực hiện đánh giá các phương pháp dựa trên các tiêu chí như độ chính xác dự đoán và thời gian suy luận ra kết quả.

4.1 Giới thiệu chung

Từ sự ra đời của các phương pháp ước lượng mô hình thay thế axit amin mà ngày càng nhiều mô hình thay thế được ước lượng và đề xuất. Các mô hình có thể ở dạng đơn ma trận, đa ma trận hay đa tần số và được ước lượng từ một bộ dữ liệu axit amin cụ thể, do vậy mô hình sẽ giải thích tốt nhất bộ dữ liệu mà dựa trên đó nó được ước lượng. Việc ước lượng mô hình phải trải qua quá trình tốn kém về mặt thời gian và không gian tính toán, ngoài ra để ước lượng được mô hình đủ tốt chúng ta cũng cần thu thập lượng dữ liệu phù hợp đủ lớn. Từ đó, vấn đề được đặt ra là: với một dữ liệu sắp hàng axit amin cho trước, làm thế nào để tìm mô hình phù hợp nhất với sắp hàng đó trong số hàng chục mô hình đang tồn tại một cách nhanh nhất mà không cần phải ước lượng mô hình mới?

Thông thường, để lựa chọn mô hình phù hợp nhất cho một sắp hàng, chúng ta thường dựa trên một số tiêu chuẩn như giá trị hợp lý, BIC hay AIC. Tuy nhiên, việc sử dụng các tiêu chuẩn thông tin này vẫn còn nhiều tranh cãi trong phân tích tiến hóa [74-76] do cây đúng và mô hình đúng không chắc đã có giá trị BIC/AIC tốt nhất [15]. ModelFinder là phương pháp phổ biến sử dụng tiêu chuẩn cực đại hợp lý để chọn ra mô

hình phù hợp nhất với dữ liệu. Về mặt kĩ thuật, giả sử chúng ta cần chọn một trong số 10 mô hình cho trước, ModelFinder sẽ thực hiện tính toán giá trị hợp lý của cây phân loài với mỗi mô hình trong tập 10 mô hình đó, rồi đưa ra mô hình cho giá trị tốt nhất. Ngoài ra, ModelFinder cũng kết hợp các mô hình thay thế cùng với các mô hình tốc độ biến đổi tại vị trí và mô hình tần số. Ví dụ, thay vì chỉ xét mô hình LG thì phương pháp ModelFinder cũng xem xét cả các tổ hợp như: LG+G4, LG+F, LG+I+G4. Từ đó, càng nhiều mô hình thì chúng ta sẽ càng mất nhiều thời gian để tìm ra được mô hình phù hợp nhất. Đồng thời, nếu sắp hàng càng dài thì cũng tiêu tốn nhiều thời gian tính toán hơn.

Như vậy, việc tìm mô hình theo tiêu chuẩn cực đại hợp lý yêu cầu lượng tính toán rất lớn do phải phân tích một loạt các mô hình khác nhau [13, 42, 75]. Các nhà nghiên cứu cũng đề xuất phương pháp MixtureFinder [77] nhằm lựa chọn nhanh các mô hình hỗn hợp phù hợp với các sắp hàng DNA, phương pháp này tiêu tốn tài nguyên tính toán hơn so với ModelFinder. Do vậy, chúng ta cần phải có một phương pháp khác nhanh chóng đưa ra được mô hình phù hợp nhất với sắp hàng axit amin cho trước mà tiêu tốn ít tài nguyên nhất.

Cách tiếp cận dựa trên học máy đã được áp dụng trong một số nghiên cứu tin sinh học và cho kết quả khả quan [15, 16, 60, 78-80]. Liên quan đến lựa chọn mô hình thay thế, một số phương pháp đã được giới thiệu như ModelTeller [16] sử dụng thuật toán Random Forest và ModelRevelator [15] sử dụng mạng học sâu ResNet-18 [53]. Hai phương pháp này hoạt động tốt trên các dữ liệu DNA với độ chính xác khá cao, ví dụ 91% cho ModelRevelator với mô hình GTR. Việc sử dụng học sâu để dự đoán mô hình thay thế cho trình tự axit amin là cần thiết và có nhiều lợi ích lớn như tiết kiệm thời gian và chi phí tính toán. Như chúng ta biết rằng, không thể trực tiếp sử dụng sắp hàng làm dữ liệu cho mô hình học máy, ModelRevelator đã tạo ra 260,000 giá trị thống kê cho mỗi sắp hàng để đào tạo mạng học sâu dựa trên kiến trúc ResNet-18 [53]. Với số lượng

thuộc tính lớn như vậy, các tác giả đã phải dùng GPU cho tác vụ đào tạo mạng ModelRevelator.

Như vậy, chúng ta có thể thấy bài toán đặt ra trong chương này sẽ được phát biểu như sau đây:

Đầu vào: Dữ liệu sắp hàng axit amin.

Bài toán: Phát triển phương pháp tính toán để trích chọn đặc trưng tối đa từ sắp hàng và xây dựng mạng học sâu để lựa chọn nhanh mô hình thay thế axit amin phù hợp nhất với sắp hàng. Yêu cầu mạng học sâu có thể được huấn luyện và chạy được trên các máy tính cá nhân thông thường.

Đầu ra: Mô hình thay thế phù hợp nhất.

Để giải quyết thách thức liên quan đến quá trình đào tạo mạng học sâu, cũng như sự phù hợp của dữ liệu, nhiều công trình nghiên cứu đã loại bỏ lớp tích chập trong kiến trúc mạng để giảm thiểu tối đa quá trình tính toán [81], [82]. Từ đó, trong chương này, luận án giới thiệu một kiến trúc học sâu rất tinh giản và phương pháp trích xuất thông tin từ sắp hàng để có thể dự đoán hiệu quả mô hình thay thế axit amin, phương pháp này được gọi là ModelDetector.

4.2 Phương pháp

4.2.1 Sinh dữ liệu mô phỏng

Như luận án đã trình bày ở Chương 1, chúng ta không có mô hình đúng và cây đúng cho dữ liệu thật. Ngoài ra, dữ liệu thật chứa nhiều yếu tố khó kiểm soát như các ký tự trống (gap), ký tự bị mất (missing), hay sự biến đổi tại các vị trí trên trình tự có thể tuân theo nhiều mô hình thay thế axit amin khác nhau đôi khi là cả mô hình hỗn hợp. Do đó, sử dụng dữ liệu thật để đào tạo mạng học sâu là vấn đề khó và thách thức. Để vượt qua vấn đề này, luận án sử dụng dữ liệu mô phỏng để đào tạo mạng, tuy nhiên, dữ liệu mô

phỏng cần được tạo ra đủ đa dạng và sát với dữ liệu thật nhất để đảm bảo mô hình học được những đặc tính cần thiết của dữ liệu. Đây là một thách thức rất lớn của bài toán học máy cũng như xét riêng trong bài toán lựa chọn mô hình thay thế axit amin. Nếu dữ liệu đủ độ đa dạng và độ bao phủ thì mô hình sẽ có hiệu quả tốt với các loại dữ liệu khác nhau.

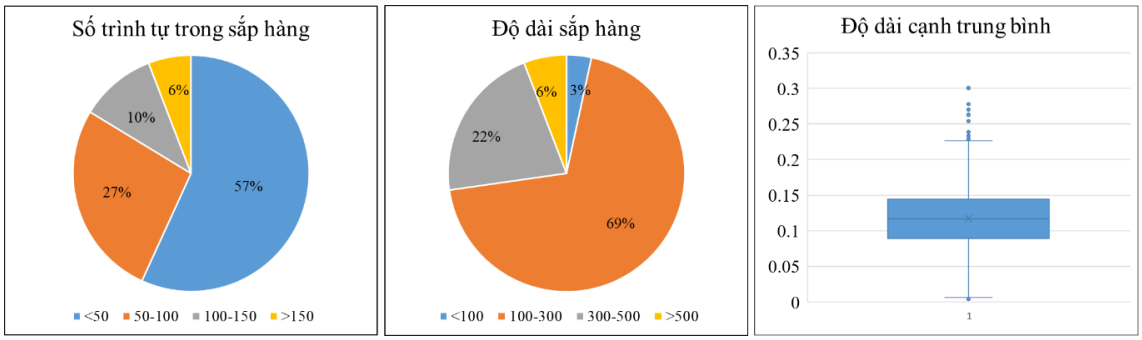
Trong phần này, nhằm đa dạng hóa dữ liệu để đào tạo mạng, luận án sử dụng sáu bộ dữ liệu thật bao gồm: các bộ dữ liệu theo loài như plants, birds, yeasts, mammals, insects và bộ dữ liệu chung HSSP. Để đánh giá mạng, ngoài những phần được tách ra từ các bộ dữ liệu trên, luận án sử dụng thêm bộ dữ liệu độc lập Pfam [35] để đánh giá khả năng và độ tin cậy của phương pháp. Một số thông tin thống kê về các bộ dữ liệu đào tạo được liệt kê trong Bảng 4.1. Bộ dữ liệu yeasts có nhiều trình tự nhất với 332 trình tự, ít nhất là bộ plants với 35 trình tự. Về độ dài trung bình của trình tự thì dài nhất là bộ mammal với 689 vị trí, ngắn nhất là bộ insect với 254 vị trí. Bộ HSSP là bộ dữ liệu chung gồm đa dạng các sắp hàng của nhiều loài khác nhau được sử dụng trong nhiều nghiên cứu khoa học, với trung bình là 76 trình tự và độ dài trung bình là 260 vị trí.

Bảng 4.1. Thông tin thống kê các bộ dữ liệu

Dataset	Số sắp hàng	Số trình tự trong sắp hàng	Độ dài trình tự trung bình	Độ dài cạnh trung bình
Plant	1000	35	330	0.064
Bird	1000	48	676	0.018
Yeast	1000	332	508	0.169
Mammal	1000	82	689	0.027
Insect	1000	134	254	0.157
HSSP	1000	76	260	0.117

Các thông tin chung về 1000 sắp hàng HSSP được thể hiện trong Hình 4.1. Luận án lựa chọn những sắp hàng có ít nhất 30 trình tự và tối thiểu 50 vị trí có sự biến đổi (variant sites). Độ dài cạnh của các cây phân loài trong 1000 sắp hàng này có phạm vi từ

0.08 đến 0.14, có năm sắp hàng mà độ dài cạnh lớn hơn 0.25. Về số trình tự trong sắp hàng, có hơn 57% sắp hàng ít hơn 50 trình tự, khoảng 6% là nhiều hơn 150 trình tự. Trong khi đó, hầu hết các sắp hàng có ít nhất 100 vị trí, chỉ khoảng 3% là ít hơn 100 vị trí.



Hình 4.1. Phân bố độ dài cạnh, số trình tự và số vị trí của 1000 sắp hàng HSSP.

Để thực hiện sinh dữ liệu cho quá trình đào tạo và kiểm tra, chương trình AliSim được sử dụng để tạo dữ liệu đi kèm với các mô hình. Trong phần này, luận án sử dụng 13 mô hình nhằm huấn luyện và kiểm tra mạng học sâu, đó là năm mô hình theo loài Q.plant, Q.bird, Q.yeast, Q.mammal, Q.insect [11] và tám mô hình chung gồm Q.pfam [11], LG [8], WAG [7], JTT [6], VT [83], PMB [32], Blosum62 [38] và Dayhoff [33].

Với mỗi tập dữ liệu 1000 sắp hàng, luận án thực hiện chia thành ba tập con trong đó một tập gồm 800 sắp hàng để tạo dữ liệu cho việc đào tạo mạng (training set), một tập gồm 100 sắp hàng dùng để tạo dữ liệu kiểm tra (test set) và một tập gồm 100 sắp hàng dùng để tạo dữ liệu dùng cho việc xác thực hiệu suất mạng trong quá trình đào tạo (validation set). Như vậy, chúng ta có tương ứng 4800, 600 và 600 sắp hàng dùng cho việc tạo dữ liệu đào tạo, kiểm tra và xác thực.

Luận án cũng sử dụng quy trình tạo dữ liệu như trình bày trong Chương 2, dữ liệu được tạo ra không có ký tự trống. Việc không tạo ký tự trống sẽ đảm bảo cho quá trình đánh phương pháp theo các kích thước sắp hàng khác nhau. Với mỗi sắp hàng thuộc năm

loài, luận án tạo dữ liệu sử dụng mô hình theo đúng loài đó và tám mô hình chung. Với mỗi sắp hàng HSSP, luận án tạo dữ liệu cùng với tất cả 13 mô hình. Các sắp hàng tạo ra có độ dài tương ứng là 100, 300, 500, 700, 900, và 1,100 vị trí. Tổng cộng, với mỗi mô hình, luận án đã tạo ra 172,800 sắp hàng cho đào tạo mạng, 5,400 sắp hàng cho kiểm tra mạng và 5,400 sắp hàng cho xác thực mạng. Cho cả 13 mô hình, luận án đã tạo ra 2,246,400 sắp hàng cho quá trình đào tạo, 70,200 sắp hàng cho việc kiểm tra và 70200 sắp hàng cho việc xác thực mô hình.

4.2.2 Phương pháp trích xuất thông tin theo cặp trình tự

Thiết lập các tổng hợp thống kê (summary statistics) từ các sắp hàng axit amin là một trong các bước quan trọng nhất của quá trình đào tạo các mạng học sâu để nhận biết các mô hình thay thế axit amin. Do có tổng cộng 20 axit amin nên mô hình *Q* bao gồm 400 hệ số tốc độ biến đổi giữa các cặp axit amin. Để trích xuất thông tin thống kê trong sắp hàng, luận án lựa chọn ngẫu nhiên $N = 1000$ cặp trình tự trong sắp hàng. Với mỗi cặp trình tự, luận án sẽ thực hiện đếm số lần thay thế axit amin trong đó và tổng hợp toàn bộ thông tin của *N* lần lặp. Cuối cùng, luận án thu được 400 giá trị đã được chuẩn hóa thể hiện tốc độ biến đổi của các axit amin trong sắp hàng. 400 giá trị này hình thành nên ma trận 20×20, đặt tên là $F_2 = \{f_2(xy)\}$, trong đó $f_2(xy)$ là số lần hai axit amin *x* và *y* xuất hiện trên cùng 1 vị trí của các cặp trình tự. Luận án không thống kê các ký tự trống trong sắp hàng vì nó không mang thông tin hữu ích cho việc huấn luyện mạng.

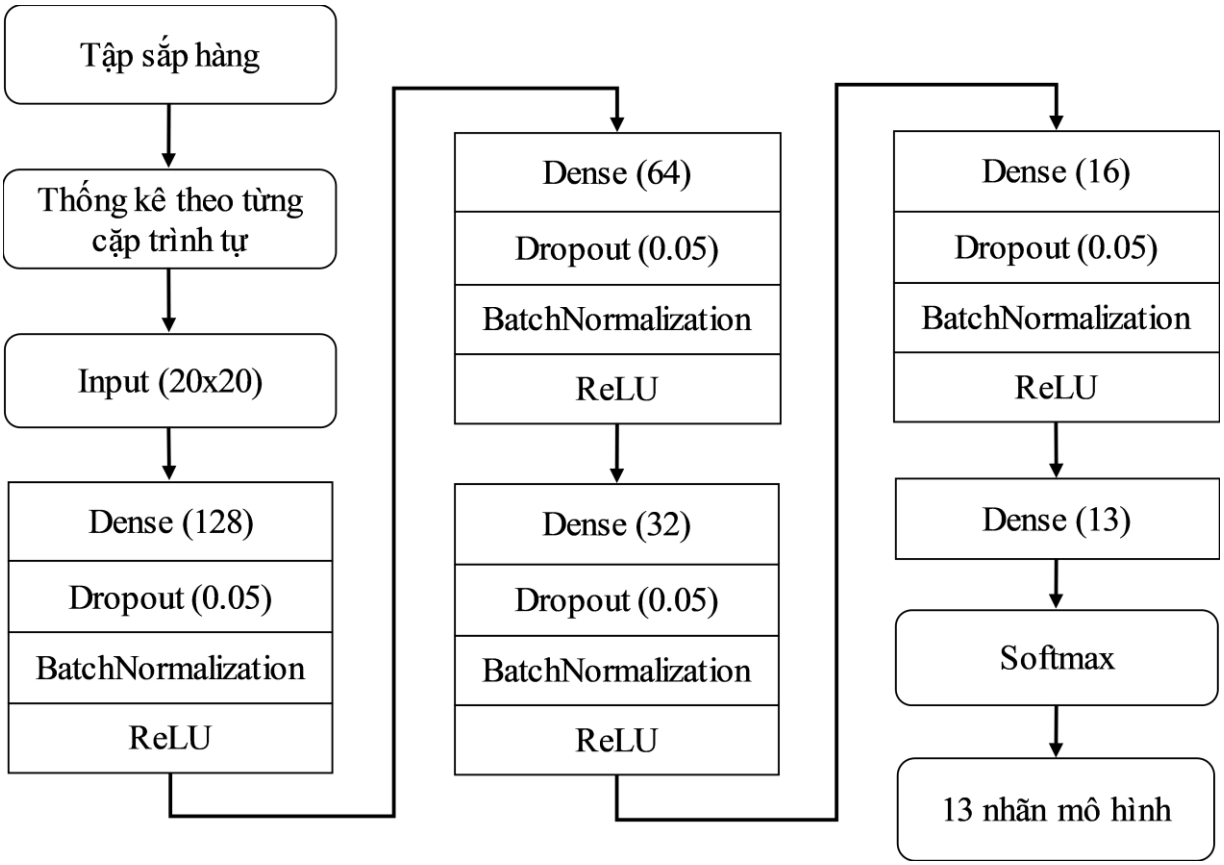
4.2.3 Kiến trúc mạng

Mục tiêu của luận án ở phần này là tìm một phương pháp dựa trên trí tuệ nhân tạo vừa có thể lựa chọn nhanh mô hình mà lại có thể hoạt động được trong môi trường máy tính cá nhân. Trước tiên, luận án thử nghiệm phương pháp học máy đơn giản như máy vector hỗ trợ SVM. Kết quả dự đoán trên tập kiểm tra cho thấy phương pháp này đạt độ chính xác 85.6%. Kết quả này tuy đạt trên 85% nhưng vẫn chưa thể so sánh với phương

pháp ModelFinder. Do vậy, luận án hướng đến xây dựng các mạng học sâu để có thể phân tích các thuộc tính của dữ liệu một cách tốt hơn và cũng chủ trương phát triển các phương pháp tính toán nhằm tối đa hóa đặc trưng của dữ liệu sắp hàng.

Như đã nói ở phần trước, luận án sẽ thiết kế một mạng học sâu đơn giản và không sử dụng đến lớp tích chập. Trong các mạng học sâu phân loại ảnh, lớp tích chập là thành phần quan trọng giúp trích xuất các thuộc tính như cạnh, viền và hình dạng của đối tượng trong ảnh. Với dữ liệu tổng hợp thống kê từ sắp hàng, từng giá trị đã mang ý nghĩa như một thuộc tính, ngoài ra mọi vị trí trong dữ liệu đều có vai trò quan trọng, do vậy luận án không dùng đến lớp tích chập để tối ưu thời gian đào tạo mạng, phương pháp này được gọi là ModelDetector, xem chi tiết kiến trúc trong Hình 4.2. ModelDetector sử dụng hàm kích hoạt ReLU cho các lớp bên trong, hàm Softmax cho lớp cuối cùng và sử dụng hàm tối ưu Adam [84]. Nhãn của mỗi sắp hàng mô phỏng chính là mô hình thay thế axit amin được sử dụng để sinh ra sắp hàng đó. Luận án thực hiện gán nhãn cho từng sắp hàng với 1 trong 13 nhãn: Q.plant, Q.bird, Q.yeast, Q.mammal, Q.insect, Q.pfam, LG, WAG, JTT, VT, PMB, Blosum62, và Dayhoff.

Với 2,246,400 sắp hàng để huấn luyện mạng, luận án dùng 100 epoch với batch size 40. Bên cạnh đó, nhằm tránh hiện tượng quá khớp (overfitting), luận án sử dụng lớp DropOut để ngẫu nhiên loại bỏ 5% số nút mạng và dùng hàm EarlyStopping callback để dừng quá trình huấn luyện ngay khi độ chính xác trên tập xác thực không tốt hơn trong 5 epoch liên tục. Để huấn luyện mạng, luận án sử dụng bộ thư viện Tensorflow 2.13.1 [85] và chạy trên một máy tính với 8 cores không bao gồm GPU. Trước đó, quá trình tối ưu tham số được thực hiện bởi hàm GridSearchCV. Hàm này sẽ quét toàn bộ các tổ hợp cấu hình batch size, dropout, optimizer, learning rate, rồi đánh giá độ chính xác (accuracy) trên tập xác thực (validation set) để lựa chọn được tổ hợp tối ưu nhất dùng cho bộ dữ liệu.

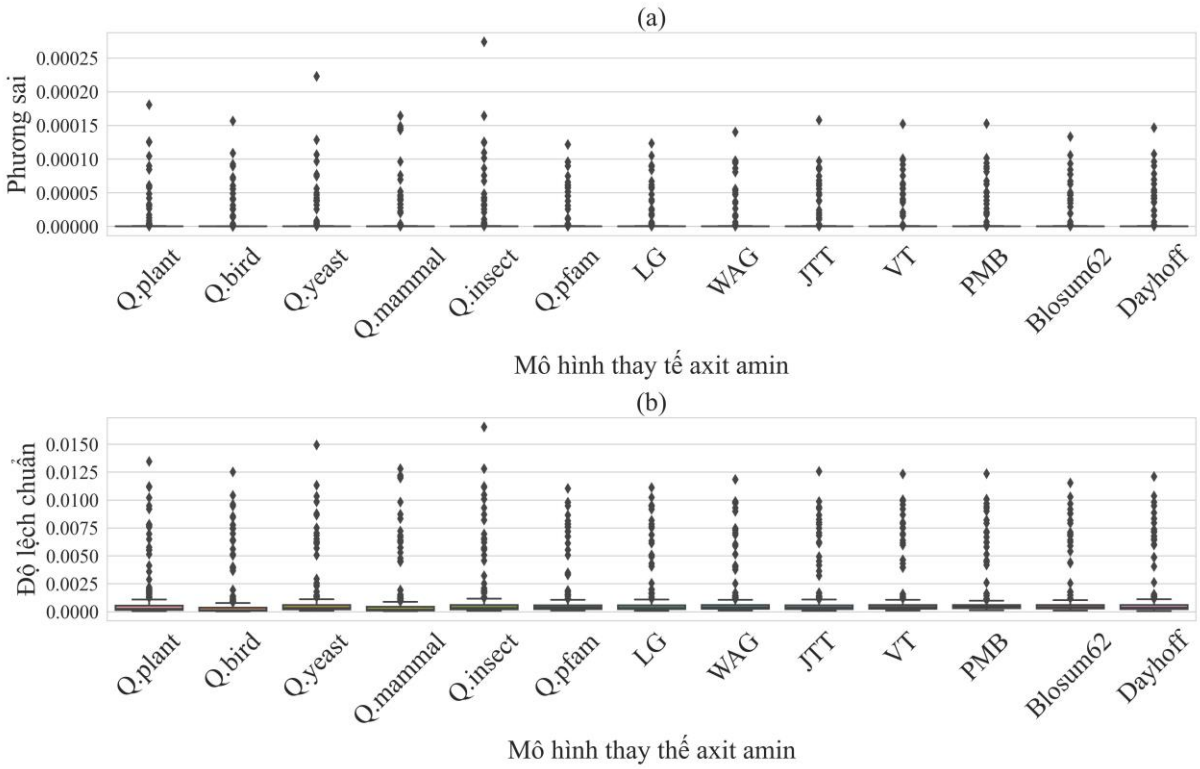


Hình 4.2. Kiến trúc mạng học sâu được sử dụng để huấn luyện ModelDetector từ tập các sắp hàng mô phỏng.

4.3 Kết quả

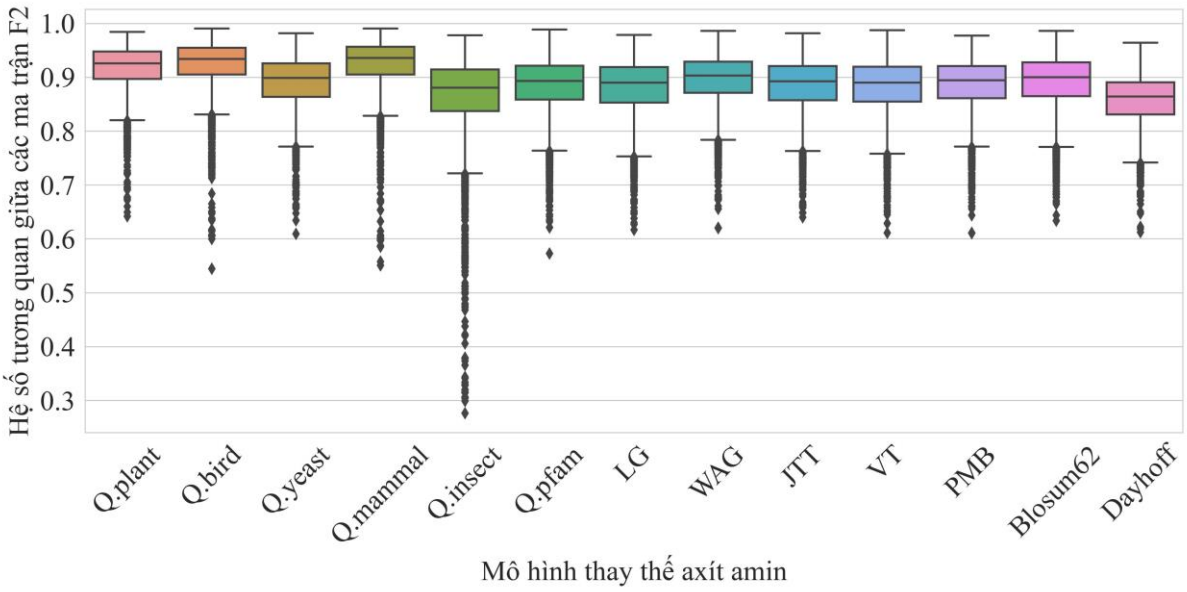
4.3.1 Phân tích các giá trị thống kê

Luận án thực hiện phân tích để xác định mối liên hệ giữa các giá trị tổng hợp thống kê từ các sắp hàng thuộc bộ dữ liệu thật và từ các sắp hàng mô phỏng. Trước tiên, luận án phân tích phân bố của những tổng hợp thống kê này bằng cách tính phương sai và độ lệch chuẩn theo từng sắp hàng, xem Hình 4.3. Hầu hết các thông tin tổng hợp thống kê có phương sai thấp, chỉ một vài giá trị là vượt quá 0.00015. Phân phối phương sai dường như tương tự nhau qua các mô hình. Sự tương tự này cũng xảy ra với phân phối của độ lệch chuẩn, chỉ một vài giá trị lớn hơn 0.01.



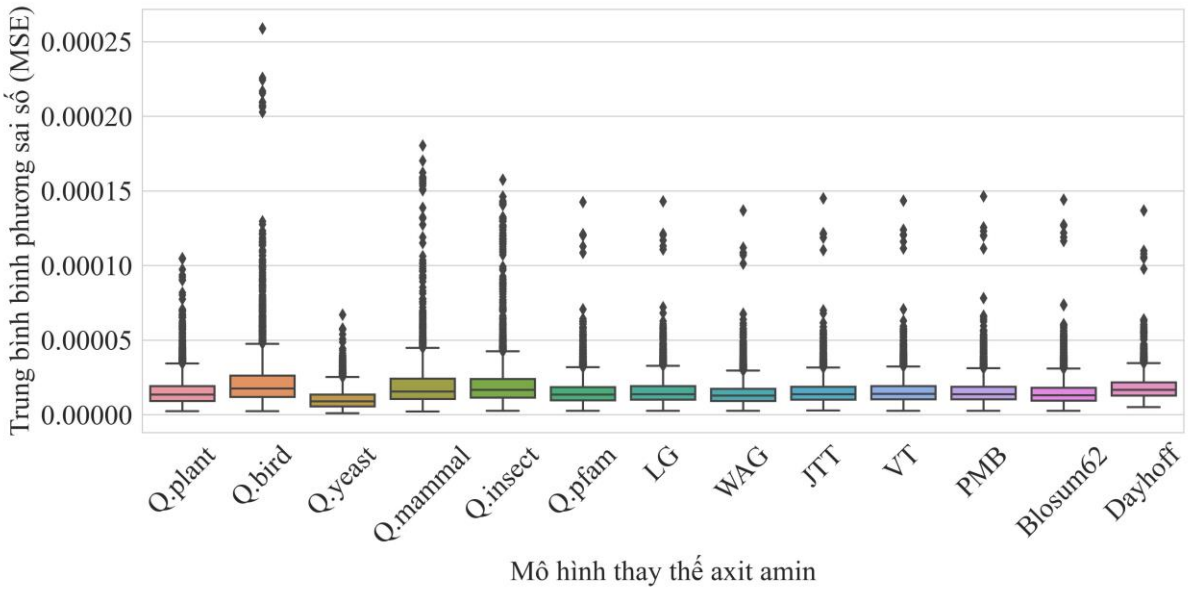
Hình 4.3**.** Phương sai (Variance) và độ lệch chuẩn (Standard deviation) của các ma trận F2 thống kê từ các sắp hàng mô phỏng.

Tiếp theo, luận án tính toán giá trị tương quan Pearson giữa ma trận F2 thu được từ sắp hàng thực và ma trận thu được từ sắp hàng mô phỏng. Kết quả trong Hình 4.4 cho thấy các ma trận F2 có độ tương quan mạnh với nhau. Giá trị trung bình từ 0.85 với mô hình Dayhoff tới 0.92 cho mô hình Q.mammal. Có tới 95% sắp hàng cho giá trị tương quan trên 0.8 cho thấy các dữ liệu mô phỏng đã tương đối sát với dữ liệu thật, chỉ còn một số trường hợp có sự khác biệt. Luận án cũng thực hiện đánh giá mức ý nghĩa của các giá trị tương quan thông qua tính toán p-value sử dụng kiểm định Student t-test. Giả thuyết Null H_0 là các ma trận F2 không tương quan với nhau. Toàn bộ *p*-value đều rất nhỏ, gần với 0 cho thấy các ma trận này thực sự có sự tương quan với nhau.



Hình 4.4. Phân bố hệ số tương quan Pearson giữa các ma trận F2 của dữ liệu thật và dữ liệu mô phỏng.

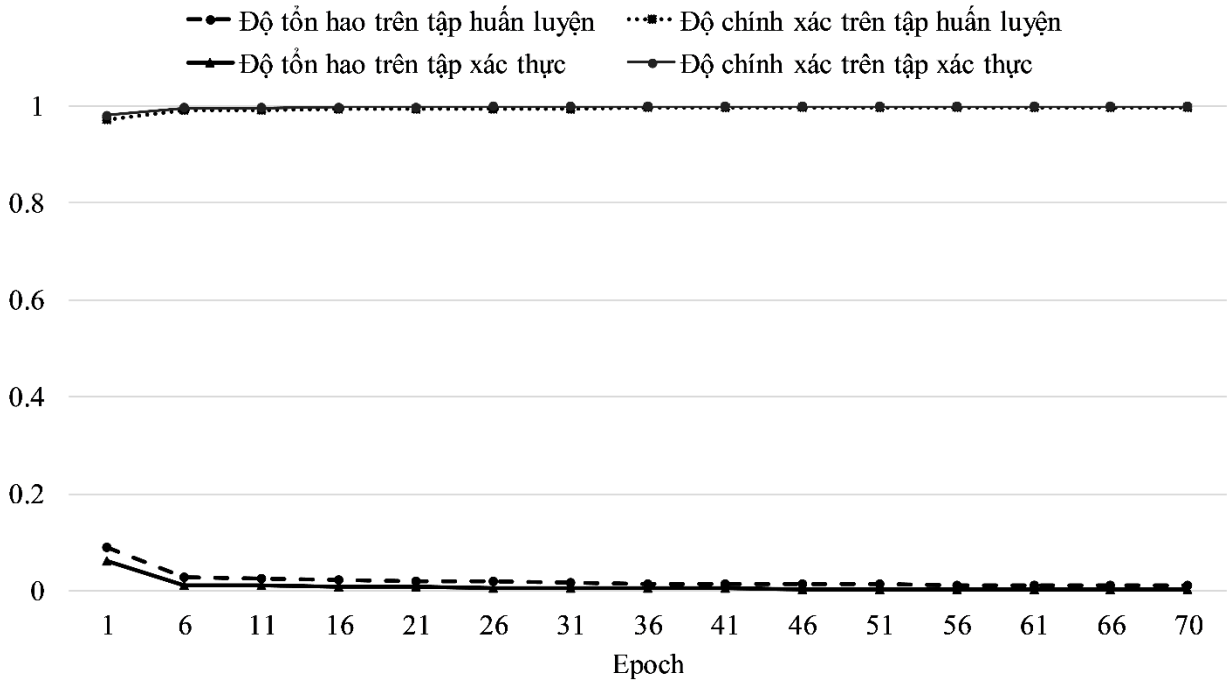
Ngoài ra, luận án cũng đánh giá thêm phân bố của các giá trị trung bình bình phương sai số (MSE) giữa các ma trận F2 từ dữ liệu thật và dữ liệu mô phỏng. Với mỗi mô hình *M*, luận án tính toán 400 giá trị MSE tương ứng với 400 phần tử của *M*, kết quả được thể hiện trong Hình 4.5. Toàn bộ các giá trị MSE đều rất nhỏ (<0.0003) một lần nữa thể hiện độ tương đồng cao giữa các sắp hàng. Đồng thời, kết quả cũng cho thấy sự hiệu quả của quá trình tạo dữ liệu mô phỏng.



Hình 4.5. Phân bố MSE giữa các ma trận F2 của dữ liệu thật và dữ liệu mô phỏng.

4.3.2 Đánh giá phương pháp trên bộ dữ liệu mô phỏng

Sau khi xây dựng kiến trúc mạng và thực hiện thu thập các tổng hợp thống kê từ dữ liệu đào tạo, luận án tiến hành quá trình huấn luyện mạng theo các thông số như đã trình bày ở các phần trước. Hình 4.6 thể hiện độ chính xác cao trong quá trình huấn luyện mạng. Độ chính xác trên tập xác thực đạt 0.99 và giá trị tổn hao trên tập xác thực giảm xuống 0.01 sau một vài epoch. Quá trình huấn luyện mạng kết thúc sau 70 epoch với độ chính xác trên tập xác thực là 0.998. Độ chính xác và độ tổn hao tương ứng giữa tập huấn luyện và tập xác thực chênh lệch rất nhỏ cho thấy mạng không bị hiện tượng quá khớp.

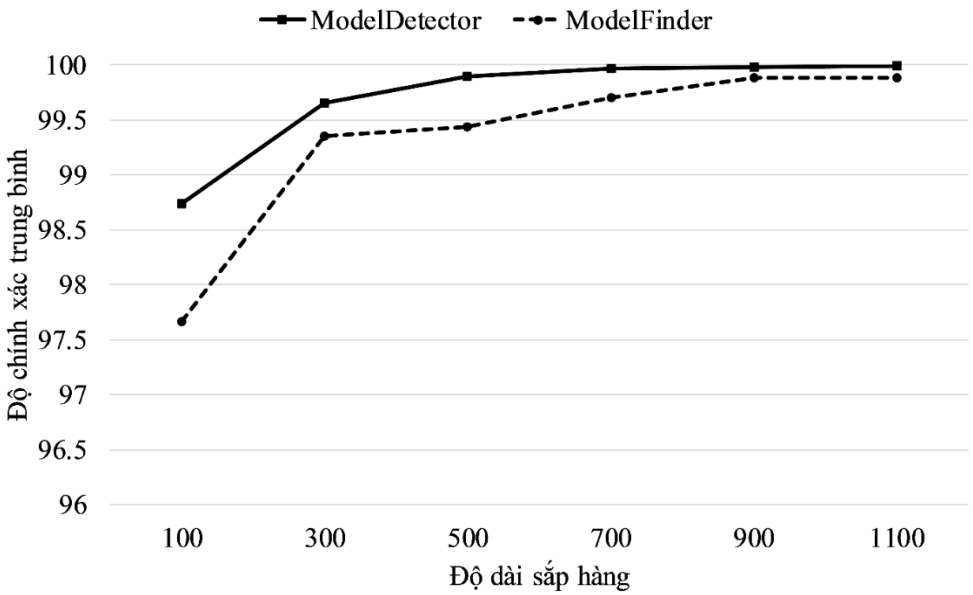


Hình 4.6. Độ chính xác và tổn hao trên tập đào tạo và tập xác thực trong quá trình huấn luyện mạng ModelDetector.

Tiếp theo, luận án thực hiện đánh giá hiệu suất của mạng ModelDetector và so sánh với phương pháp rất thông dụng dựa trên tiêu chuẩn cực đại hợp lý là ModelFinder [13]. Luận án chạy cả hai phương pháp trên tập dữ liệu kiểm tra rồi thống kê số lượng sắp hàng được các phương pháp dự đoán đúng, xem kết quả trong Hình 4.7. Ở đây, ModelFinder được thực thi trong giới hạn 13 mô hình, khi thực thi, nó sẽ tìm ra mô hình tốt nhất bằng cách quét toàn bộ các kết hợp giữa mô hình thay thế với mô hình tốc độ biến đổi tại các vị trí, ví dụ LG+G4+I hay Q.pfam+G4+I.

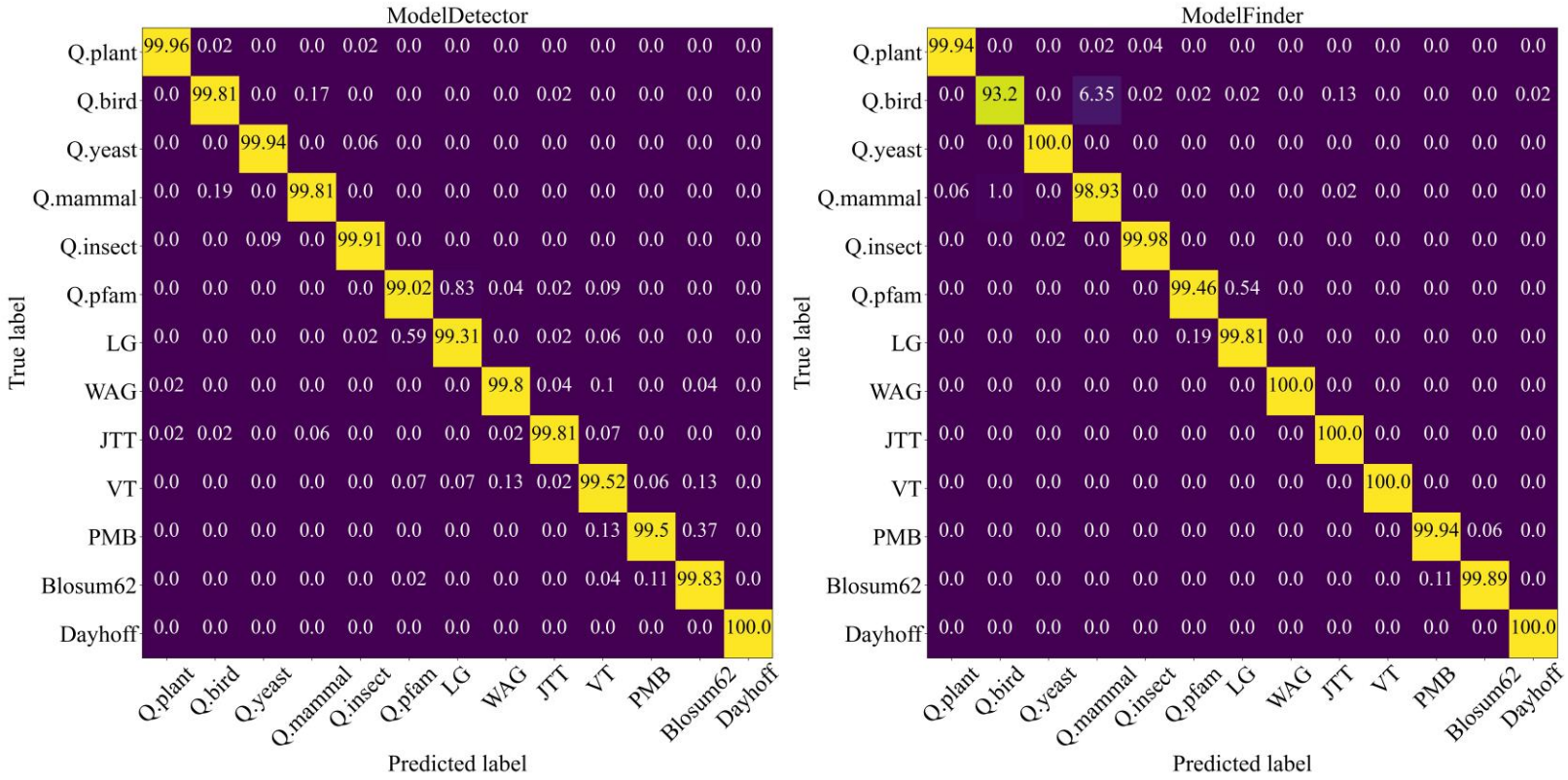
Mạng ModelDetector đã có hiệu suất tương đương so với ModelFinder với độ chính xác trung bình của hai phương pháp là 99.71% và 99.32% tương ứng. Độ chính xác có xu hướng tăng dần theo từng kích thước sắp hàng, điều này hoàn toàn hợp lý vì sắp hàng càng dài thì lượng thông tin thống kê càng lớn. Ví dụ, với những sắp hàng có

100 vị trí, độ chính xác của ModelDetector là 98.74% và độ chính xác tăng lên 99.99% khi độ dài sắp hàng là 1100 vị trí.



Hình 4.7. Độ chính xác của ModelDetector và ModelFinder trên tập dữ liệu kiểm tra.

Để chi tiết hơn kết quả dự đoán, luận án thống kê cụ thể nhãn của những sắp hàng dự đoán sai, kết quả được biểu diễn trong ma trận nhầm lẫn trong Hình 4.8. Theo đó, ModelDetector dự đoán có độ chính xác lớn hơn 99% ở tất cả các mô hình. Một số sắp hàng Q.pfam được dự đoán thành LG, kết quả này là do hai mô hình được ước lượng từ một bộ dữ liệu và khá tương đồng nhau. Phương pháp ModelFinder cũng hoạt động tốt trên hầu hết các mô hình ngoại trừ Q.bird (93.2%) với 343 trên tổng 5400 sắp hàng Q.bird được dự đoán thành Q.mammal. Kết quả kém hơn ở Q.bird có lẽ là do độ tương quan khá cao ở hai mô hình này (0.98) dẫn đến khi chạy ModelFinder thì giá trị BIC của hai mô hình chênh lệch nhau rất ít. Và như luận án cũng thảo luận ở phần trước, đây chính là vấn đề khi sử dụng các tiêu chuẩn như AIC và BIC để nhận biết mô hình tốt nhất vì mô hình đúng có thể không có AIC hay BIC tốt nhất.



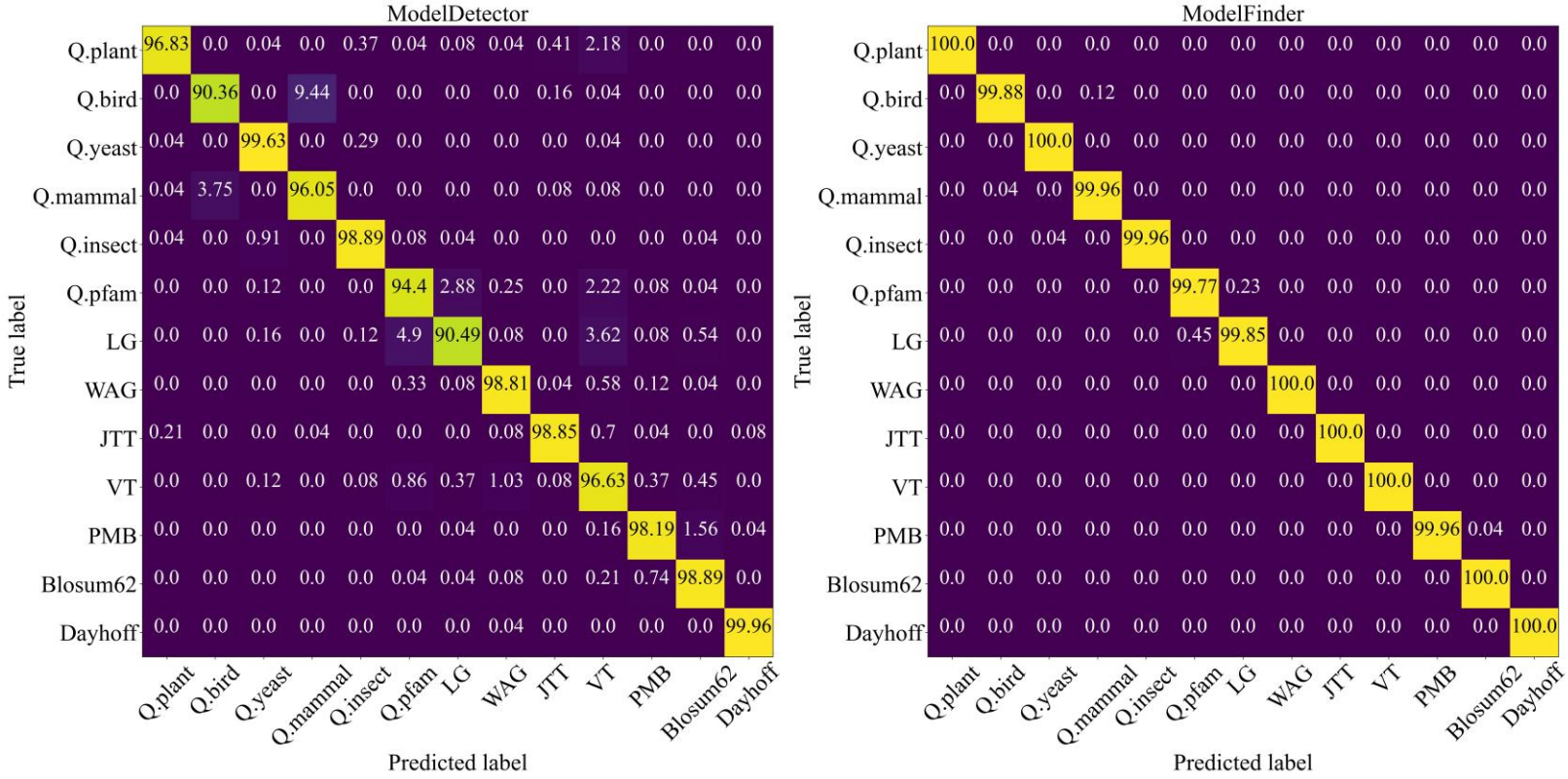
Hình 4.8. Kết quả dự đoán của hai phương pháp ModelDetector và ModelFinder trên bộ dữ liệu kiểm tra. True label: mô hình đúng của sắp hàng, Predicted label: mô hình được dự đoán bởi các phương pháp.

4.3.3 Đánh giá phương pháp trên bộ dữ liệu độc lập Pfam

Nhằm tăng tính khách quan và đánh giá toàn diện hơn phương pháp ModelDetector, luận án sử dụng thêm một bộ dữ liệu độc lập là bộ Pfam [35] để kiểm nghiệm khả năng của phương pháp trên dữ liệu thật lẫn dữ liệu mô phỏng dựa trên bộ Pfam.

Luận án chọn ra ngẫu nhiên 1000 sắp hàng Pfam trong đó mỗi sắp hàng có ít nhất 30 trình tự và 100 vị trí có sự biến đổi. Các sắp hàng này có độ dài có thể chia thành bốn nhóm như sau: nhóm một có 658 sắp hàng độ dài từ 100 đến 300 vị trí, nhóm hai có 210 sắp hàng có độ dài từ 300 đến 500 vị trí, nhóm ba gồm 74 sắp hàng có độ dài từ 500 đến 700 vị trí và nhóm bốn gồm 58 sắp hàng có độ dài trên 700 vị trí. Từ 1000 sắp hàng này, luận án sử dụng AliSim để mô phỏng bắt trước cả về độ dài lẫn các ký tự trống của sắp hàng gốc. Theo đó, cho từng mô hình thay thế, luận án sinh ra tương ứng 1, 3, 9 và 11 sắp hàng từ mỗi sắp hàng thuộc về các nhóm 1, 2, 3 và 4 như nêu trên đồng thời thực hiện loại bỏ những sắp hàng có nhiều hơn 50% ký tự trống. Kết quả là luận án đã tạo ra 31,587 sắp hàng mô phỏng theo 13 mô hình với kích thước tối thiểu 100 vị trí có sự biến đổi. Từ đó, luận án tiến hành đánh giá hiệu năng của cả hai phương pháp ModelDetector và ModelFinder trên tập sắp hàng này, kết quả chi tiết như trong Hình 4.9. Về giá trị trung bình, ModelDetector đã thực hiện dự đoán đúng 96.77% số sắp hàng trong đó thấp nhất đạt 90.36%% với sắp hàng thuộc mô hình Q.bird và 99.96% với mô hình Dayhoff. ModelFinder vẫn giữ được độ ổn định tốt khi hầu hết các sắp hàng đều được dự đoán đúng, độ chính xác trung bình là 99.95% trong đó nhiều mô hình đạt 100% như Q.plant, Q.yeast, WAG, JTT, VT, PMB, Blosum62 và Dayhoff. ModelDetector đã gán nhãn sai 9.44% sắp hàng Q.bird thành Q.mammal, điều này do sự tương đồng cao giữa hai mô hình này khi độ tương quan Pearson của chúng là 0.98. Ngoài ra, có 4.9% sắp hàng thuộc mô hình LG cũng được dự đoán thành Q.pfam, điều này xảy ra là vì cả hai mô hình này đều được ước lượng từ cùng một bộ dữ liệu. Độ chính xác của ModelDetector tăng lên

theo kích thước của sắp hàng. Với những sắp hàng có trên 500 vị trí, ModelDetector đã gán nhãn đúng 99.63%.



Hình 4.9. Kết quả dự đoán hai phương pháp trên bộ dữ liệu mô phỏng tạo từ tập Pfam. True label: mô hình đúng của sắp hàng, Predicted label: mô hình được dự đoán bởi các phương pháp.

Luận án tiếp tục lựa chọn ra 886 sắp hàng thật có tối thiểu 100 vị trí có sự biến đổi thuộc bộ Pfam để đánh giá khả năng chạy trên dữ liệu thật của phương pháp ModelDetector. Do không biết mô hình đúng của các sắp hàng, luận án thực hiện so sánh kết quả của ModelDetector với kết quả của ModelFinder để xem có bao nhiêu sắp hàng mà hai phương pháp dự đoán trùng nhau, kết quả chi tiết xem Bảng 4.2. Theo đó, số lượng sắp hàng mà hai phương pháp dự đoán cùng mô hình tăng lên theo kích thước sắp hàng. Ví dụ, hai phương pháp dự đoán đúng ở mức 60.98% sắp hàng có độ dài 100-300 vị trí và 66.67% sắp hàng có trên 500 vị trí. Xem xét giá trị BIC của mô hình tốt nhất và mô hình tốt thứ hai trong kết quả của ModelFinder, luận án thấy rằng các giá trị này có độ chênh lệch rất nhỏ. Kết quả này dường như do hiện tượng không đồng nhất về tốc độ biến đổi tại các vị trí (RHAS) nên các vị trí có thể tuân theo các mô hình thay thế khác nhau [86]. Đôi khi mô hình đúng không phải là có BIC tốt nhất, luận án tiếp tục thống kê thêm sự trùng lặp ở mô hình tốt thứ hai. Với cách thống kê này, hai phương pháp có kết quả trùng nhau ở 82.09% những sắp hàng từ 100 đến 300 vị trí và tăng tới 83.33% trên các sắp hàng ít nhất 500 vị trí.

Bảng 4.2. Số lượng sắp hàng thật thuộc bộ Pfam mà cả hai phương pháp ModelDetector và ModelFinder dự đoán cùng một mô hình

Số vị trí trên sắp hàng	Số sắp hàng mà ModelDetector trùng với mô hình tốt nhất của ModelFinder	Số sắp hàng mà ModelDetector trùng với mô hình tốt thứ hai của ModelFinder
100-300	60.98%	82.09%
300-500	62.51%	82.16%
>500	66.67%	83.33%

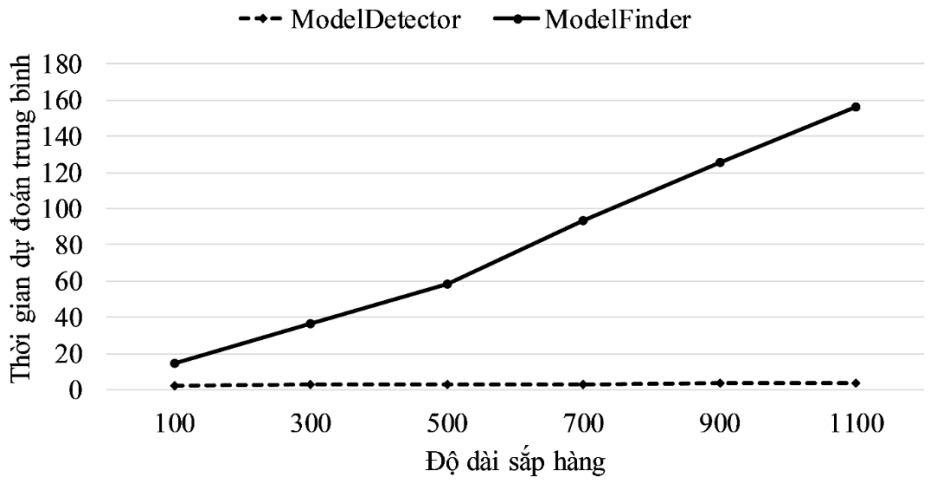
Với các sắp hàng được dự đoán khác nhau, luận án đánh giá khoảng cách nRF [51] giữa các cây được xây dựng với mô hình dự đoán bởi ModelDetector và mô hình tốt nhất của ModelFinder. Giá trị nRF trung bình là 0.121. Với mỗi sắp hàng được dự đoán trái

ngược này, luận án cũng thử nghiệm chạy IQ-TREE ba lần để xây dựng ba cây phân loài khác nhau với kịch bản là chạy với cùng một mô hình nhưng bằng các số nhân ngẫu nhiên khác nhau. Mục đích của việc này là đánh giá kết quả xây dựng cây của IQ-TREE xem có đồng nhất hay không. Kết quả là các cây này có nRF trung bình là 0.143. Từ các kết quả trên, chúng ta có thể thấy độ tương đồng cao trong kết quả dự đoán của ModelDetector và ModelFinder.

4.3.4 Phân tích thời gian dự đoán mô hình

Yếu tố thời gian chạy là một trong những ưu điểm của phương pháp dựa trên học sâu. ModelDetector được huấn luyện rất nhanh trên các máy tính có 8 lõi mà không cần bất kì phần cứng GPU nào. Việc loại bỏ lớp tích chập đã làm giảm đáng kể thời gian huấn luyện khi giờ đây, mỗi epoch chỉ còn tiêu tốn 168 giây. Tổng cộng, để thu được các tham số mạng cho 2,246,400 sắp hàng, luận án chỉ cần chạy huấn luyện trong khoảng 3.3 giờ sau 70 epoch.

Giờ đây, việc dự đoán lại càng diễn ra nhanh hơn. Luận án thực hiện thống kê thời gian chạy theo các kích thước sắp hàng khác nhau, thời gian chạy được tính theo thời gian tổng hợp thống kê và thời gian xử lý của mạng để đưa ra kết quả cuối cùng, chi tiết xem Hình 4.10. Với các dữ liệu mô phỏng, thời gian chạy của ModelDetector nhanh hơn ModelFinder ở toàn bộ số trường hợp. Thời gian chạy của ModelFinder tăng rất nhanh theo kích thước sắp hàng, ví dụ, ModelFinder mất lần lượt 14.3, 58.2 và 156.6 giây để tìm ra mô hình tốt nhất của sắp hàng có 100, 500 và 1100 vị trí. Trong khi đó, ModelDetector gần như không thay đổi nhiều thời gian chạy với những kích thước này. Cụ thể, ModelDetector cần 2.4 giây để dự đoán một sắp hàng có 100 vị trí và 3.6 giây cho một sắp hàng có 1100 vị trí.



Hình 4.10. Thời gian dự đoán (đơn vị: giây) của ModelDetector và ModelFinder.

Luận án cũng kiểm tra xem với những sắp hàng có hàng trăm nghìn đến hàng triệu vị trí thì thời gian chạy hai phương pháp này như thế nào. Luận án tạo ra các sắp hàng có 50 trình tự và độ dài tăng dần từ 1,000 đến 1,000,000 vị trí. Kết quả là, ModelDetector chỉ cần khoảng 693 giây để đưa ra kết quả dự đoán cho sắp hàng có 1,000,000 vị trí, nhanh gấp 27 lần so với ModelFinder, kết quả chi tiết xem trong Bảng 4.3.

Bảng 4.3. Thời gian dự đoán của hai phương pháp ModelDetector và ModelFinder trên các sắp hàng có từ 1,000 đến 1000,000 vị trí

Độ dài sắp hàng	ModelDetector			ModelFinder
	Tổng hợp thống kê	Dự đoán mô hình	Tổng	
1,000	1.3	1.8	3.1	10.4
10,000	6.6	1.9	8.5	37.7
50,000	32.6	1.9	34.5	262.2
100,000	67.3	1.9	69.2	622.8
500,000	332.8	1.9	334.7	8,255.3
1,000,000	691.4	1.8	693.2	18,684.5

4.4 Tổng kết chương

Trong chương này, luận án đã trình bày một kiến trúc mạng dựa trên học sâu cùng với phương pháp trích chọn thông tin tổng hợp thống kê từ sắp hàng axit amin để dự đoán mô hình thay thế của sắp hàng đó. Trong khi phương pháp ModelRevelator sử dụng tới 260,000 thuộc tính và cần dùng phần cứng GPU chuyên dụng để huấn luyện mô hình thì phương pháp của luận án, gọi là ModelDetector, chỉ cần dùng 400 giá trị tổng hợp thống kê để huấn luyện mạng và dự đoán mô hình của sắp hàng. Phương pháp ModelDetector có thời gian huấn luyện và dự đoán vượt trội so với ModelRevelator và độ chính xác có thể so sánh được với ModelFinder. Tuy nhiên, một giới hạn trong phương pháp dựa trên học sâu là chỉ có thể dự đoán được mô hình mà đã được huấn luyện. Như ModelDetector thì chỉ dự đoán dc 13 mô hình, muốn mở rộng thì bắt buộc chúng ta phải tạo thêm dữ liệu đào tạo và thực hiện lại việc huấn luyện mạng. May mắn là ModelDetector có thể được huấn luyện rất nhanh nên việc đó là hoàn toàn khả thi.

Một vấn đề nữa là mạng ModelDetector được dự đoán chỉ dựa vào các sắp hàng sinh ra với mô hình tốc độ tại các vị trí là +I+G, do vậy nó chưa thể dự đoán tốt được những sắp hàng không sử dụng mô hình tốc độ hoặc sử dụng mô hình tần số +F. Ngoài ra, các nhà nghiên cứu gần đây cũng cung cấp thêm các mô hình hỗn hợp mới nhằm tăng tính thực tế cũng như độ tin cậy của cây phân loài [77, 87-89]. Do vậy, dự đoán nhanh các mô hình hỗn hợp cũng là một hướng đi cho bài toán dựa trên học sâu này.

Dù cho vẫn còn nhiều điểm cần phải tối ưu và xem xét, nhưng luận án cho rằng phương pháp này rất hứa hẹn trong việc dự đoán mô hình thay thế axit amin và hoàn toàn có thể trở thành xu hướng tốt trong nghiên cứu tiến hóa.

KẾT LUẬN

1. Những kết quả đạt được và ý nghĩa

Phát triển và đánh giá phương pháp ước lượng mô hình thay thế axit amin luôn là bài toán quan trọng trong nghiên cứu sinh học tiến hóa. Ước lượng mô hình thay thế axit amin thường phức tạp và tốn kém cả về thời gian lẫn tài nguyên tính toán. Nhiều phương pháp đã được đề xuất để tối ưu quá trình ước lượng này. Trong luận án này, tôi đã thực hiện các phân tích lý thuyết và thực nghiệm để giải quyết ba bài toán cơ bản gồm: đánh giá phương pháp ước lượng mô hình thay thế axit amin, đề xuất phương pháp ước lượng nhanh mô hình thay thế axit amin sử dụng đa ma trận và cuối cùng là đề xuất phương pháp lựa chọn nhanh mô hình thay thế axit amin cho một sắp hàng bất kì sử dụng học sâu.

Trước tiên, luận án đưa ra một quy trình để khám phá hiệu suất các mô hình ước lượng từ dữ liệu mô phỏng. Qua các thực nghiệm trên hai bộ dữ liệu thử nghiệm, luận án đã xác nhận mối liên hệ giữa kích thước của dữ liệu đào tạo với hiệu suất mô hình đầu ra. Trong một số bài toán, đôi khi chúng ta muốn nhanh chóng có được mô hình thay thế axit amin trong thời gian hợp lý mà hiệu suất mô hình có thể chấp nhận được. Từ đó, luận án đã đưa ra một số đề xuất về khối lượng dữ liệu cũng như các thay đổi của cây phân loài khi dữ liệu thay đổi. Việc này nhằm mục đích giúp các nhà khoa học có thể có phương án tốt nhất về mặt dữ liệu cho bài toán của mình.

Tiếp đó, luận án trình bày QMix, là một phương pháp mới dùng để ước lượng mô hình thay thế axit amin sử dụng đa ma trận. Phương pháp mới này được thể hiện dưới dạng một công cụ phần mềm, với giao diện đơn giản, dễ sử dụng cũng như nhanh chóng đưa ra kết quả cho người dùng. Chỉ với một lệnh duy nhất, QMix sẽ thực hiện quá trình ước lượng và đưa ra các mô hình mới. QMix hỗ trợ các tùy chọn giúp người dùng dễ dàng sử dụng như: các thuộc tính thuận nghịch hay không thuận nghịch, mô hình thay

thế khởi tạo cũng như số lượng ma trận đầu ra. Luận án cũng đã thử nghiệm QMix trên các bộ dữ liệu và giới thiệu một số mô hình thay thế axit amin mới, cụ thể, luận án giới thiệu hai mô hình chung dựa trên thuộc tính không thuận nghịch về thời gian nT4X và nT4M tương ứng có mô hình tốc độ tại vị trí tuân theo phân phối tự do và phân phối gamma. Luận án cũng đề xuất hai mô hình thay thế đa ma trận cho bộ dữ liệu thực vật, QPlant.mix và nQPlant.mix. Các mô hình mới này đã thể hiện được những ưu điểm trong việc nghiên cứu tiến hóa trình tự axit amin như khả năng xây dựng cây có gốc, đây là một sự cải tiến so với các mô hình đa ma trận đang tồn tại.

Cuối cùng, luận án trình bày phương pháp lựa chọn mô hình thay thế axit amin mới sử dụng mạng học sâu, gọi là ModelDetector. Với phương pháp này, người dùng có thể nhanh chóng lựa chọn mô hình tối ưu nhất cho một sắp hàng cho trước với độ chính xác tương đương ModelFinder. Một trong những đặc trưng của ModelDetector là thời gian huấn luyện và dự đoán nhanh. Hiện tại, ModelDetector có thể hỗ trợ tìm mô hình tối ưu trong số 13 mô hình được huấn luyện với độ chính xác trên 96%, đây là những mô hình phổ biến nhất thường được sử dụng do đó có thể giúp ích rất nhiều cho các nhà khoa học. Quá trình thực nghiệm và so sánh kết quả với phương pháp ModelFinder dựa trên tiêu chuẩn cực đại hợp lý cho thấy đây là một hướng đi mới đầy khả quan và triển vọng trong nghiên cứu tin sinh học tiến hóa.

2. Những hạn chế và hướng phát triển tiếp theo

Ngoài những kết quả đã đạt được, luận án nhận thấy còn một số điểm hạn chế đòi hỏi những nghiên cứu và phân tích sâu hơn. Đầu tiên với quy trình đánh giá phương pháp ước lượng, luận án đang chỉ tập trung vào phương pháp ước lượng dựa trên tiêu chuẩn cực đại hợp lý, do đây đang là phương pháp thông dụng và hiệu quả, vì vậy không thể áp dụng quy trình này sang các phương pháp khác. Luận án hiện đang chỉ sử dụng hai bộ dữ liệu plant và bird để đánh giá, mặc dù hai bộ dữ liệu này được coi là tiêu chuẩn nhưng cũng cần mở rộng việc đánh giá trên các bộ dữ liệu khác như mammal, yeast hay

insect. Tiếp đó, chúng ta cũng cần xem xét kĩ hơn đến các khía cạnh khác như sự khác biệt và đa dạng của dữ liệu, liệu với tập dữ liệu đa dạng các loài thì hiệu suất của mô hình sẽ thay đổi như thế nào theo kích thước dữ liệu?

Với phương pháp ước lượng mô hình đa ma trận, câu hỏi là liệu có thể tự động xác định số lượng ma trận phù hợp nhất với dữ liệu hay không? Hiện tại, chúng ta cần xác định sẵn số lượng ma trận trước khi dùng phương pháp QMix để ước lượng mô hình. Như vậy, số ma trận đó có thể không thực sự phù hợp với sự biến đổi của dữ liệu. Trong khi đó, qua những đề xuất từ Chương 2, chúng ta có thể ước lượng được số lượng ma trận hợp lý cho một tập sắp hàng cho trước. Ngoài ra, ước lượng mô hình đa ma trận còn có thể dựa trên các thuộc tính của trình tự axit amin như: cấu trúc không gian, khả năng hòa tan trong dung môi hay các đặc trưng hóa học khác. Do vậy, chúng ta cũng cần phát triển thêm các cách thức để ước lượng mô hình đa ma trận phù hợp với thực tế hơn dựa trên các thuộc tính này.

Với phương pháp lựa chọn mô hình dựa trên học sâu, vấn đề quan trọng là sinh dữ liệu mô phỏng sát với dữ liệu thật nhất, luận án sinh dữ liệu dựa trên các tham số được ước lượng từ dữ liệu thật nhưng quá trình đánh giá ModelDetector cho thấy kết quả dự đoán trên dữ liệu thật vẫn còn hạn chế. Dữ liệu thật thường có sự biến đổi đa dạng, mỗi vị trí có thể tuân theo các mô hình khác nhau, cả mô hình đơn và mô hình hỗn hợp. Như vậy, sinh dữ liệu giống với dữ liệu thật là bài toán rất thách thức. Ngoài ra là làm sao để có thể sử dụng dữ liệu thật trong quá trình đào tạo mạng? Đây là một câu hỏi rất khó đặt ra cho những nhà nghiên cứu về tin sinh học tiến hóa. Nếu dữ liệu có sự đa dạng và khó đoán, liệu việc sử dụng mạng không có lớp tích chập có còn phù hợp hay không? Nếu không thì chúng ta cần một kiến trúc mạng học sâu như thế nào để đáp ứng được yêu cầu bài toán? Đây cũng là một câu hỏi cần giải đáp trong các nghiên cứu tiếp theo.

DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC

[CT1] N. H. Tinh, C. C. Dang, and L. S. Vinh, "Rooting Phylogenetic Trees from Protein Alignments," in *Proceedings - International Conference on Knowledge and Systems Engineering, KSE,* Oct. 2023, pp. 1–5. doi: 10.1109/KSE59128.2023.10299425.

Đây là hội nghị quốc tế uy tín thuộc danh mục Scopus.

[CT2] N. H. Tinh, C. C. Dang, and L. S. Vinh, "Estimating amino acid substitution models from genome datasets: A simulation study on the performance of estimated models," *J. Evol. Biol.,* vol. 37, no. 2, pp. 256–265, 2023, doi: 10.1093/jeb/voad017.

Đây là tạp chí uy tín quốc tế, thuộc danh mục SCIE, nằm trong nhóm Q1 theo SCImago.

[CT3] N. H. Tinh, C. C. Dang, and L. S. Vinh, "QMix: An Efficient Program to Automatically Estimate Multi-Matrix Mixture Models for Amino Acid Substitution Process," *J. Comput. Biol.,* vol. 31, no. 8, pp. 703–707, 2024, doi: 10.1089/cmb.2023.0403.

Đây là tạp chí uy tín quốc tế, thuộc danh mục SCIE, nằm trong nhóm Q2 theo SCImago.

[CT4] N. H. Tinh and L. S. Vinh, "Improving the study of plant evolution with multi-matrix mixture models," *Plant Syst. Evol.,* vol. 310, 2024, doi: 10.1007/s00606-024-01896-0.

Đây là tạp chí uy tín quốc tế, thuộc danh mục SCIE, nằm trong nhóm Q2 theo SCImago.

[CT5] N. H. Tinh and L. S. Vinh, "An efficient deep learning method for amino acid substitution model selection," *J. Evol. Biol.*, vol. 38, no. 1, pp. 129–139, 2024, doi: https://doi.org/10.1093/jeb/voae141.

Đây là tạp chí uy tín quốc tế, thuộc danh mục SCIE, nằm trong nhóm Q1 theo SCImago.

[CT6] N. H. Tinh, C. C. Dang, and L. S. Vinh, "nT4X and nT4M: Novel Time Non-reversible Mixture Amino Acid Substitution Models," *J. Mol. Evol.*, 2025, doi: https://doi.org/10.1007/s00239-024-10230-8.

Đây là tạp chí uy tín quốc tế, thuộc danh mục SCIE, nằm trong nhóm Q1 theo SCImago.

TÀI LIỆU THAM KHẢO

TIẾNG VIỆT

[1] P. T. Hổ, "Di truyền học," *Nhà xuất bản Giáo dục*, 2008.

[2] N. H. Lộc, T. T. Lệ, and H. T. M. Thi, "Giáo trình Sinh học phân tử," *Nhà xuất bản Đại học Huế*, 2007.

[3] L. S. Vinh, "Các phương pháp phân tích dữ liệu sinh học có kích thước lớn," *Nhà xuất bản Đại học Quốc gia Hà Nội*, 2019.

[4] L. Đ. Trình, "Sinh học phân tử của tế bào," *Nhà xuất bản Khoa học và Kỹ thuật*, 2001.

TIẾNG ANH

[5] D. M.O, S. R.M, and O. B.C, "A model for evolutionary change in proteins," *Atlas protein Seq. Struct.*, pp. 345–352, 1978.

[6] D. T. Jones, W. R. Taylor, and J. M. Thornton, "The rapid generation of mutation data matrices from protein sequences," *Bioinformatics*, vol. 8, no. 3, pp. 275–282, 1992, doi: 10.1093/bioinformatics/8.3.275.

[7] S. Whelan and N. Goldman, "A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach," *Mol. Biol. Evol.*, vol. 18, no. 5, pp. 691–699, 2001, doi: 10.1093/oxfordjournals.molbev.a003851.

[8] S. Q. Le and O. Gascuel, "An improved general amino acid replacement matrix," *Mol. Biol. Evol.*, vol. 25, no. 7, pp. 1307–1320, 2008, doi: 10.1093/molbev/msn067.

[9] P. S. Klosterman *et al.*, "XRate: A fast prototyping, training and annotation tool

for phylo-grammars," *BMC Bioinformatics*, vol. 7, 2006, doi: 10.1186/1471-2105-7-428.

[10] S. A., "RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models," *Bioinformatics*, vol. 22, p. 2688, 2006.

[11] B. Q. Minh, C. C. Dang, L. S. Vinh, and R. Lanfear, "QMaker: Fast and Accurate Method to Estimate Empirical Models of Protein Evolution," *Syst. Biol.*, vol. 70, no. 5, pp. 1046–1060, 2021, doi: 10.1093/sysbio/syab010.

[12] C. C. Dang *et al.*, "nQMaker: Estimating Time Nonreversible Amino Acid Substitution Models," *Syst. Biol.*, vol. 71, no. 5, pp. 1110–1123, 2022, doi: 10.1093/sysbio/syac007.

[13] S. Kalyaanamoorthy, B. Q. Minh, T. K. F. Wong, A. Von Haeseler, and L. S. Jermiin, "ModelFinder: Fast model selection for accurate phylogenetic estimates," *Nat. Methods*, vol. 14, no. 6, pp. 587–589, 2017, doi: 10.1038/nmeth.4285.

[14] Di. Darriba, D. Posada, A. M. Kozlov, A. Stamatakis, B. Morel, and T. Flouri, "ModelTest-NG: A New and Scalable Tool for the Selection of DNA and Protein Evolutionary Models," *Mol. Biol. Evol.*, vol. 37, no. 1, pp. 291–294, 2020, doi: 10.1093/molbev/msz189.

[15] S. Burgstaller-Muehlbacher, S. M. Crotty, H. A. Schmidt, F. Reden, T. Drucks, and A. von Haeseler, "ModelRevelator: Fast phylogenetic model estimation via deep learning," *Mol. Phylogenet. Evol.*, vol. 188, 2023, doi: 10.1016/j.ympev.2023.107905.

[16] S. Abadi, O. Avram, S. Rosset, T. Pupko, and I. Mayrose, "Modelteller: Model selection for optimal phylogenetic reconstruction using machine learning," *Mol. Biol. Evol.*, vol. 37, no. 11, pp. 3338–3352, 2020, doi: 10.1093/molbev/msaa154.

[17] J. D. Thompson, D. G. Higgins, and T. J. Gibson, “CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice,” *Nucleic Acids Res.*, vol. 22, no. 22, pp. 4673–4680, 1994, doi: 10.1093/nar/22.22.4673.

[18] K. K., M. K., K. K., and M. T., “MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform,” *Nucleic Acids Res.*, vol. 30, 2002.

[19] C. B. Do, M. S. P. Mahabhashyam, M. Brudno, and S. Batzoglou, “ProbCons: Probabilistic consistency-based multiple sequence alignment,” *Genome Res.*, vol. 15, no. 2, pp. 330–340, 2005, doi: 10.1101/gr.2821705.

[20] R. E. C., “MUSCLE: multiple sequence alignment with high accuracy and high throughput,” *Nucleic Acids Res.*, vol. 32, no. 5, pp. 1792–1797, 2004, [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC390337/

[21] R. R. Sokal, “A statistical method for evaluating systematic relationships,” *Univ Kans Sci Bull*, vol. 38, pp. 1409–1438, 1958.

[22] N. Saitou and M. Nei, “The neighbor-joining method: a new method for reconstructing phylogenetic trees.,” *Mol. Biol. Evol.*, vol. 4, no. 4, pp. 406–425, 1987, doi: 10.1093/oxfordjournals.molbev.a040454.

[23] L. Kannan and W. C. Wheeler, “Maximum Parsimony on Phylogenetic networks,” *Algorithms Mol. Biol.*, vol. 7, no. 1, 2012, doi: 10.1186/1748-7188-7-9.

[24] W. P. Maddison, M. J. Donoghue, and D. R. Maddison, “Outgroup Analysis and Parsimony,” *Syst. Zool.*, vol. 33, no. 1, p. 83, 1984, doi: 10.2307/2413134.

[25] H. Kishino and M. Hasegawa, “Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order

in hominoidea," *J. Mol. Evol.*, vol. 29, no. 2, pp. 170–179, 1989, doi: 10.1007/BF02100115.

[26] F. Ronquist *et al.*, "Mrbayes 3.2: Efficient bayesian phylogenetic inference and model choice across a large model space," *Syst. Biol.*, vol. 61, no. 3, pp. 539–542, 2012, doi: 10.1093/sysbio/sys029.

[27] J. P. Huelsenbeck, J. P. Bollback, and A. M. Levine, "Inferring the root of a phylogenetic tree," *Syst. Biol.*, vol. 51, no. 1, pp. 32–43, 2002, doi: 10.1080/106351502753475862.

[28] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, "Biological sequence analysis: Probabilistic models of proteins and nucleic acids," pp. 1–371, 2006, [Online]. Available: papers2://publication/uuid/28FE17E2-9BF7-4BF3-8079-5302425D060F

[29] J. Felsenstein, *Inferring Phylogenies*, 2nd ed. Sinauer, 2003.

[30] T. H. Jukes and C. R. Cantor, "Evolution of protein molecules," 1969, pp. 21–132. doi: https://doi.org/10.1016/B978-1-4832-3211-9.50009-7.

[31] M. Kimura, "A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences," *J. Mol. Evol.*, vol. 16, no. 2, pp. 111–120, 1980, doi: 10.1007/BF01731581.

[32] S. Veerassamy, A. Smith, and E. R. M. Tillier, "A Transition Probability Model for Amino Acid Substitutions from Blocks," *J. Comput. Biol.*, vol. 10, no. 6, pp. 997–1010, 2003, doi: 10.1089/106652703322756195.

[33] M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt, "A model of evolutionary change in proteins," *Atlas protein Seq. Struct.*, vol. 5, no. Suppl 3, pp. 345–352, 1978, [Online]. Available: http://www.mendeley.com/research/a-model-of-

evolutionary-change-in-proteins/

[34] S. Tavaré, “Some probabilistic and statistical problems in the analysis of DNA sequences,” *Am. Math. Soc. Lect. Math. Life Sci.*, vol. 17, pp. 57–86, 1986.

[35] S. El-Gebali *et al.*, “The Pfam protein families database in 2019,” *Nucleic Acids Res.*, vol. 47, no. D1, pp. D427–D432, 2019, doi: 10.1093/nar/gky995.

[36] R. Schneider, A. De Daruvar, and C. Sander, “The HSSP database of protein structure-sequence alignments,” *Nucleic Acids Res.*, vol. 25, no. 1, pp. 226–230, 1997, doi: 10.1093/nar/25.1.226.

[37] S. Q. Le, C. C. Dang, and O. Gascuel, “Modeling protein evolution with several amino acid replacement matrices depending on site rates,” *Mol. Biol. Evol.*, vol. 29, no. 10, pp. 2921–2936, 2012, doi: 10.1093/molbev/mss112.

[38] S. Henikoff and J. G. Henikoff, “Amino acid substitution matrices from protein blocks,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 89, no. 22, pp. 10915–10919, 1992, doi: 10.1073/pnas.89.22.10915.

[39] J. Adachi and M. Hasegawa, “Model of amino acid substitution in proteins encoded by mitochondrial DNA,” *J. Mol. Evol.*, vol. 42, no. 4, pp. 459–468, 1996, doi: 10.1007/BF02498640.

[40] Chor B and Tuller T., “ Maximum likelihood of evolutionary trees: hardness and approximation,” *Bioinformatics*, pp. 97–106, 2005.

[41] C. C. Dang, L. S. Vinh, O. Gascuel, B. Hazes, and S. Q. Le, “FastMG: a simple, fast, and accurate maximum likelihood procedure to estimate amino acid replacement rate matrices from large data sets,” *BMC Bioinformatics*, vol. 15, p. 341, 2014, doi: 10.1186/1471-2105-15-341.

[42] B. Q. Minh *et al.*, “IQ-TREE 2: New Models and Efficient Methods for

Phylogenetic Inference in the Genomic Era," *Mol. Biol. Evol.*, vol. 37, no. 5, pp. 1530–1534, 2020, doi: 10.1093/molbev/msaa015.

[43] J. H. Ran, T. T. Shen, M. M. Wang, and X. Q. Wang, "Phylogenomics resolves the deep phylogeny of seed plants and indicates partial convergent or homoplastic evolution between Gnetales and angiosperms," *Proc. R. Soc. B Biol. Sci.*, vol. 285, no. 1881, 2018, doi: 10.1098/rspb.2018.1012.

[44] E. D. Jarvis *et al.*, "Phylogenomic analyses data of the avian phylogenomics project," *Gigascience*, vol. 4, no. 1, 2015, doi: 10.1186/s13742-014-0038-1.

[45] C. C. Dang and L. S. Vinh, "Estimating amino acid substitution models for metazoan evolutionary studies," *J. Evol. Biol.*, vol. 36, no. 3, pp. 499–506, 2023, doi: 10.1111/jeb.14147.

[46] L. T. Nguyen, H. A. Schmidt, A. Von Haeseler, and B. Q. Minh, "IQ-TREE: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies," *Mol. Biol. Evol.*, vol. 32, no. 1, pp. 268–274, 2015, doi: 10.1093/molbev/msu300.

[47] Z. Yang, "Among-site rate variation and its impact on phylogenetic analyses," *Trends Ecol. Evol.*, vol. 11, no. 9, pp. 367–372, 1996, doi: 10.1016/0169-5347(96)10041-0.

[48] Z. Yang, "Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites," *Mol. Biol. Evol.*, vol. 10, no. 6, pp. 1396–1401, 1993, doi: 10.1093/oxfordjournals.molbev.a040082.

[49] G. Schwarz, "Estimating the Dimension of a Model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 2007, doi: 10.1214/aos/1176344136.

[50] H. Akaike, "A New Look at the Statistical Model Identification," pp. 215–222,

1974, doi: 10.1007/978-1-4612-1694-0_16.

[51] D. F. Robinson and L. R. Foulds, “Comparison of phylogenetic trees,” *Math. Biosci.*, vol. 53, no. 1–2, pp. 131–147, 1981, doi: 10.1016/0025-5564(81)90043-2.

[52] S. Guindon, J. F. Dufayard, V. Lefort, M. Anisimova, W. Hordijk, and O. Gascuel, “New algorithms and methods to estimate maximum-likelihood phylogenies: Assessing the performance of PhyML 3.0,” *Syst. Biol.*, vol. 59, no. 3, pp. 307–321, 2010, doi: 10.1093/sysbio/syq010.

[53] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[54] C. Sander and R. Schneider, “The HSSP database of protein structure-sequence alignments,” *Nucleic Acids Res.*, vol. 22, no. 17, pp. 3597–3599, 1994.

[55] M. J. Sanderson, M. J. Donoghue, W. H. Piel, and T. Eriksson, “TreeBASE: a prototype database of phylogenetic analyses and an interactive tool for browsing the phylogeny of life,” *Am. J. Bot.*, vol. 81, p. 183, 1994.

[56] X. X. Shen *et al.*, “Tempo and Mode of Genome Evolution in the Budding Yeast Subphylum,” *Cell*, vol. 175, no. 6, pp. 1533-1545.e20, 2018, doi: 10.1016/j.cell.2018.10.023.

[57] S. Wu, S. Edwards, and L. Liu, “Genome-scale DNA sequence data and the evolutionary history of placental mammals,” *Data Br.*, vol. 18, pp. 1972–1975, 2018, doi: 10.1016/j.dib.2018.04.094.

[58] B. Misof *et al.*, “Phylogenomics resolves the timing and pattern of insect evolution,” *Science (80-.).*, vol. 346, no. 6210, pp. 763–767, 2014, doi:

10.1126/science.1257570.

[59] A. J. Barley, J. M. Brown, and R. C. Thomson, “Impact of Model Violations on the Inference of Species Boundaries under the Multispecies Coalescent,” *Syst. Biol.*, vol. 67, no. 2, pp. 269–284, 2018, doi: 10.1093/sysbio/syx073.

[60] A. Suvorov and D. R. Schrider, “Reliable estimation of tree branch lengths using deep neural networks,” *PLoS Comput. Biol.*, vol. 20, no. 8, 2024, doi: https://doi.org/10.1371/journal.pcbi.1012337.

[61] A. Rambaut and N. C. Grassly, “Seq-gen: An application for the monte carlo simulation of dna sequence evolution along phylogenetic trees,” *Bioinformatics*, vol. 13, no. 3, pp. 235–238, 1997, doi: 10.1093/bioinformatics/13.3.235.

[62] W. Fletcher and Z. Yang, “INDELible: A flexible simulator of biological sequence evolution,” *Mol. Biol. Evol.*, vol. 26, no. 8, pp. 1879–1888, 2009, doi: 10.1093/molbev/msp098.

[63] N. Ly-Trong, S. Naser-Khdour, R. Lanfear, and B. Q. Minh, “AliSim: A Fast and Versatile Phylogenetic Sequence Simulator for the Genomic Era,” *Mol. Biol. Evol.*, vol. 39, no. 5, p. msac092, 2022, doi: 10.1093/molbev/msac092.

[64] N. Ly-Trong, G. M. J. Barca, and B. Q. Minh, “AliSim-HPC: parallel sequence simulator for phylogenetics,” *Bioinformatics*, vol. 39, no. 9, 2023, doi: 10.1093/bioinformatics/btad540.

[65] S. Q. Le and O. Gascuel, “Accounting for solvent accessibility and secondary structure in protein phylogenetics is clearly beneficial,” *Syst. Biol.*, vol. 59, no. 3, pp. 277–287, 2010, doi: 10.1093/sysbio/syq002.

[66] S. Q. Le, N. Lartillot, and O. Gascuel, “Phylogenetic mixture models for proteins,” *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 363, no. 1512, pp. 3965–3976, 2008, doi:

10.1098/rstb.2008.0180.

[67] H. C. Wang, K. Li, E. Susko, and A. J. Roger, “A class frequency mixture model that adjusts for site-specific amino acid frequencies and improves inference of protein phylogeny,” *BMC Evol. Biol.*, vol. 8, no. 1, 2008, doi: 10.1186/1471-2148-8-331.

[68] J. M. Koshi and R. A. Goldstein, “Context-dependent optimal substitution matrices,” *Protein Eng. Des. Sel.*, vol. 8, no. 7, pp. 641–645, 1995, doi: 10.1093/peds/8.7.641.

[69] J. L. Thorne, N. Goldman, and D. T. Jones, “Combining protein evolution and secondary structure,” *Mol. Biol. Evol.*, vol. 13, no. 5, pp. 666–673, 1996, doi: 10.1093/oxfordjournals.molbev.a025627.

[70] N. Goldman, J. L. Thorne, and D. T. Jones, “Assessing the impact of secondary structure and solvent accessibility on protein evolution,” *Genetics*, vol. 149, no. 1, pp. 445–458, 1998, doi: 10.1093/genetics/149.1.445.

[71] L. S. Quang, O. Gascuel, and N. Lartillot, “Empirical profile mixture models for phylogenetic reconstruction,” *Bioinformatics*, vol. 24, no. 20, pp. 2317–2323, 2008, doi: 10.1093/bioinformatics/btn445.

[72] H. Kishino, T. Miyata, and M. Hasegawa, “Maximum likelihood inference of protein phylogeny and the origin of chloroplasts,” *J. Mol. Evol.*, vol. 31, no. 2, pp. 151–160, 1990, doi: 10.1007/BF02109483.

[73] S. Naser-Khdour, B. Quang Minh, and R. Lanfear, “Assessing Confidence in Root Placement on Phylogenies: An Empirical Study Using Nonreversible Models for Mammals,” *Syst. Biol.*, vol. 71, no. 4, pp. 959–972, 2022, doi: 10.1093/sysbio/syab067.

[74] D. C. Jhwueng, S. Huzurbazar, B. C. O'Meara, and L. Liu, "Investigating the performance of AIC in selecting phylogenetic models," *Stat. Appl. Genet. Mol. Biol.*, vol. 13, no. 4, pp. 459–475, 2014, doi: 10.1515/sagmb-2013-0048.

[75] T. K. Seo and J. L. Thorne, "Information criteria for comparing partition schemes," *Syst. Biol.*, vol. 67, no. 4, pp. 616–632, 2018, doi: 10.1093/sysbio/syx097.

[76] E. Susko and A. J. Roger, "On the Use of Information Criteria for Model Selection in Phylogenetics," *Mol. Biol. Evol.*, vol. 37, no. 2, pp. 549–562, 2020, doi: 10.1093/molbev/msz228.

[77] H. Ren, T. K. F. Wong, B. Q. Minh, and R. Lanfear, "MixtureFinder: Estimating DNA Mixture Models for Phylogenetic Analyses," *Mol. Biol. Evol.*, vol. 42, no. 1, 2025, doi: 10.1093/molbev/msae264.

[78] M. K. K. Leung, A. Delong, B. Alipanahi, and B. J. Frey, "Machine learning in genomic medicine: A review of computational problems and data sets," *Proc. IEEE*, vol. 104, no. 1, pp. 176–197, 2016, doi: 10.1109/JPROC.2015.2494198.

[79] A. Kan, "Machine learning applications in cell image analysis," *Immunol. Cell Biol.*, vol. 95, no. 6, pp. 525–530, 2017, doi: 10.1038/icb.2017.16.

[80] G. Kandoi, M. L. Acencio, and N. Lemke, "Prediction of druggable proteins using machine learning and systems biology: A mini-review," *Front. Physiol.*, vol. 6, no. DEC, 2015, doi: 10.3389/fphys.2015.00366.

[81] S. Dekel, Y. Keller, and A. Bar-Hillel, "Deep Convolutional Tables: Deep Learning Without Convolutions," *IEEE Trans. Neural Networks Learn. Syst.*, 2023, doi: 10.1109/TNNLS.2023.3270402.

[82] R. Sunkara and T. Luo, "No More Strided Convolutions or Pooling: A New CNN

Building Block for Low-Resolution Images and Small Objects," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 13715 LNAI, pp. 443–459, 2023, doi: 10.1007/978-3-031-26409-2_27.

[83] T. Müller and M. Vingron, "Modeling amino acid replacement," *J. Comput. Biol.*, vol. 7, no. 6, pp. 761–776, 2000, doi: 10.1089/10665270050514918.

[84] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, 2015.

[85] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," *Proc. 12th USENIX Symp. Oper. Syst. Des. Implementation, OSDI 2016*, pp. 265–283, 2016.

[86] T. Le Kim and V. Le Sy, "mPartition: A Model-Based Method for Partitioning Alignments," *J. Mol. Evol.*, vol. 88, no. 8–9, pp. 641–652, 2020, doi: 10.1007/s00239-020-09963-z.

[87] H. Banos *et al.*, "GTRpmix: A Linked General Time-Reversible Model for Profile Mixture Models," *Mol. Biol. Evol.*, vol. 41, no. 9, 2024, doi: 10.1093/molbev/msae174.

[88] N. Ly-Trong, C. Bielow, N. De Maio, and B. Q. Minh, "CMAPLE: Efficient Phylogenetic Inference in the Pandemic Era," *Mol. Biol. Evol.*, vol. 41, no. 7, 2024, doi: 10.1093/molbev/msae134.

[89] T. K. F. Wong, C. Cherryh, A. G. Rodrigo, M. W. Hahn, B. Q. Minh, and R. Lanfear, "MAST: Phylogenetic Inference with Mixtures Across Sites and Trees," *Syst. Biol.*, vol. 73, no. 2, pp. 375–391, 2024, doi: 10.1093/sysbio/syae008.