ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

ĐẶNG CAO CƯỜNG

CÁC PHƯƠNG PHÁP XÂY DỰNG MA TRẬN BIẾN ĐỔI AXÍT AMIN

LUẬN ÁN TIẾN SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội – 2013

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

ĐẶNG CAO CƯỜNG

CÁC PHƯƠNG PHÁP XÂY DỰNG MA TRẬN BIẾN ĐỔI AXÍT AMIN

Chuyên ngành: Khoa học Máy tính
Mã số: 62.48.01.01

LUẬN ÁN TIẾN SĨ CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC:

1. TS. Lê Sỹ Vinh
2. TS. Lê Sĩ Quang

Hà Nội – 2013

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

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 TS. Lê Sỹ Vinh và TS. Lê Sĩ Quang.

Tôi xin bày tỏ lòng biết ơn sâu sắc tới TS. Lê Sỹ Vinh, TS. Lê Sĩ Quang và giáo sư Oliver Gascuel, những người đã có những định hướng giúp tôi thành công trong việc nghiên cứu của mình. Các thầy cũng đã động viên và chỉ bảo giúp tôi vượt qua những khó khăn để tôi hoàn thành được luận án này. Tôi cũng chân thành cảm ơn thầy Hoàng Xuân Huấn, thầy đã cho tôi nhiều kiến thức quý báu về nghiên cứu khoa học và cuộc sống. Những sự chỉ bảo quý giá của các thầy đã giúp tôi hoàn thành tốt luận án này.

Tôi cũng xin 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.

MỤC LỤC

Danh mục các ký hiệu và chữ viết tắt

l	*Chiều dài của một sắp hàng*
m	*Số lượng chuỗi có trong một sắp hàng*
N	*Số lượng sắp hàng trong một tập các sắp hàng*
$\mathbf{S}$	*Tập hợp 20 axít amin*
q_{ij}	*Tốc độ biến đổi tức thời giữa axít amin i và axít amin j*
π_i	*Tần số của axít amin i*
r_{ij}	*Hệ số hoán đổi giữa axít amin i và axít amin j*
α	*Tham số định hình của phân phối gamma*
$\mathbf{A}$	*Tập các sắp hàng*
D	*Một sắp hàng đa chuỗi*
D^a	*Sắp hàng đa chuỗi thứ a trong một tập các sắp hàng*
D_i	*Vị trí thứ i trong sắp hàng đa chuỗi D*
$\mathbf{Q}$	*Ma trận tốc độ biến đổi tức thời*
$\mathbf{\Pi}$	*Véc tơ tần số của 20 axít amin*
$\mathbf{R}$	*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*
EM	*Thuật toán cực đại hoá kỳ vọng (expectation maximization)*
ML	*Phương pháp cực đại khả năng (maximum likelihood)*
STT	*Số thứ tự*
RF	*Khoảng cách Robinson-Fould*

Danh mục các bảng

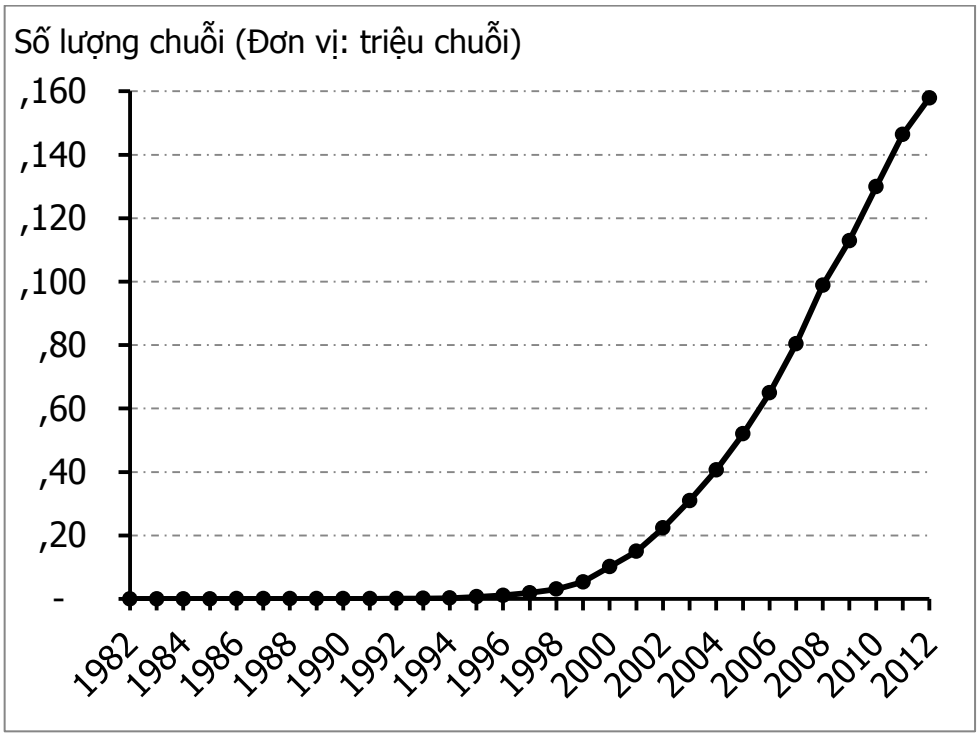
Danh mục các hình vẽ, đồ thị

Danh mục các thuật toán

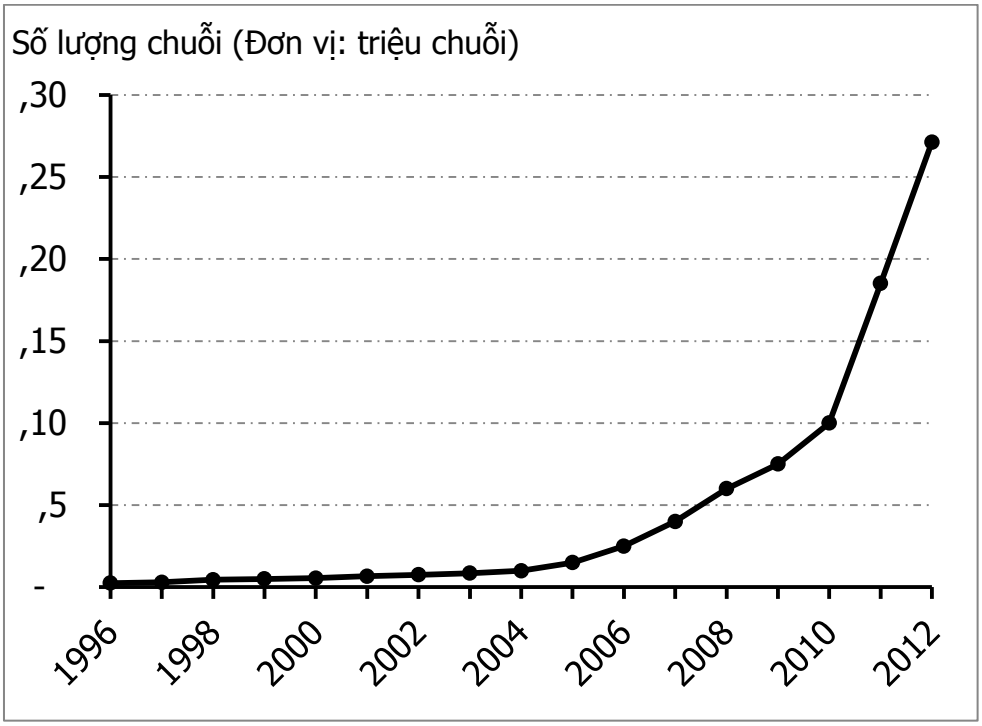
MỞ ĐẦU

Ứng dụng công nghệ thông tin để nghiên cứu và giải quyết các bài toán trong sinh học phân tử đang rất được quan tâm. Tin sinh học là lĩnh vực nghiên cứu kết hợp cả hai ngành công nghệ thông tin và sinh học phân tử. Tin sinh học đang được đầu tư lớn do khả năng mang lại sự tiến bộ về khoa học và hiệu quả kinh tế thông qua việc thúc đẩy sự phát triển công nghệ sinh học và ứng dụng trong y tế, nông nghiệp và các lĩnh vực khác.

Trong sinh học phân tử có hai loại dữ liệu phổ biến và quan trọng nhất là chuỗi ADN và chuỗi prôtêin. Số lượng các chuỗi này đang liên tục tăng dần hàng ngày với tốc độ chóng mặt. Hình 0.1 và Hình 0.2 minh họa số lượng chuỗi ADN và chuỗi prôtêin qua các năm của hai cơ sở dữ liệu Genbank và UniProt tương ứng.



Hình 0.1: Biểu đồ số lượng chuỗi ADN theo năm của cơ sở dữ liệu Genbank *(Nguồn: http://www.ncbi.nlm.nih.gov/genbank/)*.



Hình 0.2: Biểu đồ số lượng chuỗi prôtêin theo năm của cơ sở dữ liệu UniProt *(Nguồn: http://www.uniprot.org/).*

Các bài toán liên quan đến chuỗi prôtêin như sắp hàng đa chuỗi, tìm kiếm chuỗi tương đồng, xây dựng cây phân loài đều là các bài toán cơ bản và quan trọng của tin sinh học. Tất cả các bài toán này đều cần đến một thành phần rất quan trọng là mô hình (ma trận) biến đổi axít amin. Mô hình biến đổi axít amin có số lượng tham số lớn (khoảng 200 tham số) và thường khó có thể ước lượng trực tiếp trong quá trình phân tích dữ liệu. Chúng ta thường ước lượng trước một mô hình chung (general model) và mô hình này được sử dụng cho mọi bộ dữ liệu prôtêin. Mô hình chung đầu tiên là PAM [21] và gần đây nhất là LG [49].

Quá trình ước lượng mô hình biến đổi axít amin là một quá trình phức tạp và trải qua nhiều bước tính toán khác nhau, mỗi bước là một bài toán khó. Ba bước chính của quá trình ước lượng mô hình là:

1. Xây dựng cây phân loài từ tập các sắp hàng đa chuỗi. Các thuật toán xây dựng cây dùng trong quá trình ước lượng mô hình còn tốn rất nhiều thời gian. Ví dụ phải mất vài ngày để ước lượng được mô hình LG [17].
2. Xác định các ràng buộc liên quan đến mô hình. Độ chính xác của mô hình hiện tại vẫn còn hạn chế do việc mô hình hoá đã loại bỏ một số điều kiện ràng buộc trong sinh học phân tử.
3. Xây dựng các mô hình riêng biệt cho các loài sinh vật khác nhau. Đây là một bước rất quan trọng bởi vì trong nhiều trường hợp các mô hình chung không mô hình hoá được hết các đặc điểm biến đổi riêng biệt của các loài.

Từ đó, luận án tập trung vào giải quyết các bài toán ở ba bước chính trên. Cụ thể là:

1. Đề xuất một số phương pháp mới để tăng tốc độ quá trình xây dựng cây, giảm bớt số bước tối ưu cấu trúc cây, từ đó giúp giảm thời gian ước lượng mô hình.
2. Sử dụng thêm các ràng buộc trong sinh học phân tử vào quá trình mô hình hoá. Việc này sẽ giúp nâng cao tính chính xác của mô hình biến đổi axít amin khi phân tích dữ liệu.
3. Xây dựng một hệ thống ước lượng tự động mô hình biến đổi axít amin từ dữ liệu của người dùng, qua đó giúp người dùng có thể ước lượng các mô hình riêng biệt cho các loài sinh vật khác nhau.
4. Bên cạnh đó, luận án cũng xây dựng thử nghiệm mô hình biến đổi axít amin cho riêng vi rút cúm và kiểm nghiệm tính hiệu quả của mô hình mới này.

Các kết quả của luận án đã được công bố trong 03 bài báo ở tạp chí SCI quốc tế [17, 18, 48] và 02 báo cáo ở hội nghị quốc tế [20, 54]. Ngoài phần kết luận, luận án được tổ chức như sau:

Chương 1 giới thiệu khái quát về chuỗi ADN, chuỗi axít amin và các phép biến đổi trên chuỗi axít amin. Sau đó là phần giới thiệu về bài toán mô hình hoá quá trình biến đổi axít amin và bài toán ước lượng mô hình biến đổi axít amin. Tiếp theo

là phần trình bày về hai cách tiếp cận chính để ước lượng mô hình biến đổi axít amin là phương pháp đếm và phương pháp cực đại khả năng (maximum likelihood). Phần cuối của chương này giới thiệu về phương pháp xây dựng cây phân loài bằng phương pháp cực đại khả năng và các phương pháp so sánh hai mô hình biến đổi axít amin.

Chương 2 đề xuất phương pháp ước lượng nhanh mô hình biến đổi axít amin. Luận án đề xuất hai phương pháp chia tách nhỏ dữ liệu đầu vào. Hai phương pháp này giúp giảm thời gian xây dựng cây phân loài, một bước chiếm rất nhiều thời gian trong quá trình ước lượng mô hình biến đổi axít amin. Các thực nghiệm đã chứng tỏ được hiệu quả của hai phương pháp này.

Chương 3 của luận án giới thiệu mô hình biến đổi axít amin sử dụng nhiều ma trận, một cải tiến mới so với các mô hình đơn ma trận hiện nay. Mô hình mới này sử dụng thêm các ràng buộc trong sinh học phân tử giúp tăng cường khả năng mô hình hoá các quá trình biến đổi của các chuỗi axít amin. Các thực nghiệm với hai bộ dữ liệu HSSP và TreeBase đã chứng tỏ mô hình biến đổi đa ma trận có độ chính xác cao hơn các mô hình hiện tại.

Chương 4 đề xuất một thuật toán ước lượng mô hình biến đổi axít amin cải tiến giúp giảm 50% thời gian ước lượng mô hình. Có được điều này chính là do thuật toán mới đã tìm cách giảm bớt số bước tối ưu cấu trúc cây phân loài – một bước chiếm nhiều thời gian trong quá trình ước lượng. Chương này cũng giới thiệu hệ thống ước lượng mô hình tự động cài đặt thuật toán cải tiến trên.

Chương 5 trình bày mô hình biến đổi axít amin cho vi rút cúm, gọi là mô hình FLU. Phần sau của chương là các kết quả so sánh mô hình FLU với các mô hình khác. Qua các thực nghiệm, mô hình FLU đã chứng tỏ được hiệu quả cao hơn hẳn các mô hình hiện tại khi phân tích dữ liệu vi rút cúm.

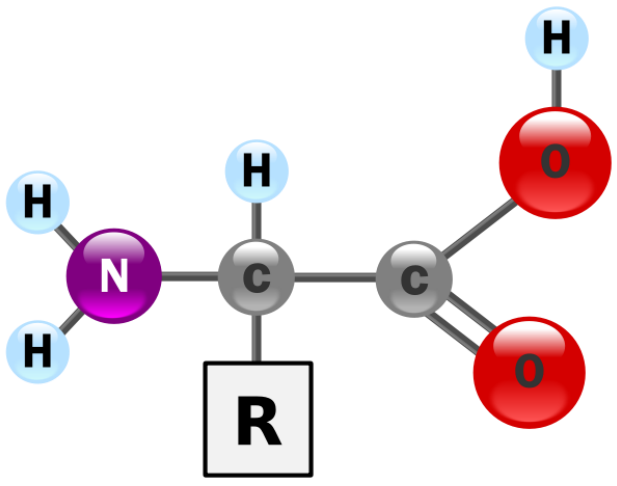
Chương 1. BÀI TOÁN ƯỚC LƯỢNG SỰ BIẾN ĐỔI CỦA AXÍT AMIN

1.1. Giới thiệu chung

Trong phần này chúng tôi sẽ trình bày các khái niệm cơ bản về ADN, axít amin, sắp hàng đa chuỗi và cây phân loài.

1.1.1. ADN và axít amin

Trong sinh học phân tử, Axít Deoxyribo Nucleic (viết tắt ADN) mang thông tin di truyền mã hóa cho hoạt động sinh trưởng và phát triển của các loài sinh vật [4, 5]. ADN được cấu tạo từ nhiều phân tử nhỏ gọi là các nuclêotít. Có 4 loại nuclêotít là: Adenine (A), Thymine (T), Cytosine (C), và Guanine (G). Các nuclêotít kết hợp với nhau thành một mạch dài nhờ các liên kết phôtphođieste để tạo thành một chuỗi nuclêotít (còn gọi là chuỗi pôlinuclêotít). ADN có cấu tạo gồm hai chuỗi nuclêotít xoắn kép với nhau, trong đó các nuclêotít giữa 2 chuỗi liên kết với nhau bằng liên kết hiđrô theo nguyên tắc bổ sung: A với T và G với C [1].



Hình 1.1: Minh họa cấu tạo của một phân tử axít amin.

Axít amin là một hợp chất hữu cơ được cấu tạo bởi ba thành phần: nhóm amin ($-NH_2$), nhóm cacboxyl (-COOH) và nhóm biến đổi R quyết định tính chất của axít amin [1, 16]. Hình 1.1 minh họa cấu tạo chung của một axít amin. Các axít amin kết hợp với nhau thành một mạch dài nhờ các liên kết péptít (còn gọi là chuỗi pôlipéptít) để tạo thành một chuỗi axít amin hay còn gọi là chuỗi prôtêin. Các chuỗi này có thể xoắn cuộn hoặc gấp theo nhiều cách để tạo thành các bậc cấu trúc không gian khác nhau của chuỗi prôtêin [5].

Mối quan hệ giữa nuclêotít và axít amin được thể hiện qua quá trình tổng hợp prôtêin. Trong một chuỗi nuclêotít mã hóa prôtêin, mỗi bộ ba nuclêotít liên tiếp được gọi là một *codon*. Mỗi *codon* có thể mã hóa một axít amin hoặc là tín hiệu kết thúc của một quá trình tổng hợp prôtêin [44]. Có tất cả 64 *codon*, trong đó có 61 *codon* mã hóa cho các axít amin, 3 *codon* còn lại được gọi là *stop-codon* (xem thêm Bảng 1.1).

Bảng 1.1: Danh sách 64 *codon*. Mỗi *codon* mã hoá một axít amin.

	T		C		A		G		
	Codon	**Axít amin**	**Codon**	**Axít amin**	**Codon**	**Axít amin**	**Codon**	**Axít amin**	
T	TTT	Phe	TCT	Ser	TAT	Tyr	TGT	Cys	**T**
	TTC	Phe	TCC	Ser	TAC	Tyr	TGC	Cys	**C**
	TTA	Leu	TCA	Ser	TAA	STOP	TGA	STOP	**A**
	TTG	Leu	TCG	Ser	TAG	STOP	TGG	Trp	**G**
C	CTT	Leu	CCT	Pro	CAT	His	CGT	Arg	**T**
	CTC	Leu	CCC	Pro	CAC	His	CGC	Arg	**C**
	CTA	Leu	CCA	Pro	CAA	Gln	CGA	Arg	**A**
	CTG	Leu	CCG	Pro	CAG	Gln	CGG	Arg	**G**
A	ATT	Ile	ACT	Thr	AAT	Asn	AGT	Ser	**T**
	ATC	Ile	ACC	Thr	AAC	Asn	AGC	Ser	**C**
	ATA	Ile	ACA	Thr	AAA	Lys	AGA	Arg	**A**
	ATG	Met	ACG	Thr	AAG	Lys	AGG	Arg	**G**
G	GTT	Val	GCT	Ala	GAT	Asp	GGT	Gly	**T**
	GTC	Val	GCC	Ala	GAC	Asp	GGC	Gly	**C**
	GTA	Val	GCA	Ala	GAA	Glu	GGA	Gly	**A**
	GTG	Val	GCG	Ala	GAG	Glu	GGG	Gly	**G**

Do có nhiều *codon* cùng mã hoá một axít amin nên số axít amin được mã hoá chỉ là 20 [16]. Tên đầy đủ và viết tắt của 20 axít amin được liệt kê đầy đủ trong Bảng 1.2.

Bảng 1.2: Danh sách 20 axít amin.

STT	Tên axít 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

1.1.2. Các phép biến đổi trên chuỗi axít amin

Theo thuyết tiến hoá của Darwin thì các sinh vật đều có chung một nguồn gốc [19]. Sự giống nhau giữa các sinh vật có thể được thể hiện bằng sự giống nhau ở kiểu hình, kiểu gen hoặc các chuỗi nuclêotít, axít amin. Hai chuỗi axít amin ở hai sinh vật khác nhau cùng tiến hoá từ một chuỗi axít amin tổ tiên thì gọi là hai chuỗi axít amin *tương đồng*. Hai chuỗi axít amin tương đồng có các khác biệt là do có các biến đổi trong quá trình tiến hoá. Các biến đổi trên chuỗi axít amin có thể do các

biến đổi ở vùng mã hoá của chuỗi ADN trước quá trình tổng hợp prôtêin hoặc do biến đổi tại các bước phiên mã, dịch mã của quá trình tổng hợp prôtêin. Các phép biến đổi thông thường được chia làm ba loại chính là [45]:

- **Thay thế**: một axít amin này bị thay thế bằng một axít amin khác.
- **Xoá**: một hoặc một số axít amin bị xoá khỏi chuỗi prôtêin.
- **Chèn**: một hoặc một số axít amin được chèn vào chuỗi prôtêin.

Hình 1.2 minh hoạ một ví dụ các phép biến đổi trên hai chuỗi axít amin. Cột 1, 2 và 3 chứa các axít amin khác nhau thể hiện các phép thay thế. Các ký tự trống (-) trên cột 4 và 6 thể hiện các phép chèn hoặc xoá đã xảy ra.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Chuỗi 1	**E**	**H**	**A**	**-**	D	**N**	E	M	C	Q	L	K	P	L	P
Chuỗi 2	**F**	**G**	**D**	**R**	D	**-**	E	M	C	Q	L	K	P	L	P

Hình 1.2: Một ví dụ các phép biến đổi trên hai chuỗi axít amin tương đồng.

1.1.3. Sắp hàng đa chuỗi axít amin

Quá trình biến đổi làm cho các chuỗi axít amin tương đồng khác nhau cả về nội dung cũng như độ dài. Sắp hàng đa chuỗi sẽ giúp làm rõ các phép biến đổi giữa các chuỗi axít amin. Sắp hàng đa chuỗi có thể được hiểu như một ma trận các axít amin, trong đó mỗi hàng chính là một chuỗi axít amin; còn mỗi cột (vị trí) chứa các axít amin tương đồng của các chuỗi (xem thêm Hình 1.3). Chúng ta có thể sử dụng sắp hàng đa chuỗi để xây dựng cây phân loài giúp đánh giá nguồn gốc tiến hóa của các chuỗi [44]. Kích thước của một sắp hàng đa chuỗi được hiểu là số lượng chuỗi có trong sắp hàng đó, còn chiều dài của một sắp hàng đa chuỗi chính là chiều dài của các chuỗi trong sắp hàng. Hình 1.3 minh hoạ một ví dụ của một sắp hàng đa chuỗi với bốn chuỗi axít amin của bốn loài linh trưởng. Sắp hàng có chiều dài là 15.

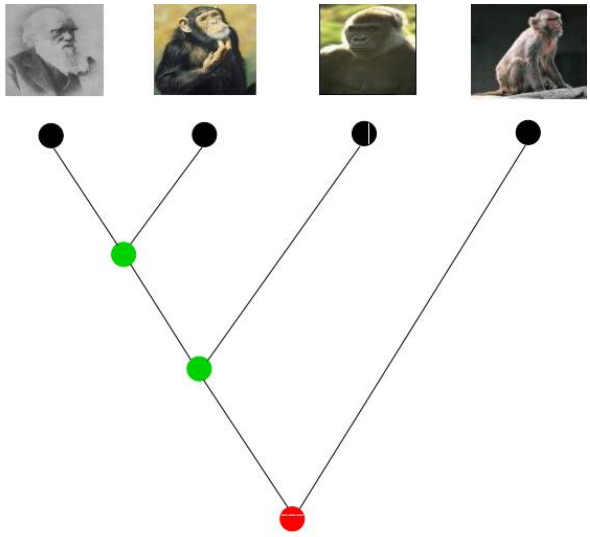
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Người	**E**	**H**	**D**	**-**	**N**	**D**	**E**	M	C	Q	L	K	P	L	P
Tinh tinh	**F**	**H**	**D**	**R**	**-**	**D**	**E**	M	C	Q	L	K	P	L	P
Khỉ đột	**F**	**G**	**D**	**R**	**-**	**D**	**E**	M	C	Q	L	K	P	L	P
Vượn	**F**	**G**	**D**	**R**	**-**	**V**	**H**	M	C	Q	L	K	P	L	P

Hình 1.3: Minh họa một sắp hàng đa chuỗi axít amin của bốn loài linh trưởng.

1.1.4. Cây phân loài

Cây phân loài (cây tiến hóa) là một dạng sơ đồ phân nhánh thể hiện quá trình tiến hóa của các loài sinh vật và cho biết sự tương đồng và khác biệt về giữa chúng. Các sinh vật liên kết với nhau trong cây được cho là có cùng một tổ tiên chung.



Hình 1.4: Một ví dụ về cây phân loài giữa bốn loài linh trưởng.

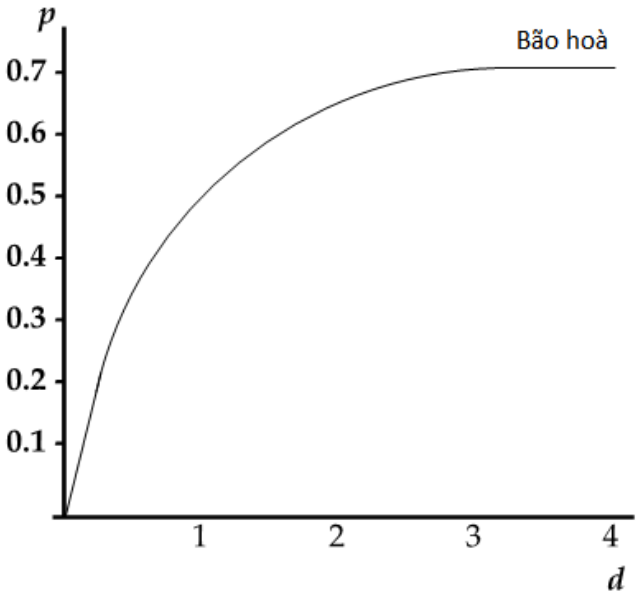
Trong cây phân loài mỗi nút lá biểu diễn cho một loài sinh vật hiện tại, mỗi nút cha đại diện cho tổ tiên gần nhất của các nút con. Độ dài cạnh có thể được hiểu như là ước lượng khoảng cách về thời gian giữa các loài. Trong luận án này, nếu không có chú thích thêm thì cây phân loài được gọi tắt là cây. Hình 1.4 minh họa một cây phân loài thể hiện mối quan hệ giữa một số loài linh trưởng.

1.2. Mô hình hoá quá trình biến đổi axít amin

1.2.1. Sự khác biệt giữa hai chuỗi tương đồng

Có sự khác nhau giữa hai chuỗi axít amin tương đồng cùng tiến hóa từ một tổ tiên chung là do có các biến đổi giữa các axít amin trong quá trình tiến hóa. Hai loại khoảng cách thường dùng để đo sự khác biệt giữa hai chuỗi axít amin tương đồng **x** và **y** là khoảng cách quan sát và khoảng cách di truyền [44]:

- **Khoảng cách quan sát** giữa hai chuỗi axít amin **x** và **y** là tỷ lệ giữa số vị trí trên hai chuỗi có các axít amin không giống nhau so với chiều dài chuỗi.
- **Khoảng cách di truyền** giữa hai chuỗi axít amin **x** và **y** là tỷ lệ giữa số lượng thực tế các biến đổi đã xảy ra giữa hai chuỗi trong quá trình tiến hoá so với chiều dài chuỗi.



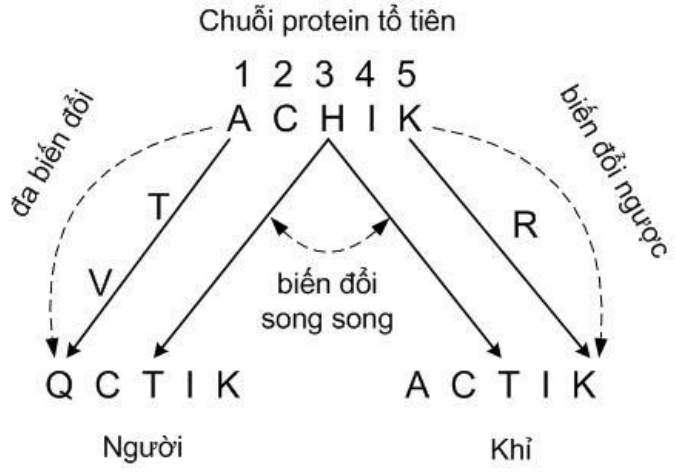
Hình 1.5: Quan hệ giữa khoảng cách di truyền (*d*) và khoảng cách quan sát (*p*).

Nếu khoảng cách di truyền nhỏ thì nó có thể được ước lượng tương đối chính xác bằng khoảng cách quan sát (xem minh họa trong Hình 1.5) [44]. Tuy nhiên, nếu có nhiều phép biến đổi xảy ra tại một vị trí trên chuỗi axít amin thì ước lượng khoảng cách di truyền bằng khoảng cách quan sát cho độ chính xác thấp. Việc

không ước lượng được khoảng cách di truyền bằng khoảng cách quan sát là do sự phức tạp của quá trình biến đổi axít amin giữa hai chuỗi (xem Hình 1.6) [60].

Có ba hiện tượng xảy ra trong quá trình biến đổi của các chuỗi axít amin làm cho khoảng cách quan sát nhỏ hơn rất nhiều so với khoảng cách di truyền là [60]:

- **Đa biến đổi (multiple substitutions)**: Có nhiều phép biến đổi cùng xảy ra tại một vị trí nhưng chúng ta chỉ quan sát được nhiều nhất 1 phép biến đổi (vị trí 1 trong Hình 1.6).
- **Biến đổi song song (parallel substitutions)**: Hai phép biến đổi giống hệt nhau cùng xảy ra tại một ví trí trên hai chuỗi con. Chúng ta không quan sát được phép biến đổi này vì trên hai chuỗi con không có sự khác biệt (vị trí 3 trong Hình 1.6).
- **Biến đổi ngược (back substitutions)**: Có nhiều phép biến đổi xảy ra nhưng axít amin ban đầu và cuối cùng lại giống nhau, chúng ta không quan sát được biến đổi nào giữa hai chuỗi con (vị trí 5 trong Hình 1.6).



Hình 1.6: Những hiện tượng phức tạp trong quá trình biến đổi các axít amin.

Giả sử chúng ta có hai chuỗi prôtêin của người là ‘QCTIK’ và khỉ là ‘ACTIK’ cùng được biến đổi từ một chuỗi prôtêin tổ tiên. Khi so sánh sự khác biệt giữa hai chuỗi này chúng ta chỉ thấy một phép biến đổi Q↔A ở vị trí số 1. Tuy nhiên, đã có ba phép biến đổi (A↔T↔V↔Q) xảy ra ở vị trí số 1; hai phép biến đổi (H↔T,

H↔T) xảy ra ở vị trí số 3 và hai phép biến đổi (K↔R↔K) xảy ra ở vị trí số 5. Khoảng cách quan sát được tính là p = 1/5 = 0,2; trong khi khoảng cách thực tế (khoảng cách di truyền) là d = (3+2+2)/7 = 1,4 tương đương có trung bình 1,4 phép biến đổi trên mỗi vị trí của chuỗi. Như vậy, cách phân tích sự khác biệt bằng quan sát không cho kết quả chính xác về quá trình biến đổi giữa hai chuỗi. Để ước lượng khoảng cách di truyền, chúng ta phải sử dụng mô hình xác suất ngẫu nhiên để mô phỏng quá trình biến đổi giữa các axít amin.

1.2.2. Mô hình Markov cho quá trình biến đổi axít amin

Xét quá trình biến đổi giữa các axít amin tại một vị trí trên chuỗi prôtêin. Quá trình biến đổi này là ngẫu nhiên và liên tục theo thời gian 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} chính là tập 20 axít amin như trong Bảng 1.2. Quá trình biến đổi axít amin có thể được mô hình hóa bởi một quá trình Markov [22, 28, 44] với các thuộc tính sau đây:

- Độc lập với quá khứ (memoryless): Tốc độ biến đổi từ axít amin *x* thành axít amin *y* không phụ thuộc vào quá trình biến đổi trước đó của axít amin *x*.
- Đồng nhất (homologous): Tốc độ biến đổi giữa các axít amin là đồng nhất trong toàn bộ quá trình biến đổi.
- Liên tục (continuous): Quá trình biến đổi giữa các axít amin có thể diễn ra bất cứ thời điểm nào trong suốt quá trình biến đổi.
- Ổn định (stationary): Tần số của các axít amin là không đổi trong suốt quá trình biến đổi. Gọi **Π** = $\{\pi_i\}$ với i = 1, … 20 là véc tơ tần số xuất hiện của 20 axít amin, khi đó $\sum_{i=1}^{20} \pi_i = 1$ và các π_i không đổi theo thời gian.

Gọi $\mathbf{P}(t) = \{p_{ij}(t),\ i \in \mathbf{S},\ j \in \mathbf{S}\}$ là ma trận xác suất chuyển giữa các axít amin sau một khoảng thời gian t; $p_{ij}(t)$ là xác suất chuyển từ axít amin i ($i \in \mathbf{S}$) sang axít amin j ($j \in \mathbf{A}$) sau một khoảng thời gian t. **P** có kích thước 20*20 và với mỗi axít amin i, ta có:

$$\sum_{j \in \boldsymbol{S}} p_{ij}(t) = 1 \tag{1.1}$$

và $p_{ij}(t) > 0$ với $\forall t > 0$.

$\mathbf{P}(t)$ cũng thỏa mãn công thức Chapman-Kolmogorov:

$$\mathbf{P}(t+s) = \mathbf{P}(t) + \mathbf{P}(s) \tag{1.2}$$

với các điều kiện khởi tạo là:

$$p_{ii}(0) = 1, \ cho \ \forall i = j$$

$$p_{ij}(0) = 0, \ cho \ \forall i \ \neq j$$

Với giá trị Δt nhỏ, ma trận xác suất chuyển $\mathbf{P}(\Delta t)$ có thể được tính xấp xỉ tuyến tính theo khai triển Taylor như sau:

$$\mathbf{P}(\Delta t) \approx \mathbf{P}(0) + \Delta t * \mathbf{Q} \tag{1.3}$$

trong đó $\mathbf{Q} = \{q_{ij}, \ i \in \boldsymbol{S}, \ j \in \boldsymbol{S}\}$ là ma trận tốc độ biến đổi tức thì (instantaneous substitution rate matrix) giữa các axít amin; $\mathbf{Q}$ có kích thước 20*20 và q_{ij} là tốc độ biến đổi tức thì từ axít amin i sang axít amin j.

Xét một axít amin i, để đảm bảo điều kiện tổng xác suất chuyển từ i đến các trạng thái khác bằng 1 sau một khoảng thời gian t bất kì (công thức 1.1) thì các giá trị của $\mathbf{Q}$ phải thỏa mãn điều kiện:

$$\sum_{j \in \boldsymbol{S}} q_{\boldsymbol{ij}} = 0 \text{ hay } q_{ii} = -\sum_{j \in \boldsymbol{S}, j \neq i} q_{\boldsymbol{ij}} \tag{1.4}$$

Chúng ta có thể coi q_{ij} là lượng biến đổi từ axít amin i sang axít amin j trong một đơn vị thời gian, còn q_{ii} là tổng lượng biến đổi rời khỏi axít amin i. Giá trị q_{ij} càng lớn thể hiện tốc độ biến đổi từ axít amin i sang axít amin j càng lớn.

Dựa vào công thức Chapman-Kolmogorov (công thức 1.2), chúng ta có thể tính $\mathbf{P}(t)$ từ $\mathbf{Q}$ và t như sau:

$$\mathbf{P}(t) = e^{t\mathbf{Q}}. \tag{1.5}$$

Chúng ta gọi

$$\mu = -\sum_{j \in \mathbf{S}} \pi_j q_{jj} \tag{1.6}$$

là tổng số lượng biến đổi axít amin trong một đơn vị thời gian.. Ta có $d = \mu t$ là tổng số lượng biến đổi axít amin sau một khoảng thời gian t. Ma trận tốc độ biến đổi **Q** được chuẩn hóa sao cho tổng số lượng axít amin biến đổi trong một đơn vị thời gian bằng 1 ($\mu = 1$). Tức là, $p_{ij}(t)$ là xác xuất axít amin i biến đổi thành axít amin j nếu có d biến đổi giữa axít amin i và axít amin j.

Quá trình biến đổi axít amin thường được giả sử có tính thuận nghịch theo thời gian (time reversible), tức là số lượng biến đổi từ axít amin i sang axít amin j bằng với số lượng biến đổi từ axít amin j sang axít amin i (mặc dù tần số xuất hiện của hai axít amin i, j có thể khác nhau). Điều này được thể hiện bằng công thức:

$$\pi_i q_{ij} = \pi_j q_{ji} \tag{1.7}$$

hay

$$\frac{q_{ij}}{\pi_j} = \frac{q_{ji}}{\pi_i}$$

Ta kí hiệu $r_{ij} = \frac{q_{ij}}{\pi_j}$; $r_{ji} = \frac{q_{ji}}{\pi_i}$ và gọi r_{ij}, r_{ji} $(r_{ij} = r_{ji})$ là hệ số hoán đổi (exchangeability coefficient) giữa hai axít amin i và j. Hệ số hoán đổi (hay tốc độ biến đổi tương đối) giữa hai axít amin i và j càng lớn thể hiện sự biến đổi giữa hai axít amin i và j xảy ra càng nhiều và ngược lại.

Ma trận tốc độ biến đổi tức thì **Q** có thể được biểu diễn bởi ma trận hoán đổi $\mathbf{R} = \{r_{ij}\}$ và vectơ tần số xuất hiện $\mathbf{\Pi} = \{\pi_i\}$ như sau:

$$q_{ij} = \begin{cases} \pi_j r_{ij} & nếu\ i \neq j \\ -\sum_{x \neq i} q_{ix} & nếu\ i = j \end{cases} \tag{1.8}$$

hoặc có thể viết gọn dưới dạng: $\mathbf{Q} = \mathbf{\Pi} * \mathbf{R}$. Chúng ta cũng thấy ma trận hệ số hoán đổi **R** có dạng đối xứng qua đường chéo chính. Như vậy chúng ta có thể ước lượng

Π và **R** thay cho ước lượng **Q**. Hình 1.7 minh họa hai thành phần **R** và **Π** của mô hình LG [49], trong đó ma trận **R** là 19 dòng đầu tiên và véc tơ **Π** là dòng cuối cùng. Do **R** có dạng đối xứng nên chúng ta chỉ cần lưu trữ một nửa ma trận nằm dưới đường chéo chính.

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. Số tham số cần ước lượng của **R** là 19 * 20/2 - 1 = 189, do **R** là ma trận đối xứng và được chuẩn hoá (công thức 1.6 và 1.8). Để ước lượng **Q** chúng ta cần phải ước lượng tổng cộng 208 tham số. Trong nhiều nghiên cứu về mô hình biến đổi axít amin, ma trận biểu diễn tốc độ biến đổi tức thì **Q** còn được gọi là mô hình **Q**.

Mô hình **Q** được sử dụng trong hầu hết các hệ thống phân tích chuỗi prôtêin. Cụ thể, **Q** được sử dụng để phân tích sự khác biệt và tính khoảng cách di truyền giữa các chuỗi prôtêin. Mô hình **Q** là thành phần cơ bản và quan trọng nhất trong các hệ thống xây dựng cây tiến hóa sử dụng các phương pháp xác suất thống kê [28, 66]. Ngoài ra, **R** cũng có thể được sử dụng như ma trận điểm (score matrix) trong các hệ thống sắp hàng đa chuỗi prôtêin. Chúng ta có thể xem thêm các ứng dụng của **Q** trong tài liệu [59].

1.3. Bài toán ước lượng mô hình biến đổi axít amin

Quá trình biến đổi của các axít amin có thể được mô hình hoá bởi mô hình **Q**. Các tham số của mô hình **Q** có thể được ước lượng từ các sắp hàng đa chuỗi axít amin. Bài toán xây dựng mô hình biến đổi axít amin từ các sắp hàng đa chuỗi axít amin được tóm tắt ngắn gọn như sau:

Dữ liệu vào: Dữ liệu đầu vào là một tập các sắp hàng đa chuỗi axít amin. Các sắp hàng thường có độ dài từ vài chục đến vài chục nghìn axít amin. Tập các sắp hàng đa chuỗi được ký hiệu là $\mathbf{A} = \{D^1, \ldots D^N\}$, trong đó N là số lượng sắp hàng còn D^a ($1 \leq a \leq N$) là ký hiệu sắp hàng thứ a trong tập **A**.

	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	
	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

Hình 1.7: Mô hình biến đổi axít amin LG [49]**.**

Bài toán: Ước lượng mô hình biến đổi axít amin mô tả các biến đổi axít amin trong quá trình tiến hoá trên các chuỗi prôtêin đầu vào. Chúng ta cần đề xuất các phương pháp cho kết quả chính xác cao với thời gian thực hiện chấp nhận được.

Dữ liệu ra: Một mô hình biến đổi axít amin **Q** thể hiện quá trình biến đổi của các chuỗi axít amin ở dữ liệu đầu vào **A**.

Ước lượng mô hình **Q** là một bài toán phức tạp bởi chúng ta phải ước lượng khoảng 200 tham số. Nhiều phương pháp xây dựng mô hình **Q** đã được nghiên cứu và đề xuất trong suốt hơn 30 năm qua. Các phương pháp có thể chia theo hai hướng tiếp cận chính: phương pháp đếm (counting approach) và phương pháp cực đại khả năng (maximum likelihood approach).

1.4. Các phương pháp ước lượng mô hình biến đổi axít amin

1.4.1. Phương pháp đếm

Trong phương pháp đếm, các tham số cần ước lượng của mô hình được tính toán một cách trực tiếp từ dữ liệu. Hai ma trận phổ biến được ước lượng bằng phương pháp đếm là PAM và BLOSUM.

1.4.1.1. Ma trận PAM (Point Accepted Mutation)

Ma trận PAM là mô hình biến đổi axít amin đầu tiên được xây dựng vào năm 1978 [21]. Tác giả của mô hình PAM là Dayhoff và các cộng sự đã sử dụng bộ dữ liệu gồm 71 nhóm prôtêin, trong đó mỗi nhóm bao gồm các chuỗi prôtêin có quan hệ gần nhau (giống nhau ít nhất 85%). Sự giống nhau cao giữa các chuỗi prôtêin giúp đảm bảo các biến đổi trực tiếp giữa các axít amin (ví dụ A → R) chiếm phần lớn, còn các biến đổi gián tiếp (ví dụ A→ X → R) chỉ chiếm phần nhỏ.

Ma trận PAM1 cho biết xác suất thay thế giữa các axít amin nếu có khoảng 1% tổng số axít amin bị biến đổi. Các giá trị của ma trận PAM1 cho biết xác suất

biến đổi từ axít amin *i* thành axít amin *j* sau một đơn vị thời gian. Các phần từ không nằm trên đường chéo chính của ma trận được tính bởi công thức sau [21]:

$$\mathrm{PAM1}(i,j) = \frac{\lambda m_j b_{ij}}{\sum_{i \in S} b_{ij}} \tag{1.9}$$

trong đó m_j là độ đột biến của axít amin *j*, được tính tương đối so với các axít amin khác (xem thêm Bảng 1.3); b_{ij} là số lần biến đổi giữa hai axít amin *i* và *j* quan sát được từ dữ liệu còn λ là hằng số được chọn sao cho tổng số biến đổi trên toàn bộ dữ liệu là 1%. Các phần tử nằm trên đường chéo chính của ma trận PAM được chọn sao cho tổng của bất kỳ cột nào cũng bằng một.

Bảng 1.3: Danh sách độ đột biến tương đối của 20 axít amin. Độ đột biến của Ala (A) được đặt là 100. Asn (N) và Ser (S) là 2 axít amin có độ đột biến lớn nhất còn Trp (W) và Cys (C) là 2 axít amin có độ đột biến nhỏ nhất.

Axít amin	Độ đột biến		Axít amin	Độ đột biến
Asn	134		His	66
Ser	120		Arg	65
Asp	106		Lys	56
Glu	102		Pro	56
Ala	100		Gly	49
Thr	97		Tyr	41
Ile	96		Phe	41
Met	94		Leu	40
Gln	93		Cys	20
Val	74		Trp	18

Ma trận PAM1 được sử dụng làm cơ sở để tính toán các ma trận PAM khác với giả định rằng những đột biến lặp đi lặp lại sẽ tuân theo cùng một mô hình như của ma trận PAM1 và nhiều phép thay thế có thể xảy ra ở cùng một vị trí. Dayhoff đã xây dựng đến ma trận PAM250. Con số kèm theo ma trận PAM càng cao thể hiện khoảng cách tiến hóa càng lớn. Ví dụ PAM150 được sử dụng cho các chuỗi có khoảng cách xa hơn, có nhiều khác biệt và biến đổi hơn so với PAM100. Hình 1.8 minh họa ma trận PAM250 với các hệ số được nhân 100 lần [10].

Năm 1992, khi số lượng các chuỗi prôtêin được thu thập nhiều hơn, nhóm nghiên cứu của Jones đã áp dụng phương pháp đếm tương tự như Dayhoff nhưng trên một tập dữ liệu lớn hơn để xây dựng mô hình JTT [40]. Mô hình JTT được sử dụng rộng rãi đối với các phân tích về cây phát sinh loài.

	A	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
A	13	6	9	9	5	8	9	12	6	8	6	7	7	4	11	11	11	2	4	9
R	3	17	4	3	2	5	3	2	6	3	2	9	4	1	4	4	3	7	2	2
N	4	4	6	7	2	5	6	4	6	3	2	5	3	2	4	5	4	2	3	3
D	5	4	8	11	1	7	10	5	6	3	2	5	3	1	4	5	5	1	2	3
C	2	1	1	1	52	1	1	2	2	2	1	1	1	1	2	3	2	1	4	2
Q	3	5	5	6	1	10	7	3	7	2	3	5	3	1	4	3	3	1	2	3
E	5	4	7	11	1	9	12	5	6	3	2	5	3	1	4	5	5	1	2	3
G	12	5	10	10	4	7	9	27	5	5	4	6	5	3	8	11	9	2	3	7
H	2	5	5	4	2	7	4	2	15	2	2	3	2	2	3	3	2	2	3	2
I	3	2	2	2	2	2	2	2	2	10	6	2	6	5	2	3	4	1	3	9
L	6	4	4	3	2	6	4	3	5	15	34	4	20	13	5	4	6	6	7	13
K	6	18	10	8	2	10	8	5	8	5	4	24	9	2	6	8	8	4	3	5
M	1	1	1	1	0	1	1	1	1	2	3	2	6	2	1	1	1	1	1	2
F	2	1	2	1	1	1	1	1	3	5	6	1	4	32	1	2	2	4	20	3
P	7	5	5	4	3	5	4	5	5	3	3	4	3	2	20	6	5	1	2	4
S	9	6	8	7	7	6	7	9	6	5	4	7	5	3	9	10	9	4	4	6
T	8	5	6	6	4	5	5	6	4	6	4	6	5	3	6	8	11	2	3	6
W	0	2	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	55	1	0
Y	1	1	2	1	3	1	1	1	3	2	2	1	2	15	1	2	2	3	31	2
V	7	4	4	4	4	4	4	4	5	4	15	10	4	10	5	5	5	72	4	17

Hình 1.8: Ma trận PAM250 thể hiện xác suất biến đổi giữa các axít amin (các giá trị được nhân với 100). Ví dụ xác suất biến đổi từ A sang R là 3% và từ A sang N là 4%.

1.4.1.2. Ma trận BLOSUM (BLOcks SUbstitution Matrix)

Ma trận BLOSUM được giới thiệu lần đầu tiên bởi Henikoff và Henikoff vào năm 1992 [37]. Ma trận này được dùng chủ yếu cho bài toán sắp hàng đa chuỗi. Các tác giả đã sử dụng bộ dữ liệu BLOCKS [36], đây là bộ dữ liệu chứa các chuỗi prôtêin do chính nhóm tác giả xây dựng. Họ đã tìm các đoạn bảo tồn (conserved regions) để từ đó tính ra các tần số xuất hiện của các axít amin và xác suất biến đổi giữa các cặp các axít amin. Sau đó, các tác giả tính giá trị log-odds cho mỗi cặp biến đổi axít amin có thể có.

Tương tự như với ma trận PAM, cũng có nhiều ma trận BLOSUM được xây dựng và các ma trận này thường được ký hiệu là BLOSUM45, BLOSUM62, BLOSUM80. BLOSUM*k* có nghĩa là các đoạn đa sắp hàng mà các chuỗi giống nhau ít nhất *k*% được sử dụng. Tuy nhiên, ngược lại với PAM, giá trị số đi kèm ma trận BLOSUM thể hiện độ tương đồng của dữ liệu, BLOSUM80 được dùng cho dữ liệu có độ tương đồng cao hơn BLOSUM45.

1.4.2. Phương pháp cực đại khả năng (maximum likelihood)

1.4.2.1. Giới thiệu chung

Một trong các nhược điểm chính của các phương pháp đếm là chúng thường chỉ áp dụng tốt được cho các tập dữ liệu có độ tương đồng cao [49]. Để khắc phục hạn chế trên, phương pháp cực đại khả năng (maximum likelihood, viết tắt là ML) đã được đề xuất để xây dựng mô hình **Q** [6, 64]. Một số nghiên cứu đã chỉ ra rằng phương pháp cực đại khả năng có thể giúp tránh các lỗi có tính hệ thống và giúp tận dụng các thông tin trong các sắp hàng đa chuỗi prôtêin hiệu quả hơn so với các phương pháp đếm [63]. Năm 1996, nhóm tác giả Adachi và Haseqawa sử dụng phương pháp ML để phân tích các chuỗi prôtêin ti thể của 20 loài động vật có xương sống để xây dựng mô hình mtREV [6]. Nhóm tác giả cho thấy mô hình mtREV tốt hơn các mô hình khác khi phân tích quá trình tiến hóa giữa các loài sinh vật dựa vào các chuỗi prôtêin ti thể.

Tuy nhiên, thời gian tính toán là một trong những cản trở lớn nhất trong việc áp dụng phương pháp ML trên những tập dữ liệu prôtêin lớn. Nhóm tác giả Whelan và Goldman đã đề xuất phương pháp ML xấp xỉ và áp dụng trên cơ sở dữ liệu gồm 3905 chuỗi prôtêin và xây dựng mô hình WAG vào năm 2002 [63]. Mô hình WAG cho kết quả tốt hơn các mô hình khác khi được dùng để phân tích quá trình tiến hóa giữa các sinh vật dựa vào các chuỗi prôtêin.

Gần đây nhất, vào năm 2008, nhóm tác giả Le và Gascuel đã cải tiến phương pháp của Whelan và Goldman bằng cách kết hợp thêm thông tin về tính không đồng

nhất trong tốc độ biến đổi theo vị trí vào quá trình xây dựng mô hình **Q**. Nhóm tác giả đã áp dụng phương pháp đề xuất trên cơ sở dữ liệu prôtêin Pfam bao gồm khoảng 50000 chuỗi với khoảng 6,5 triệu axít amin để xây dựng mô hình LG [49]. LG hiện được cho là mô hình chung tốt nhất để phân tích các chuỗi prôtêin.

1.4.2.2. Ước lượng mô hình bằng phương pháp cực đại khả năng

Giả sử $D = \{D_1, \ldots D_l\}$ là một sắp hàng đa chuỗi có chiều dài l trong đó D_i ($1 \leq i \leq l$) là vị trí thứ i của sắp hàng. Gọi T là cây phân loài tương ứng với sắp hàng đa chuỗi D. Sử dụng mô hình **Q** như đã trình bày ở phần 1.2.1, giá trị likelihood của **Q** và T đối với D được tính theo công thức [44]:

$$L(\mathbf{Q},T \mid D) = \prod_{i=1}^{l} L(\mathbf{Q},T \mid D_i) \tag{1.10}$$

trong đó $L(\mathbf{Q},T \mid D_i)$ là likelihood của **Q** và T đối với vị trí D_i, giá trị này có thể tính một cách hiệu quả bằng một thuật toán cắt tỉa của Felsenstein (xem chi tiết trong tài liệu [26]).

Phương pháp cực đại khả năng để ước lượng mô hình biến đổi axít amin được giới thiệu lần đầu bởi Adachi và Haseqawa [6]. Giả sử chúng ta có một bộ dữ liệu gồm N sắp hàng đa chuỗi prôtêin ký hiệu là $\mathbf{A} = \{D^1, \ldots D^N\}$. Ký hiệu $\mathbf{T} = \{ T^1, T^2, \ldots T^N \}$ là tập các cây, trong đó mỗi $T^a \in \mathbf{T}$ là cây tương ứng được xây dựng từ sắp hàng D^a với mô hình **Q**. Giá trị likelihood của mô hình **Q** và **T** được tính theo công thức:

$$L(\mathbf{Q}, \mathbf{T}) = \prod_{a=1}^{N} L(\mathbf{Q},T^a \mid D^a). \tag{1.11}$$

Mô hình **Q** khi đó được ước lượng bằng cách tìm cực đại của giá trị likelihood $L(\mathbf{Q}, \mathbf{T})$ theo công thức sau:

$$\mathbf{Q} = \arg\max_{\mathbf{Q}} \{L(\mathbf{Q}, \mathbf{T})\} \tag{1.12}$$

Quá trình tìm cực đại cho giá trị likelihood $L(\mathbf{Q}, \mathbf{T})$ theo công thức 1.11 là một bài toán rất khó vì chúng ta phải tối ưu cùng lúc các tham số của mô hình **Q** cùng tất cả các cây phân loài **T** (bao gồm cả cấu trúc và độ dài các cạnh. Các nghiên cứu đã chỉ ra rằng các hệ số của **Q** được ước lượng tương đối chính xác khi sử dụng cây phân loài gần tối ưu [63]. Vì vậy, công thức 1.11 có thể được đơn giản hóa và xấp xỉ bởi:

$$L(\mathbf{Q}, \mathbf{T}) = \prod_{a=1}^{N} L(\mathbf{Q} \mid T^{*a}, D^{a}) \tag{1.13}$$

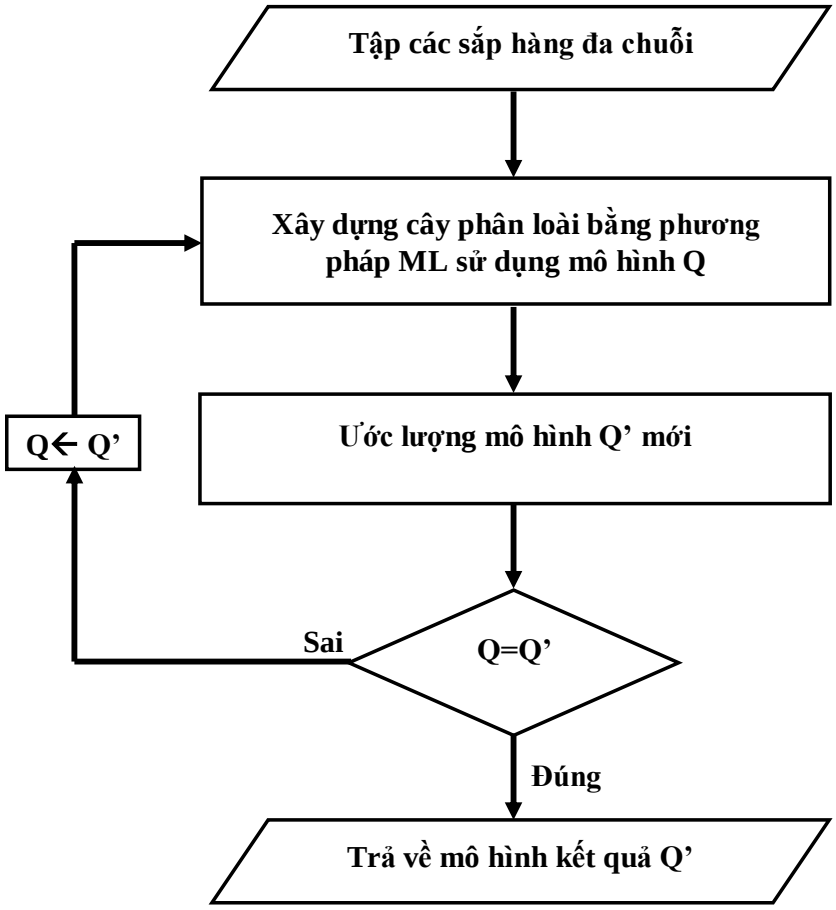
với T^{*a} là cây phân loài *gần tối ưu* của D^{a}. Do đó công thức để ước lượng mô hình **Q** có dạng:

$$\mathbf{Q} = \arg\max_{\mathbf{Q}} \left\{ \prod_{a=1}^{N} L(\mathbf{Q} \mid T^{*a}, D^{a}) \right\} \tag{1.14}$$

Lược đồ thuật toán ước lượng mô hình biến đổi axít amin bằng phương pháp cực đại khả năng được trình bày ở Hình 1.9 (xem chương 2 để biết thêm chi tiết về thuật toán).

1.5. Xây dựng cây phân loài bằng phương pháp ML

Một trong các bước quan trọng trong việc ước lượng **Q** là xây dựng các cây phân loài. Có nhiều phương pháp đã được đề xuất để xây dựng cây phân loài từ một sắp hàng đa chuỗi như phương pháp dựa vào khoảng cách [30, 52], phương pháp Maximum parsimony (MP) [29] hay phương pháp cực đại khả năng (ML) [26]. Hiện nay phương pháp ML được sử dụng phổ biến và rộng rãi vì thường cho kết quả tốt hơn các phương pháp khác [28, 35, 56, 58].



Hình 1.9: Lược đồ quá trình ước lượng mô hình biến đổi axít amin bằng phương pháp ML.

Trong phương pháp ML, cây "tốt nhất" được hiểu là cây có giá trị likelihood lớn nhất. Giá trị likelihood của một cây *T* đối với một mô hình biến đổi **Q** và dữ liệu *D* được tính như sau:

$$L(T \mid \mathbf{Q}, D) = \prod_{i=1}^{l} L(T \mid \mathbf{Q}, D_i) \qquad (1.15)$$

Như vậy chúng ta sẽ cần tìm cây *T* (bao gồm cấu trúc cây và độ dài các cạnh) sao cho giá trị likelihood theo công thức 1.15 đạt cực đại.

Bài toán tối ưu cây *T* là một bài toán NP-khó [15, 28] do số lượng cây có cấu trúc khác nhau tương ứng với cùng một sắp hàng là (2n-5)!!. Số lượng này tăng

nhanh theo số lượng chuỗi. Một số phương pháp tìm kiếm gần đúng đã được đề xuất [33, 34, 61].

1.6. Các phương pháp so sánh hai mô hình

1.6.1. So sánh bằng việc xây dựng cây ML

Phương pháp so sánh hai mô hình dựa trên các cây phân loài xây dựng bằng phương pháp ML là cách so sánh phổ biến nhất. Cả hai mô hình cùng được sử dụng để xây dựng cây phân loài bằng phương pháp ML với cùng một tập các sắp hàng đa chuỗi.

Gọi M_1, M_2 là hai mô hình cần so sánh. Với mỗi sắp hàng D^a, cây phân loài tương ứng với M_1 là T^a_1 , với M_2 là T^a_2. Giá trị likelihood của hai cây tương ứng là $L(T^a_1)$ và $L(T^a_2)$. Nếu $L(T^a_1) > L(T^a_2)$ thể hiện M_1 tốt hơn M_2. Ngược lại, nếu $L(T^a_1) < L(T^a_2)$ thể hiện M_2 tốt hơn M_1. Trong thực tế, để việc tính toán dễ dàng hơn người ta thường so sánh giá trị log(likelihood) với log là hàm lôgarít tự nhiên.

1.6.2. So sánh cấu trúc cây

Phương pháp so sánh cấu trúc cây không dùng để đánh giá mô hình nào tốt hơn mà được dùng để chỉ ra sự khác biệt giữa hai mô hình khi sử dụng để xây dựng cây phân loài. Chúng ta so sánh cấu trúc hai cây xây dựng từ cùng một sắp hàng với hai mô hình khác nhau. Hai cây có cấu trúc càng giống nhau thể hiện hai mô hình càng giống nhau.

Để đo sự khác biệt giữa cấu trúc của hai cây, chúng tôi sử dụng khoảng cách Robinson-Fould (RF) [51]. Khoảng cách RF giữa cấu trúc của hai cây là tỷ lệ giữa số phân vùng chỉ có ở một trong hai cây trên tổng số phân vùng của cả hai cây. Khoảng cách RF có khoảng giá trị từ 0,0 đến 1,0. Giá trị RF giữa hai cây càng nhỏ thì cấu trúc của hai cây càng giống nhau.

1.6.3. So sánh độ tương quan Pearson

Độ tương quan Pearson giữa hai ma trận hệ số hoán đổi R_1 của mô hình M_1, R_2 của mô hình M_2 sẽ giúp đánh giá mối quan hệ tuyến tính giữa các hệ số tương ứng của hai ma trận. Độ tương quan Pearson có khoảng giá trị từ -1,0 đến 1,0. Độ tương quan bằng -1,0 thể hiện hai ma trận có tương quan cùng giảm, ngược lại nếu độ tương quan bằng 1,0 thể hiện hai ma trận có tương quan cùng tăng. Độ tương quan bằng 0 thể hiện hai ma trận không có tương quan với nhau. Chúng ta cũng có so sánh tương tự với véc tơ tần số xuất hiện các axít amin của hai mô hình.

1.7. Kết luận chương

Các chuỗi axít amin (hay prôtêin) là một thành phần vô cùng quan trọng của sự sống. Với sự phát triển của công nghệ sinh học, số lượng chuỗi axít amin mới được thu thập đang tăng theo cấp số nhân. Quá trình tiến hoá và biến đổi giữa các chuỗi axít amin diễn ra rất phức tạp. Để nghiên cứu và phân tích sự khác biệt giữa các chuỗi prôtêin, chúng ta có thể sử dụng mô hình Markov để mô hình hoá một cách hiệu quả quá trình biến đổi giữa các axít amin.

Mục đích của bài toán ước lượng ma trận biến đổi axít amin là ước lượng các tham số của mô hình **Q**. Mô hình **Q** biểu diễn sự biến đổi axít amin theo mô hình Markov, **Q** là một thành phần rất quan trọng của nhiều bài toán liên quan đến chuỗi prôtêin như: sắp hàng đa chuỗi, tìm kiếm chuỗi tương đồng, xây dựng cây phân loài. Do đó có thể nói bài toán ước lượng ma trận biến đổi axít amin là một bài toán cơ bản và quan trọng của tin sinh học.

Hai nhóm phương pháp chính để ước lượng mô hình **Q** là nhóm phương pháp đếm và nhóm phương pháp cực đại khả năng. Phương pháp đếm thì nhanh nhưng chỉ áp dụng cho các chuỗi prôtêin có độ tương đồng cao còn phương pháp cực đại khả năng cho kết quả tốt hơn nhưng quá trình ước lượng mô hình còn tốn nhiều thời gian. Ngày nay, chúng ta thường sử dụng các phương pháp cực đại khả năng để ước

lượng mô hình biến đổi axít amin. Nhiều phương pháp đã được đề xuất và áp dụng trên các tập dữ liệu khác nhau tạo ra các mô hình khác nhau để phân tích các chuỗi prôtêin.

Chương 2. PHƯƠNG PHÁP ƯỚC LƯỢNG NHANH MÔ HÌNH BIẾN ĐỔI AXÍT AMIN BẰNG PHƯƠNG PHÁP CỰC ĐẠI KHẢ NĂNG

2.1. Giới thiệu

Phương pháp cực đại khả năng cho kết quả tốt nhưng lại yêu cầu khối lượng tính toán lớn nên rất khó áp dụng cho các bộ dữ liệu lớn. Một trong những bước tốn thời gian nhất của quá trình xây dựng mô hình **Q** là bước xây dựng cây phân loài từ các sắp hàng đa chuỗi. Chương này đề xuất một cách tiếp cận mới để vượt qua trở ngại này bằng cách chia tách các sắp hàng đa chuỗi lớn thành những sắp hàng nhỏ nhưng vẫn giữ được các thông tin để ước lượng các ma trận. Thực nghiệm với hai bộ dữ liệu chuẩn của vi rút cúm và Pfam cho thấy phương pháp cải tiến này có thể chạy nhanh hơn so với phương pháp tốt nhất hiện nay từ ba đến sáu lần trong khi các ma trận ước lượng gần như không khác biệt. Như vậy, phương pháp cải tiến này sẽ cho phép việc ước lượng các ma trận từ những tập dữ liệu rất lớn.

2.2. Ước lượng mô hình bằng phương pháp cực đại khả năng

Trong mục này chúng tôi trình bày và phân tích các bước để ước lượng mô hình bằng phương pháp cực đại khả năng theo lược đồ đã trình bày ở Hình 1.9.

2.2.1. Mô tả phương pháp

Phương pháp ước lượng mô hình **Q** từ tập dữ liệu **A** = $\{D^1, \ldots D^N\}$ gồm ba bước chính: *Xây dựng cây bằng ML, Ước lượng các tham số của mô hình* và *So sánh mô hình.* Cụ thể các bước như sau:

- *Xây dựng cây bằng ML*: Xây dựng cây phân loài từ các sắp hàng đa chuỗi sử dụng mô hình **Q** bằng phương pháp ML như đã trình bày trong mục 1.5 của chương 1.

- *Ước lượng các tham số của mô hình:* ước lượng mô hình **Q'** mới từ các sắp hàng đa chuỗi và cây tương ứng bằng thuật toán cực đại kỳ vọng (expectation maximization) [41].
- *So sánh mô hình:* So sánh **Q** và **Q'**. Nếu **Q'** ≈ **Q,** kết thúc và **Q'** là mô hình kết quả. Nếu không, thay **Q** bằng **Q'** và quay lại bước *Xây dựng cây.*

2.2.2. Phân tích phương pháp

Chúng tôi tiến hành thực nghiệm với 200 sắp hàng lớn nhất của bộ dữ liệu chuẩn Pfam [9] thì thấy bước *Xây dựng cây bằng ML* chiếm phần lớn thời gian (35 giờ) trong khi bước *Ước lượng các tham số của mô hình* chỉ chiếm phần nhỏ thời gian (4 giờ) [20]. Tiến hành thực nghiệm tương tự với 1373 sắp hàng đa chuỗi của vi rút cúm [20] cũng cho kết quả tương tự: bước *Xây dựng cây bằng ML* chiếm 273 giờ trong khi bước *Ước lượng các tham số của mô hình* chỉ chiếm 10 giờ [20].

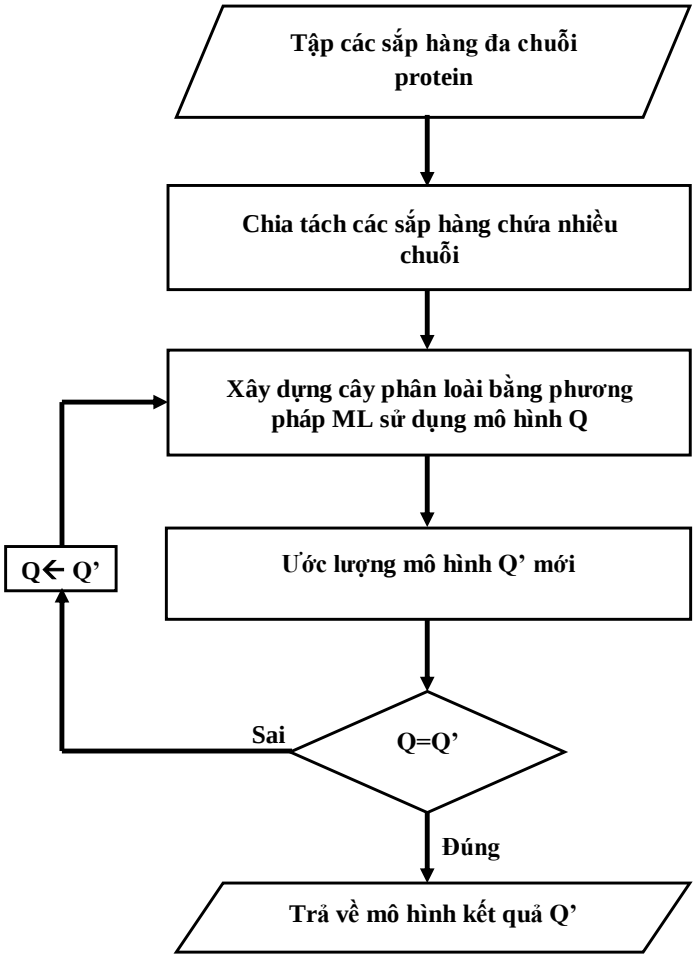
Như vậy, từ thực nghiệm chúng tôi nhận thấy với các sắp hàng có kích thước lớn thì bước xây dựng cây bằng ML thường chiếm nhiều thời gian nhất. Điều này có thể được giải thích là do bài toán *Xây dựng cây bằng ML* đã được chứng minh là bài toán NP-khó [15, 28]. Cụ thể hơn, với mỗi sắp hàng gồm m chuỗi ($m \geq 3$), số lượng cây phân loài dạng nhị phân không gốc là [25]: $\prod_{i=3}^{m}(2i-5)$.

Bảng 2.1: Số lượng cây nhị phân không gốc tương ứng với số chuỗi axít amin *m*.

m	**Số lượng cây nhị phân không gốc**
3	1
4	3
5	15
6	105
7	945
8	10395
9	135135
10	2027025

Chúng ta có thể thấy số lượng cây tương ứng của một sắp hàng tăng với cấp số nhân theo số lượng chuỗi của sắp hàng đó (xem Bảng 2.1). Nhiều phương pháp gần đúng khác nhau được đề xuất để xây dựng cây ML [34, 46, 57, 61], tuy nhiên các phương pháp vẫn còn chạy chậm với các bộ dữ liệu lớn.

Như vậy việc giảm kích thước của một sắp hàng có thể giúp giảm thời gian xây dựng cây tương ứng. Từ những phân tích trên chúng tôi đề xuất một phương pháp mới để ước lượng nhanh mô hình biến đổi axít amin như mô tả trong Hình 2.1 sau đây.



Hình 2.1: Lược đồ phương pháp ước lượng nhanh mô hình biến đổi axít amin.

2.3. Các phương pháp chia tách dữ liệu

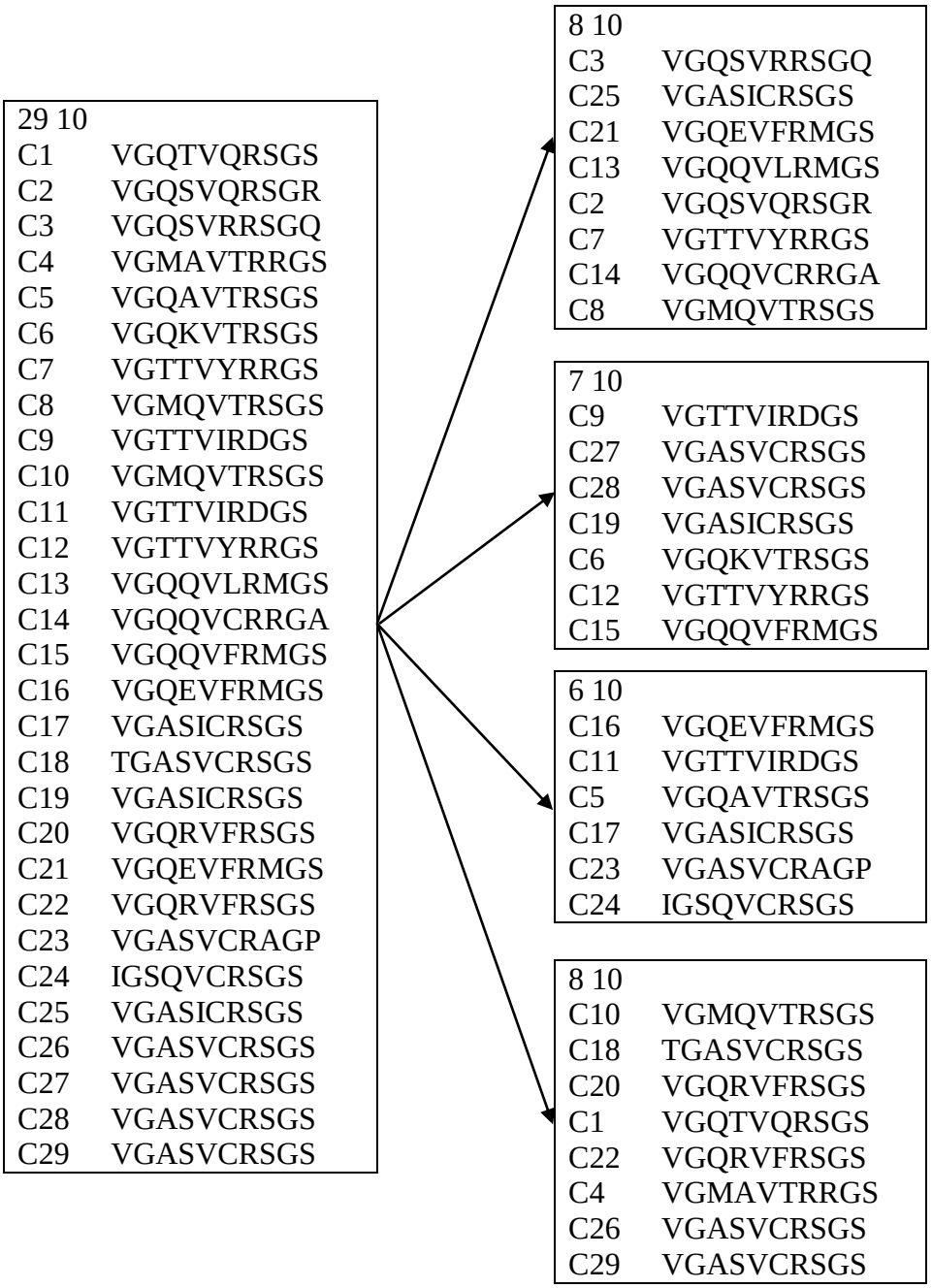
Dựa vào các phân tích của mục trước, luận án trình bày hai phương pháp để tăng tốc quá trình xây dựng cây phân loài. Ý tưởng ở đây là chia nhỏ các sắp hàng kích thước lớn thành nhiều sắp hàng kích thước nhỏ hơn.

2.3.1. Phương pháp chia tách ngẫu nhiên

Xét một sắp hàng D^a gồm m chuỗi và một số nguyên dương k ($k \geq 4$) làm ngưỡng chia tách. Các chuỗi của D^a được tách ngẫu nhiên thành các sắp hàng nhỏ có số lượng chuỗi nằm trong đoạn từ k đến $2k$. Các sắp hàng nhỏ này sẽ được sử dụng để ước lượng mô hình **Q**. Giả sử M là mô hình được ước lượng từ các sắp hàng không chia tách thì M_k^{R} sẽ là mô hình được ước lượng từ các sắp hàng được chia tách ngẫu nhiên với ngưỡng k. Ví dụ $\mathrm{LG}_8^{\mathrm{R}}$ là mô hình được ước lượng với cùng bộ dữ liệu như mô hình LG nhưng các sắp hàng có kích thước từ 8 đến 16 chuỗi. Các bước cụ thể của phương pháp chia tách sắp hàng ngẫu nhiên được trình bày ở Thuật toán 2.1. Minh họa của phương pháp này với $k = 4$ được trình bày ở Hình 2.2.

procedure Thuật toán chia tách ngẫu nhiên;
input: Một sắp hàng D^a với m chuỗi axít amin và số nguyên dương $k \geq 4$;
output: Các sắp hàng con với kích thước từ k đến $2k$;
begin
    while (số lượng chuỗi trong $D^a \geq k + 4$)
        - Sinh ngẫu nhiên một số tự nhiên s thoả mãn $k \leq s \leq 2k$;
        - Chọn ngẫu nhiên s chuỗi trong D^a để tạo thành một sắp hàng con;
        - Loại bỏ các chuỗi đã chọn ra khỏi D^a;
    endwhile;
end;

Thuật toán 2.1: Thuật toán chia tách sắp hàng ngẫu nhiên.



Hình 2.2: Minh họa thuật toán chia tách sắp hàng ngẫu nhiên với *k*=4.

2.3.2. Phương pháp chia tách dựa theo cấu trúc cây

Phương pháp này dựa theo tư tưởng của thuật toán BIONJ [30] với độ phức tạp là O(m^3) với m là số chuỗi. Ý tưởng của thuật toán là: các chuỗi lần lượt được gộp lại nếu như số lượng chuỗi trong nhóm mới nằm trong đoạn từ k đến 2k. Chi

tiết phương pháp chia tách dựa theo cấu trúc cây được trình bày trong Thuật toán 2.2 sau đây:

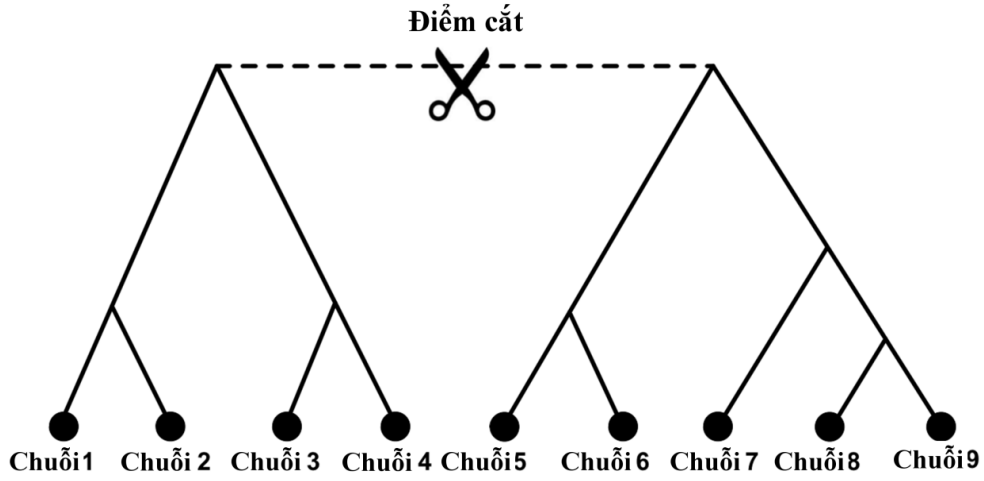
procedure Thuật toán chia tách dựa theo cấu trúc cây;
input: Một sắp hàng D^a với m chuỗi axít amin và số nguyên dương $k \geq 4$;
output: Các sắp hàng con với kích thước từ k đến $2k$;
begin
Mỗi chuỗi của D^a được coi như một nhóm. Tính khoảng cách giữa hai nhóm một dựa vào ma trận khoảng cách và thuật toán BIONJ [30];
repeat
Tìm hai nhóm có khoảng cách nhỏ nhất, giả sử là G_1 và G_2. Gọi m_1 và m_2 là số lượng chuỗi của G_1 và G_2 tương ứng;
if $m_1 + m_2 \leq 2k$ **then**
Kết hợp G_1 và G_2 thành một nhóm mới;
Tính toán lại khoảng cách giữa nhóm mới này và các nhóm khác theo thuật toán BIONJ [30];
else / / $m_1 > k$ hoặc $m_2 > k$
if $m_1 > k$ **then**
Xem G_1 là một sắp hàng con;
else / / $m_2 > k$
Xem G_2 là một sắp hàng con;
endif
endif
until (chỉ còn một nhóm G_0);
Giả sử m_0 là số lượng chuỗi của G_0.
if $m_0 \geq 3$ **then**
Xem G_0 là một sắp hàng con;
else
Kết hợp G_0 vào một sắp hàng con trước đó
end;

Thuật toán 2.2: Thuật toán chia tách sắp hàng dựa theo cấu trúc cây.

Lưu ý, bước cuối cùng khi chỉ còn lại một nhóm, nếu kích thước nhóm này lớn hơn 3 thì nó sẽ được coi là một sắp hàng, nếu không thì nó sẽ được kết hợp vào một sắp hàng trước đó. Bước này giúp đảm bảo không có sắp hàng nào có ít hơn 4 chuỗi.

Giả sử M là mô hình được ước lượng từ các sắp hàng không chia tách thì M_k sẽ là mô hình được ước lượng từ các sắp hàng được chia tách dựa theo cấu trúc cây với ngưỡng k. Hình 2.3 minh họa một cách chia tách dựa theo cấu trúc cây với k = 4

trong đó một sắp hàng gồm 9 chuỗi được chia tách thành hai sắp hàng nhỏ hơn có 4 và 5 chuỗi.



Hình 2.3: Minh họa thuật toán chia tách sắp hàng dựa trên cấu trúc cây với *k*=4.

2.3.3. Nhận xét về các phương pháp chia tách sắp hàng

Cả hai phương pháp chia tách đều giúp giảm thời gian xây dựng cây do số lượng cấu trúc cây khác nhau giảm rõ rệt. Cụ thể, với *k*=12 số lượng cấu trúc cây khác nhau còn khoảng 650 triệu, với *k*=8 số lượng cấu trúc cây khác nhau giảm còn 10395, với *k*=4 thì số lượng cấu trúc cây khác nhau chỉ còn là 3.

Phương pháp chia tách ngẫu nhiên có thể tạo ra các sắp hàng nhỏ chứa các chuỗi prôtêin có quan hệ xa. Điều này có thể dẫn tới các cây phân loài tương ứng với các sắp hàng nhỏ này có độ chính xác không cao [11, 14, 28] từ đó làm giảm độ chính xác cuả mô hình **Q**. Phương pháp chia tách dựa theo cấu trúc cây sẽ tạo ra các sắp hàng nhỏ chứa các nhánh của cây lớn (cây từ sắp hàng ban đầu), do đó các sắp hàng nhỏ sẽ ít có khả năng chứa các chuỗi prôtêin có quan hệ xa. Chính điều này sẽ giúp nâng cao độ chính xác của mô hình **Q**.

2.4. Kết quả thực nghiệm

Để đánh giá hai phương pháp đề xuất, chúng tôi đã thử nghiệm cả hai phương pháp trên hai bộ dữ liệu Pfam [9] và vi rút cúm với cách đánh giá chuẩn như trong các nghiên cứu trước đây [18, 49, 63]. Cụ thể, chúng tôi so sánh thời gian ước lượng các mô hình và kết quả xây dựng cây phân loài của các mô hình đó theo tiêu chuẩn ML. Hai phương pháp chia tách sắp hàng được thử nghiệm với các ngưỡng k bằng 4, 8, 16 và 32.

2.4.1. Dữ liệu kiểm tra

2.4.1.1. Bộ dữ liệu vi rút cúm

Đây là bộ dữ liệu các chuỗi prôtêin vi rút cúm đã được sử dụng để ước lượng mô hình FLU [18]. Các chuỗi prôtêin vi rút cúm được tải về từ cơ sở dữ liệu của NCBI (phiên bản năm 2011). Sau đó dữ liệu được tiến hành loại bỏ các chuỗi trùng lặp và chia thành các nhóm lớn theo kiểu vi rút (cúm A, cúm B và cúm C). Mỗi nhóm lớn được tiếp tục chia thành các nhóm nhỏ theo từng kiểu prôtêin. Tiếp theo các nhóm nhỏ được chia ngẫu nhiên thành các tập từ 5 đến 99 chuỗi. Các tập sau đó được xây dựng thành các sắp hàng đa chuỗi bằng phần mềm MUSCLE [23]. Bộ dữ liệu FLU bao gồm 1373 sắp hàng với tổng số 71087 chuỗi.

2.4.1.2. Bộ dữ liệu Pfam

Pfam là bộ dữ liệu các sắp hàng đa chuỗi prôtêin đã được sử dụng để ước lượng mô hình LG [49]. Các sắp hàng này được lấy từ hai cơ sở dữ liệu Swiss-Prot và TrEMBL [12]. Pfam bao gồm 3912 sắp hàng với 49637 chuỗi, các sắp hàng đều có ít nhất 5 chuỗi và chiều dài ít nhất 50.

2.4.2. Kết quả với bộ dữ liệu vi rút cúm

Ở thí nghiệm này, bộ dữ liệu vi rút cúm được chia tách ngẫu nhiên thành hai tập con, một tập để ước lượng mô hình gồm 687 sắp hàng và một tập để kiểm tra gồm 686 sắp hàng.

Bảng 2.2: Thời gian ước lượng mô hình của phương pháp chia tách ngẫu nhiên với bộ dữ liệu vi rút cúm. $\mathbf{FLU}_k^{\mathbf{R}}$ là mô hình ước lượng từ các sắp hàng được chia nhỏ bằng phương pháp chia tách ngẫu nhiên với ngưỡng *k*.

Mô hình	Bước xây dựng cây bằng ML	Bước ước lượng tham số mô hình	Tổng thời gian
FLU_4^R	7,8	19,5	27,3
FLU_8^R	11,1	18,8	29,9
FLU_{16}^R	22,9	17,5	40,4
FLU_{32}^R	65,9	15,1	81
FLU	273,5	10,3	283,8

Bảng 2.3: Thời gian ước lượng mô hình của phương pháp chia tách dựa theo cấu trúc cây với bộ dữ liệu vi rút cúm. FLU_k là mô hình ước lượng từ các sắp hàng được chia nhỏ bằng phương pháp chia tách dựa theo cấu trúc cây với ngưỡng *k*.

Mô hình	Bước xây dựng cây bằng ML	Bước ước lượng tham số mô hình	Tổng thời gian
FLU_4	14,7	6,7	21,4
FLU_8	40,5	8,3	48,7
FLU_{16}	86,6	9,2	95,9
FLU_{32}	170,2	10,2	180,4
FLU	273,5	10,3	283,8

Bảng 2.2 và Bảng 2.3 lần lượt cho thấy thời gian cần thiết để ước lượng các mô hình từ bộ dữ liệu FLU sử dụng hai phương pháp chia tách với các ngưỡng *k* khác nhau. Thời gian ước lượng của FLU là khoảng 284 giờ (xấp xỉ 12 ngày), trong khi FLU_8^R chỉ cần khoảng 30 giờ. Như vậy là đã nhanh hơn xấp xỉ 10 lần. Đối với

phương pháp tách dựa trên cây, thời gian ước lượng FLU_8 là gần 49 giờ, tương đương nhanh hơn khoảng sáu lần.

Bảng 2.4: So sánh kết quả các mô hình của phương pháp chia tách ngẫu nhiên trên bộ dữ liệu vi rút cúm. M_1: mô hình thứ nhất; M_2: mô hình thứ hai; M_1-M_2: Khác biệt về giá trị trung bình log-likelihood trên một vị trí giữa hai mô hình M_1 và M_2; $M_1>M_2$: M_1 tốt hơn M_2; $M_1<M_2$: M_2 tốt hơn M_1; $T_1 \neq T_2$: cây ước lượng bởi M_1 và M_2 có cấu trúc khác nhau.

M_1	M_2	M_1-M_2	$M_1>M_2$	$M_1<M_2$	$T_1 \neq T_2$
FLU	FLU_4^R	0,029	545	141	592
FLU	FLU_8^R	0,025	537	149	590
FLU	FLU_{16}^R	0,018	513	173	588
FLU	FLU_{32}^R	0,006	421	265	565

Bảng 2.5: So sánh kết quả các mô hình của phương pháp chia tách dựa theo cấu trúc cây trên bộ dữ liệu vi rút cúm. M_1: mô hình thứ nhất; M_2: mô hình thứ hai; M_1-M_2: Khác biệt về giá trị trung bình log-likelihood trên một vị trí giữa hai mô hình M_1 và $M2$; $M_1>M_2$: M_1 tốt hơn M_2; $M_1<M_2$: M_2 tốt hơn M_1; $T_1 \neq T_2$: cây ước lượng bởi M_1 và M_2 có cấu trúc khác nhau.

M_1	M_2	M_1-M_2	$M_1>M_2$	$M_1<M_2$	$T_1 \neq T_2$
FLU	FLU_4	0,011	290	396	588
FLU	FLU_8	0,009	293	393	593
FLU	FLU_{16}	0,003	294	392	568
FLU	FLU_{32}	0,000	303	383	446

Bảng 2.4 và Bảng 2.5 trình bày các kết quả so sánh mô hình FLU với các mô hình ước lượng nhanh trong việc xây dựng lại cây ML cho dữ liệu vi rút cúm. Với phương pháp chia tách ngẫu nhiên FLU_{32}^R là xấp xỉ với FLU (khác biệt log-likelihood trung bình nhỏ hơn 0,01), riêng các mô hình FLU_4^R, FLU_8^R, FLU_{16}^R thì không tốt bằng FLU. Với phương pháp chia theo cấu trúc cây, các mô hình FLU_8, FLU_{16} và FLU_{32} tốt tương đương FLU. Như vậy có thể thấy là chia theo cây tốt hơn chia ngẫu nhiên với cùng một ngưỡng *k*.

2.4.3. Kết quả với bộ dữ liệu Pfam

Ở thí nghiệm này, chúng tôi sử dụng 200 sắp hàng lớn nhất từ bộ dữ liệu Pfam để ước lượng mô hình và chọn ngẫu nhiên 500 sắp hàng từ phần còn lại để làm dữ liệu kiểm tra. Các mô hình kết quả sau đó được dùng để xây dựng cây bằng phương pháp ML cho 500 sắp hàng kiểm tra. Chúng tôi so sánh các phương pháp trên cả hai tiêu chuẩn là thời gian ước lượng và hiệu quả của mô hình.

Bảng 2.6: Thời gian ước lượng mô hình của phương pháp chia tách ngẫu nhiên với bộ dữ liệu Pfam. $\mathtt{LG}_k^{\mathtt{R}}$ là mô hình ước lượng từ các sắp hàng được chia nhỏ bằng phương pháp chia tách ngẫu nhiên với ngưỡng *k*.

Mô hình	Bước xây dựng cây bằng ML	Bước ước lượng tham số mô hình	Tổng thời gian
LG_4^R	1,5	8,7	10,2
LG_8^R	2,2	8,6	10,9
LG_{16}^R	4,8	7,7	12,4
LG_{32}^R	18,7	5,2	23,9
LG	35,1	4,4	39,5

Bảng 2.7: Thời gian ước lượng mô hình của phương pháp chia tách tách dựa theo cấu trúc cây với bộ dữ liệu Pfam. LG_k là mô hình ước lượng từ các sắp hàng được chia nhỏ bằng phương pháp chia tách dựa theo cấu trúc cây với ngưỡng *k*.

Mô hình	Bước xây dựng cây bằng ML	Bước ước lượng tham số mô hình	Tổng thời gian
LG_4	3,4	5,2	8,7
LG_8	7,7	4,9	12,6
LG_{16}	13,9	4,9	18,8
LG_{32}	21,9	4,8	26,7
LG	35,1	4,4	39,5

Bảng 2.6 và Bảng 2.7 cho thấy tổng thời gian ước lượng tăng khi *k* tăng. Với *k*=8, thời gian ước lượng của LG_8^R và LG_8 đều ít hơn xấp xỉ 3 lần so với thời gian

ước lượng LG. Chúng ta cũng có thể thấy rằng thời gian chạy của *Bước xây dựng cây* tăng khi giá trị *k* tăng.

Bảng 2.8: So sánh kết quả của phương pháp chia tách ngẫu nhiên với bộ dữ liệu Pfam. M_1: mô hình thứ nhất; M_2: mô hình thứ hai; M_1-M_2: Khác biệt về giá trị trung bình log-likelihood trên một vị trí giữa hai mô hình M_1 và $M2$; $M_1>M_2$: M_1 tốt hơn M_2; $M_1<M_2$: M_2 tốt hơn M_1; $T_1 \neq T_2$: cây ước lượng bởi M_1 và M_2 có cấu trúc khác nhau.

M_1	M_2	M_1-M_2	$M_1>M_2$	$M_1<M_2$	$T_1 \neq T_2$
LG	LG_4^R	0,009	293	207	170
LG	LG_8^R	0,006	279	221	164
LG	LG_{16}^R	-0,001	256	244	159
LG	LG_{32}^R	-0,004	192	308	100

Bảng 2.9: So sánh kết quả của phương pháp chia dựa theo cấu trúc cây với bộ dữ liệu Pfam. M_1: mô hình thứ nhất; M_2: mô hình thứ hai; M_1-M_2: Khác biệt về giá trị trung bình log-likelihood trên một vị trí giữa hai mô hình M_1 và M_2; $M_1>M_2$: M_1 tốt hơn M_2; $M_1<M_2$: M_2 tốt hơn M_1; $T_1 \neq T_2$: cây ước lượng bởi M_1 và M_2 có cấu trúc khác nhau.

M_1	M_2	M_1-M_2	$M_1>M_2$	$M_1<M_2$	$T_1 \neq T_2$
LG	LG_4	0,008	288	212	158
LG	LG_8	-0,004	179	321	121
LG	LG_{16}	-0,003	160	340	103
LG	LG_{32}	-0,001	193	307	79

Bảng 2.8 và Bảng 2.9 thể hiện các kết quả so sánh mô hình LG với các mô hình ước lượng nhanh trong việc xây dựng lại cây ML cho 500 sắp hàng kiểm tra. Bảng 2.8 cho thấy các mô hình LG_{16}^R và LG_{32}^R của phương pháp chia tách ngẫu nhiên tốt hơn so với LG. Bảng 2.9 cho thấy các mô hình LG_8, LG_{16} và LG_{32} của phương pháp chia tách dựa trên cây cũng tốt hơn so với LG. Thời gian ước lượng mô hình với hai phương pháp chia ngẫu nhiên và chia theo cây là tương đương nhau. Về mặt hiệu quả của mô hình thì với ngưỡng $k \geq 8$, phương pháp chia theo cây cho kết quả tốt hơn các phương pháp khác.

2.5. Kết luận chương

Ước lượng mô hình biến đổi axít amin là một bài toán rất quan trọng trong nghiên cứu về các chuỗi prôtêin. Nhiều phương pháp ước lượng mô hình khác nhau đã được đề xuất. Phương pháp ML cho kết quả tốt hơn so với các phương pháp khác. Tuy nhiên, phương pháp này rất chậm và khó áp dụng được với các bộ dữ liệu lớn.

Chúng tôi giới thiệu hai phương pháp giúp giảm thời gian ước lượng mô hình. Ý tưởng cốt lõi là phân chia các sắp hàng lớn thành các sắp hàng nhỏ hơn để giảm thời gian xây dựng cây.

Các thực nghiệm với hai bộ dữ liệu Pfam và vi rút cúm cho thấy phương pháp chia tách dựa trên cấu trúc cây cho kết quả tốt. Mô hình LG_8 có hiệu quả tương đương LG trong khi thời gian ước lượng nhanh hơn ba lần. Mô hình FLU_8 có hiệu quả tương đương FLU trong khi thời gian ước lượng nhanh hơn sáu lần. Do đó, phương pháp chia tách dựa trên cây với ngưỡng k=8 được chúng tôi khuyên dùng. Các kết quả nghiên cứu của chương này đã được công bố tại hội nghị quốc tế KSE năm 2011 (công trình khoa học số 3).

Chương 3. XÂY DỰNG MÔ HÌNH BIẾN ĐỔI ĐA MA TRẬN

Phần lớn các mô hình biến đổi axít amin sử dụng một ma trận để mô hình hoá sự biến đổi giữa các axít amin. Tuy nhiên quá trình biến đổi ở các vị trí trên chuỗi axít amin là không giống nhau và phụ thuộc vào nhiều yếu tố. Trong hầu hết các trường hợp, một ma trận là không đủ để mô hình hóa sự phức tạp của quá trình biến đổi giữa các axít amin. Ở chương này, chúng tôi sẽ nghiên cứu việc sử dụng mô hình với nhiều ma trận cho các vị trí khác nhau trên chuỗi axít amin.

3.1. Tính không đồng nhất của tốc độ biến đổi theo vị trí

Nhiều nghiên cứu đã chỉ ra rằng tốc độ biến đổi có tính không đồng nhất, tức là tốc độ biến đổi giữa các vị trí khác nhau trong cùng một chuỗi có sự khác biệt đáng kể [44]. Hiện tượng này thường được giải thích bởi sự hiện diện của các nhu cầu tiến hóa khác nhau ở các vị trí khác nhau [44]. Để không bỏ qua hiện tượng quan trọng này, chúng ta cần sử dụng một mô hình phân phối để biểu diễn tốc độ biến đổi axít amin tại các vị trí khác nhau trong chuỗi prôtêin [44].

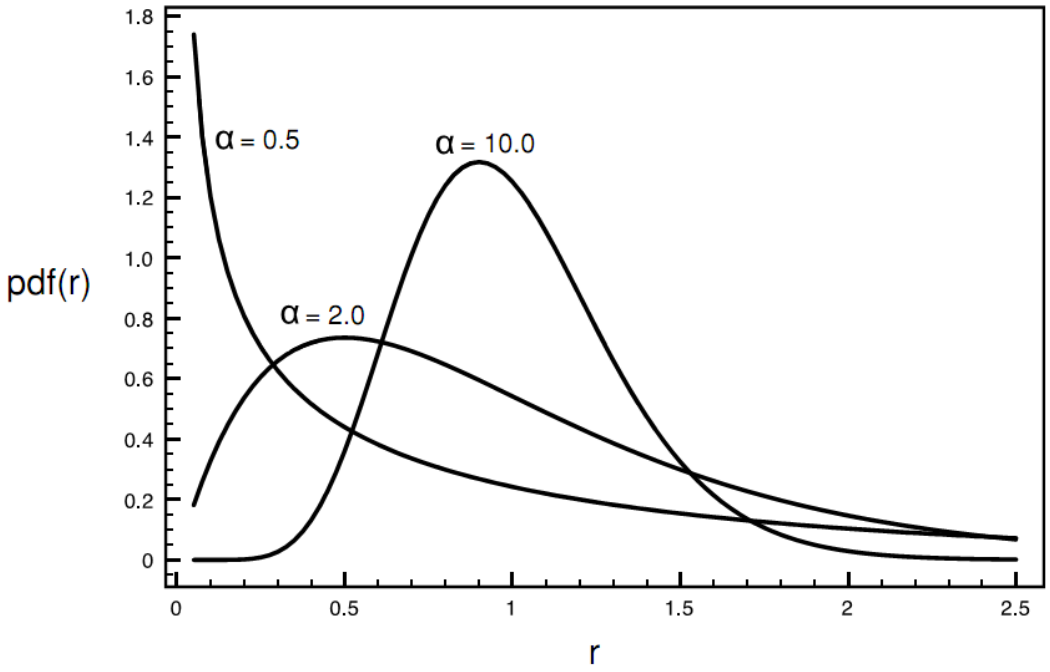
Tính không đồng nhất của tốc độ biến đổi axít amin tại các vị trí khác nhau có thể được mô hình hoá bằng một phân phối gamma (Γ) với kỳ vọng là 1,0 và phương sai là 1/α (α>0) theo công thức sau:

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

trong đó $\Gamma(\alpha) = \int_0^{\infty} e^{-t} t^{\alpha-1} \mathrm{d}t$.

Mức độ không đồng nhất tốc độ biến đổi ở các vị trí được điều chỉnh bằng cách thay đổi tham số định hình α của phân phối như thể hiện trong Hình 3.1. Với α<1 phân phối có dạng chữ L trong khi nếu α>1 thì phân phối sẽ có dạng hình chuông. Giá trị α nhỏ sẽ thể hiện tính không đồng nhất mạnh của tốc độ biến đổi. Ví dụ, với α = 0,5 (được gọi là nhỏ) thì tốc độ biến đổi rất nhanh ở chỉ một vài vị trí,

còn phần lớn các vị trí còn lại là biến đổi rất chậm. Trong khi đó, nếu α = 10 (được gọi là lớn) thì tính không đồng nhất sẽ rất yếu, có nghĩa là tất cả mọi vị trí đều có tốc độ biến đổi gần bằng nhau.



Hình 3.1: Các dạng phân phối gamma với các tham số α khác nhau [44].

Trong thực tế, thay vì sử dụng phân phối liên tục chúng ta thường sử dụng phân phối rời rạc với một số hữu hạn *c* các phân lớp, mỗi phân lớp có xác suất tương ứng là q_1, q_2, ..., q_c. Cụ thể hơn, chúng ta có thể sử dụng từ 4 đến 8 phân lớp rời rạc là đủ để xấp xỉ hàm liên tục của phân phối gamma [65].

3.2. Mô hình biến đổi đa ma trận

Trước khi trình bày về mô hình đa ma trận chúng tôi xin nhắc lại một số khái niệm và ký hiệu của mô hình chuẩn đã trình bày ở chương 1. Với mô hình chuẩn ta cần ước lượng 208 tham số của mô hình **Q**. Ký hiệu *D* là một sắp hàng, *T* là cây cây phân loài tương ứng của *D* được xây dựng bằng ML với mô hình **Q**. Khi đó likelihood của **Q** và *T* đối với *D* được tính theo công thức:

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

trong đó $D = \{D_1, \ldots D_l\}$ là một sắp hàng đa chuỗi có chiều dài l và D_i ($1 \le i \le l$) là vị trí thứ i của sắp hàng. Yang đã giới thiệu một mô hình hỗn hợp dựa trên một ma trận duy nhất nhưng tốc độ của các vị trí biến thiên theo một phân phối gamma rời rạc với c phân loại tốc độ có trọng số bằng nhau [64]. Likelihood được tính bằng công thức:

$$L(\mathbf{Q}, T, \alpha \mid D) = \prod_{i=1}^{l} \left(\frac{1}{c} \sum_{k=1}^{c} L(\Gamma(\alpha, k)\mathbf{Q}, T \mid D_i) \right) \tag{3.2}$$

với $\Gamma(\alpha, k)$ là tốc độ thứ k của một phân bố gamma rời rạc với tham số α. Các trọng số của các tốc độ đều bằng $1/c$. Cả T và α được ước lượng bằng phương pháp ML từ tập dữ liệu đầu vào.

Mô hình đa ma trận đã được đề xuất trong một số nghiên cứu [32, 42]. Với mô hình đa ma trận này, likelihood được tính như sau:

$$L(\mathbf{Q} = \{Q_1, .., Q_M\}, T, W = \{w_1, ..., w_M\} \mid D) = \prod_{i=1}^{l} \left(\sum_{m=1}^{M} w_m L(Q_m, T \mid D_i) \right) \tag{3.3}$$

trong đó M là số lượng ma trận và w_m là trọng số của ma trận Q_m với điều kiện $\sum_{m=1}^{M} w_m = 1$.

Các nghiên cứu gần đây [50, 62] đã kết hợp mô hình của Yang (công thức 3.2) với công thức 3.3 ở trên để tạo thành mô hình đa ma trận:

$$\begin{aligned} &L(\mathbf{Q} = \{Q_1, .., Q_M\}, T, W = \{w_1, ..., w_M\}, \alpha \mid D) \\ &= \prod_{i=1}^{l} \left(\sum_{m=1}^{M} \frac{w_m}{c} \sum_{k=1}^{c} L(\Gamma(\alpha, k) Q_m, T \mid D_i) \right) \end{aligned} \tag{3.4}$$

Trong đó điều kiện $\sum_{m=1}^{M} w_m = 1$ vẫn được giữ nguyên.

Như vậy, công thức 3.4 thể hiện hai cấp độ hỗn hợp, một cho giá trị các loại tốc độ phân phối gamma và một cho các ma trận thay thế. Các tác giả Le và Gascuel [50] đã sử dụng công thức 3.4 để ước lượng các mô hình EX2 (gồm hai ma trận) và UL3 (gồm ba ma trận).

Mặc dù các mô hình EX2, UL3 là tốt nhưng lại yêu cầu khối lượng tính toán và bộ nhớ lớn. Điều này chủ yếu là do số lượng lớn các phân loại vị trí, ví dụ như UL3 có tới 12 phân loại vị trí và 4 phân loại gamma. Để đơn giản hóa công thức 3.4, chúng tôi chỉ sử dụng bốn phân loại tốc độ và bốn ma trận tương ứng (c = 4, M = 4). Các trọng số của cả 4 phân loại đều bằng ¼. Mô hình với bốn ma trận này được đặt tên là LG4M. Giả sử **Q** = (Q_1, Q_2, Q_3, Q_4) là tập bốn ma trận, khi đó likelihood của **Q**, cây phân loài T và tham số α được tính bằng công thức:

$$L(\mathbf{Q},T,\alpha \mid D) = \prod_{i=1}^{l}\left(\frac{1}{4}\sum_{k=1}^{4} L(\Gamma(\alpha,k)Q_k,T \mid D_i)\right) \qquad (3.5)$$

Công thức 3.5 này là một sự kết hợp giữa công thức 3.2 của Yang và công thức 3.4 của các mô hình hỗn hợp hai cấp. Thay vì dùng chung một ma trận như trong mô hình của Yang, mỗi tốc độ có ma trận riêng và mỗi ma trận được áp dụng chỉ cho một loại tốc độ thay vì cho tất cả các tốc độ như trong mô hình hỗn hợp hai cấp. Như vậy, mô hình theo công thức 3.5 là tổng quát hơn so với mô hình của Yang, nhưng vẫn giữ các tham số tự do được ước tính từ các dữ liệu (tức là α và T) như trong mô hình của Yang.

Mô hình LG4M trong công thức 3.5 sử dụng một phân phối gamma rời rạc để phân lớp các tốc độ biến đổi theo vị trí. Chúng tôi loại bỏ phân phối gamma để có mô hình LG4X tổng quát hơn. Likelihood khi đó được tính như sau:

$$L(\mathbf{Q}, T, P=\{\rho_1,\rho_2,\rho_3,\rho_4\}, W=\{w_1,w_2,w_3,w_4\} \mid D)$$
$$= \prod_{i=1}^{l}\left(\sum_{k=1}^{4} w_k L(\rho_k Q_k, T \mid D_i)\right) \quad (3.6)$$

trong đó w_k , ρ_k là các trọng số, tốc độ của Q_k thoả mãn $\sum_{k=1}^{4} w_k = 1$ và $\sum_{k=1}^{4} w_k \rho_k = 1$. Như vậy LG4X chỉ còn có 3 trọng số w_k và 3 tốc độ ρ_k là các tham số cần ước lượng.

3.3. Thuật toán ước lượng mô hình đa ma trận

Giả sử, chúng ta có một tập N sắp hàng prôtêin ký hiệu là $\mathbf{A} = \{D^1,\ldots, D^N\}$, với D^a là sắp hàng thứ a. Mục tiêu của thuật toán là ước lượng mô hình gồm 4 ma trận $\mathbf{Q}^* = (Q_1^*, Q_2^*, Q_3^*, Q_4^*)$ sao cho likelihood sau đạt cực đại:

$$\mathbf{Q}^* = \underset{\mathbf{Q}=(Q_1,Q_2,Q_3,Q_4),T,P,W}{\arg\max} \left\{\prod_{a=1}^{N} L\left(\mathbf{Q}, T^a, \rho^a, w^a \mid D^a\right)\right\}, \quad (3.7)$$

trong đó $T = (T^1, \ldots, T^N)$, $P = (\rho^1, \ldots, \rho^N)$ và $W = (w^1, \ldots, w^N)$ lần lượt là các cây, tốc độ và trọng số tương ứng của N sắp hàng; $L\left(T^a, \mathbf{Q}, \rho^a, w^a; D^a\right)$ là likelihood của $\mathbf{Q}$, cây T^a, tốc độ $\rho^a = (\rho_1^a, \ldots, \rho_4^a)$ và trọng số $w^a = (w_1^a, \ldots, w_4^a)$ đối với D^a. Để ước lượng $\mathbf{Q}^* = (Q_1^*, Q_2^*, Q_3^*, Q_4^*)$ thì cần ước lượng T^*, P^* và W^*. LG4M được ước lượng theo công thức 3.5 và LG4X được ước lượng theo công thức 3.6. Với mỗi sắp hàng D^a, các tốc độ ρ^a và trọng số w^a của LG4M tuân theo phân phối gamma rời rạc với 4 trọng số bằng nhau, trong khi với LG4X, các tham số được ước lượng theo điều kiện $\sum_1^4 w_k^a = 1$ và $\sum_1^4 w_k^a \rho_k^a = 1$.

Dựa theo các nghiên cứu [49, 50, 63], chúng tôi thấy có thể ước lượng $\mathbf{Q}^* = (Q_1^*, Q_2^*, Q_3^*, Q_4^*)$ từ công thức 3.7 qua 2 bước:

1. Cho các ma trận Q_i giá trị khởi tạo ban đầu, ước lượng T^*, P^* và W^* bằng ML dựa theo công thức 3.5 cho LG4M và 3.6 cho LG4X.
2. Từ các giá trị T^*, P^* và W^* đã có, ước lượng $\mathbf{Q}^* = (Q_1^*, Q_2^*, Q_3^*, Q_4^*)$ theo công thức 3.7.

Hai bước này được lặp đi lặp lại cho đến khi T^*, P^* và W^* không thay đổi.

Do tốc độ, trọng số và các cây là độc lập với nhau nên ta có thể tối ưu T^*, P^* và W^* cho từng D^a một cách độc lập theo công thức:

$$\forall D^a : (T^a, \rho^a, w^a) = \arg\max_{T,P,W} \left\{ L\left(T, \mathbf{Q}, P, W \mid D^a\right) \right\}. \quad (3.8)$$

Sau khi có T^*, P^*, W^* chúng ta sẽ ước lượng **Q*** để likelihood của dữ liệu đạt cực đại.

$$\mathbf{Q}^* = \left(Q_1^*, Q_2^*, Q_3^*, Q_4^*\right) = \arg\max_{\mathbf{Q}=(Q_1,Q_2,Q_3,Q_4)} \left\{ \prod_{a=1}^{N} L\left(T^a, \mathbf{Q}, \rho^a, w^a \mid D^a\right) \right\}. \quad (3.9)$$

Tuy nhiên trong thực tế chúng ta không thể tối ưu trực tiếp (Q_1^*, Q_2^*, Q_3^*, Q_4^*) từ công thức 3.9 do số lượng tham số tự do của **Q** quá lớn (4*208 tham số). Do đó chúng tôi áp dụng phương pháp xấp xỉ của Le và các cộng sự [49, 50] cho công thức 3.9 bằng cách chỉ sử dụng các phân lớp tốc độ có xác suất lớn nhất thay vì tính tổng trên tất cả các phân lớp tốc độ. Khi đó mô hình $\mathbf{Q}^* = (Q_1^*, Q_2^*, Q_3^*, Q_4^*)$ được ước lượng theo công thức:

$$\mathbf{Q}^* = \left(Q_1^*, Q_2^*, Q_3^*, Q_4^*\right) = \arg\max_{\mathbf{Q}=(Q_1,Q_2,Q_3,Q_4)} \left\{ \prod_{a=1}^{N} \prod_{i=1}^{l} L\left(T^a, \rho_{c_i}^a Q_{c_i} \mid D_i^a\right) \right\} \quad (3.10)$$

trong đó D_i^a là vị trí thứ i của sắp hàng D^a, c_i là phân lớp tốc độ có xác suất lớn nhất (được tính trong quá trình ước lượng cây) cho D_i^a, ρ_{ci} là tốc độ của c_i tương ứng với Q_{ci}. Công thức 3.10 có thể viết lại dưới dạng:

$$\forall k = 1...4,\ Q_k^* = \arg\max_{Q_k} \left\{ \prod_{a=1}^{N} \prod_{i=1:c_i=k}^{l} L(T^a, \rho_k^a Q_k \mid D_i^a) \right\} \quad (3.11)$$

Như vậy, các ma trận Q_k được ước lượng độc lập.

Thuật toán 3.1 tóm tắt toàn bộ các bước ước lượng mô hình LG4X và LG4M.

procedure Thuật toán ước lượng mô hình đa ma trận;
input: Tập N sắp hàng **A** = { D^1, …, D^N }, mô hình khởi tạo ban đầu $\mathbf{Q}_{\mathbf{start}}$;
output: Mô hình **Q** = {Q_1, Q_2, Q_3, Q_4};
begin
  $Q_1 = Q_2 = Q_3 = Q_4 = \mathbf{Q}_{\mathbf{start}}$;
  repeat
    foreach sắp hàng D^a trong **A**
      - $T^a \leftarrow$ Cây phân loài của D^a xây dựng bằng ML với **Q**;
      - Ước lượng các tốc độ $\rho^a = \rho_1^a, \ldots, \rho_4^a$ và các trọng số $w^a = w_1^a, \ldots, w_4^a$ dựa theo công thức 3.8;
      - Phân lớp cho vị trí $D^a{}_i$ của D^a vào tập $L_{c_i}^a$ sao cho thỏa mãn $c_i = \arg\max_{k=1...4} w_k L(T^a, \rho_k^a Q_k \mid D_i^a)$;
      - Chia các sắp hàng D^a và cây T^a thành 4 sắp hàng và 4 cây con theo phân lớp ở trên, các cây con được nhân với các tốc độ $\rho_1^a, \ldots, \rho_4^a$ tương ứng: $(L_1^a, T^a * \rho_1^a)$, $(L_2^a, T^a * \rho_2^a)$, $(L_3^a, T^a * \rho_3^a)$, $(L_4^a, T^a * \rho_4^a)$;
    end foreach;
    for (k = 1...4)
      Ước lượng mô hình Q^*_k từ các sắp hàng và cây con thuộc phân lớp k ở trên $(L_k^a, T^a * \rho_k^a)$ bằng thuật toán cực đại kỳ vọng [41] với Q_k là mô hình khởi tạo ban đầu của thuật toán cực đại kỳ vọng;
    endfor;
  until ($Q_k \approx Q^*_k$ với mọi k)**;**
  Q ← **Q'**;
end;

Thuật toán 3.1: Thuật toán ước lượng mô hình LG4M và LG4X.

3.4. Kết quả thực nghiệm

3.4.1. Dữ liệu kiểm tra

Để ước lượng LG4M và LG4X, chúng tôi sử dụng bộ dữ liệu HSSP [55]. HSSP gồm 1771 sắp hàng, trung bình mỗi sắp hàng có 56 chuỗi và chiều dài 254. 1471 sắp hang được chọn ngẫu nhiên để ước lượng LG4M và LG4X, 300 sắp hàng còn lại dùng cho việc kiểm tra.

Để đánh giá các mô hình với dữ liệu thực tế, chúng tôi sử dụng bộ dữ liệu TreeBase [53]. TreeBase chứa các sắp hàng đã được sử dụng cho các bài toán phát sinh loài trong các bài báo đã công bố trên các tạp chí uy tín. TreeBase có tất cả 84 sắp hàng với kích thước khác nhau, từ nhỏ (7 chuỗi và chiều dài 232) đến rất lớn (62 chuỗi và chiều dài 11544).

3.4.2. Tiêu chuẩn đánh giá AIC

Do mỗi mô hình có số tham số tự do khác nhau nên chúng tôi dùng tiêu chuẩn AIC [7] để đánh giá, công thức tính AIC như sau:.

$$AIC(M, D^a) = 2 * LL(M, T^a|D^a) - 2 * \#parameters(M) \qquad (3.12)$$

với *LL(M, T^a |D^a)* là log-likelihood của mô hình *M* và cây xây dựng được là T^a còn *#parameters(M)* là số lượng các tham số tự do của mô hình *M*. Giá trị AIC càng lớn càng tốt. Tất cả các mô hình thử nghiệm đều có cùng các tham số là độ dài các cạnh của cây, 1 tham số α cho tùy chọn phân phối gamma (trừ LG4X) hoặc 6 tham số cho các tốc độ tự do và trọng số (LG4X). Ngoài ra, các mô hình EX2 (UL3) còn có thêm một (hai) tham số hỗn hợp tương ứng.

Với mỗi mô hình *M*, chúng tôi tính giá trị AIC trung bình trên mỗi vị trí cho tất cả các sắp hàng thử nghiệm:

$$AIC/\text{vi tri}(M,\mathbf{A}) = \frac{\sum_{a=1}^{N} AIC(M,D^a)}{\sum_{a=1}^{N} l^a}\ , \qquad (3.13)$$

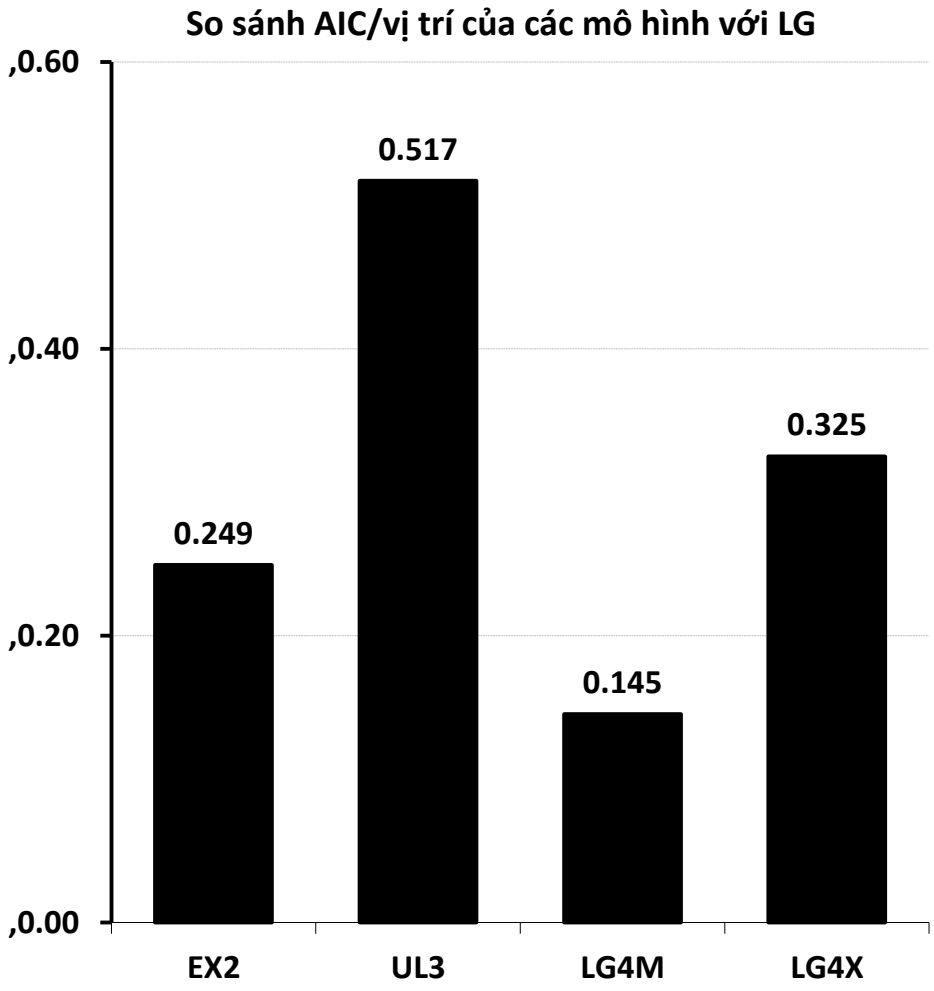
với N là số lượng sắp hàng có trong **A**, l^a là chiều dài của sắp hàng D^a. Chúng tôi so sánh từng cặp mô hình M_1 và M_2 với nhau và đếm số sắp hàng D^a mà AIC(M_1,D^a) > AIC(M_2,D^a) (M_1 tốt hơn M_2 với sắp hàng D^a).

3.4.3. So sánh kết quả của các mô hình

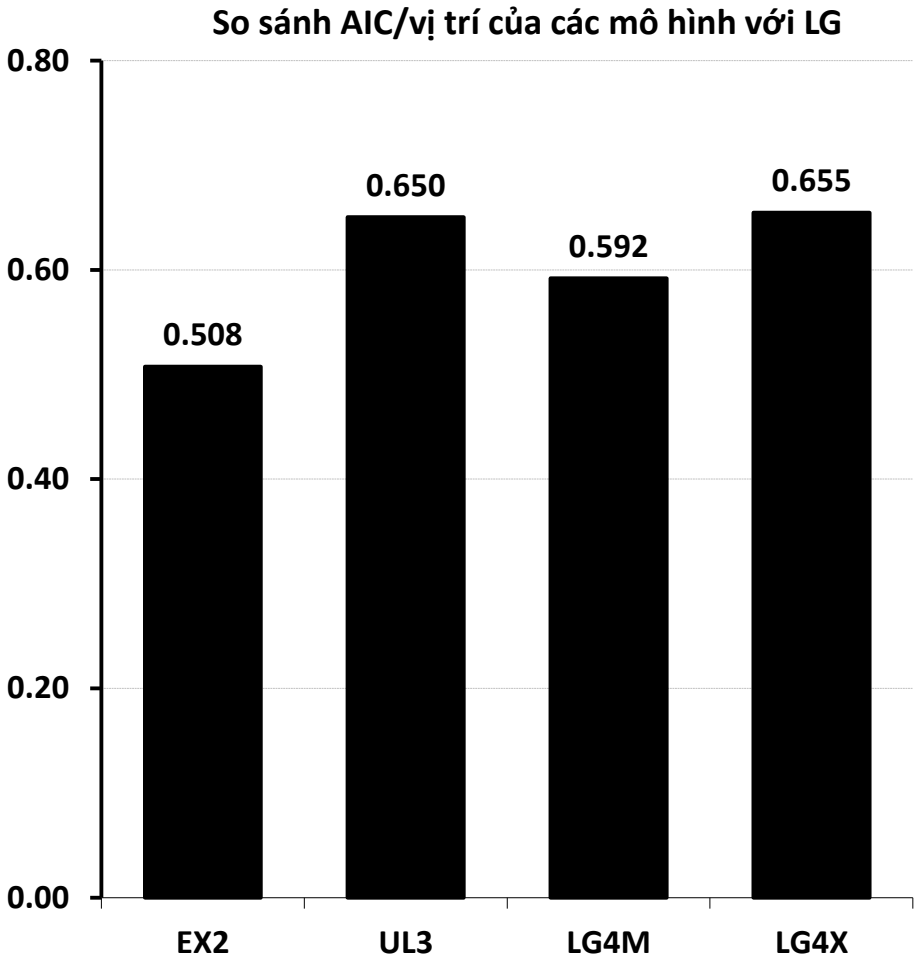
Trong mục này, chúng tôi đánh giá kết quả của các mô hình mới bằng cách so sánh với các mô hình khác sử dụng 84 sắp hàng TreeBase [53] và 300 sắp hàng HSSP [55]. LG4M và LG4X được so sánh với các mô hình đơn ma trận (LG) và các mô hình hỗn hợp hai cấp EX2, UL3 [50]. Tất cả các mô hình (trừ LG4X) đều sử dụng với bốn loại phân phối gamma cho tốc độ biến đổi trên từng vị trí.

Chúng tôi so sánh kết quả của LG4M, LG4X với LG, EX2 và UL3 trên hai tiêu chí giá trị trung bình AIC trên một vị trí và cấu trúc cây. Tất cả các so sánh được chạy với cây khởi tạo là BioNJ [30] và thuật toán tìm kiếm cây SPR [27].

Đầu tiên, chúng tôi so sánh các mô hình với LG trên tiêu chí giá trị trung bình AIC trên một vị trí của tất cả các cây xây dựng bằng phương pháp ML (Hình 3.2 và Hình 3.3). Chúng ta có thể thấy LG4M tốt hơn LG với khoảng cách trung bình AIC trên một vị trí là 0,145 và 0,592 tương ứng với TreeBase và HSSP. Với LG4X, mô hình này tốt hơn LG đáng kể với khoảng cách trung bình AIC trên một vị trí khá lớn là 0,325 và 0,655 tương ứng với TreeBase và HSSP.



Hình 3.2: So sánh giá trị trung bình AIC/vị trí của các mô hình với LG trên bộ dữ liệu TreeBase.



Hình 3.3: So sánh giá trị trung bình AIC/vị trí của các mô hình với LG trên bộ dữ liệu HSSP.

Tiếp tục so sánh LG4X với LG4M (Bảng 3.1), chúng tôi thấy LG4X tốt hơn hẳn LG4M trên TreeBase (khoảng cách trung bình AIC trên một vị trí là 0,180). Với HSSP (Bảng 3.2) thì khoảng cách này khá nhỏ (0,063), nguyên nhân có thể là do cả hai mô hình đều cùng được ước lượng từ một bộ dữ liệu. So sánh LG4X với EX2, UL3 thì thấy LG4X tốt hơn EX2 với khoảng cách trung bình AIC trên một vị trí là 0,076 và 0,147 tương ứng với TreeBase và HSSP. Trên bộ dữ liệu HSSP, LG4X tốt tương đương UL3 khi hơn 0,004 điểm trung bình AIC trên một vị trí. Còn trên bộ dữ liệu TreeBase, LG4X kém UL3 0,192 điểm trung bình AIC trên một vị trí (xem thêm Bảng 3.1 và Bảng 3.2).

Nhận xét chung lại, các mô hình đa ma trận mới đề xuất tốt tương đương các mô hình hỗn hợp hai cấp EX2 và UL3 trên phương diện điểm trung bình AIC trên một vị trí và các mô hình này đều tốt hơn các mô hình đơn ma trận (LG).

Chúng tôi cũng so sánh các mô hình trên từng cấu trúc cây xây dựng được bằng cách đếm số lượng các sắp hàng của M_1 có cấu trúc khác với M_2. Thử nghiệm này là cần thiết vì nếu các mô hình mới xây dựng cây có cùng cấu trúc như các mô hình hiện có thì những nỗ lực giới thiệu các mô hình mới có sẽ ít ý nghĩa.

Các kết quả thử nghiệm với tiêu chí so sánh cấu trúc cây được trình bày trong Bảng 3.1 và Bảng 3.2. Cụ thể, với 84 sắp hàng TreeBase, số lượng cây của LG4M có giá trị log-likelihood tốt hơn LG là 51 (chiếm 61%). Còn LG4X tốt hơn LG ở 72 cây, chiếm 86%. LG4M chỉ tốt hơn LG4X ở một cây duy nhất còn LG4X tốt hơn EX2 và UL3 ở lần lượt 67 cây (80%) và 39 cây (46%). Các mô hình LG4M và LG4X cũng cho các cây có cấu trúc khác biệt so với các mô hình còn lại (xem thêm Bảng 3.1).

Bảng 3.1: So sánh log-likelihood và cấu trúc cây giữa các mô hình trên 84 sắp hàng TreeBase.

M_1	M_2	AIC/vị trí	$\neq M_1 > M_2$	$\neq M_1 < M_2$	$\neq T_1 > T_2$	$\neq T_1 < T_2$
LG4M	LG	0.145	51	33	37	20
LG4X	LG	0.325	72	12	48	10
LG4X	LG4M	0.180	83	1	52	0
LG4X	EX2	0.076	67	17	44	11
LG4X	UL3	-0.192	39	45	24	35

Với 300 sắp hàng HSSP, số lượng cây của LG4M có giá trị log-likelihood cao hơn LG là 270 (chiếm 90%). Còn LG4X tốt hơn LG ở 287 cây, chiếm 96%. So sánh LG4M và LG4X thì LG4M tốt hơn LG4X ở 93 cây, chiếm 31%. LG4X tốt hơn EX2 và UL3 ở lần lượt 241 cây (80%) và 199 cây (50%). Như vậy, có thể thấy LG4X tốt tương đương UL3 - mô hình phức tạp hơn và có sử dụng thông tin về cấu trúc bậc ba của các chuỗi. Các mô hình LG4M và LG4X cũng xây dựng được các cây có cấu trúc khác biệt so với các mô hình còn lại (xem thêm Bảng 3.2).

Bảng 3.2: So sánh log-likelihood và cấu trúc cây giữa các mô hình trên 300 sắp hàng HSSP.

M_1	M_2	AIC/vị trí	$\neq M_1 > M_2$	$\neq M_1 < M_2$	$\neq T_1 > T_2$	$\neq T_1 < T_2$
LG4M	LG	0,592	270	30	251	27
LG4X	LG	0,655	287	13	257	10
LG4X	LG4M	0,063	207	93	166	83
LG4X	EX2	0,147	241	59	200	51
LG4X	UL3	0,004	199	201	165	99

3.4.4. So sánh dung lượng bộ nhớ sử dụng và thời gian chạy

Để so sánh dung lượng bộ nhớ sử dụng và thời gian chạy của hai mô hình LG4M và LG4X với mô hình một ma trận (LG) và mô hình hỗn hợp (EX2, UL3). Chúng tôi xây dựng cây phân loài cho các sắp hàng của bộ dữ liệu TreeBase bằng chương trình PhyML [33]. Máy tính thực nghiệm sử dụng bộ vi xử lý Intel Xeon E5440 tốc độ 2.83GHz, bộ nhớ RAM 16GB.

Bảng 3.3: Kết quả so sánh dung lượng bộ nhớ sử dụng (GB) và thời gian chạy (giờ) của các mô hình với bộ dữ liệu TreeBase.

Mô hình	Dung lượng bộ nhớ sử dụng với 1 sắp hàng (GB)	Thời gian chạy với 1 sắp hàng (giờ)	Tổng thời gian chạy (giờ)
LG4M	2	8	60
LG4X	2	11	85
LG	2	6	55
EX2	4	51	280
UL3	6	53	380

Kết quả thực nghiệm cho thấy cả hai mô hình LG4M và LG4X yêu cầu cùng một dung lượng bộ nhớ giống như các mô hình đơn ma trận, trong khi các mô hình EX2 và UL3 lần lượt cần nhiều hơn hai và ba lần dung lượng bộ nhớ. Cụ thể, để xây dựng cây với sắp hàng lớn nhất của bộ dữ liệu TreeBase (có 62 chuỗi và chiều dài là 11544), LG4X cần 2GB trong khi UL3 cần đến 6GB.

Về tốc độ, LG4M có thời gian tính toán tương đương LG, còn LG4X chạy chậm hơn từ 1,5 đến 1,8 lần. Tuy nhiên cả LG4M và LG4X đều chạy nhanh hơn rất nhiều so với các mô hình hỗn hợp (xem thêm Bảng 3.3).

3.5. Kết luận chương

Trong chương này, chúng tôi đã đề xuất hai mô hình mới là LG4M và LG4X. Ý tưởng chính là sử dụng nhiều ma trận khác nhau cho các loại tốc độ tiến hóa khác nhau, kết hợp với sử dụng một phân phối tự do để thay thế cho các phân phối gamma chuẩn của tốc độ biến đổi trên từng vị trí. Các thực nghiệm với bộ dữ liệu TreeBase cho thấy rằng LG4M và LG4X xây dựng được các cây có giá trị log-likelihood cao hơn và cấu trúc khác so với các mô hình đơn ma trận.

Cả LG4M và LG4X đều cho kết quả tốt hơn so với các mô hình đơn ma trận trong khi yêu cầu cùng một lượng tài nguyên tính toán, đây hứa hẹn sẽ là sự thay thế hợp lý cho các mô hình đơn ma trận. Hai mô hình này cũng có thể được tích hợp vào các phần mềm xây dựng cây phân loài hiện tại một cách dễ dàng. Các kết quả nghiên cứu của chương này đã được công bố trên tạp chí quốc tế *Molecular Biology and Evolution* năm 2012 (công trình khoa học số 5).

Chương 4. HỆ THỐNG ƯỚC LƯỢNG MÔ HÌNH TỰ ĐỘNG

4.1. Mở đầu

Nhiều mô hình biến đổi axít amin chung đã được đề xuất như JTT [40], WAG [63] và LG [49] và cho hiệu quả tốt trong phần lớn các trường hợp. Ngoài ra, một số mô hình cho các tập dữ liệu riêng biệt đã được đề xuất như HIVw và HIVb cho vi rút HIV [47]; FLU cho vi rút cúm [18], mtREV cho prôtêin ty thể [6]. Các mô hình riêng biệt này thường cho kết quả tốt hơn các mô hình chung khi áp dụng cho các nhóm prôtêin tương ứng [6, 18, 47]. Do đó, việc ước lượng mô hình cho các tập dữ liệu riêng biệt là cần thiết.

Chúng tôi muốn xây dựng một hệ thống tự động để đáp ứng nhu cầu trên. Hệ thống cần phục vụ được cùng lúc nhiều người dùng và thời gian chờ của người dùng càng ngắn càng tốt. Do đó chúng tôi đã nghiên cứu và áp dụng một cải tiến khác để tăng tốc quá trình ước lượng mô hình.

Trong phương pháp ước lượng mô hình **Q,** bước tối ưu cấu trúc cây bằng ML được lặp lại nhiều lần. Các nghiên cứu đã chỉ ra rằng ước lượng mô hình với các cây gần tối ưu cũng cho các mô hình có chất lượng tốt. Từ đây chúng tôi đề xuất một phương pháp ước lượng nhanh với chỉ một lần tối ưu cấu trúc cây.

4.2. Phương pháp ước lượng nhanh

Chúng tôi thống kê với nhiều tập dữ liệu và bộ tham số khác nhau thì số lần lặp ước lượng lại ma trận **Q** trung bình là 3 và bước xây dựng cây bằng ML là tốn thời gian nhất [20]. Từ những phân tích này, thuật toán được cải tiến như sau:

- Chỉ tối ưu cấu trúc cây một lần duy nhất ở lần lặp 2.
- Thay thế tần số axít amin trong mô hình khởi tạo ban đầu bằng tần số axít amin của dữ liệu.
- Sử dụng 4 phân loại tốc độ gamma.

Các bước cụ thể của thuật toán ước lượng nhanh mô hình biến đổi axít amin được trình bày trong Thuật toán 4.1 sau đây:

procedure Thuật toán ước lượng nhanh;
input: Tập N sắp hàng $\mathbf{A} = \{D^1, \ldots D^N\}$ và mô hình khởi tạo ban đầu $\mathbf{Q_{start}}$;
output: Mô hình **Q**;
begin
Thay thế tần số axít amin trong $\mathbf{Q_{start}}$ bằng tần số tính từ dữ liệu;
$\mathbf{Q} \leftarrow \mathbf{Q_{start}}$;
for (i = 1 .. 3)
foreach sắp hàng D^a trong **A**
if (i == 1) **then**
$T^a \leftarrow$ Cây phân loài của D^a xây dựng bằng thuật toán BioNJ [30];
endif;
if (i == 2) **then**
Tối ưu cấu trúc của T^a với **Q** bằng thuật toán SPR [27];
endif;
- Tối ưu độ dài các cạnh của T^a với **Q**;
- Tối ưu tham số của phân phối gamma với 4 phân lớp tốc độ biến đổi theo vị trí;
- Tách D^a thành 4 sắp hàng con D_1^a, D_2^a, D_3^a, D_4^a dựa theo xác suất của các phân phối tốc độ theo vị trí.
- Tạo ra 4 cây con T_1^a, T_2^a, T_3^a, T_4^a có cấu trúc giống T^a, các cạnh của 4 cây con được nhân tỷ lệ theo các tốc độ đã ước lượng của mỗi phân loại theo phân phối gamma;

end foreach;
Ước lượng ma trận **Q'** từ các sắp hàng và cây con ở trên bằng thuật toán EM [41] với **Q** là ma trận khởi tạo ban đầu;
$\mathbf{Q} \leftarrow \mathbf{Q'}$;
endfor;
end;

Thuật toán 4.1: Thuật toán ước lượng nhanh mô hình biến đổi axít amin.

4.3. Kết quả thực nghiệm

4.3.1. Dữ liệu kiểm tra

Chúng tôi sử dụng ba bộ dữ liệu để tiến hành các thực nghiệm. Bộ dữ liệu thứ nhất là Pfam [9] gồm 3912 sắp hàng. Đây là bộ dữ liệu đã dùng để ước lượng mô hình LG [49]. Bộ dữ liệu thứ hai là TreeBase [53] với 84 sắp hàng để kiểm tra mô hình LG. Bộ dữ liệu thứ ba là FLU đã sử dụng để ước lượng mô hình biến đổi axít amin cho vi rút cúm [18].

4.3.2. Kết quả với bộ dữ liệu Pfam

Để đánh giá phương pháp cải tiến, chúng tôi ước lượng lại mô hình LG với đúng tập dữ liệu đã công bố. Gọi mô hình ước lượng lại là LG'. Kết quả cho thấy LG' gần như giống hệt với LG (độ tương quan Pearson bằng 0,996).

4.3.2.1. So sánh thời gian ước lượng mô hình

So sánh phương pháp mới và cũ, chúng tôi thấy tổng thời gian ước lượng mô hình giảm xấp xỉ hai lần. Trong đó chủ yếu là giảm ở bước xây dựng cây (xem thêm Bảng 4.1).

Bảng 4.1: So sánh thời gian ước lượng lại mô hình LG với hai phương pháp. Quá trình ước lượng mô hình dừng sau 3 lần lặp.

Lần lặp	Bước	Thời gian (giờ)	
		Phương pháp cũ	Phương pháp mới
1	Xây dựng cây	31,1	2,0
	Ước lượng tham số mô hình	5,9	5,9
2	Xây dựng cây	30,7	31,4
	Ước lượng tham số mô hình	6,7	6,7
3	Xây dựng cây	30,3	1,6
	Ước lượng tham số mô hình	6,7	6,7
Tổng thời gian:		**111,4**	**54,2**

4.3.2.2. So sánh hiệu quả của mô hình

So sánh về hiệu quả xây dựng lại cây bằng phương pháp ML, hai mô hình cho kết quả tương đương khi chênh lệch trung bình log-likelihood trên một vị trí là không đáng kể (0,003). So sánh giá trị log-likelihood của từng cây xây dựng được, chúng tôi thấy LG tốt hơn LG' ở 37/84 sắp hàng (chiếm 44%), còn LG' tốt hơn LG ở 47/84 sắp hàng (chiếm 56%).

Chúng tôi đã cho chạy thủ tục bootstrap 500 lần để đánh giá độ tin cậy của mô hình LG'. Gọi **R** và **Π** là hai thành phần của mô hình LG, **R'** và **Π'** là hai thành phần của mô hình LG'. Kết quả cho thấy:

1. Giá trị trung bình của độ lệch tương đối giữa véc tơ tần số **Π** và **Π'** là rất nhỏ, chỉ khoảng 0,4%.
2. Giá trị trung bình của độ lệch tương đối của ma trận tốc độ biến đổi tương đối **R** và **R'** thì lớn hơn (4%) nhưng là khá nhỏ.

4.3.3. Kết quả với bộ dữ liệu FLU

Mô hình FLU100 được ước lượng bằng phương pháp cũ từ 100 sắp hàng được chọn ngẫu nhiên từ bộ dữ liệu FLU. Còn mô hình FLU100' được ước lượng bằng phương pháp mới đề xuất với cùng 100 sắp hàng trên. Mô hình FLU100' rất gần với mô hình FLU100 (độ tương quan Pearson là 0,999), FLU100' cũng gần với mô hình FLU trong bài báo đã công bố [18] (độ tương quan Pearson là 0,987).

4.3.3.1. So sánh thời gian ước lượng mô hình

Tương tự như kết quả với bộ dữ liệu Pfam, tổng thời gian ước lượng mô hình FLU100' cũng giảm khoảng 2 lần. Trong đó chủ yếu là giảm ở bước *Xây dựng cây* (xem thêm Bảng 4.2).

Bảng 4.2: So sánh thời gian ước lượng lại mô hình FLU với hai phương pháp. Quá trình ước lượng mô hình dừng sau 3 lần lặp.

Lần lặp	Bước	Thời gian (giờ)	
		Phương pháp cũ	Phương pháp mới
1	Xây dựng cây	14,3	0,5
	Ước lượng tham số mô hình	0,6	0,6
2	Xây dựng cây	11,0	15,9
	Ước lượng tham số mô hình	0,5	0,5
3	Xây dựng cây	7,7	0,2
	Ước lượng tham số mô hình	6,7	6,7
Tổng thời gian:		**34,3**	**17,9**

4.3.3.2. So sánh kết quả của mô hình

Chúng tôi chọn ngẫu nhiên trong bộ dữ liệu FLU ra 200 sắp hàng không trùng lặp với 100 sắp hàng của FLU100 và tiến hành xây dựng cây bằng phần mềm PhyML [33] với FLU100 và FLU100'.

So sánh log-likelihood của 200 cây xây dựng bởi hai mô hình, chúng tôi thấy chênh lệch giá trị trung bình log-likelihood trên một vị trí cũng rất nhỏ, gần như không đáng kể (0,006). So sánh chi tiết hơn, mô hình FLU100 tốt hơn mô hình FLU100' ở 71 trên tổng số 200 sắp hàng (chiếm 36%), còn mô hình FLU100' tốt hơn mô hình FLU100 ở 129 sắp hàng (chiếm 64%).

Chúng tôi cũng cho chạy thủ tục bootstrap 1000 lần để đánh giá độ tin cậy của mô hình FLU100'. Tương tự như với mô hình LG và LG', gọi **R** và **Π** là hai thành phần của mô hình FLU100, **R'** và **Π'** là hai thành phần của mô hình FLU100'. Chúng tôi có một số nhận xét như sau:

1. Giá trị trung bình của độ lệch tương đối giữa véc tơ tần số **Π** và **Π'** là 2,9%, lớn hơn của LG' nhưng vẫn chấp nhận được.
2. Giá trị trung bình của độ lệch tương đối của ma trận tốc độ biến đổi tương đối **R** và **R'** thì khá lớn (18,5%).

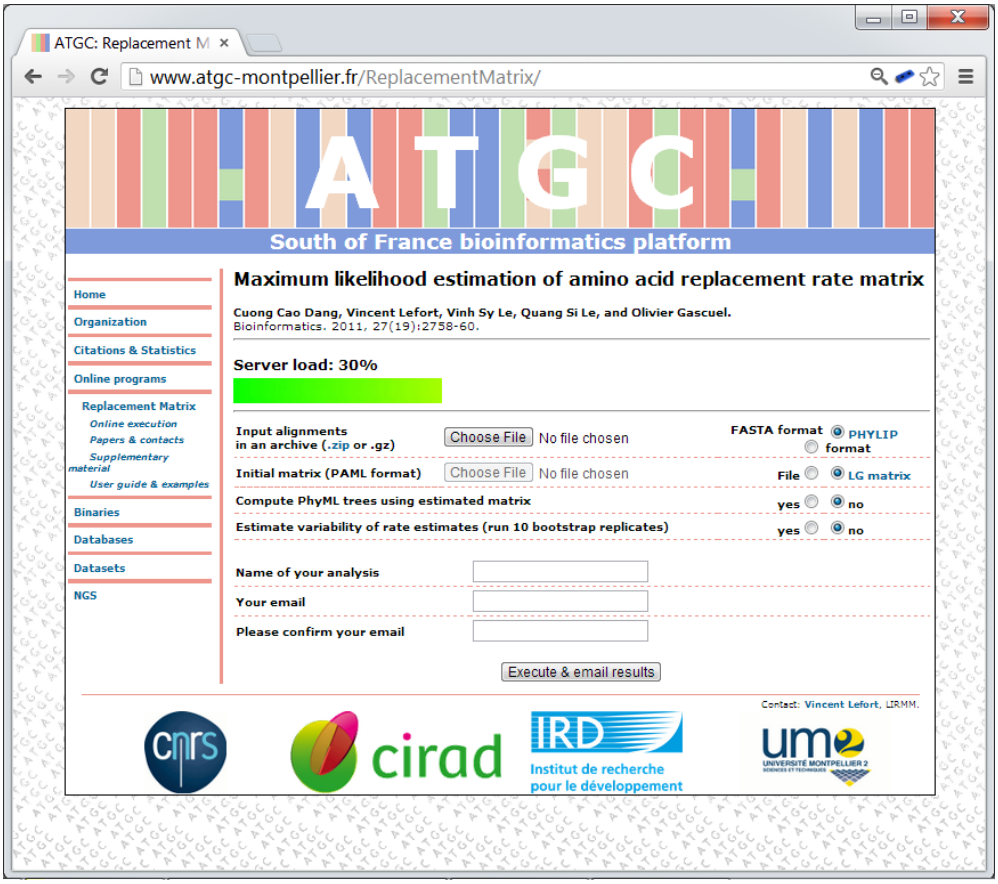
Từ các thực nghiệm với hai bộ dữ liệu Pfam và FLU cho thấy tần số các axít amin đều được ước lượng khá chính xác. Tuy nhiên, việc ước lượng chính xác các hệ số hoán đổi là không dễ, chúng ta có thể thấy chúng có độ lệch tương đối cao trong các kết quả, đặc biệt là với bộ dữ liệu FLU. Nguyên nhân của vấn đề này là do quá trình biến đổi axít amin có một phần ẩn và các giá trị này không thể được tính trực tiếp từ các chuỗi (trái ngược với các tần số), đặc biệt đối với các cặp axít amin mà hiếm khi được liên kết với nhau nhưng lại có nhiều trong các sắp hàng của vi rút cúm.

4.4. Hệ thống ước lượng mô hình tự động

Chúng tôi kết hợp với Viện nghiên cứu LIRMM, Cộng hoà Pháp để xây dựng hệ thống ước lượng mô hình tự động ứng dụng phương pháp ước lượng nhanh đã trình bày ở trên. Người dùng có thể tải lên một tập các sắp hàng prôtêin mà họ quan tâm và sẽ nhận qua thư điện tử mô hình và một số thống kê, so sánh với các mô hình thông dụng khác. Hệ thống còn có tuỳ chọn bootstrap không tham số để đánh giá độ tin cậy và ổn định của kết quả. Cây phân loài được ước lượng bằng ma trận kết quả cũng được cung cấp như một tùy chọn. Hệ thống là sự kết hợp và tinh chỉnh của các phần phần mềm ML mới nhất như PhyML 3.0 [33] và XRATE 2.0 [41] và được chạy trên một hệ thống cluster. Minh họa giao diện chính của hệ thống ước lượng mô hình tự động như trong Hình 4.1.

Sau khi mô hình được ước lượng, hệ thống gửi mô hình kết quả qua email cho người dùng cùng với một số kết quả thống kê và so sánh. Hai lựa chọn bổ sung có sẵn là:

1. Thực hiện một nghiên cứu bootstrap để đánh giá độ ổn định của ma trận **Q**.
2. Chạy PhyML 3.0 với **Q** và với ma trận kết quả cùng các tùy chọn tiêu chuẩn để xây dựng các cây phân loài của tất cả các sắp hàng đầu vào. Các cây này được dự đoán là sẽ có sự khác biệt đáng kể so với các cây xây dựng bởi ma trận $\mathbf{Q}_{\mathbf{start}}$ hay LG. Để tiết kiệm thời gian tính toán, cây được xây dựng từ kết quả của bước 3.



Hình 4.1: Hệ thống trực tuyến ước lượng ma trận biến đổi axít amin.

4.5. Kết luận chương

Chương này của luận án đã trình bày một cải tiến khác của phương pháp ước lượng ma trận giúp giảm đáng kể thời gian thực hiện (trung bình còn 1/2 so với phương pháp cũ). Phương pháp cải tiến đã được kiểm thử với hai bộ dữ liệu Pfam [9] và FLU [18]. Mô hình ước lượng bằng phương pháp cải tiến gần như giống hệt với mô hình được ước lượng bằng phương pháp cũ (độ tương quan Pearson > 0,999). Giá trị log-likelihood chênh lệch giữa hai mô hình là không đáng kể. Các cấu trúc cây cũng không có nhiều khác biệt giữa mô hình ước lượng lại và mô hình đã công bố.

Chương này cũng trình bày hệ thống trực tuyến tự động ước lượng ma trận biến đổi từ dữ liệu của người dùng. Kết quả nghiên cứu của chương này đã được công bố trên tạp chí quốc tế *Bioinformatics* năm 2011 (công trình khoa học số 2).

Chương 5. MÔ HÌNH BIẾN ĐỔI AXÍT AMIN CHO VI RÚT CÚM

5.1. Giới thiệu về vi rút cúm và sự cần thiết của các mô hình biến đổi axít amin riêng biệt cho từng loài

Các mô hình biến đổi axít amin chung như PAM [21], JTT [39], WAG [63], LG [49] được xây dựng dựa trên một tập các chuỗi axít amin từ các loài sinh vật khác nhau. Tuy nhiên, những nghiên cứu mới nhất gần đây cho thấy các mô hình chung này không cho kết quả tốt nhất khi phân tích dữ liệu prôtêin của một số loài sinh vật, như vi rút HIV [47]. Nguyên nhân là vì các mô hình chung không thể phản ánh đầy đủ bản chất sinh học, hóa học cũng như quá trình tiến hóa của một số loài riêng biệt.

Do đó, một hướng mới đang được các nhà nghiên cứu quan tâm và phát triển là xây dựng các mô hình biết đổi axít amin riêng cho từng loài sinh vật. Năm 2007, Nickle và đồng nghiệp áp dụng phương pháp cực đại khả năng để xây dựng mô hình biến đổi axít amin cho vi rút HIV [47]. Nhóm tác giả xây dựng hai mô hình, HIVw để mô phỏng quá trình biến đổi của vi rút bên trong người bệnh, và HIVb để mô phỏng quá trình biến đổi của vi rút giữa các người bệnh. Các kết quả của nhóm tác giả cho thấy HIVb và HIVw tốt hơn các mô hình chung khác.

Trong những năm gần đây, dịch bệnh do vi rút cúm đang xảy ra trên toàn thế giới. Từ đó nổi lên vấn đề cần phải nghiên cứu toàn diện về loại vi rút nguy hiểm này, đặc biệt là các nghiên cứu về quá trình tiến hóa, lan truyền và lây nhiễm của chúng [3, 8, 24, 31, 38].

Vi rút cúm là một loại vi rút RNA và thuộc họ Orthomyxoviridae [2, 13, 43]. Chúng được chia thành ba loại là: cúm A, cúm B và cúm C, trong đó có cúm A là phổ biến và nguy hiểm nhất. Vi rút cúm A đã gây ra nhiều vấn đề nghiêm trọng cho

sức khỏe con người và kinh tế xã hội, đặc biệt là dịch H5N1 (cúm gia cầm) và H1N1. Bảng 5.1 liệt kê các dịch cúm lớn của con người đã xảy ra trên thế giới.

Bảng 5.1: Danh sách các dịch cúm lớn xảy ra với con người.

Tên dịch cúm	Năm xảy ra	Tổn thất về con người	Chủng vi rút gây bệnh
Asiatic (Russian) Flu	1889–1890	1 triệu	H2N2
Spanish Flu	1918–1920	50 triệu	H1N1
Asian Flu	1957–1958	1,5 đến 2 triệu	H2N2
Hong Kong Flu	1968–1969	1 triệu	H3N2
Russian Flu	1977-1978	Không có số liệu	H1N1
Swine Flu	2009–2010	18,209	H1N1

Do đó trong chương này, luận án đề xuất mô hình FLU cho vi rút cúm để giúp tăng cường sự hiểu biết của chúng ta về sự tiến hóa của loại vi rút này. Mô hình FLU được xây dựng với phương pháp ước lượng nhanh đã đề xuất trong Chương 2. Các kết quả thực nghiệm đã chỉ ra rằng FLU tốt hơn hẳn các mô hình hiện tại khi phân tích prôtêin của vi rút cúm.

5.2. Ước lượng mô hình FLU

Chúng tôi sử dụng bộ dữ liệu chuẩn của vi rút cúm đã được sử dụng trong bài báo [18], kết hợp với phương pháp chia tách sắp hàng theo cấu trúc cây ở chương 2 để ước lượng mô hình FLU. Ngưỡng chia tách được chọn bằng 8 (k=8), có nghĩa là các sắp hàng sau khi được chia tách sẽ có kích thước từ 8 đến 16 chuỗi. Tổng số sắp hàng trước khi chia chia tách là 992, số lượng sắp hàng sau khi chia tách là 3970. Tiếp tục thực hiện các bước ước lượng mô hình như trong chương 2, chúng tôi có một mô hình biến đổi axít amin cho vi rút cúm gọi là FLU.

5.3. Kết quả thực nghiệm

Mô hình FLU và được so sánh với 14 mô hình được sử dụng rộng rãi nhất hiện nay, danh sách 14 mô hình có thể xem trong Bảng 5.2.

5.3.1. Phân tích và đánh giá mô hình

Mô hình biến đổi axít amin **Q** bao gồm ma trận hệ số hoán đổi (**R**) và tần số xuất hiện của 20 axít amin (**Π**). Chúng tôi phân tích FLU bằng cách so sánh hai thành phần này của FLU với hai thành phần tương ứng của các mô hình khác. Bảng 5.2 cho thấy độ tương quan Pearson thấp giữa FLU và các mô hình khác. Điều này chứng tỏ FLU rất khác so với các mô hình hiện tại.

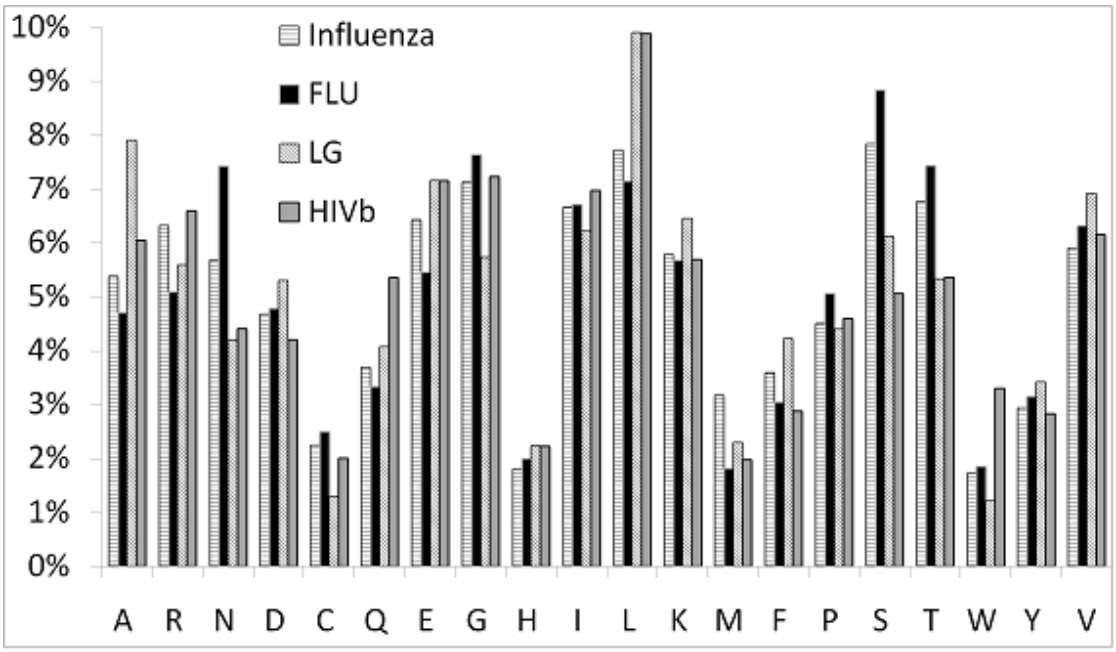
Bảng 5.2: Độ tương quan Pearson giữa mô hình FLU và 14 mô hình phổ biến hiện có. Các giá trị tương quan thấp cho thấy mô hình FLU là rất khác biệt so với các mô hình hiện có.

Mô hình	R	Π
JTT	0.874	0.802
HIVb	0.865	0.718
HIVw	0.835	0.840
WAG	0.820	0.766
LG	0.811	0.718
CpREV	0.810	0.751
Blosum62	0.757	0.747
MtREV	0.756	0.481
RtREV	0.750	0.666
VT	0.746	0.771
MtMam	0.735	0.480
DCMut	0.727	0.694
Dayhoff	0.727	0.694
MtArt	0.692	0.460

Tiếp theo, Hình 5.1 so sánh tần số axít amin của các mô hình và dữ liệu thực nghiệm (ký hiệu là Influenza). Chúng tôi nhận thấy tần số axít amin của FLU và dữ liệu (Influenza) là gần giống nhau nhưng tương đối khác so với hai mô hình còn lại.

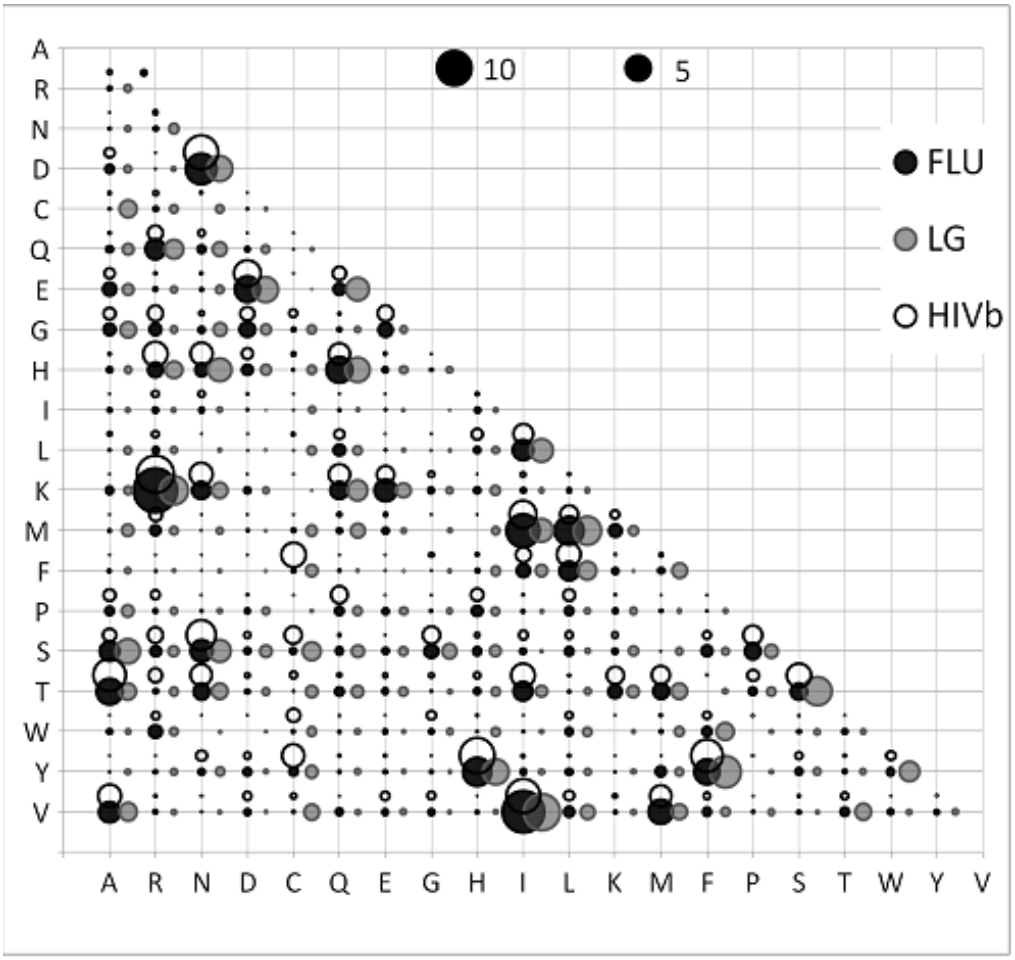
Độ tương quan Pearson giữa các tần số axít amin của FLU và dữ liệu (0,943) cao hơn nhiều độ tương quan Pearson giữa FLU với mô hình HIVb (0,718) và LG (0,718). Đáng chú ý, chúng tôi quan sát thấy sự khác biệt lớn giữa các tần số axít amin của dữ liệu và những mô hình còn lại. Ví dụ: tần số của Alanine (A) trong dữ liệu (~5%) là thấp hơn nhiều so với LG (~8%), tần số của Leucine (L) trong dữ liệu

(~7%) cũng thấp hơn nhiều so với LG (~10%) và HIVb(~10%). Những kết quả này chứng tỏ rằng FLU thể hiện tần số axít amin của các chuỗi prôtêin vi rút cúm chính xác hơn các mô hình khác.

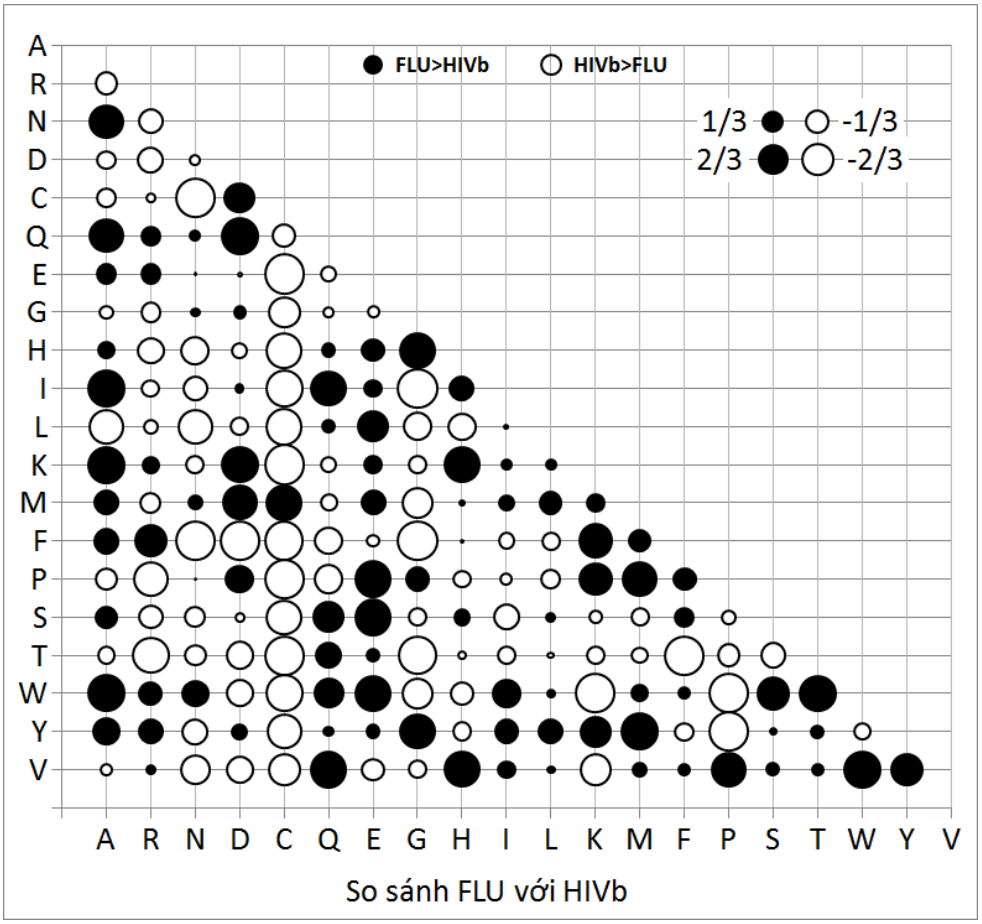


Hình 5.1: So sánh tần số xuất hiện của 20 axít amin trong dữ liệu thực nghiệm (được ký hiệu là Influenza) với các mô hình FLU, LG và HIVb.

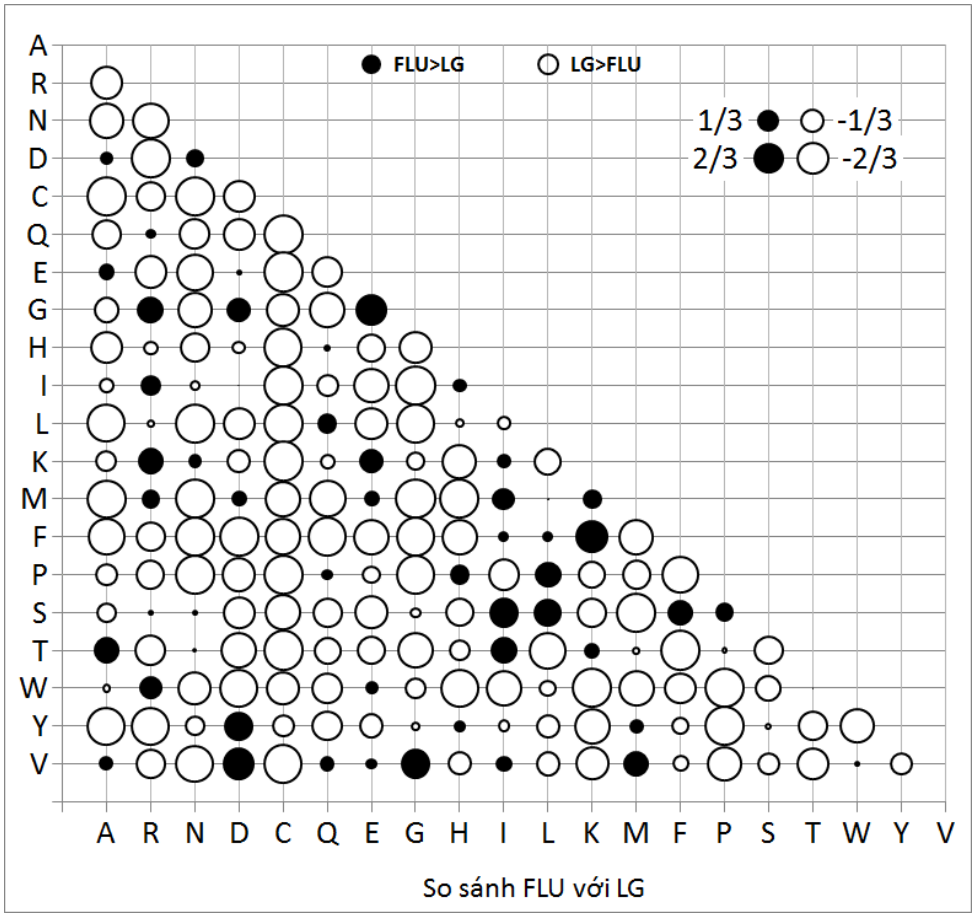
Các hệ số hoán đổi của mô hình FLU, HIVb và LG được minh họa trong Hình 5.2. Về nguyên tắc, các mô hình này đều mô tả được tính chất sinh học, hóa học và vật lý tương tự của các axít amin. Ví dụ các hệ số hoán đổi lớn giữa K (axít amin phân cực, tích điện dương) và R (axít amin phân cực, tích điện dương) hay hệ số hoán đổi nhỏ giữa K và C (axít amin không phân cực, trung tính). Tuy nhiên, chúng khác nhau đáng kể khi chúng ta nhìn vào sự khác biệt tương đối của chúng (xem thêm ở Hình 5.3 và Hình 5.4).



Hình 5.2: Các hệ số hoán đổi trong mô hình FLU, LG và HIVb. Các hình tròn màu đen, xám, trắng thể hiện các hệ số hoán đổi tương ứng của FLU, LG và HIVb.



Hình 5.3: So sánh tương quan các hệ số hoán đổi giữa FLU và HIVb. Các hình tròn hiển thị sự khác biệt tương đối giữa hệ số hoán đổi trong FLU và HIVb. Các hình tròn màu đen thể hiện hệ số của FLU lớn hơn HIVb, màu trắng thể hiện hệ số của HIVb lớn hơn FLU. Giá trị 1/3 hoặc 2/3 có nghĩa hệ số của FLU lớn hơn HIVb 2 hoặc 5 lần. Giá trị -1/3 hoặc -2/3 có nghĩa hệ số của HIVb lớn hơn FLU 2 hoặc 5 lần.



Hình 5.4: So sánh tương quan các hệ số hoán đổi giữa FLU và LG. Các hình tròn hiển thị sự khác biệt tương đối giữa hệ số hoán đổi trong FLU và LG. Các hình tròn màu đen thể hiện hệ số của FLU lớn hơn LG, màu trắng thể hiện hệ số của LG lớn hơn FLU. Giá trị 1/3 hoặc 2/3 có nghĩa rằng hệ số của FLU lớn hơn LG 2 hoặc 5 lần. Giá trị -1/3 hoặc -2/3 có nghĩa rằng hệ số của LG lớn hơn FLU 2 hoặc 5 lần.

Bảng 5.3: Độ lệch tương đối giữa các hệ số hoán đổi của FLU so với HIVb và LG. Giá trị ở hàng "*Hai lần*" và cột "*FLU>LG*" cho biết số hệ số hoán đổi trong FLU lớn hơn ít nhất hai lần hệ số tương ứng trong LG. Giải thích tương tự cho các ô còn lại.

	FLU>HIVb	**HIVb>FLU**	**FLU>LG**	**LG>FLU**
Hai lần	52	48	20	106
Năm lần	32	25	3	67

Bảng 5.3 tóm tắt độ lệch tương đối giữa các hệ số hoán đổi của FLU với HIVb và LG. Ví dụ, có 67 trong tổng số 190 hệ số của LG lớn hơn ít nhất năm lần những hệ số tương ứng của FLU. Những phân tích trên giúp chúng ta có thể đưa ra kết luận là mô hình FLU có các hệ số hoán đổi và tần số axít amin khác biệt rất lớn so với các mô hình hiện có.

5.3.2. So sánh hiệu quả của FLU với các mô hình khác

FLU được so sánh với các mô hình khác trong việc xây dựng cây phân loài bằng ML cho các sắp hàng prôtêin vi rút cúm.

5.3.2.1. Thử nghiệm toàn cục

Trong thử nghiệm toàn cục, FLU và các mô hình JTT, WAG, LG cùng được dùng để xây dựng cây phân loài bằng ML cho tất cả 3970 sắp hàng. Do FLU được ước lượng và thử nghiệm trên cùng một bộ dữ liệu nên FLU chứa nhiều hơn các mô hình khác 208 tham số tự do. Vì vậy, để so sánh FLU và các mô hình khác, chúng tôi sử dụng tiêu chuẩn AIC với điểm phạt là 208 tham số [7].

Bảng 5.4 cho thấy giá trị AIC trung bình của FLU cao hơn các mô hình còn lại. Ví dụ, AIC của FLU cao hơn HIVb là 0,088/vị trí, tương đương FLU tốt hơn HIVb trung bình 13 điểm log-likelihood cho mỗi sắp hàng có độ dài 300.

Bảng 5.4: Giá trị AIC trung bình trên mỗi vị trí của FLU so với các mô hình khác (sắp xếp theo thứ tự giảm dần). FLU có giá trị AIC trung bình trên mỗi vị trí tốt nhất.

Mô hình	Giá trị trung bình AIC/vị trí
FLU	-9.241
HIVb	-9.329
JTT	-9.334
HIVw	-9.369
CpREV	-9.390
VT	-9.399
LG	-9.401
WAG	-9.405
Blosum62	-9.463
Dayhoff	-9.486
DCMut	-9.487
RtREV	-9.491
MtREV	-9.827
MtMam	-9.888
MtArt	-9.925

5.3.2.2. Thử nghiệm chéo

Trong thử nghiệm chéo, Tập dữ liệu D được chia ngẫu nhiên thành hai tập D_1và D_2, một tập để huấn luyện, tập còn lại để kiểm tra. Đầu tiên FLU_1 (hoặc FLU_2) được ước lượng từ D_1 (hoặc D_2). Sau đó FLU_1 (hoặc FLU_2) được sử dụng để xây dựng cây ML cho các sắp hàng của D_2 (hoặc D_1). Kết quả chúng tôi thu được 3970 cây phân loài được xây dựng với FLU_1 hoặc FLU_2. Để đơn giản, chúng ta gọi FLU là mô hình tổng thể cho cả FLU_1 và FLU_2 trong thử nghiệm này. Do chúng tôi tiến hành ước lượng và kiểm tra mô hình trên hai bộ dữ liệu độc lập nên có thể so sánh trực tiếp giá trị log-likelihood của cây xây dựng bằng FLU với giá trị log-likelihood của cây xây dựng bằng các mô hình khác.

Bảng 5.5 cho thấy FLU tốt hơn hẳn các mô hình khác. FLU xây dựng cây phân loài có giá trị log-likelihood tốt nhất cho 2499/3970 sắp hàng (chiếm 63%), tốt thứ hai cho 482/3970 sắp hàng (chiếm12%).

Bảng 5.5: So sánh xây dựng cây của FLU với 14 mô hình khác. Các cột *1st*, *2nd*, … *15th* cho biết số lượng sắp hàng mà mô hình đứng ở thứ hạng tương ứng trên tổng số 15 mô hình thử nghiệm. Ví dụ, mô hình FLU đứng ở thứ hạng đầu tiên với 2499, đứng vị trí thư hai với 482 trên tổng số 3970 sắp hàng. Cột *LogLK*/vị trí cho biết giá trị trung bình của log-likelihood trên một vị trí của mỗi mô hình.

Mô hình	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	12th	13th	14th	15th	LogLK/vị trí
FLU	2499	482	489	170	119	101	51	22	12	12	13	0	0	0	0	-4.621
HIVb	874	871	1113	411	157	109	157	91	53	28	105	1	0	0	0	-4.664
JTT	309	913	1203	1350	111	65	13	5	1	0	0	0	0	0	0	-4.667
HIVw	176	1230	307	357	223	267	287	195	271	208	64	385	0	0	0	-4.684
LG	88	152	264	633	562	415	333	361	221	406	433	95	7	0	0	-4.701
CpREV	13	54	111	433	1341	813	557	281	179	130	48	10	0	0	0	-4.695
VT	7	54	223	355	639	1160	953	300	224	55	0	0	0	0	0	-4.699
WAG	1	192	195	137	591	643	790	1189	183	44	5	0	0	0	0	-4.703
Dayhoff	1	11	39	50	90	108	259	425	712	548	1523	163	18	20	3	-4.743
RtREV	1	0	1	4	8	12	68	233	574	1296	577	1167	12	12	5	-4.745
Blosum62	1	7	11	24	82	181	380	545	1029	522	536	617	18	17	0	-4.731
MtREV	0	0	0	0	0	0	0	3	2	10	25	23	3158	626	123	-4.914
DCMut	0	4	14	46	47	93	120	317	498	701	615	1463	30	20	2	-4.743
MtMam	0	0	0	0	0	3	2	3	10	10	26	40	402	2528	946	-4.944
MtArt	0	0	0	0	0	0	0	0	1	0	0	6	325	747	2891	-4.962

Cây phân loài xây dựng với FLU cũng có giá trị log-likelihood trung bình cao nhất, cao hơn khoảng 0,043 điểm log-likelihood so với mô hình tốt nhất thứ hai là HIVb (xem thêm Bảng 5.6). Trong tổng số 3970 cây thì có trên 84,5% cây xây dựng với FLU tốt hơn (theo giá trị log-likelihood) cây xây dựng với các mô hình còn lại.

Bảng 5.6: So sánh từng đôi giữa FLU với các mô hình HIVb, HIVw, JTT và LG. M_1 - M_2: trung bình log-likelihood khác nhau giữa cây xây dựng với M_1 và M_2, giá trị dương (âm) có nghĩa M_1 là tốt hơn (kém hơn) so với M_2. $M_1 > M_2$: số sắp hàng trên tổng số 3970 sắp hàng mà M_1 tốt hơn M_2. $M_2 > M_1$: số lượng sắp hàng trên tổng số 3970 sắp hàng mà M_2 tốt hơn M_1.

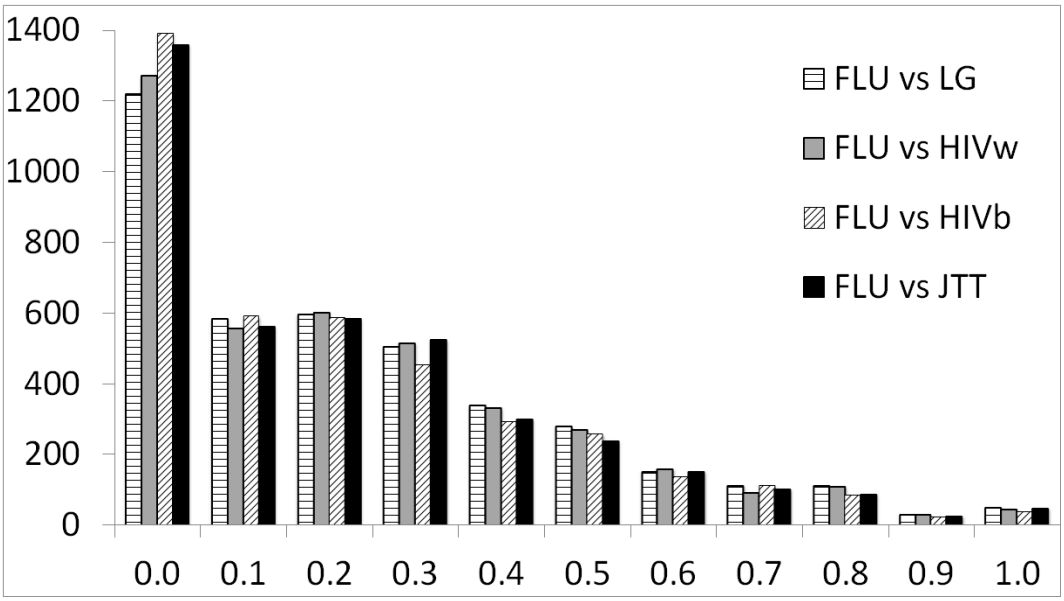
M_1	M_2	$M_1 - M_2$	$M_1 > M_2$	$M_2 > M_1$
FLU	HIVb	0.043	3356	614
FLU	JTT	0.046	3357	613
FLU	HIVw	0.063	3371	599
FLU	LG	0.080	3367	603

5.3.2.3. Phân tích và đánh giá cây

Để đo sự khác biệt giữa cấu trúc của hai cây, chúng tôi sử dụng khoảng cách Robinson-Fould (RF) [51]. Khoảng cách RF giữa cấu trúc của hai cây là tỷ lệ giữa số phân vùng chỉ có ở một trong hai cây trên tổng số phân vùng của cả hai cây. Như vậy, khoảng cách RF có khoảng giá trị từ 0,0 đến 1,0. Giá trị RF càng nhỏ thì cấu trúc càng giống nhau.

So sánh cây xây dựng bởi FLU với với các mô hình khác, chúng tôi thấy phần lớn các cây có cấu trúc khác nhau (khoảng cách RF > 0). Cụ thể: với HIVb là 2579 cây (~65%), với HIVw là 2699 cây (~68%), với JTT là 2612 cây (~66%) và với LG là 2751 cây (~69%).

Hình 5.5 cho thấy chi tiết số lượng các cây xây dựng với FLU có cấu trúc khác cây xây dựng với các mô hình khác. Cụ thể, khoảng cách RF bằng 0,2 ở ~600 cây (tương đương khoảng 15% tổng số cây), khoảng cách RF bằng 0,4 ở ~340 cây (tương đương khoảng 8.5% tổng số cây).



Hình 5.5: Khoảng cách Robinson-Foulds (RF) giữa các cây của FLU với HIVb, HIVw, JTT và LG. Trục hoành thể hiện khoảng cách RF, trục tung thể hiện số lượng cây.

Độ dài trung bình các cạnh của cây xây dựng với FLU cũng dài hơn những cây xây dựng với các mô hình khác: FLU là 0,074 trong khi LG là 0,028, JTT là 0,047. Phát hiện này cho thấy cây xây dựng với FLU thể hiện được nhiều biến đổi ẩn trong quá trình tiến hóa của vi rút cúm hay có thể nói FLU mô tả tốt hơn các đặc điểm của quá trình tiến hóa vi rút cúm so với các mô hình chung.

5.3.3. Tính bền vững của mô hình

Chúng tôi phân tích tính bền vững của vi rút cúm bằng cách đo độ tương quan Pearson giữa 3 mô hình FLU, FLU_1 và FLU_2 (xem mục 5.3.2.2. Thử nghiệm chéo).

Bảng 5.7 cho thấy mối tương quan rất cao (độ tương quan Pearson lớn hơn 0,990) giữa FLU, FLU_1 và FLU_2 ở cả hệ số hoán đổi (**R**) và tần số axít amin (**Π**). Như vậy, bộ dữ liệu D là đủ lớn để ước lượng một mô hình biến đổi axít amin cho prôtêin cúm.

Bảng 5.7: Độ tương quan Pearson giữa 3 mô hình FLU, FLU_1 và FLU_2.

	R	Π
FLU với FLU_1	0,9995	0,9998
FLU với FLU_2	0,9995	0,9998
FLU_1với FLU_2	0,9981	0,9994

Chúng tôi cũng đánh giá ảnh hưởng của yếu tố thời gian của quá trình tiến hóa của vi rút cúm trên FLU. Chúng tôi chia tập dữ liệu D thành hai tập con gần bằng nhau là D_{t1} gồm các chuỗi prôtêin trước năm 2004 và D_{t2} gồm các chuỗi prôtêin từ năm 2004 trở đi. Sau đó, hai tập con D_{t1} và D_{t2} này được sử dụng để ước lượng hai mô hình FLU_{t1} và FLU_{t2} tương ứng. Cả hai mô hình FLU_{t1} và FLU_{t2} đều rất giống nhau (độ tương quan Pearson lớn hơn 0,998). Hơn thế, cả hai cũng đều rất giống với FLU (độ tương quan Pearson lớn hơn 0,998). Độ tương quan cao chỉ ra rằng ảnh hưởng của các yếu tố thời gian của quá trình tiến hóa tới việc ước lượng mô hình biến đổi axít amin là không đáng kể. Như vậy, FLU có thể được áp dụng để phân tích các prôtêin của vi rút cúm mới xuất hiện cũng như đã xuất hiện từ lâu.

5.4. Kết luận chương

Vi rút cúm là rất nguy hiểm cho các sinh vật nói chung và loài người nói riêng. Do đó các nghiên cứu y sinh học về vi rút này là rất cần thiết. Tuy nhiên các mô hình chung hiện tại chưa đáp ứng được các nhu cầu nghiên cứu đó. Do vậy một mô hình biến đổi axít amin dành riêng cho vi rút cúm sẽ là một thành phần quan trọng hỗ trợ cho các nghiên cứu này.

Chúng tôi đã ước lượng mô hình FLU và các phân tích cho thấy FLU mô hình hoá các đặc điểm tiến hóa của vi rút cúm tốt hơn so với các mô hình hiện tại. Các thử nghiệm toàn cục và thử nghiệm chéo đều khẳng định FLU tốt hơn các mô hình hiện tại trong việc xây dựng cây ML.

KẾT LUẬN

Các nghiên cứu về chuỗi axít amin đóng vai trò quan trọng trong sinh học phân tử và tin sinh học. Mô hình biến đổi axít amin là một thành phần có vai trò rất quan trọng trong nghiên cứu chuỗi axít amin. Phương pháp cực đại khả năng là một trong những phương pháp tốt nhất hiện nay để ước lượng mô hình biến đổi axít amin. Tuy nhiên các phương pháp hiện tại vẫn còn gặp nhiều hạn chế về thời gian thực hiện cũng như độ chính xác.

Luận án đã đề xuất hai cải tiến quan trọng để giảm thời gian của phương pháp ước lượng mô hình biến đổi axít amin hiện tại. Đề xuất đầu tiên là hai phương pháp chia tách nhỏ dữ liệu đầu vào giúp giảm đáng kể thời gian ước lượng mô hình. Đề xuất thứ hai là giảm bớt các bước tối ưu tham số khi xây dựng cây phân loài giúp giảm 50% thời gian ước lượng mô hình. Độ chính xác của các phương pháp cải tiến tương đương với phương pháp cũ.

Luận án cũng đưa ra một mô hình đa ma trận mới giúp mô hình hoá tốt hơn quá trình biến đổi của các chuỗi axít amin. Mô hình này cũng đã chứng tỏ được những ưu việt của nó so với các mô hình hiện tại khi độ chính xác được cải thiện đáng kể trong khi thời gian chạy vẫn tương đương với mô hình đơn ma trận.

Luận án đã xây dựng một hệ thống ước lượng mô hình tự động giúp ước lượng các ma trận biến đổi axít amin từ dữ liệu của người dùng. Hệ thống là kết quả nghiên cứu kết hợp cùng Viện nghiên cứu LIRMM, Cộng hoà Pháp. Hệ thống hoạt động được gần hai năm và đã có nhiều người sử dụng.

Chúng tôi cũng xây dựng mô hình FLU cho vi rút cúm. Mô hình FLU đã được tích hợp vào phần mềm xây dựng cây phân loài PhyML và đã chứng tỏ được hiệu quả khi phân tích các chuỗi axít amin của vi rút cúm. Mô hình này giúp tăng cường hiểu biết về vi rút cúm, giúp chúng ta có cách đối phó hữu hiệu hơn với loại vi rút rất nguy hiểm này.

Như vậy luận án đã tập trung phân tích và đề xuất những cải tiến cho các thành phần quan trọng nhất của phương pháp xây dựng mô hình biến đổi axít amin gồm: Dữ liệu đầu vào (Chương 2), Mô hình biến đổi (Chương 3) và Xây dựng cây phân loài bằng ML (Chương 4). Những cải tiến này đã giúp giảm đáng kể thời gian xây dựng và tăng độ chính xác của ma trận. Các kết quả của từng chương có thể gộp lại thành một kết quả thống nhất là những cải tiến cho phương pháp xây dựng ma trận biến đổi axít amin. Tuỳ vào điều kiện bài toán cụ thể mà chúng ta có thể lựa chọn áp dụng một hay nhiều cải tiến.

DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN

1. Cuong DC, Quang LS, Gascuel O, and Vinh LS (2010), "FLU, an amino acid substitution model for influenza proteins", *BMC Evolutionary Biology* Vol. 10 (1), pp. 99-110.
2. Cuong DC, Lefort V, Vinh LS, Quang LS and Gascuel O (2011), "ReplacementMatrix: a web server for maximum-likelihood estimation of amino acid replacement rate matrices", *Bioinformatics* Vol. 27 (19), pp. 2758–2760.
3. Dat LV, Cuong DC, Quang LS and Vinh LS (2011), "A Fast and Efficient Method for Estimating Amino Acid Substitution Models", *Proc. of the 2011 Third International Conference on Knowledge and Systems Engineering*, pp. 85 –91.
4. Sau NV, Cuong DC, Quang LS and Vinh LS (2011), "Protein Type Specific Amino Acid Substitution Models for Influenza Viruses", *Proc. of the 2011 Third International Conference on Knowledge and Systems Engineering*, pp. 98 –103.
5. Quang LS, Cuong DC, and Gascuel O (2012), "Modeling Protein Evolution with Several Amino Acid Replacement Matrices Depending on Site Rates", *Mol Biol Evol* Vol. 29 (10), pp. 2921–2936.

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. Phạm Thị Trân Châu, Trần Thị Áng (2006), *Hóa sinh học*, Nhà xuất bản Giáo dục.
2. Nguyễn Tiến Dũng (2008), "Vài nét về virut cúm gia cầm H5N1," *Tạp chí Khoa học Kỹ thuật Thú y* Tập 15 (4), pp. 80–86.
3. Lê Thanh Hòa, Trương Nam Hải, Nông Văn Hải, Đinh Duy Khang, Phan Văn Chi, Quyền Đình Thi, Lê Trần Bình (2009), "Nguồn gen và cơ chế tiến hoá phân tử của virus cúm A/H1N1 - 2009 gây đại dịch ở người hiện nay," *Tạp chí Công nghệ Sinh học* Tập 7 (2), pp. 133–153.
4. Phạm Thành Hổ (2008), *Di truyền học*, Nhà xuất bản Giáo dục.
5. Lê Đức Trình (2001), *Sinh học phân tử của tế bào*, Nhà xuất bản Khoa học và Kỹ thuật.

Tiếng Anh

6. Adachi J and Hasegawa M (1996), "Model of amino acid substitution in proteins encoded by mitochondrial DNA", *Journal of Molecular Evolution* Vol. 42 (4), pp. 459–468.
7. Akaike H (1974), "A new look at the statistical model identification", *IEEE Transactions on Automatic Control* Vol. 19 (6), pp. 716– 723.
8. Bao Y, Bolotov P, Dernovoy D, Kiryutin B, Zaslavsky L, Tatusova T, Ostell J, and Lipman D (2008), "The influenza virus resource at the National Center for Biotechnology Information", *Journal of Virology* Vol. 82 (2), pp. 596–601.

9. Bateman A, Birney E, Cerruti L, Durbin R, Etwiller L, Eddy SR, Griffiths-Jones S, Howe KL, Marshall M, and Sonnhammer ELL (2002), "The Pfam Protein Families Database", *Nucl. Acids Res.* Vol. 30 (1), pp. 276–280.

10. Baxevanis AD and Ouellette BFF (2001), *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins, 2nd Edition*, Wiley-Blackwell.

11. Bergsten J (2005), "A review of long-branch attraction", *Cladistics* Vol. 21 (2), pp. 163–193.

12. Boeckmann B, Bairoch A, Apweiler R, Blatter M-C, Estreicher A, Gasteiger E, Martin MJ, et al. (2003), "The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003", *Nucl. Acids Res.* Vol. 31 (1), pp. 365–370.

13. Bouvier NM and Palese P (2008), "The biology of influenza viruses", *Vaccine* Vol. 26, pp. 49–53.

14. Brinkmann H, Giezen M van der, Zhou Y, Raucourt GP de, and Philippe H (2005), "An Empirical Assessment of Long-Branch Attraction Artefacts in Deep Eukaryotic Phylogenomics", *Syst Biol* Vol. 54 (5), pp. 743–757.

15. Chor B and Tuller T (2005), "Maximum likelihood of evolutionary trees: hardness and approximation", *Bioinformatics* Vol. 21 (1), pp. 97–106.

16. Creighton TE (1992), *Proteins: Structures and Molecular Properties, 2nd Edition*, W. H. Freeman.

17. Cuong DC, Lefort V, Vinh LS, Quang LS, and Gascuel O (2011), "ReplacementMatrix: a web server for maximum-likelihood estimation of amino acid replacement rate matrices", *Bioinformatics* Vol. 27 (19), pp. 2758–2760.

18. Cuong DC, Quang LS, Gascuel O, and Vinh LS (2010), "FLU, an amino acid substitution model for influenza proteins", *BMC Evolutionary Biology* Vol. 10 (1), pp. 99–110.

19. Darwin C (1928), *The Origin of Species*, Hayes Barton Press.

20. Dat LV, Cuong DC, Quang LS, and Vinh LS (2011), "A Fast and Efficient Method for Estimating Amino Acid Substitution Models", *Proc. of the 2011 Third International Conference on Knowledge and Systems Engineering*, pp. 85 –91.

21. Dayhoff M, Schwartz R, and Orcutt B (1978), "A Model of Evolutionary Change in Proteins", *Atlas of protein sequence and structure* Vol. 5, pp. 345–351.

22. Durbin R, Eddy SR, Krogh A, and Mitchison G (1998), *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press.

23. Edgar RC (2004), "MUSCLE: multiple sequence alignment with high accuracy and high throughput", *Nucleic Acids Research* Vol. 32 (5), pp. 1792–1797.

24. Fauci AS (2005), "Race against time", *Nature* Vol. 435 (7041), pp. 423–424.

25. Felsenstein J (1978), "The Number of Evolutionary Trees", *Syst Biol* Vol. 27 (1), pp. 27–33.

26. Felsenstein J (1981), "Evolutionary trees from DNA sequences: A maximum likelihood approach", *Journal of Molecular Evolution* Vol. 17, pp. 368–376.

27. Felsenstein J (1989), "PHYLIP - Phylogeny Inference Package (Version 3.2)", *Cladistics* Vol. 5, pp. 164–166.

28. Felsenstein J (2004), *Inferring phylogenies*, Sinauer Associates.

29. Fitch WM (1971), "Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology", *Syst Biol* Vol. 20 (4), pp. 406–416.

30. Gascuel O (1997), "BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data", *Mol. Biol. EVol.* Vol. 14 (7), pp. 685–695.

31. Ghedin E, Sengamalay NA, Shumway M, Zaborsky J, Feldblyum T, Subbu V, Spiro DJ, et al. (2005), "Large-scale sequencing of human influenza reveals the dynamic nature of viral genome evolution", *Nature* Vol. 437 (7062), pp. 1162–1166.

32. Goldman N, Thorne JL, and Jones DT (1998), "Assessing the impact of secondary structure and solvent accessibility on protein evolution.", *Genetics* Vol. 149 (1), pp. 445–458.

33. Guindon S, Dufayard J-F, Lefort V, Anisimova M, Hordijk W, and Gascuel O (2010), "New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0", *Syst Biol* Vol. 59 (3), pp. 307–321.

34. Guindon S and Gascuel O (2003), "A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood", *Systematic Biology* Vol. 52 (5), pp. 696–704.

35. Hasegawa M and Fujiwara M (1993), "Relative efficiencies of the maximum likelihood, maximum parsimony, and neighbor-joining methods for estimating protein phylogeny", *Mol. Phylogenet. EVol.* Vol. 2 (1), pp. 1–5.

36. Henikoff S and Henikoff JG (1991), "Automated assembly of protein blocks for database searching", *Nucleic Acids Res.* Vol. 19 (23), pp. 6565–6572.

37. Henikoff S and Henikoff JG (1992), "Amino acid substitution matrices from protein blocks", *Proc. Natl. Acad. Sci. U.S.A.* Vol. 89 (22), pp. 10915–10919.

38. Janies D, Hill AW, Guralnick R, Habib F, Waltari E, and Wheeler WC (2007), "Genomic analysis and geographic visualization of the spread of avian influenza (H5N1)", *Systematic Biology* Vol. 56 (2), pp. 321–329.

39. Jones DT, Taylor WR, and Thornton JM (1994), "A mutation data matrix for transmembrane proteins", *FEBS Letters* Vol. 339 (3), pp. 269–275.

40. Jones DT, Taylor WR, and Thornton JM (1992), "The rapid generation of mutation data matrices from protein sequences", *Computer applications in the biosciences : CABIOS* Vol. 8 (3), pp. 275 –282.

41. Klosterman PS, Uzilov AV, Bendaña YR, Bradley RK, Chao S, Kosiol C, Goldman N, and Holmes I (2006), "XRate: a fast prototyping, training and annotation tool for phylo-grammars", *BMC Bioinformatics* Vol. 7, pp. 428–453.

42. Koshi JM and Goldstein RA (1995), "Context-dependent optimal substitution matrices", *Protein Eng.* Vol. 8 (7), pp. 641–645.

43. Lamb RA and Choppin PW (1983), "The Gene Structure and Replication of Influenza Virus", *Annual Review of Biochemistry* Vol. 52 (1), pp. 467–506.

44. Lemey P, Salemi M, and Vandamme A-M (Editors) (2009), *The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing*, Cambridge University Press.

45. Li W-H (1997), *Molecular Evolution*, Sinauer Associates.

46. Minh BQ, Vinh LS, von Haeseler A, and Schmidt HA (2005), "pIQPNNI: parallel reconstruction of large maximum likelihood phylogenies", *Bioinformatics* Vol. 21 (19), pp. 3794–3796.

47. Nickle DC, Heath L, Jensen MA, Gilbert PB, Mullins JI, and Kosakovsky Pond SL (2007), "HIV-specific probabilistic models of protein evolution", *PloS One* Vol. 2 (6), pp. 503-514.

48. Quang LS, Cuong DC, and Gascuel O (2012), "Modeling Protein Evolution with Several Amino Acid Replacement Matrices Depending on Site Rates", *Mol Biol Evol* Vol. 29 (10), pp. 2921–2936.

49. Quang LS and Gascuel O (2008), “An Improved General Amino Acid Replacement Matrix”, *Molecular Biology and Evolution* Vol. 25 (7), pp. 1307 –1320.

50. Quang LS, Lartillot N, and Gascuel O (2008), “Phylogenetic mixture models for proteins”, *Philos Trans R Soc Lond B Biol Sci* Vol. 363 (1512), pp. 3965–3976.

51. Robinson DF and Foulds LR (1981), “Comparison of phylogenetic trees”, *Mathematical Biosciences* Vol. 53 (1), pp. 131–147.

52. Saitou N and Nei M (1987), “The neighbor-joining method: a new method for reconstructing phylogenetic trees.”, *Mol Biol Evol* Vol. 4 (4), pp. 406–425.

53. Sanderson M, Donoghue M, Piel W, and Eriksson T (1994), “TreeBASE: a prototype database of phylogenetic analyses and an interactive tool for browsing the phylogeny of life”, *American Journal of Botany* Vol. 81 (6), pp. 183–193.

54. Sau NV, Cuong DC, Quang LS, and Vinh LS (2011), “Protein Type Specific Amino Acid Substitution Models for Influenza Viruses”, *Proc. of the 2011 Third International Conference on Knowledge and Systems Engineering*, pp. 98 –103.

55. Schneider R, de Daruvar A, and Sander C (1997), “The HSSP database of protein structure-sequence alignments.”, *Nucleic Acids Res* Vol. 25 (1), pp. 226–230.

56. Spencer M, Susko E, and Roger AJ (2005), “Likelihood, Parsimony, and Heterogeneous Evolution”, *Mol Biol Evol* Vol. 22 (5), pp. 1161–1164.

57. Strimmer K and Haeseler A von (1996), “Quartet Puzzling: A Quartet Maximum-Likelihood Method for Reconstructing Tree Topologies”, *Mol Biol Evol* Vol. 13 (7), pp. 964-969.

58. Tateno Y, Takezaki N, and Nei M (1994), “Relative efficiencies of the maximum-likelihood, neighbor-joining, and maximum-parsimony methods when substitution rate varies with site.”, *Mol Biol Evol* Vol. 11 (2), pp. 261–277.

59. Thorne JL (2000), “Models of protein sequence evolution and their applications”, *Current Opinion in Genetics & Development* Vol. 10, pp. 602–605.

60. Vinh LS (2005), *Phylogeny Reconstructions Come of Age*, Ph.D. Thesis, University of Düsseldorf, Düsseldorf, Germany.

61. Vinh LS and Haeseler A von (2004), “IQPNNI: Moving Fast Through Tree Space and Stopping in Time”, *Mol Biol Evol* Vol. 21 (8), pp. 1565–1571.

62. Wang H-C, Li K, Susko E, and Roger A (2008), “A class frequency mixture model that adjusts for site-specific amino acid frequencies and improves inference of protein phylogeny”, *BMC Evolutionary Biology* Vol. 8 (1), pp. 331–344.

63. Whelan S and Goldman N (2001), “A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach”, *Molecular Biology and Evolution* Vol. 18 (5), pp. 691–699.

64. Yang Z (1993), “Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites”, *Molecular Biology and Evolution* Vol. 10 (6), pp. 1396–1401.

65. Yang Z (1994), “Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods”, *J. Mol. EVol.* Vol. 39 (3), pp. 306–314.

66. Yang Z (2006), *Computational molecular evolution*, Oxford University Press.