ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

TẠ VĂN NHÂN

ÁP DỤNG PHƯƠNG PHÁP **DÓNG HÀNG TRÌNH TỰ** CHO BÀI TOÁN DỰ ĐOÁN BIẾN THỂ GEN

Chuyên ngành: Khoa học dữ liệu

Mã số: 8904468.01QTD

LUẬN VĂN THẠC SỸ KHOA HỌC

NGƯỜI HƯỚNG DẪN KHOA HỌC
PGS.TS. NGUYỄN THỊ HỒNG MINH

Hà Nội - Năm 2021

MỞ ĐẦU

Giải trình tự DNA đang ngày càng trở nên nhanh chóng và kinh tế. Tuy nhiên, để ráp các trình tự thu được dựa trên bộ gen tham chiếu và tìm kiếm các biến thể, chúng ta cần có những hệ thống đủ mạnh để xử lý và phân tích dữ liệu. Phương pháp dóng hàng trình tự là một giải pháp hữu hiệu cho vấn đề này. Đã có nhiều kết quả nghiên cứu liên quan tới phương pháp cũng như phát triển công cụ dóng hàng trình tự được công bố. Tuy nhiên vấn đề về thời gian thực hiện, mức độ chính xác và phạm vi áp dụng của các kĩ thuật dóng hàng vẫn còn là những chủ đề cần được phát triển.

Mục đích của luận văn là nghiên cứu sâu kĩ thuật dóng hàng trình tự, đề xuất cải tiến để tăng hiệu quả về thời gian của thuật toán cũng như khả năng triển khai trên các hạ tầng tính toán phổ dụng. Đồng thời áp dụng thuật toán để khám phá trong phạm vi rộng hơn các biến thể gen so với một số nghiên cứu trước đây, khám phá mức độ ảnh hưởng của các biến thể đến chức năng của Protein.

Cụ thể trong nghiên cứu này, chúng tôi phát triển thuật toán dóng hàng dựa trên chuyển dạng Burrows-Wheeler và thuật toán Smith-Waterman. Trong đó, các mã giả được viết chi tiết để có thể triển khai bằng các ngôn ngữ lập trình khác nhau. Chúng tôi sử dụng ngôn ngữ Go với kỹ thuật song song và đồng thời để triển khai thuật toán dóng hàng trình tự dựa trên chuyển dạng Burrows-Wheeler, các chương trình được triển khai có thể chạy trên các hệ thống tính toán hiệu năng cao nhiều bộ xử lí, và cũng có thể chạy trên các máy tính cá nhân với khả năng tận dụng tất cả các logic proccessor của bộ xử lí. Kết quả thực nghiệm thuật toán bằng chương trình của chúng tôi

được so sánh với kết quả nhận được từ công cụ BWA-MEM nhằm kiểm nghiệm tính chính xác của thuật toán mà chúng tôi phát triển. Đồng thời, việc thử nghiệm này cũng giúp hiểu rõ hơn về các tham số cho phù hợp với dữ liệu để sử dụng thuận lợi các công cụ dóng hàng trên các hệ thống đã có.

Trong chương 1, luận văn giới thiệu một số kiến thức cơ sở về sinh học phân tử, tin sinh học, các công nghệ giải trình tự. Từ những kiến thức cơ sở đó, những nghiên cứu sâu về phương pháp dóng hàng trình tự và những đề xuất cải tiến được trình bày chi tiết trong chương 2, bao gồm cả phần phương pháp và phần thực nghiệm. Cuối cùng, trong chương 3 trình bày những kết quả áp dụng các phương pháp và công cụ dóng hàng để tìm biến thể gen của bệnh tâm thần phân liệt (Schizophrenia), một hội chứng rối loạn tâm thần nghiêm trọng có liên quan đến nhiều gen với yếu tố di truyền cao. Dữ liệu được tiền xử lý và khớp với bộ gen tham chiếu sử dụng thuật toán dựa trên chuyển dạng Burrows-Wheeler. Sau đó, thuật toán dóng hàng Smith-Waterman sắp xếp lại các Haplotype ở một số vùng hoạt động giúp kết quả dóng hàng ban đầu chính xác hơn. Các quá trình được triển khai trên nền tảng Galaxy và máy chủ Linux 64CPUs. Kết quả những biến thể tìm được trên các gen sẽ được so sánh với một số kết quả nghiên cứu của một số nhà khoa học và tổ chức đã được công bố.

Mục lục

Danh sách hình vẽ

Danh sách bảng

CHƯƠNG 1

KIẾN THỨC CƠ SỞ

1.1 Một số khái niệm về sinh học phân tử và di truyền

1.1.1 Các phân tử của một tế bào

Một tế bào gồm có các phân tử nhỏ có chức năng mang theo năng lượng hoặc truyền tín hiệu như: adenosine triphosphate (ATP), epinephrine, chất dẫn truyền thần kinh (neurotransmitters), nước, đường, acid béo, amino acid, và đơn phân tử nucleotide. Các phân tử này có thể tồn tại độc lập hoặc liên kết với nhau tạo thành các đại phân tử. Hai đại phân tử được biết đến trong sinh học là protein và nucleic acid:

Protein là một chuỗi dài được tạo thành bởi liên kết từ hàng trăm đến hàng nghìn amino acid. Sự đa dạng của protein bắt nguồn từ các cách kết hợp khác nhau của 20 loại amino acid mà trình tự của chúng xác định cấu trúc 3 chiều độc đáo của mỗi protein. Các cấu trúc riêng của protein quy định các chức năng cụ thể của chúng như: kháng thể, enzyme, điều khiển, thành phần cấu trúc, vận chuyển, dự trữ [1].

[1]https://ghr.nlm.nih.gov/primer/howgenswork/protein

Deoxyribonucleic acid (DNA) là vật chất di truyền ở người và hầu hết các sinh vật khác mang thông tin (dưới dạng bộ ba mã di truyền) về cách thức, thời gian, và vị trí để sản xuất mỗi loại protein. Gần như mọi tế bào trong cơ thể người đều có cùng một DNA mà thường nằm trong nhân tế bào (nơi nó được gọi là DNA nhân), nhưng một lượng nhỏ DNA cũng có thể được tìm thấy trong ty thể (nơi nó được gọi là DNA ty thể hoặc mtDNA) và lục lạp [2].

Gen là một đơn vị chức năng của DNA. Gen thường bao gồm hai phần: vùng mã hóa (coding region) xác định trình tự amino acid của protein, vùng điều hòa (regulatory region) kiểm soát thời gian và tế bào mà protein được tạo thành [Lod+07]. Hiện nay, các nhà khoa học đã tìm ra khoảng 21000 gen mã hóa protein chiếm 2% hệ gen con người, 98% còn lại là vùng điều hòa biểu hiện gen và những chức năng bí ẩn chưa được khám phá [Per+18].

Ribonucleic acid (RNA) là một phân tử polymer được liên kết bởi các nucleotide. RNA tương đối giống DNA về cấu trúc và cấu tạo, chúng chỉ khác nhau ở hai điểm: RNA có dạng sợi đơn gập vào chính nó trong khi DNA có dạng sợi xoắn kép; T trong DNA được thay thế bằng Uracil (U) trong RNA.

1.1.2 Luận thuyết trung tâm

Luận thuyết trung tâm đưa ra các quy tắc chung cho việc chuyển giao thông tin giữa DNA, RNA và protein [Cri70]. Tế bào chuyển đổi thông tin được mã hóa trong DNA để tạo ra protein thông qua hai quá trình: phiên mã (transcription) và dịch mã (translation). Trong quá trình phiên mã, vùng mã hóa của gen được sao chép thành một phiên bản RNA. Đối với tế bào nhân thực, sản phẩm RNA ban đầu được xử lý thành các phân tử RNA thông tin nhỏ hơn (mRNA), các mRNA này sau đó di chuyển đến tế bào chất. Tại đây, cỗ máy phân tử phức tạp mang tên ribosome chứa trong nó cả mRNA và protein thực hiện quá trình dịch mã. Ribosome lắp ráp và liên kết các amino acid với nhau theo một thứ tự chính xác do mRNA quy định [Lod+07].

[2] https://ghr.nlm.nih.gov/primer/basics/dna

1.1.3 Nhiễm sắc thể

Phần lớn DNA nằm trong nhân của tế bào nhân thực được gấp nhiều lần thành hình dạng các nhiễm sắc thể và được tái tạo trong quá trình phân chia tế bào. Mỗi nhiễm sắc thể chứa một phân tử DNA liên kết với một số protein nhất định. Khi tế bào phân chia, một cỗ máy phân chia gồm nhiều protein (được gọi là replisome) tách DNA trong nhiễm sắc thể thành hai sợi, mỗi sợi được dùng làm khuôn để tổng hợp nên sợi bổ sung ngược của nó. Kết quả là chúng ta có hai chuỗi DNA mới giống chuỗi ban đầu. Ngoại trừ tế bào trứng và tinh trùng, mỗi tế bào người có 23 cặp nhiễm sắc thể, một nửa được di truyền từ bố, nửa còn lại được di truyền từ mẹ, do đó có hai bản sao của một gen được di truyền từ bố và mẹ (lưỡng bội, hay diploid). Riêng đối với các nhiễm sắc thể giới tính, nữ giới sở hữu hai nhiễm sắc thể X trong khi nam giới sở hữu một nhiễm sắc thể X và một nhiễm sắc thể Y. Đối với tế bào trứng và tinh trùng, quá trình phân chia được gọi là meiosis, nó trải qua hai giai đoạn phân bào chia một tế bào ban đầu thành 4 tế bào con trong đó mỗi tế bào con chỉ chứa một bản sao của mỗi nhiễm sắc thể (đơn bội, hay haploid).

1.1.4 Đột biến

Đột biến là sự thay đổi trình tự các nucleotide xảy ra trong quá trình nhân đôi của DNA gây ra sự thay đổi về chức năng của protein. Các đột biến có thể có lợi, có hại, hoặc không ảnh hưởng đến sinh vật. Đột biến có thể di truyền nếu chúng nằm trong các tế bào có khả năng góp phần hình thành con cái, các tế bào này còn được gọi là tế bào mầm (germ-line cell). Ví dụ tế bào chứng, tinh trùng và các tế bào tiền thân của chúng là những tế bào mầm. Ngược lại, đột biến không di truyền nếu chúng nằm trong các tế bào không đóng góp vào sự hình thành con cái như các tế bào cơ thể, loại tế bào này được gọi là tế bào soma (somatic cell). Tuy nhiên, các đột biến trong tế bào

soma vẫn có thể tích lũy theo thời gian gây ra bệnh cho người, chẳng hạn căn bệnh ung thư.

1.2 Các công nghệ giải trình tự DNA

1.2.1 Giải trình tự Sanger

Công nghệ giải trình tự đầu tiên được Frederick Sanger cùng các cộng sự nghiên cứu từ năm 1977 và được thương mại hóa vào năm 1986. Phương pháp giải trình tự Sanger dựa trên việc sao chép DNA trong ống nghiệm (vitro replication). Trên thực tế, bộ gen người đầu tiên đã được giải trình trong một thập kỷ nhờ công nghệ giải trình tự Sanger.

1.2.2 Giải trình tự thế hệ tiếp theo (NGS)

Công nghệ giải trình tự thế hệ tiếp theo có thể giải trình tự DNA với tốc độ rất nhanh và giá thành ngày càng giảm, nhờ đó đã tạo ra những thành tựu khoa học ấn tượng và các ứng dụng sinh học mới. Cho đến năm 2008, các công cụ giải trình tự thế hệ tiếp theo có thể tạo ra một lượng dữ liệu trong 24h tương đương với vài trăm máy giải trình Sanger nhưng chỉ cần một người duy nhất vận hành [Sch08]. Dựa trên nguyên tắc giải trình tự Sanger, phương pháp giải trình tự thế hệ tiếp theo đã có những cải tiến lớn. Trong mỗi chu kỳ, nếu như công nghệ giải trình tự Sanger chỉ cho phép thao tác trên từng fragment thì công nghệ NGS có thể thao tác song song trên nhiều fragment.

1.2.3 Các loại trình tự nhận được từ máy giải trình tự

Như đã đề cập trong công nghệ NGS, các fragment được tổng hợp tách biệt theo cùng một chiều. Dựa vào chiều và khoảng cách các read ta có ba loại trình tự có thể được xuất ra từ máy giải trình tự:

- Single-end reads: toàn bộ các read được tổng hợp theo cùng một chiều thuận hoặc nghịch.
- Paired-end reads: hai read của cùng một đoạn DNA được tổng hợp theo cả hai chiều thuận và nghịch, chúng có thể có hoặc không có trình tự chung (overlapping sequence). Khi kết hợp hai read thuận và nghịch của cùng một đoạn DNA có thể tạo ra một read dài hơn single-end read. Hơn nữa, loại dữ liệu này tạo thuận lợi hơn cho việc lắp ráp trình tự và tìm các biến thể.
- Mate pairs: nếu hai read được tổng hợp từ cùng một đoạn DNA không có trình tự chung thì chiều dài của fragment sẽ bằng tổng chiều dài của hai read, đoạn trình tự ở giữa (inner distance), và adapter. Nếu không tính đến adapter ta có một insert size. Mate pairs là dạng đặc biệt của paired-end reads khi insert size dài cỡ 2000 bp đến 5000 bp.

1.3 Các bài toán tin sinh học

1.3.1 Một số bài toán phổ biến

Trong những năm gần đây, chúng ta đã nghe nói nhiều đến liệu pháp gen (gen therapy) hoặc chỉnh sửa gen (genome editing). Để thực hiện được điều đó, các nhà khoa học đã không ngừng khám phá những thông điệp còn ẩn dấu trong các trình tự sinh học. Ví dụ, tần số xuất hiện của một mẫu trình tự trong một đoạn DNA, cơ chế nhân đôi DNA, và quá trình khử amin cho chúng ta manh mối về vùng bắt đầu sao chép của DNA (gọi là oriC) [Com15]. Bài toán định vị oriC không chỉ giúp ta hiểu cách tế bào tái tạo mà còn giúp con người giải quyết các vấn đề y sinh khác nhau. Một ứng dụng của nó là phương pháp trị liệu gen, người ta cấy một bộ gen nhỏ được biến đổi gen (được gọi là vector virus) vào thành tế bào. Trong nông nghiệp, vectơ virus mang gen nhân tạo đã được sử dụng để tạo ra cà chua chịu sương giá và ngô kháng thuốc trừ sâu. Năm 1990, trong bài báo "Sự bắt đầu", bác sỹ William French Anderson đã công bố liệu pháp gen lần đầu

tiên được thực hiện thành công trên người khi nó cứu sống của một bé gái bốn tuổi bị rối loạn suy giảm miễn dịch kết hợp nghiêm trọng [And90]. Một ví dụ khác, quá trình phiên mã DNA thành các mRNA cho chúng ta tín hiệu về việc xác định các gen quy định chức năng của protein. Quá trình này đóng vai trò cốt yếu trong bài toán phân tích biểu hiện gen.

1.3.2 Bài toán dự đoán ảnh hưởng của biến thể gen

1.3.2.1 Một số cách tiếp cận và hạn chế

Phân tích liên kết (Linkage analysis): là bản đồ ánh xạ liên kết hiển thị thông tin di truyền liên quan đến các nhóm liên kết (cặp nhiễm sắc thể) trong bộ gen [Pev15]. Đơn vị ánh xạ là centiMorgans (cM), dựa trên tần số tái tổ hợp giữa các điểm đánh dấu đa hình như SNPs hoặc vi tế bào (1 cM bằng một sự kiện tái tổ hợp trong 100 meioses). Đối với bộ gen người, tỉ lệ tái tổ hợp khoảng từ 1 đến 2 cM/Mb. Trong các nghiên cứu liên kết, vị trí của gen bệnh được xác định dựa trên mối liên kết của nó với các vị trí đánh dấu di truyền trên nhiễm sắc thể. Phương pháp này được áp dụng thành công cho các bệnh đơn gen, tiêu biểu là bệnh Huntington [Gus89]. Tuy nhiên, phân tích liên kết còn tồn tại một số hạn chế [ADL08]:

- Các gen bệnh chưa được biết trước gây khó khăn cho việc khoanh vùng vị trí của chúng trên các nhiễm sắc thể. Đặc biệt đối với các bệnh phổ biến có liên quan đến nhiều gen, phương pháp phân tích liên kết chưa cho thấy sự hiệu quả.

- Thường có nhiều allele gây bệnh trong một gen, do đó cần một phả hệ đủ lớn để nghiên cứu.

Nghiên cứu liên kết toàn hệ gen (Genome-Wide Association Studies)

1.3.2.2 Giải trình tự bộ gen người

GWAS và Linkage chỉ sử dụng thông tin từng phần, trong khi giải trình tự bộ gen người phân tích toàn bộ mối quan hệ giữa biến thể gen và kiểu hình đang được áp dụng thành công cho các bệnh phức tạp [KB13]. Nếu mảng SNP trong GWAS tạo ra dữ liệu chỉ vài trăm nghìn đến vài triệu biến thể thì WGS và WES cho phép khám phá nhiều hơn các biến thể. Giống như hàng nghìn mẫu bình thường và mẫu bệnh đã được GWAS nghiên cứu trong những năm gần đây, số lượng lớn mẫu tương tự hiện đang được giải trình tự cho các rối loạn phức tạp như tâm thần phân liệt, rối loạn lưỡng cực và tự kỷ.

1.4 Dóng hàng trình tự

Bài toán mà chúng ta quan tâm là dự đoán sự ảnh hưởng của biến thể gen đến kiểu hình. Các phương pháp còn được chia ra làm ba nhóm chính dựa trên: thông tin tiến hóa hay còn gọi là bảo thủ trình tự (sequence conservation); cấu trúc 3D của protein; các thuộc tính khác nhau được tính toán trực tiếp từ trình tự amino acid [Kha+15]. Như đã đề cập đến sự hạn chế của một số cách tiếp cận và sự ưu việt của giải trình tự bộ gen, chúng ta tập trung vào nhóm phương pháp bảo thủ trình tự. Trong đó, **các thuật toán dóng hàng trình tự** là cơ sở của **nhóm phương pháp bảo thủ trình tự** sử dụng tối đa dữ liệu **giải trình tự bộ gen người**. Các trình tự bảo thủ là các trình tự giống nhau hoặc tương tự trong Protein, DNA, và RNA.

CHƯƠNG 2

PHÁT TRIỂN CÁC THUẬT TOÁN DÓNG HÀNG TRÌNH TỰ

2.1 Thuật toán dựa trên chuyển dạng Burrows-Wheeler

2.1.1 Một số cấu trúc dữ liệu

2.1.1.1 Mảng hậu tố (Suffix Arrays)

Mảng hậu tố được giới thiệu lần đầu tiên vào năm 1993 bởi Udi Manber và gen Myers như là một sự thay thế hiệu quả về bộ nhớ cho cây hậu tố trước đó [MM93]. Với một chuỗi T chứa các chữ cái ta cần đưa ra một mảng chứa các số nguyên không âm lưu lại vị trí các ký tự đầu tiên của các hậu tố thuộc T. Để thuận lợi cho việc dóng hàng sau này ký tự \$ được thêm vào cuối chuỗi T. Lý do ta chọn ký tự \$ là khi sắp xếp lại chuỗi T theo thứ tự bảng chữ cái thì \$ sẽ đứng ở vị trí đầu tiên. Ngoài ra, chúng ta có một số khái niệm sau:

- *Suffix String (SS)* là chuỗi con của T bắt đầu từ một vị trí nào đó và kết thúc tại \$.

- *Suffix Matrix (SM)* là một ma trận mà các hàng là các *SS* được sắp xếp theo thứ tự bảng chữ cái.

- *Suffix Arrays (SA)* là mảng số nguyên không âm chứa thứ tự của các ký tự đầu thiên thuộc *SS* trên T.

Để xây dựng *mảng hậu tố* của chuỗi T trước tiên ta cần sắp xếp tất cả các chuỗi con *SS* theo thứ tự trong bảng chữ cái. Hon et al. đã công bố một thuật toán hiệu quả cho việc sắp xếp này với thời gian $\mathcal{O}(|T|.log(|T|)$ sử dụng bộ nhớ $\mathcal{O}(|T|.log(|\Sigma|))$ bits, với $|\Sigma|$ là số ký tự duy nhất của chuỗi T không kể ký tự \$ [Hon+07]. Sau khi có được *mảng hậu tố SA* ta có thể sử dụng nó để nhanh chóng xác định vị trí của một trình tự mỗi lần nó xuất hiện trong T. Hơn nữa, các chuỗi con *SS* sau khi sắp xếp được nhóm lại ở các vị trí liên tiếp nếu chúng có chuỗi con tiền tố giống nhau. Ví dụ trong hình 2.1, các chuỗi con của chuỗi ATCATGATC\$ được nhóm lại trong ma trận *SM*. Hai chuỗi ATC\$ và ATCATGATC\$ có chuỗi con chung ATC nên chúng được nhóm lại ở các vị trí gần nhau 1 và 2 tương ứng với các giá trị 6 và 0 trong *mảng hậu tố SA*. Để giảm không gian lưu trữ, ta có thể chỉ cần lưu *mảng hậu tố một phần* (*Partial Suffix Arrays*).

2.1.1.2 Ma trận chuyển dạng Burrows-Wheeler

Năm 1994, Michael Burrows và David Wheeler đã công bố phương pháp chuyển đổi chuỗi bằng cách sắp xếp theo khối cho thuật toán nén dữ liệu không mất thông tin, chuỗi được chuyển đổi còn được gọi là *chuyển dạng Burrows-Wheeler (BWT)* [BW94]. Để nhận được BWT của một chuỗi T ban đầu ta cần tạo ra một ma trận BWT (BWTM) như sau:

- **Bước 1**: Tạo ra tập hợp các chuỗi con *SS* của chuỗi T, sắp xếp các chuỗi này theo thứ tự bảng chữ cái từ trên xuống dưới ta được ma trận *SM*.

- **Bước 2**: Xây dựng ma trận $BWTM$ bằng cách thêm vào sau các chuỗi SS các tiền tố tương ứng của chúng trên chuỗi T.

BWTM	Index	SA
$ATCATGAT**C**	0	9
ATC$ATCAT**G**	1	6
ATCATGATC**$**	2	0
ATGATC$AT**C**	3	3
C$ATCATGA**T**	4	8
CATGATC$A**T**	5	2
GATC$ATCA**T**	6	5
TC$ATCATG**A**	7	7
TCATGATC$**A**	8	1
TGATC$ATC**A**	9	4

Hình 2.1: *Mảng hậu tố* của chuỗi ATCATGATC$ được được tạo thành bởi sự sắp xếp lại các *chuỗi hậu tố* của nó theo thứ tự bảng chữ cái. Kết quả ta được *mảng hậu tố SA* sắp xếp lại thứ tự các ký tự đầu tiên của các chuỗi con SS trên chuỗi ban đầu. Mặt khác, khi thêm các tiền tố tương ứng với các chuỗi con SS vào ngay sau nó ta được ma trận BWT. Cột cuối cùng của $BWTM$ được in đậm đại diện cho chuyển dạng Burrows-Wheeler.

Để thực hiện quá trình giải nén, ta không chỉ dựa vào BWT mà còn cần đến cột đầu tiên của $BWTM$ (FC). Ban đầu, có thể thấy việc nén dữ liệu không hiệu quả vì ngoài BWT ta còn phải lưu FC. Tuy nhiên, vấn đề này được giải quyết trong thuật toán dóng hàng khi ta chỉ cần lưu các vị trí của các phần tử đầu tiên thuộc tập hợp {A, T, G, C} nằm trên FC (FO). Trước hết chúng ta xét hai tính chất của $BWTM$ là tính chất chu trình và tính chất đầu cuối. Tính chất chu trình Đối với cách xây dựng $BWTM$, các dòng của ma trận là các chuỗi được thu được từ T bởi cách xoay vòng các chuỗi hậu tố và tiền tố theo một chu trình. Quay lại ví dụ với chuỗi ATCATGATC$, ký tự $ nằm ở vị trí đầu tiên tại hàng đầu tiên của $BWTM$ do ta đã sắp xếp lại các SS tại bước 1, phần 2.1.1.2. Giả sử ban đầu chỉ có hai cột FC và BWT, ta cần tìm ký tự thứ hai của dòng 1. Rõ ràng, nếu chọn $ là hậu tố thì nó sẽ đứng trước một chuỗi tiền tố thuộc T, ký tự đầu tiên của chuỗi tiền tố này chính là ký tự thứ hai của dòng 1. Khi chưa quay vòng, $ đứng ở vị trí cuối cùng của chuỗi T, do đó ký tự cần tìm

là A ở đầu dòng 3. Dễ thấy dòng 3 cũng chính là chuỗi T ban đầu (Xem hình 2.1). Giả sử ta cần tìm ký tự thứ hai của dòng 3. Với một ký tự A thuộc FC ta có thể tìm thấy 3 ký tự A thuộc BWT. Để lựa chọn ký tự tiếp theo là A ở dòng nào ta phải xét đến tính chất thứ hai. Tính chất đầu cuối Sự xuất hiện lần thứ k của một kí tự trong cột đầu và lần xuất hiện thứ k của ký tự này trong cột cuối tương ứng với cùng vị trí của ký tự này trong chuỗi T. Nói cách khác, các ký tự giống nhau trên FC có thứ tự trước sau giống với thứ tự trước sau của chúng trên BWT. Đặt số thứ tự của các ký tự giống nhau bằng các chỉ số nguyên dương. Đối với chuỗi ATCATGATC$ ta có FC là $\$_1A_1A_2A_3C_1C_2G_1T_1T_2T_3$, BWT là $C_1G_1\$_1C_2T_1T_2T_3A_1A_2A_3$ (Xem hình 2.2). Như vậy, từ A_2 ở dòng 3 có thể suy ra vị trí tiếp theo của nó là T_2 ở đầu dòng 9. Quá trình này được lặp lại cho đến khi tìm ra chuỗi ban đầu.

Từ tính chất chu trình và *mảng hậu tố* SA, ta có thể xây dựng chuyển dạng Burrows-Wheeler của chuỗi T với thời gian tuyến tính [OS09]. Ký hiệu BWT_i, T_i tương ứng là ký tự thứ i của BWT và T, SA_i là giá trị tại vị trí thứ i của mạng hậu tố, ta có:

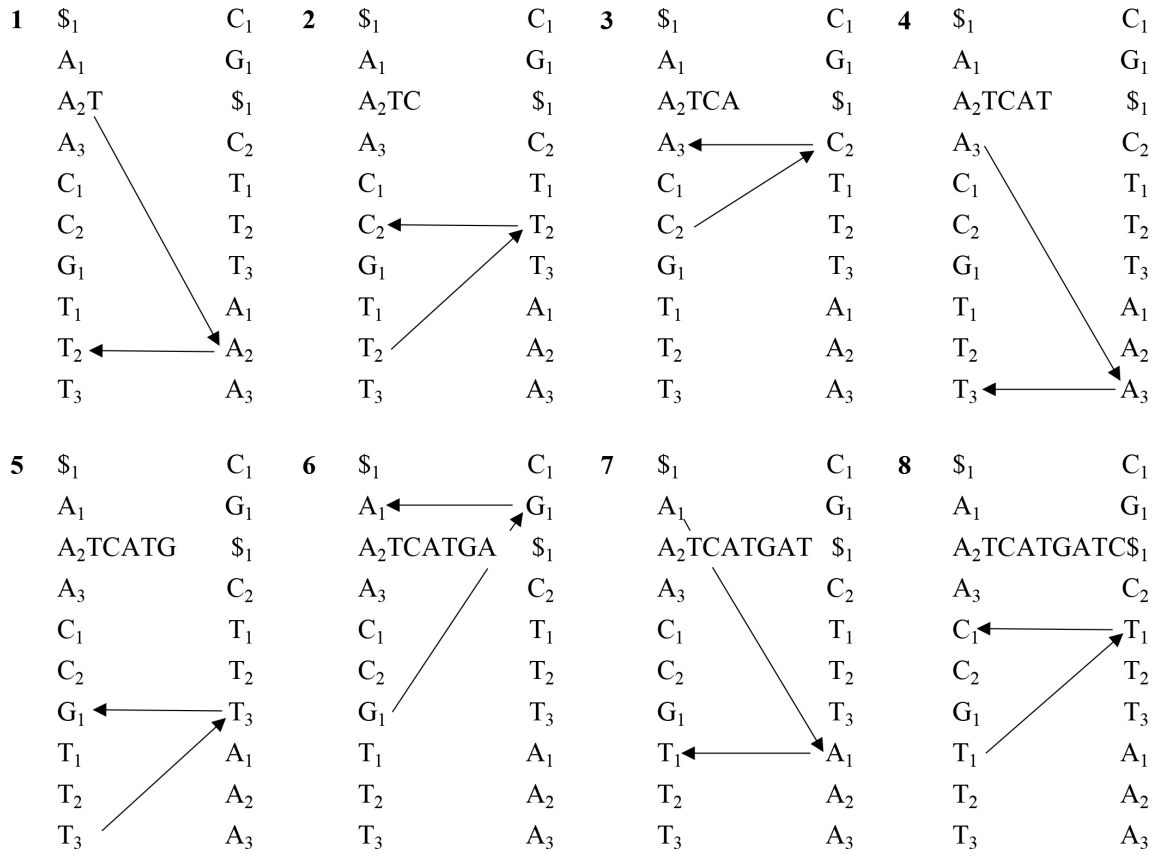
$$BWT_i = \begin{cases} T_{SA_i-1} & \text{nếu } SA_i > 0 \\ T_{|T|-1} = \$ & \text{nếu } SA_i = 0 \end{cases}$$

với $i = 0, 1, ... , |T| - 1$.

Ngoài ra, khi sắp xếp lại BWT theo thứ tự bảng chữ cái ta nhận được FC. Các ký tự giống nhau trên FC được nhóm lại thành các cụm. Nhờ tính chất đầu cuối, ta chỉ cần lưu vị trí đầu tiên của các ký tự này (FO) cho các thuật toán về sau.

2.1.1.3 Ma trận điểm kiểm tra (Checkpoint Arrays)

Sau khi xây dựng được chuỗi BWT của trình tự T, vị trí của một ký tự thuộc chuỗi BWT có thể được tìm thấy trên chuỗi T. Để thực hiện điều này, ta cần đếm số lần xuất hiện của một ký tự trên BWT mà thứ tự của nó thuộc đoạn [0, n], với n = 0, ..., $|T| - 1$. Tuy nhiên thao tác này tiêu tốn khá nhiều thời gian. Do đó, một *ma trận điểm*



Hình 2.2: **Tính chất chu trình:** ở bước 1, hậu tố bắt đầu từ T ở dòng 3 được đặt lên trước tiền tố bắt đầu bằng A ở dòng 9. Tương tự, ở bước 2, hậu tố bắt đầu bằng C ở dòng 3 được đặt lên trước tiền tố AT ở dòng 6. **Tính chất đầu cuối:** ở bước 3, C_2 của FC tương ứng với C_2 của BWT, từ đó ta tìm ra ký tự tiếp theo ở dòng 3 là A.

kiểm tra (C) được sử dụng để lưu các giá trị đếm này. Giống như *mảng hậu tố*, ta cũng có thể chỉ cần lưu một phần của ma trận điểm kiểm tra để tiết kiệm không gian lưu trữ. Ví dụ trong hình 2.3 là *ma trận điểm kiểm tra* của chuỗi BWT CG$CTTTAAA. Đối với bộ gen người, nếu cả bốn cột A, T, C, G được lưu đầy đủ thì dung lượng cần thiết để lưu trữ ma trận này (15GB) gấp đến 5 lần dung lượng lưu trữ một bộ gen người (3GB). Do đó, ta có thể lưu một ma trận con của C mà chỉ chứa các dòng cách nhau một khoảng nào đó, chẳng hạn 100 đơn vị. Khi ấy dung lượng cần thiết dùng cho việc lưu trữ một phần *ma trận điểm kiểm tra* chỉ xấp xỉ 150MB mà tốc độ tính toán

Index	BWT	A	T	C	G
0	C	0	0	1	0
1	G	0	0	1	1
2	$	0	0	1	1
3	C	0	0	2	1
4	T	0	1	2	1
5	T	0	2	2	1
6	T	0	3	2	1
7	A	1	3	2	1
8	A	2	3	2	1
9	A	3	3	2	1

Hình 2.3: *Ma trận điểm kiểm tra* của chuỗi ATCATGATC$ lưu số lần xuất hiện của các ký tự trong tập hợp {A, T, G, C} mà thứ tự của nó trên BWT thuộc đoạn [0, n], với n = -1, 0, ..., 9.

vẫn được nâng cao.

2.1.2 Thuật toán

2.1.2.1 Thuật toán khớp chính xác

P. Ferragina và G. Manzini đã đưa ra thuật toán tìm kiếm lùi (backward search) đếm số lần xuất hiện của một mẫu P trên chuỗi T với thời gian $\mathcal{O}(|P| + occ)$ [FM05]. Trong đó, occ là số lần xuất hiện của mẫu P trong chuỗi T. Thuật toán này còn được gọi là FM-index, các ký tự được tìm kiếm ngược từ cuối lên đầu. Vị trí của mỗi kí tự này được xác định trên một đoạn thuộc FC. Gọi $FO(symbol)$ là thứ tự của vị trí đầu tiên mà kí tự $symbol$ xuất hiện trong FC, $CO(symbol, i)$ là số lần kí tự $symbol$ xuất hiện trong BWT từ vị trí 0 đến vị trí thứ i, với i = -1, 0, ..., |T| - 1. Mỗi lần xét một kí tự trong xâu mẫu P, thứ tự tương ứng với các vị trí trên và dưới của kí tự $symbol$ đang tìm kiếm trên FC được cập nhật như sau:

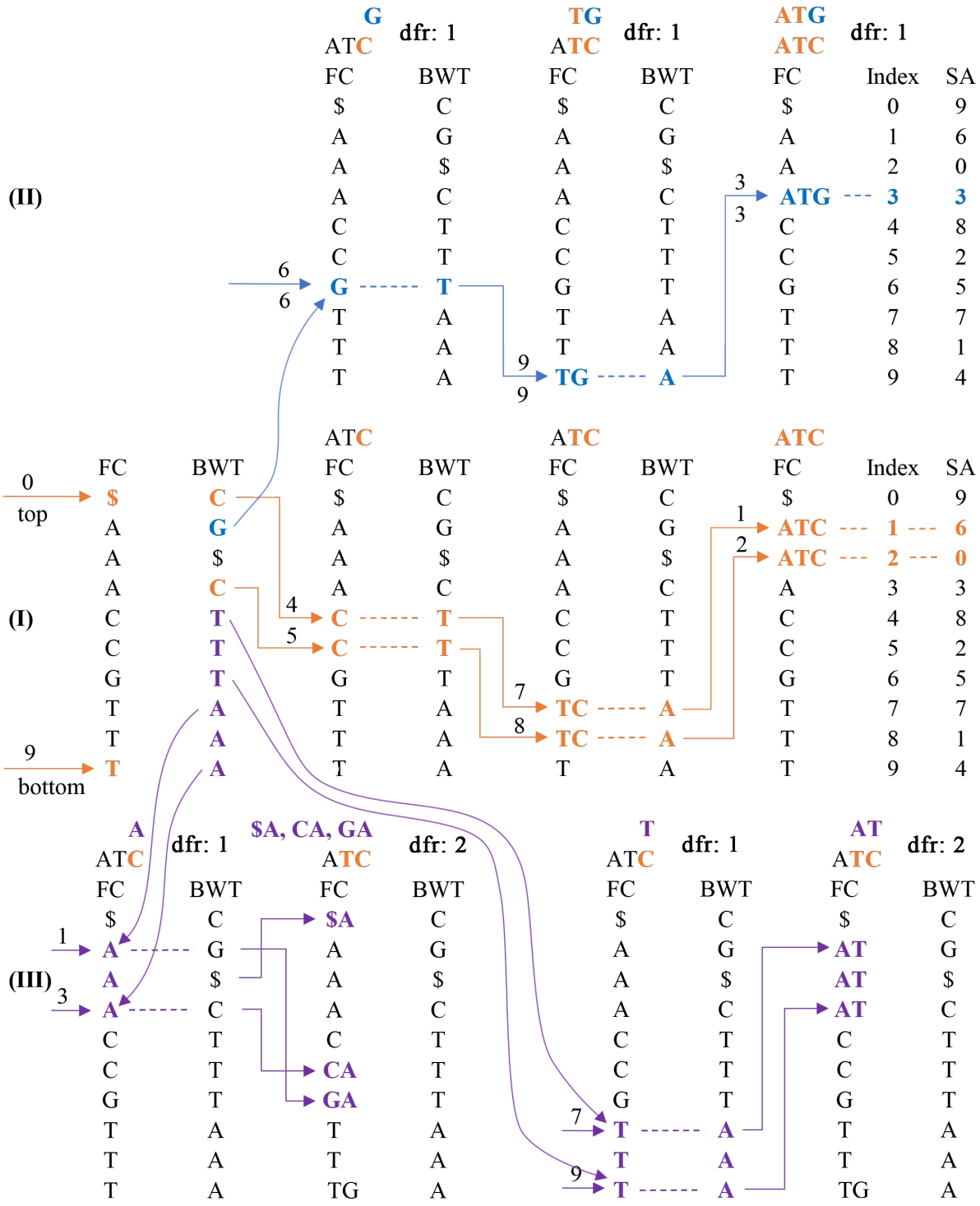
$$top \leftarrow FO(symbol) + CO(symbol, top - 1) \tag{2.1}$$

$$bottom \leftarrow FO(symbol) + CO(symbol, bottom) - 1 \tag{2.2}$$

Các vị trí khớp của mẫu P trên chuỗi T chính bằng giá trị của mảng hậu tố SA theo thứ tự tương ứng với các vị trí trên FC ở bước cuối cùng. Ở đây, hàm $CO(symbol, i)$ có thể được thay thế bằng cách gọi giá trị đếm kí tự *symbol* được lưu sẵn trong *ma trận điểm kiểm tra* để tăng tốc cho thuật toán. Trường hợp sử dụng *mảng hậu tố một phần*, ta tiếp tục quay lui các vị trí trong đoạn [*top*, *bottom*] cho đến khi các vị trí này tồn tại trong mảng hậu tố đó. Khi ấy, giá trị của vị trí khớp bằng giá trị tồn tại trong *mảng hậu tố* cộng thêm số bước quay lui. Ví dụ ta cần khớp chính xác mẫu ATC với chuỗi $ATCATGATC$ theo thứ tự từ cuối lên đầu C, T, A (Xem hình 2.4, (I)). Đoạn bắt đầu được chọn là [0, 9] gồm toàn bộ chiều dài của FC. Các ký tự C thuộc FC xuất hiện trong đoạn [0, 3] trên BWT. Để cập nhật đoạn [*top*, *bottom*] mới trên FC của bước sau theo công thức 2.1 và 2.2, ta cần tính $FO(C)$, $CO(C, -1)$ và $CO(C, 3)$. Các giá trị này lần lượt là 4, 0, và 2. Từ đó, đoạn mới của ký tự C được cập nhật trên FC là [4, 5]. Tương tự, các đoạn [*top*, *bottom*] được cập nhật tương ứng với các ký tự T và A là [7, 8] và [1, 2]. Đoạn cuối cùng [1, 2] nằm trên BWT cũng là các vị trí khớp chính xác của mẫu ATC với chuỗi đã cho. Sau khi đối chiếu với các giá trị trong *mảng hậu tố* ta được các vị trí khớp chính xác của hai chuỗi là 0 và 6.

2.1.2.2 Thuật toán khớp xấp xỉ

Đối với thuật toán tìm mẫu khớp chính xác, các kí tự thuộc mẫu P đều phải giống các kí tự trên chuỗi trình tự T theo thứ tự. Ngược lại, các kí tự thuộc mẫu khớp xấp xỉ có thể khác các kí tự trên chuỗi đã cho theo trình tự, miễn là số lượng ký tự không giống này nhỏ hơn một ngưỡng khác biệt cho phép (Khác biệt ở đây có thể là một không khớp (mismatch) hoặc một khoảng trống (gap). Khoảng trống với ý nghĩa là các Insertion, hoặc các Deletion liên tiếp, hay còn gọi là Indels). Mặc dù vậy, thuật toán khớp chính xác vẫn được áp dụng cho việc tìm các mẫu khớp xấp xỉ. Vẫn với ví dụ hai chuỗi ATC và $ATCATGATC$, ta tiến hành khớp xấp xỉ với ngưỡng khác biệt là 1. Nếu bắt đầu bằng ký tự C trên BWT, ta có hai chuỗi khớp chính xác như trên, trường hợp này hiển nhiên thỏa mãn (Xem hình 2.4, (I)).



Hình 2.4: Quá trình tìm kiếm lùi mẫu *ATC* trong chuỗi *ATCATGATC*. (I) Mẫu khớp chính xác với chuỗi đã cho tại vị trí 0 và 6. (II) Mẫu khớp xấp xỉ tại vị trí 3 với ngưỡng khác biệt là 1. (III) Mẫu không khớp xấp xỉ vì vượt quá ngưỡng khác biệt cho phép là 1.

Bây giờ ta xét ba ký tự còn lại là G, A, và T. Đối với ký tự G, vì G không giống C nên khác biệt được tính là 1. Tiếp tục cập nhật vị trí của G trên FC (Xem hình 2.4, (II)) ta tìm được đoạn mới [6, 6]. Các ký tự về sau tương ứng với đoạn mới này là T và A giống với các ký tự thứ hai và thứ nhất của mẫu ATC, do đó số khác biệt vẫn là 1. Cuối cùng, ta tìm được vị trí khớp xấp xỉ là 3, trong đó mẫu ATC có ký tự C khác với ký tự G của chuỗi $ATCATGATC$. Đối với các ký tự T và A, đến bước thứ 3 số khác biệt đã là 2, do đó ta không tìm được các vị trí khớp xấp xỉ với ngưỡng khác biệt là 1 (Xem hình 2.4, (III)).

TGCGATAGTA TGCGATAGTA
GC - A - - GT GC - - - AGT

Hình 2.5: Cho điểm dóng hàng theo mô hình phạt khoảng trống Affine khi khớp mẫu $GCAGT$ với chuỗi $TGCGATAGTA$. Dóng hàng bên trái có 2 khoảng mở, trong khi dóng hàng bên phải chỉ có một khoảng mở. Do đó, dóng hàng bên phải có điểm cao hơn vì ta phạt khoảng mở nặng hơn phạt khoảng kéo dài. Dóng hàng bên phải được lựa chọn với chỉ một Indel là ba lần xóa nucleotide liên tiếp.

2.2 Thuật toán Smith-Waterman

Thuật toán dóng hàng đa trình tự dựa trên thuật toán Smith-Waterman có thể sử dụng để dóng hàng toàn bộ hoặc dóng hàng địa phương các trình tự. Tuy nhiên, trong trường hợp hai trình tự có độ dài chênh lệch lớn, dóng hàng địa phương thể hiện sự liên quan đến sinh học tốt hơn. Ta vẫn áp dụng cách cho điểm Indels theo mô hình phạt khoảng trống Affine như phần **??**. Ngoài ra, các ma trận điểm còn được sử dụng để cho điểm các khớp và không khớp ví dụ như PAM250, BLOSUM62. Trước tiên, ta xây dựng thuật toán đệ quy dóng hàng hai trình tự dựa trên đồ thị Manhattan ba cấp, sau đó là các phương pháp cải tiến thuật toán. Cuối cùng, thuật toán tham lam được thêm vào để giải quyết bài toán dóng hàng đa trình tự.

2.3 Thực nghiệm thuật toán

2.3.1 Thuật toán song song với Golang

Với lý thuyết về dóng hàng dựa trên chuyển dạng Burrows-Wheeler ở phần 2.1, chúng tôi đã triển khai thuật toán bằng ngôn ngữ Go (Golang), một ngôn ngữ hỗ trợ mạnh cho việc tính toán song song và đồng thời. Mục đích của thực nghiệm thuật toán là để hiểu rõ hơn về thuật toán cũng như cách thiết lập các tham số cần thiết lập ban đầu.

Golang là ngôn ngữ lập trình hỗ trợ tự động khai thác sự hoạt động của các nhân (core) máy tính mà không phụ thuộc vào việc cấp phát của hệ điều hành. Các threads trên các nhân được triển khai bởi các goroutine. Chúng có thể làm việc cùng nhau tại chính xác một thời điểm hoặc đợi nhau trong một hàng đợi. Mặt khác, các goroutine hoạt động trên nguyên tắc không chia sẻ biến nên giao tiếp của chúng được đồng bộ hóa bằng các kênh đệm (buffered channel) hoặc các kênh không đệm (unbuffered channel). Ngoài ra, ta có thể khởi tạo số lượng goroutine lớn hơn hoặc bằng số các logical processor của máy tính. Khi chúng bằng nhau thời gian thực thi bằng thời gian chạy của goroutine chậm nhất. Ngược lại, khi số lượng các goroutine lớn hơn, chúng được đồng bộ hóa để tối ưu thời gian và đảm bảo không có nhân nào nhàn rỗi.

2.3.1.1 Tham số và đầu vào

Chuyển dạng Burrows-Wheeler, mảng hậu tố, và *ma trận điểm kiểm tra* được tính trước khi thực hiện thuật toán. Ta thay đổi khoảng cách c giữa cấc giá trị trong *mảng hậu tố* và khoảng cách k giữa thứ tự trong *ma trận điểm kiểm tra* để đánh giá mức độ thay đổi giữa thời gian chạy và bộ nhớ được sử dụng. Tiếp theo, ta chạy thuật toán với các giá trị khác biệt và so sánh kết quả với kết quả thu được từ công cụ BWA-MEM. Các kết quả giống nhau khi các tham số của công cụ được thiết lập như sau:

- T (Không xuất các dóng hàng với điểm thấp hơn một số nguyên cho trước) = 0.
- k (các khớp ngắn hơn một số nguyên cho trước được loại bỏ) = 0.

2.3.1.2 Kết quả

Thuật toán được triển khai trên máy ảo của nền tảng Google Cloud với cấu hình 8 vCPUs và 52 GB bộ nhớ. Với *mảng hậu tố một phần*, không có sự khác biệt đáng kể về thời gian chạy và không gian làm việc khi các giá trị c và k khác nhau. Với *ma trận điểm kiểm tra*, thời gian chạy đo được chậm hơn 104.5 lần và bộ nhớ yêu cầu lớn hơn 6.5 lần khi c và k bằng 1 so với khi chúng cùng bằng 100 (Xem bảng **??**). Tuy nhiên, ta chỉ cần chạy thuật toán một lần để tìm *ma trận điểm kiểm tra*, chúng được lưu lại để tái sử dụng nhiều lần.

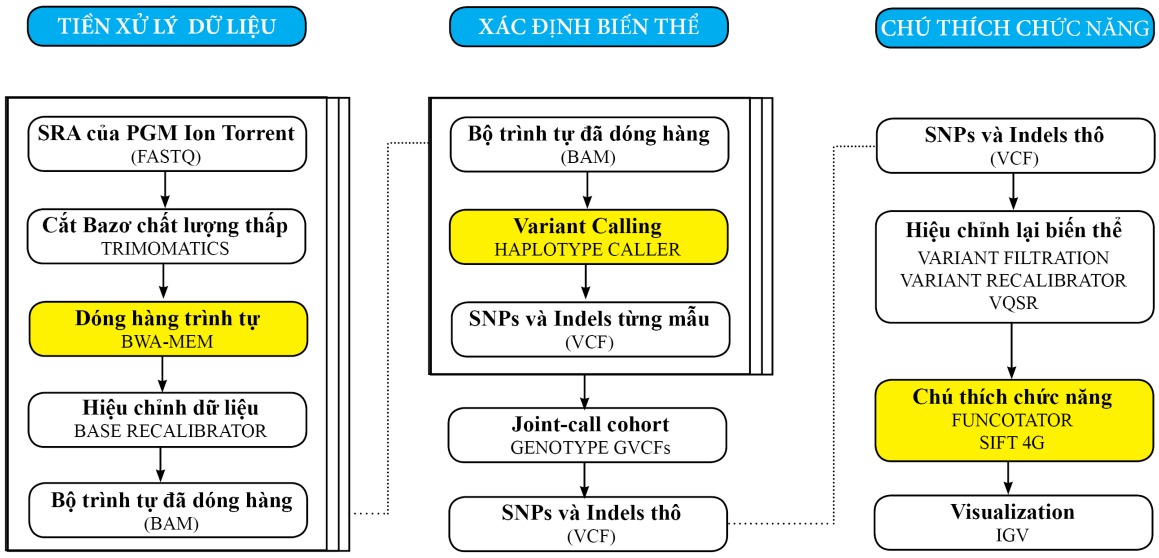
Với ngưỡng khác biệt là 3, ta tìm được 367,946 trình tự khớp với bộ gen tham chiếu (Xem bảng **??**). Các trình tự khớp có thể gồm cả chuỗi thuận và chuỗi nghịch được gán nhãn 0 và 16 tương ứng. Sau khi dóng hàng, ta có thể gọi ra các nucleotide khớp và không khớp với bộ gen tham chiếu tại các vị trí khác nhau (Xem bảng **??**).

CHƯƠNG 3

ỨNG DỤNG THUẬT TOÁN TRONG DỰ ĐOÁN BIẾN THỂ GEN

Ở chương này, ta áp dụng thuật toán dóng hàng trình tự dựa trên chuyển dạng Burrows-Wheeler và Smith-Waterman để xác định các biến thể gen, đồng thời kết hợp với các phương pháp bổ xung để xử lý dữ liệu và tìm ra sự ảnh hưởng của các biến thể đến chức năng của protein. Phạm vi tìm kiếm được trình bày chi tiết hơn ở phần **??**. Quy trình làm việc được chia ra thành 3 giai đoạn: tiền xử lý dữ liệu, xác định biến thể, và chú thích chức năng (Xem hình 3.1). Chú ý rằng, thuật toán dựa trên chuyển dạng Burrows-Wheeler được sử dụng trong phần mềm BWA-MEM dùng để dóng hàng bộ dữ liệu được xuất ra từ máy giải trình tự với bộ gen tham chiếu. Từ đó ta có được bộ trình tự với các tọa độ khớp và các vị trí biến thể ban đầu. Mặt khác, thuật toán Smith-Waterman được sử dụng trong phần mềm Haplotype Caller ở bước dóng hàng lại mỗi Haplotype với Haplotype tham chiếu. Ngoài ra, thuật toán này còn được tích hợp trong phần mềm SIFT 4G ở bước tìm trình tự tương đồng dựa trên cơ sở dữ liệu lớn về protein [Vas+16]. Các biến thể cuối cùng được hiển thị thông qua phần mềm Integrative Genomics Viewer (IGV) [TRM13].

Sau quá trình lọc biến thể ta thu được tập các biến thể sử dụng cho

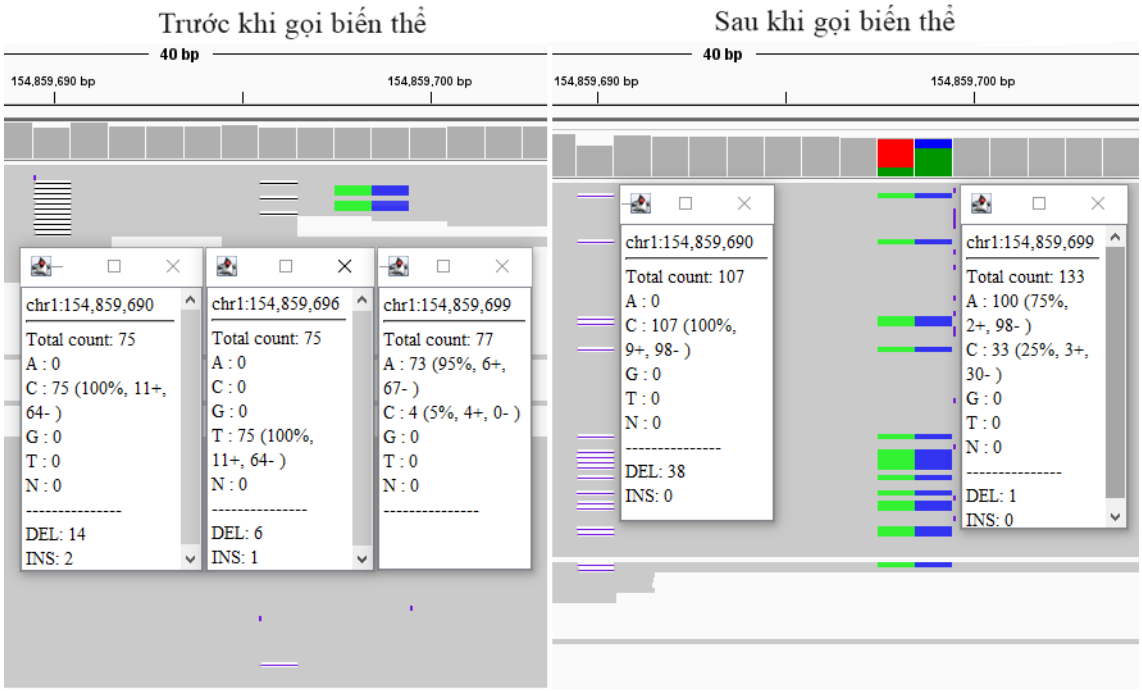


Hình 3.1: Quy trình làm việc xác định ảnh hưởng của biến thể gen gồm ba giai đoạn. **Giai đoạn 1: tiền xử lý dữ liệu**. Thuật toán dóng hàng trình tự dựa trên chuyển dạng Burrows-Wheeler được sử dụng để khớp bộ trình tự lưu trữ với bộ gen tham chiếu. **Giai đoạn 2: xác định biến thể**. Thuật toán Smith-Waterman được sử dụng để dóng hàng lại các haplotype ở bước gọi biến thể. **Giai đoạn 3: chú thích chức năng**. Lọc ra các biến thể và đánh giá mức độ ảnh hưởng của chúng đến protein, từ đó đưa ra các gen có khả năng bị đột biến.

quá trình đánh giá ảnh mức độ ảnh hưởng đến kiểu hình (Xem hình 3.3). Nhận thấy số lượng SNPs khá nhỏ so với Indels, số lượng biến thể C thành T là lớn nhất với 105563 vị trí. Mặt khác, tỷ lệ Ts/Tv (transition-to-transversion) = 67.7/32.3 = 2.10 phù hợp với tiêu chí đánh giá cho dữ liệu WGS của người. Điều này cũng chứng tỏ tỷ lệ dương tính giả thấp và dữ liệu không bị lệch (bias).

Tập biến thể cuối cùng được chú thích chức năng, ta xác định được 7362 biến thể trên tổng số 5059 gen có thể làm thay đổi chức năng của protein. Một số gen đột biến được tìm thấy trùng hợp với kết quả của các nghiên cứu trước đó.

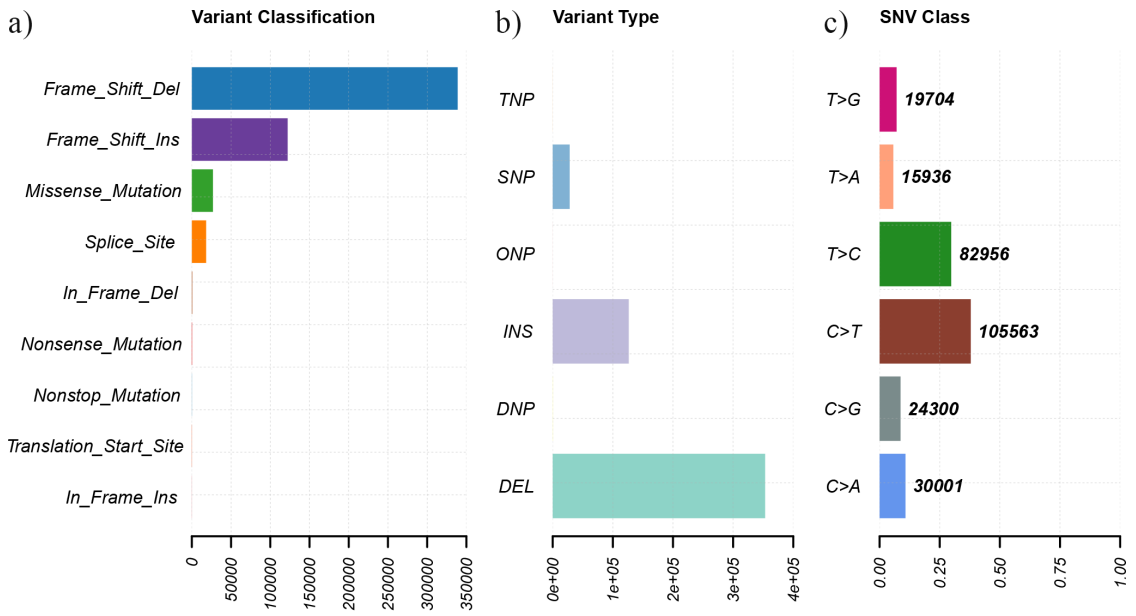
Trong số các biến thể gen đã được tìm thấy, ta sẽ hiển thị đột biến FMN1 trên mẫu SRR5344691 để xem xét một cách trực quan kết quả



Hình 3.2: Dóng hàng trước và sau khi gọi biến thể của mẫu SRR5344691 tại các vị trí trên nhiễm sắc thể số 1. Ở tọa độ 15485969, từ 14 xóa và 2 chèn trở thành 38 xóa. Tại tọa độ 154859696 các xóa và chèn được loại bỏ. Tại tọa độ 154859699, tần số nucleotide thay đổi cao hơn từ 5% C tới 25% C.

dóng hàng tìm biến thể (Xem hình **??**).

- 1: Biến thể nằm trên nhiễm sắc thể thứ 15.
- 2: Đột biến được đánh dấu gạch đỏ thuộc nhánh dài q13.2 của nhiễm sắc thể.
- 3: Tại vị trí đột biến, mã hóa của bazơ trên bộ gen tham chiếu là G.
- 4: Trình tự amino acid, bộ ba GGA mã hóa cho amino acid Serine(S).
- 5: Vùng bảo tồn của 99 loài động vật có xương sống và con người. Trong đó chiều cao của khối thể hiện mức độ bảo tồn. Với đột biến FMN1, vị trí của biến thể nằm trong vùng bảo tồn cao.



Hình 3.3: Các biểu đồ thống kê các biến thể sau quá trình lọc biến thể. a) Phân loại biến thể b) Loại biến thể: số lượng SNPs nhỏ nhất so với thêm và xóa c) Các biến thể nucleotides đơn: số lượng biến thể C thành T lớn nhất tại 105536 vị trí.

- 6: Các trình tự sau khi được dóng hàng đều có vị trí 32964177 là nucleotide C, bazơ này khác với G của bộ gen tham chiếu. Tại đây bộ ba GGA mã hóa amino acid Serine (S) chuyển thành GCA mã hóa cho Cysteine (C) làm thay đổi chức năng của protein Formin-1.

Trong các bảng thống kê đột biến gen sau đây, cột Chr là tên nhiễm sắc thể, Pos là vị trí của biến thể, cột Allele chứa sự thay đổi các Nucleotide, cột gen hiển thị tên gen, cột Amino thống kê các amino acid thay đổi, cột Score chứa điểm được tính bởi công cụ SIFT4G (từ 0 đến 0.05 với các biến thể có ảnh hưởng làm thay đổi protein), cột dbSNP cho biết các đột biến gen đã có trong cơ sở dữ liệu hay là những đột biến mới. Kết quả các gen trùng hợp được thể hiện bằng tỷ lệ tổng số gen đột biến giống nhau trên số lượng gen đột biến của các nghiên cứu có sẵn. Đối với các nghiên cứu cùng phương pháp tỷ lệ gen đột biến được tìm thấy là 3/4 trong công bố của Giacopuzzi et al. (2017) [Gia+17] và 4/7 trong công bố của Nishioka et al. (2018) [Nis+18] (Xem bảng 3.1). Đối với nghiên cứu biểu hiện gen của Tom Walsh et al. (2008) [Wal+08] ta có 14 trên 34 gen (Xem bảng **??**). Mặt khác, ở nghiên cứu về GWAS năm 2019, Laura m. Huckins và các cộng sự [Huc+19] đã công bố 70 gen đột biến trong số 413 gen đột biến được tìm thấy, trong đó có 17 gen có trong kết quả của luận văn (Xem bảng **??**). Toàn bộ các gen tìm được còn được đối chiếu với dữ liệu trên UniProtKB [1] bao gồm các gen đột biến đã được thí nghiệm, ta có 11/57 gen giống nhau (Xem bảng **??**). Ngoài ra, ta còn có 9/16 gen xuất hiện trong nghiên cứu bằng phương pháp DNMs của Daniel P.Howrigan et al. (2020) [How+20] (Xem bảng **??**). Số lượng đột biến cũng có thể được thu gọn nếu chúng ta giới hạn các kết quả bằng các tiêu chí khác như tần số allele, đường chuyển hóa, hay các biến thể mới chưa có trong cơ sở dữ liệu.

[1] https://www.uniprot.org/

Bảng 3.1: Những gen giống với các nghiên cứu cùng phương pháp giải trình tự exome. Tìm được 3 trong số 4 gen được Giacopuzzi công bố năm 2017, 4 trong số 7 gen thuộc nghiên cứu của Nishioka năm 2018.

Chr	**Pos**	**Allele**	**Gen**	**Amino**	**Score**	**dbSNP**
Giacopuzzi et al. (2017) - WES (3/4) [Gia+17]						
12	5854114	G-A	ANO2	R-W	0	rs767675843
15	32964177	G-C	FMN1	S-C	0.002	rs762291357
15	33067169	C-T	FMN1	G-E	0.039	rs11072170
19	42326316	G-C	MEGF8	A-P	0.049	novel (l)
19	42336114	G-A	MEGF8	A-T	0	novel (l)
19	42356089	G-A	MEGF8	G-S	0	novel (l)
19	42368605	G-T	MEGF8	G-W	0.032	novel (l)
19	42376182	T-C	MEGF8	F-L	0	novel (l)
Nishioka et al. (2018) - WES (4/7) [Nis+18]						
7	1.06E+08	G-A	CDHR3	V-M	0	rs35008315 (l)
7	1.06E+08	G-C	CDHR3	Q-H	0.025	rs34426483 (l)
11	73235159	C-T	P2RY2	R-C	0.004	rs1626154
12	21887888	C-T	ABCC9	D-N	0.008	rs757681761
12	78175348	G-A	NAV3	S-N	0.001	novel
12	78200517	A-G	NAV3	R-G	0.002	novel

KẾT LUẬN

Trong luận văn này, sau phần tổng hợp và trình bày một số kiến thức cơ sở cần thiết liên quan tới tin sinh học, thuật toán dóng hàng dựa trên chuyển dạng Burrows-Wheeler được tìm hiểu kĩ và trình bày một cách tường minh hơn so với các bài báo gốc. Ngôn ngữ Go với kỹ thuật song song và đồng thời được chọn để thử nghiệm thuật toán, phát huy tối đa những điểm mạnh khi làm việc với dữ liệu lớn. Kết quả nhận được trong phần thử nghiệm thuật toán giống với kết quả nhận được từ công cụ BWA-MEM với cùng dữ liệu đầu vào.

Thuật toán dóng hàng trình tự dựa trên chuyển dạng Burrows-Wheeler và thuật toán Smith-Waterman được áp dụng trong bài toán dự đoán biến thể gen đã ráp lại các trình tự lưu trữ một cách chính xác. Từ dữ liệu dóng hàng ta thu được tập các SNPs và Indels với tỉ lệ Ts/Tv phù hợp cho thấy độ tin cậy của chương trình. Đồng thời các dữ liệu này cũng có thể sử dụng để thực hiện phân tích tiếp theo cho dự đoán các biến thể. Theo phạm vi tìm kiếm rộng được xác định từ đầu, một số lượng lớn các biến thể trên các gen đã được tìm thấy, trong đó có nhiều đột biến gen giống với các công bố trước đây. Dựa trên các tiêu chí khác nhau mà chúng ta có thể đưa ra các tập biến thể nhỏ hơn cho các công việc phân tích và kiểm chứng bằng thực nghiệm.

Về mặt phương pháp, có thể giảm hơn nữa các dương tính giả nếu thực hiện thêm bước khử trùng lặp cho dữ liệu Ion Torrent. Mặt khác, phạm vi biến thể có thể được thu hẹp dựa vào các tiêu chí xác định trước. Ngoài ra, giải trình tự gen để xác định biến thể có thể được

kết hợp với các phương khác như: phân tích liên kết, biểu hiện gen, GWAS... để cho ra kết quả chính xác hơn. Tuy vậy, các công cụ, thuật toán chỉ giúp phân tích, khoanh vùng một tập hợp có khả năng cao các biến thể gen. Để đánh giá kết quả một cách chắc chắn ta vẫn cần thực hiện các thí nghiệm sinh hóa.

Chúng tôi cho rằng các phương pháp được nghiên cứu trong luận văn có tính áp dụng thực tiễn cao. Qua những kĩ thuật này, kết quả thu được có thể làm cơ sở cho việc dự đoán các biến thể gen di truyền và không di truyền ở người, động vật, thực vật cũng như tìm kiếm các gen tương đồng trên người và động vật. Những dự đoán này rất có ý nghĩa trong hỗ trợ cho các nghiên cứu thực nghiệm, giúp tăng tính khả thi cũng như hiệu quả của các thực nghiệm. Xa hơn nữa, việc phân tích dữ liệu giải trình tự cũng giúp tìm ra nguyên nhân gây bệnh, dự đoán khả năng mắc bệnh do di truyền trong một phả hệ, hoặc áp dụng trong hỗ trợ điều trị bệnh sử dụng trình tự nhắm mục tiêu. Từ đây, những dịch vụ xác định gen tiềm năng, tầm soát ung thư và các bệnh di truyền có thể được triển khai trong các hệ thống bệnh viện và y tế dự phòng.

Tài liệu tham khảo

[ADL08] Altshuler, D., Daly, M. J., and Lander, E. S. "Genetic Mapping in Human Disease". In: *Science* vol. 322, no. 5903 (Nov. 7, 2008), pp. 881–888. pmid: 18988837.

[And90] Anderson, W. F. "September 14, 1990: The Beginning". In: *Human Gene Therapy* vol. 1, no. 4 (Dec. 1, 1990), pp. 371–372.

[BW94] Burrows, M. and Wheeler, D. J. *A Block-Sorting Lossless Data Compression Algorithm*. 1994.

[Com15] Compeau, P. *BIOINFORMATICS ALGORITHMS, VOL.I*. 2nd Edition. La Jolla, CA: Active Learning Publishers, Jan. 1, 2015. 384 pp.

[Cri70] Crick, F. "Central Dogma of Molecular Biology". In: *Nature* vol. 227, no. 5258 (5258 Aug. 1970), pp. 561–563.

[FM05] Ferragina, P. and Manzini, G. "Indexing Compressed Text". In: *Journal of the ACM* vol. 52, no. 4 (July 1, 2005), pp. 552–581.

[Gia+17] Giacopuzzi, E. et al. "Exome Sequencing in Schizophrenic Patients with High Levels of Homozygosity Identifies Novel and Extremely Rare Mutations in the GABA/Glutamatergic Pathways". In: *PLOS ONE* vol. 12, no. 8 (Aug. 7, 2017), e0182778.

[Gus89] Gusella, J. F. "Location Cloning Strategy for Characterizing Genetic Defects in Huntington's Disease and Alzheimer's Disease". In: *The FASEB Journal* vol. 3, no. 9 (1989), pp. 2036–2041.

[Hon+07] Hon, W.-K., Lam, T.-W., Sadakane, K., Sung, W.-K., and Yiu, S.-M. "A Space and Time Efficient Algorithm for Constructing Compressed Suffix Arrays". In: *Algorithmica* vol. 48, no. 1 (May 1, 2007), pp. 23–36.

[How+20] Howrigan, D. P. et al. "Exome Sequencing in Schizophrenia-Affected Parent–Offspring Trios Reveals Risk Conferred by Protein-Coding de Novo Mutations". In: *Nature Neuroscience* vol. 23, no. 2 (2 Feb. 2020), pp. 185–193.

[Huc+19] Huckins, L. M. et al. "Gene Expression Imputation across Multiple Brain Regions Provides Insights into Schizophrenia Risk". In: *Nature Genetics* vol. 51, no. 4 (4 Apr. 2019), pp. 659–674.

[KB13] Kilpinen, H. and Barrett, J. C. "How Next-Generation Sequencing Is Transforming Complex Disease Genetics". In: *Trends in Genetics* vol. 29, no. 1 (Jan. 1, 2013), pp. 23–30.

[Kha+15] Khafizov, K., Ivanov, M. V., Glazova, O. V., and Kovalenko, S. P. "Computational Approaches to Study the Effects of Small Genomic Variations". In: *Journal of Molecular Modeling* vol. 21, no. 10 (Sept. 8, 2015), p. 251.

[Lod+07] Lodish, H., Berk, A., Kaiser, C. A., Krieger, M., Scott, M. P., Bretscher, A., Ploegh, H., and Matsudaira, P. *Molecular Cell Biology*. 6th edition. New York: W. H. Freeman, June 15, 2007. 973 pp.

[MM93] Manber, U. and Myers, G. "Suffix Arrays: A New Method for On-Line String Searches". In: *SIAM Journal on Computing* vol. 22, no. 5 (Oct. 1, 1993), pp. 935–948.

[Nis+18] Nishioka, M. et al. "Identification of Somatic Mutations in Monozygotic Twins Discordant for Psychiatric Disorders". In: *npj Schizophrenia* vol. 4, no. 1 (1 Apr. 13, 2018), pp. 1–7.

[OS09] Okanohara, D. and Sadakane, K. *A Linear-Time Burrows-Wheeler Transform Using Induced Sorting*. Vol. 5721. Aug. 25, 2009, p. 101. 90 pp.

[Per+18] Pertea, M., Shumate, A., Pertea, G., Varabyou, A., Chang, Y.-C., Madugundu, A. K., Pandey, A., and Salzberg, S. L. "Thousands of Large-Scale RNA Sequencing Experiments Yield a Comprehensive New Human Gene List and Reveal Extensive Transcriptional Noise". In: *bioRxiv* (May 29, 2018), p. 332825.

[Pev15] Pevsner, J. *Bioinformatics and Functional Genomics*. 3rd Edition. Chichester, West Sussex, UK ; Hoboken, New Jersey: Wiley-Blackwell, Oct. 26, 2015. 1160 pp.

[Sch08] Schuster, S. C. "Next-Generation Sequencing Transforms Today's Biology". In: *Nature Methods* vol. 5, no. 1 (1 Jan. 2008), pp. 16–18.

[TRM13] Thorvaldsdóttir, H., Robinson, J. T., and Mesirov, J. P. "Integrative Genomics Viewer (IGV): High-Performance Genomics Data Visualization and Exploration". In: *Briefings in Bioinformatics* vol. 14, no. 2 (Mar. 1, 2013), pp. 178–192.

[Vas+16] Vaser, R., Adusumalli, S., Leng, S. N., Sikic, M., and Ng, P. C. "SIFT Missense Predictions for Genomes". In: *Nature Protocols* vol. 11, no. 1 (1 Jan. 2016), pp. 1–9.

[Wal+08] Walsh, T. et al. "Rare Structural Variants Disrupt Multiple Genes in Neurodevelopmental Pathways in Schizophrenia". In: *Science* vol. 320, no. 5875 (Apr. 25, 2008), pp. 539–543. pmid: 18369103.

Appendices

Mục từ tra cứu