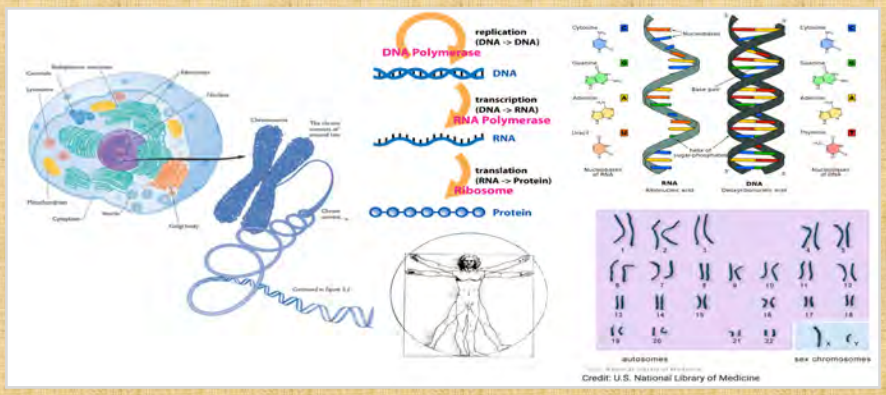
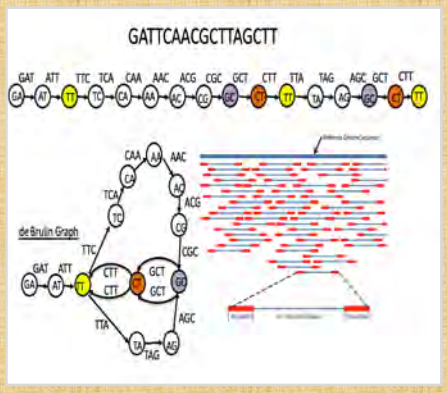
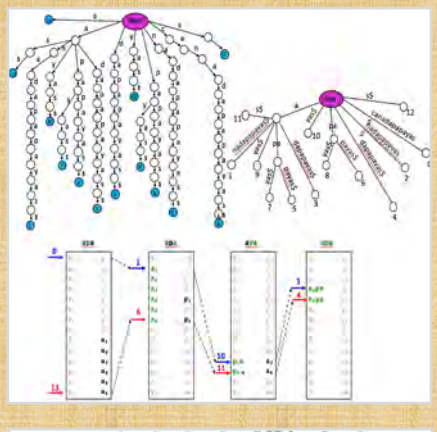
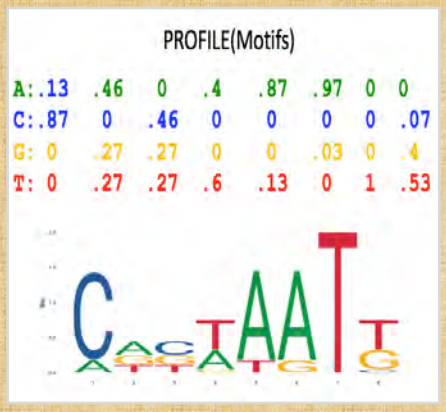
ชีวสารสนเทศ 1

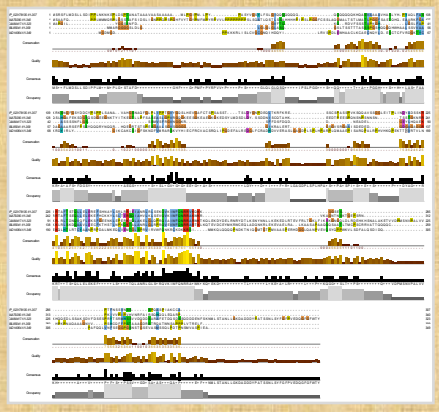
แนวทางอัลกอริทึม

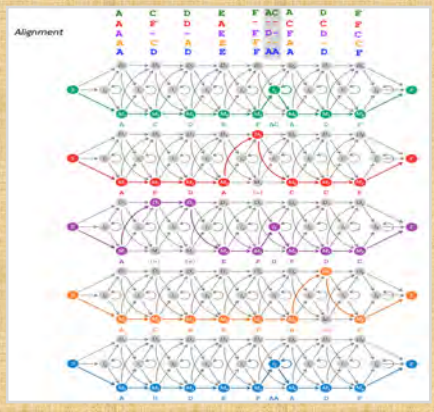


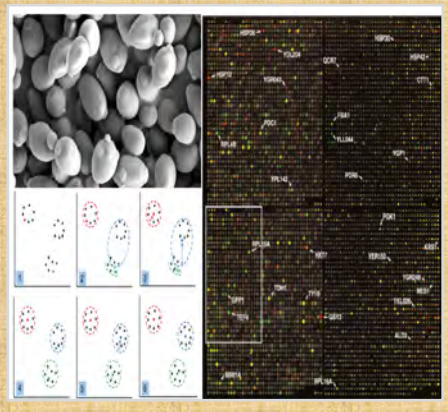


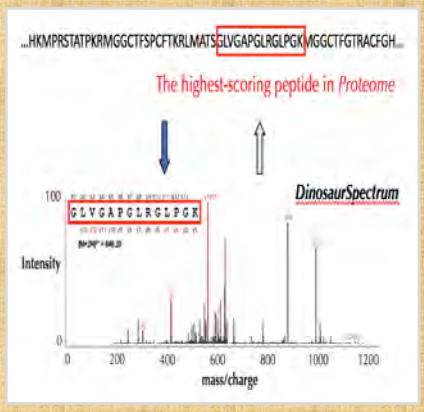












ดวงดาว วิชาดากุล

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

ชีวสารสนเทศ 1

แนวทางอัลกอริทึม

ดวงดาว วิชาดากุล
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

ดวงดาว วิชาดากุล

ชีวสารสนเทศ 1 แนวทางอัลกอริทึม / ดวงดาว วิชาดากุล

1. ชีวสารสนเทศ
2. Bioinformatics

พิมพ์ครั้งที่ 2 จำนวน 10 เล่ม พ.ศ. 2564

จัดพิมพ์โดย
ดวงดาว วิชาดากุล
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย
พญาไท กรุงเทพฯ 10330

ออกแบบปก : ดวงดาว วิชาดากุล
ออกแบบรูปเล่ม : ดวงดาว วิชาดากุล

พิมพ์ที่ ศูนย์ถ่ายเอกสาร เมตตาปริ้นติ้ง ศาลายา โทรศัพท์ 089 508 3756
135/330 หมู่ 6 ถนน พุทธมณฑลสาย 4 ตำบลศาลายา อำเภอพุทธมณฑล นครปฐม 73170

คำนำ

ตำราเรื่อง ชีวสารสนเทศ 1 แนวทางอัลกอริทึม จัดทำขึ้นเพื่อเผยแพร่ความรู้ให้กับนิสิต นักศึกษา และบุคคลทั่วไปที่สนใจศาสตร์ในสหสาขาวิชาชีวสารสนเทศ ซึ่งเกี่ยวข้องกับการประยุกต์ใช้องค์ความรู้ในสาขาวิชาวิศวกรรมคอมพิวเตอร์ วิทยาศาสตร์คอมพิวเตอร์ คณิตศาสตร์ และสถิติ ในการแก้ปัญหาในเชิงอัลกอริทึมกับโจทย์ทางชีววิทยาและอณูชีววิทยาทั้งในทางการแพทย์และเทคโนโลยีชีวภาพ โดยตำรานี้ใช้ประกอบการเรียนการสอนในรายวิชา 2110495 Advanced Topics in Computer Engineering I (Bioinformatics I) หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เนื้อหาในตำราประกอบด้วยความรู้เกี่ยวกับอณูชีววิทยาพื้นฐาน เช่น ดีเอ็นเอ อาร์เอ็นเอ โปรตีน ยีน จีโนม ความเชื่อตามหลักชีววิทยาระดับโมเลกุล (central dogma of molecular biology) เทคโนโลยีโอมิกส์ และการหาลำดับเบส ที่จำเป็นต่อการทำความเข้าใจข้อมูลในรูปแบบที่หลากหลายและสามารถนำมาวิเคราะห์หรือสร้างอัลกอริทึมเพื่อวิเคราะห์โดยกระบวนการทางคอมพิวเตอร์ได้ องค์ความรู้ในตำรานี้เกิดจากการเรียบเรียงและรวบรวมจากแหล่งความรู้ต่างๆ โดยเฉพาะหนังสือ *Bioinformatics algorithms: an active learning approach* 2nd edition 2 เล่มในชุดหนังสือของ Compeau, P.E.C, & Pevzner, P.A. (2015) ตัวอย่างโจทย์ทางอัลกอริทึมจากโรซาลินด์ (http://rosalind.info) ตัวอย่างโจทย์วิจัยร่วมสมัยและผลงานวิจัยที่ตีพิมพ์ในวารสารวิชาการนานาชาติ คอร์สออนไลน์แบบสั้นที่เน้นการให้ความรู้เกี่ยวกับเทคโนโลยีที่เกี่ยวข้องจาก EMBL-EBI Train Online (https://www.ebi.ac.uk/training/on-demand) รวมทั้งตัวอย่างผลงานวิจัยและหัวข้อวิจัยที่ทางผู้จัดทำดำเนินการอยู่และที่เสร็จสิ้นแล้ว โดยหวังเป็นอย่างยิ่งว่าตำราเล่มนี้จะเป็นประโยชน์ต่อนิสิต นักศึกษา และบุคคลทั่วไปที่มีความสนใจศาสตร์ในสาขานี้

อ.ดร. ดวงดาว วิชาดากุล
ผู้จัดทำ

สารบัญ

สารบัญรูปภาพ

บทที่ 1 ทำความรู้จักชีวสารสนเทศ

วัตถุประสงค์

- เพื่อให้นิสิตเห็นที่มา เข้าใจความหมาย ความสำคัญ และองค์ความรู้ที่เกี่ยวข้องกับชีวสารสนเทศ
- เพื่อปูพื้นความรู้ทางอณูชีววิทยาและเทคโนโลยีที่เกี่ยวข้อง ที่จำเป็นต่อความเข้าใจโจทย์ทางชีววิทยา ชีวการแพทย์ และเทคโนโลยีชีวภาพ
- เพื่อให้นิสิตได้เห็นตัวอย่างโจทย์ทางชีววิทยา ชีวการแพทย์ และเทคโนโลยีชีวภาพ รวมทั้งแนวทางในการแก้ปัญหาโจทย์เหล่านี้จากมุมมองของวิศวกรรมและวิทยาศาสตร์คอมพิวเตอร์ คณิตศาสตร์ และสถิติ
- เพื่อให้นิสิตได้ทำความรู้จักกับฐานข้อมูลสาธารณะ ตัวอย่างข้อมูลทางชีวสารสนเทศ รูปแบบการเข้าถึงข้อมูล และการประมวลผลข้อมูลพื้นฐาน

ผลลัพธ์ที่คาดหวัง

- นิสิตสามารถอธิบายที่มา ความหมาย ความสำคัญ และองค์ความรู้ที่เกี่ยวข้องกับชีวสารสนเทศได้
- นิสิตสามารถอธิบายความเชื่อทางชีววิทยาระดับโมเลกุล ศัพท์พื้นฐานทางชีววิทยา และอณูชีววิทยา เช่น จีโนม โครโมโซม ยีน ดีเอ็นเอ นิวคลีโอไทด์ อาร์เอ็นเอ โปรตีน กรดแอมิโน โคดอน และศัพท์เทคนิคที่เกี่ยวข้อง เช่น เทคโนโลยีโอมิกส์ จีโนมิกส์ ทรานสคริปโทมิกส์ โพรทีโอมิกส์ เมตาโบโลมิกส์
- นิสิตสามารถยกตัวอย่างโจทย์ทางอณูชีววิทยา ชีวการแพทย์ และเทคโนโลยีชีวภาพ รวมทั้งแนวทางในการแก้ปัญหาโจทย์เหล่านี้จากมุมมองของวิศวกรรมและวิทยาศาสตร์คอมพิวเตอร์ คณิตศาสตร์ และสถิติ
- นิสิตสามารถเขียนโปรแกรมเพื่อประมวลผลตัวอย่างข้อมูลทางชีวสารสนเทศได้

เนื้อหาโดยสรุป

แนะนำเนื้อหาวิชาโดยเริ่มจากคำอธิบายและความสำคัญของจีโนมิกส์ (genomics) การหาลำดับเบสจีโนมมนุษย์ ขนาดของข้อมูลจีโนมมนุษย์ การหาลำดับเบสในระดับจีโนมกับการวินิจฉัยและรักษาโรค ตัวอย่างเทคโนโลยีที่เกี่ยวข้องกับการหาลำดับเบสจีโนม งานวิจัยที่เกี่ยวข้องในเชิงชีวสารสนเทศ เช่น อัลกอริทึมที่ใช้ในการประกอบร่างจีโนม (genome assembly) การทำนายตำแหน่งของยีนในจีโนม (gene prediction) การหาโมทิฟควบคุม (regulatory motif finding) การหาลำดับนิวคลีโอไทด์ที่มีความเหมือนหรือคล้ายคลึงกัน เป็นต้น ตัวอย่างโปรแกรมหรือเครื่องมือทางชีวสารสนเทศที่มีการใช้งานกันอย่างกว้างขวาง เช่น โปรแกรม BLAST [1] งานวิจัยที่เกี่ยวข้อง เช่น การหาการแปรผันของลำดับเบส (variation) ในจีโนม และการอนุมานการเกิดโรค โครงการ 1,000

จีโนม (http://www.internationalgenome.org) โครงการ 100,000 จีโนม ของประเทศอังกฤษ (UK) (https://www.genomicsengland.co.uk/about-genomics-england/the-100000-genomes-project/) และโครงการ 100K Genome Asia (http://www.genomeasia100k.com) หลักการพื้นฐานทางชีววิทยาและอณูชีววิทยา จีโนม โครโมโซม ยีน ดีเอ็นเอ อาร์เอ็นเอ โปรตีน ความเชื่อตามหลักชีววิทยาระดับโมเลกุล (central dogma of molecular biology) การเข้ารหัส การถอดรหัส และการแปลรหัสดีเอ็นเอ อาร์เอ็นเอ และโปรตีน เทคโนโลยีโอมิกส์ (omics) อื่นๆ นอกเหนือจากจีโนมิกส์ ข้อมูลมหัต (big data) กับจีโนมิกส์ สถาปัตยกรรมคลาวด์ที่สนับสนุนการประมวลผลข้อมูลชีวสารสนเทศ งานวิจัยในเชิงชีวสารสนเทศ การประยุกต์ใช้ชีวสารสนเทศกับการแพทย์ การเกษตร และเทคโนโลยีชีวภาพ

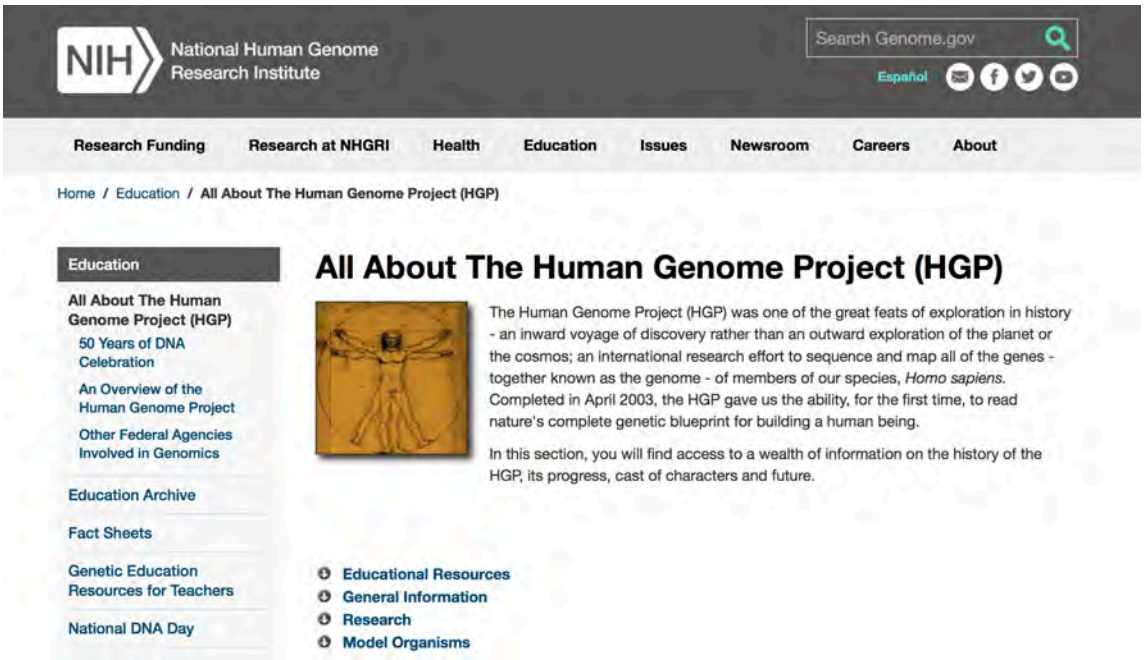
บทที่ 1 ทำความรู้จักกับชีวสารสนเทศ

บทนำเกี่ยวกับจีโนมิกส์และจีโนม

รูปที่ 1.1 ยูทูบแนะนำความหมายของจีโนมิกส์และจีโนม

(*ที่มา*: https://www.youtube.com/watch?v=mmgIClg0Y1k)

คลิปยูทูบในลิงค์รูปที่ 1.1 อธิบายความหมายของจีโนมิกส์ โดยเริ่มต้นอธิบายเกี่ยวกับจีโนม (genome) หรือรหัสพันธุกรรมทั้งหมดของสิ่งมีชีวิตหนึ่งๆ โดยรหัสพันธุกรรมเหล่านี้ถูกเข้ารหัสอยู่ในแต่ละโครโมโซมของเซลล์สิ่งมีชีวิตในรูปแบบของลำดับกรดดีออกซีไรโบนิวคลิอิก (deoxyribonucleic acid) หรือดีเอ็นเอ (DNA) สายเกลียวคู่ (double stranded helix) ซึ่งประกอบด้วยลำดับนิวคลีโอไทด์ (nucleotide) 4 ประเภท คือ อะดีนีน (adenine; A), ไทมีน (thymine; T), ไซโทซีน (cytosine; C) และ กัวนีน (guanine; G) โดยในกรณีของจีโนมมนุษย์ลำดับนิวคลีโอไทด์ที่หาลำดับเบสได้นั้นมีจำนวนประมาณ 3 พันล้านคู่เบส (base pairs) ซึ่งเทียบเท่ากับจำนวนอักษรในดิกชันนารีมาตรฐานถึง 400 เล่มรวมกัน โดยในรหัสพันธุกรรมทั้งหมดนี้จะมีเฉพาะบางบริเวณที่เป็นยีนที่สามารถแปลรหัสต่อไปเป็นโปรตีนที่มีฟังก์ชันการทำงานจำเพาะ เช่น โปรตีนที่เกี่ยวข้องกับกล้ามเนื้อ โปรตีนที่เกี่ยวข้องกับการย่อยอาหาร โดยอัตราส่วนของบริเวณที่เป็นยีนเหล่านี้มีเพียง 2-3% ในจีโนม และอีก 97-98% นั้นมีทั้งส่วนที่เป็นลำดับไม่กำหนดรหัส (noncoding sequence) และบริเวณยีนที่สามารถถอดรหัสไปเป็นอาร์เอ็นไม่กำหนดรหัส (noncoding RNA) โดยอาร์เอ็นเอไม่กำหนดรหัสเหล่านี้ไม่แปลรหัสต่อไปเป็นโปรตีนแต่มีฟังก์ชันการทำงาน โดยสรุป จีโนมิกส์คือการศึกษาจีโนมในมิติต่างๆ เช่น ค้นหาบริเวณที่เป็นยีน บริเวณที่เกิดการซ้ำของลำดับนิวคลีโอไทด์ ความสำคัญของบริเวณเหล่านี้ และฟังก์ชันการทำงานต่างๆ ที่เกี่ยวข้อง

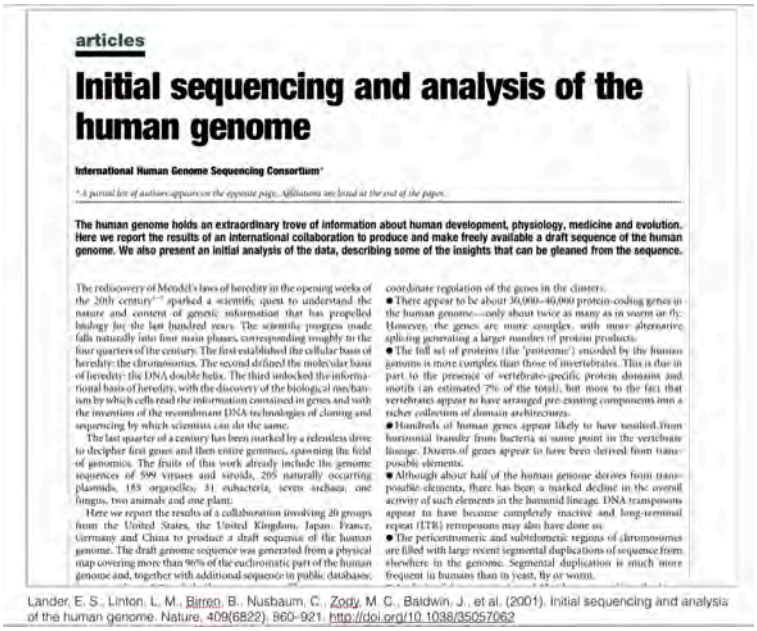


รูปที่ 1.2 โครงการจีโนมมนุษย์

(*ที่มา*: https://www.genome.gov/10001772/all-about-the--human-genome-project-hgp/)

โครงการจีโนมมนุษย์ (Human Genome Project: HGP) (รูปที่ 1.2) เป็นความร่วมมือของหน่วยงานวิจัยและศูนย์วิจัยระหว่างประเทศ โดยมีเป้าหมายเพื่อหาลำดับเบสของจีโนมมนุษย์ ทราบตำแหน่งยีนบนโครโมโซม และทำความเข้าใจฟังก์ชันการทำงานของยีนทั้งหมดที่อยู่ในจีโนม โดย International Human Genome Sequencing Consortium ได้ตีพิมพ์โครงร่างแรกของจีโนมมนุษย์ในเดือนกุมภาพันธ์ปี ค.ศ. 2001 [2] ในนิตยสารเนเจอร์ (Nature) (รูปที่ 1.3) โดยจีโนมมนุษย์มีขนาดประมาณ 3 พันล้านคู่เบส (base pairs) และมีจำนวนยีนที่ทราบจากการหาลำดับเบสอยู่ที่ประมาณ 20,000 ถึง 25,000 ยีน โครงการจีโนมมนุษย์นี้มีจุดเริ่มต้นมาจากอัลเฟรด สเตอร์เตวานต์ (Alfred Sturtevant) ที่ทำการสร้างแผนที่ยีนของแมลงหวี่ (*Drosophila*) ในปี ค.ศ. 1911-1913 ซึ่งนำมาสู่ผลการศึกษาจีโนมในระดับโมเลกุลและงานวิจัยทางด้านอณูชีววิทยาหรือชีววิทยาระดับโมเลกุล โดยเฉพาะการค้นพบโครงสร้างดีเอ็นเอเกลียวคู่ของ ฟรานซิส คริก (Francis Crick) และเจมส์ วัตสัน (James Watson) ในปี ค.ศ. 1953 ที่นำไปสู่การได้รับรางวัลโนเบลของทั้งสองท่านในสาขาสรีรวิทยาหรือการแพทย์ในปี ค.ศ. 1962

ผลงานตีพิมพ์จีโนมมนุษย์โครงร่างแรกนี้เป็นจุดเริ่มต้นของมิติวิจัยและพัฒนาต่างๆ ที่เกี่ยวข้อง เช่น เครื่องมือและเทคโนโลยีในการหาลำดับเบส กลุ่มวิจัยใหม่ กฎหมายใหม่ที่จำเป็น จีโนมของยีสต์ จีโนมของเชื้อแบคทีเรียอีโคไล จีโนมของหนอน จีโนมของแมลงวันผลไม้ และจีโนมของสิ่งมีชีวิตอื่นๆ อีกมากมาย รวมถึงการเพิ่มของข้อมูลปริมาณมหาศาลในฐานข้อมูลออนไลน์สาธารณะต่างๆ ซึ่งเป็นการพลิกโฉมและสนับสนุนการวิจัยและพัฒนาสาขาชีววิทยา อณูชีววิทยา เทคโนโลยีชีวภาพ การเกษตร และการแพทย์อย่างก้าวกระโดด



articles

Initial sequencing and analysis of the human genome

International Human Genome Sequencing Consortium*

* A partial list of authors appears on the opposite page. Affiliations are listed at the end of the paper.

The human genome holds an extraordinary trove of information about human development, physiology, medicine and evolution. Here we report the results of an international collaboration to produce and make freely available a draft sequence of the human genome. We also present an initial analysis of the data, describing some of the insights that can be gleaned from the sequence.

The rediscovery of Mendel's laws of heredity in the opening weeks of the 20th century[1–3] sparked a scientific quest to understand the nature and content of genetic information that has propelled biology for the last hundred years. The scientific progress made falls naturally into four main phases, corresponding roughly to the four quarters of the century. The first established the cellular basis of heredity: the chromosomes. The second defined the molecular basis of heredity: the DNA double helix. The third unlocked the informational basis of heredity, with the discovery of the biological mechanism by which cells read the information contained in genes and with the invention of the recombinant DNA technologies of cloning and sequencing by which scientists can do the same.

The last quarter of a century has been marked by a relentless drive to decipher first genes and then entire genomes, spawning the field of genomics. The fruits of this work already include the genome sequences of 599 viruses and viroids, 205 naturally occurring plasmids, 185 organelles, 31 eubacteria, seven archaea, one fungus, two animals and one plant.

Here we report the results of a collaboration involving 20 groups from the United States, the United Kingdom, Japan, France, Germany and China to produce a draft sequence of the human genome. The draft genome sequence was generated from a physical map covering more than 96% of the euchromatic part of the human genome and, together with additional sequence in public databases,

coordinate regulation of the genes in the clusters.

- There appear to be about 30,000–40,000 protein-coding genes in the human genome—only about twice as many as in worm or fly. However, the genes are more complex, with more alternative splicing generating a larger number of protein products.
- The full set of proteins (the 'proteome') encoded by the human genome is more complex than those of invertebrates. This is due in part to the presence of vertebrate-specific protein domains and motifs (an estimated 7% of the total), but more to the fact that vertebrates appear to have arranged pre-existing components into a richer collection of domain architectures.
- Hundreds of human genes appear likely to have resulted from horizontal transfer from bacteria at some point in the vertebrate lineage. Dozens of genes appear to have been derived from transposable elements.
- Although about half of the human genome derives from transposable elements, there has been a marked decline in the overall activity of such elements in the hominid lineage. DNA transposons appear to have become completely inactive and long-terminal repeat (LTR) retrotransposons may also have done so.
- The pericentromeric and subtelomeric regions of chromosomes are filled with large recent segmental duplications of sequence from elsewhere in the genome. Segmental duplication is much more frequent in humans than in yeast, fly or worm.

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., et al. (2001). Initial sequencing and analysis of the human genome. Nature, 409(6822), 860-921. http://doi.org/10.1038/35057062

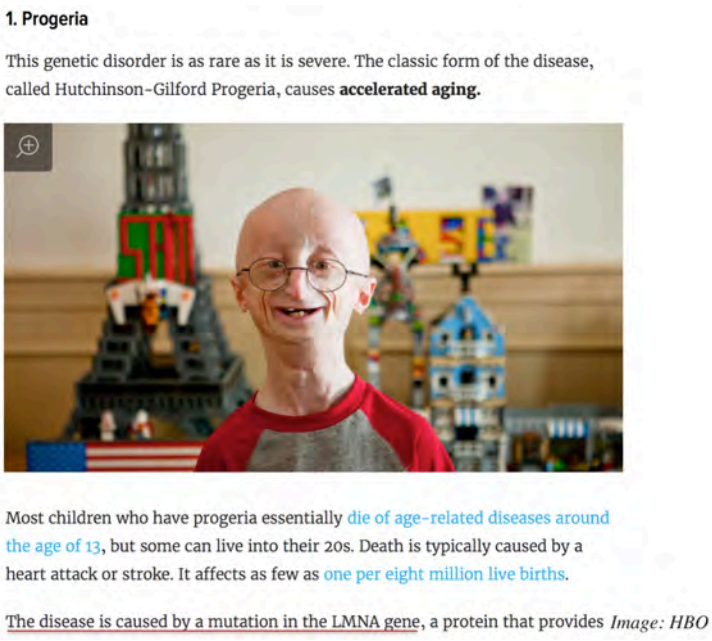
รูปที่ 1.3 ผลงานตีพิมพ์โครงร่างแรกของจีโนมมนุษย์ในเดือนกุมภาพันธ์ปี ค.ศ. 2001 ในนิตยสารเนเจอร์ (Nature) [2]

การประยุกต์ใช้จีโนมในการวิจัยและวินิจฉัยโรค

การแปรผันทางพันธุกรรมกับการเกิดโรค

โดยเฉลี่ยแต่ละ 1000 เบสของจีโนมมนุษย์แต่ละคนมีความแตกต่างกันประมาณ 1 ตำแหน่ง ซึ่งตำแหน่งที่แตกต่างกันเหล่านี้ มีผลต่อลักษณะปรากฏหรือฟีโนไทป์ (phenotype) เช่น สีตา ความสูง ความเสี่ยงต่อการมีคอเลสเตอรอลสูง และการเกิดโรคทางพันธุกรรม แสดงดังรูปที่ 1.4 คือกลุ่มอาการโพรจีเรีย (Progeria syndrome) ซึ่งเป็นโรคที่พบได้น้อยในระดับ 1 ใน 8 ล้านคน โดยมีลักษณะอาการคือเด็กจะมีรูปร่างแคระแกร็น ผิวหนังเหี่ยวย่นเหมือนคนชรา และมักเสียชีวิตเมื่ออายุประมาณ 13 ปี บางรายอาจมีอายุยืนกว่านั้น กลุ่มอาการนี้รายงานครั้งแรกในปี ค.ศ. 1886 [3] และมีรายงานเกี่ยวกับสาเหตุของโรคโดยเกิดจากการแปรผันของเบสลำดับที่ 1824 ของยีน Lamin A/C (*LMNA*) ในปี ค.ศ. 2003 [4] โดยนิวคลีโอไทด์ไซโทซีนถูกแทนที่ด้วยนิวคลีโอไทด์ไทมีน ซึ่งการแปรผันในลำดับเบสเดี่ยวที่ตำแหน่งนี้มีผลต่อการสร้างเอ็มอาร์เอ็นเอ (mRNA; messenger RNA) ที่สั้นกว่าปกติและแปลรหัสเป็นโปรตีนที่ผิดปกติ

รูปที่ 1.5 แสดงภาวะขนดก (Hypertrichosis) ที่ผู้ป่วยมีขนดกและยาวมากกว่าคนปกติโดยอาจครอบคลุมพื้นที่ทั้งร่างกายหรือเฉพาะบางส่วนของร่างกาย [5] สาเหตุของโรคเกิดจากการเปลี่ยนแปลงการจัดเรียงตัวของลำดับเบสในบริเวณใกล้กับยีน *SOX3* [6] การเกิดจำนวนชุดซ้ำของดีเอ็นเอในบริเวณโครโมโซม 17q24.2-q24.3 [7] การแปรผันเชิงโครงสร้างในบริเวณที่ใกล้เคียงกับยีน *TRPS1* ในโครโมโซมที่ 8 [8] และการเกิดการแปรผันเชิงโครงสร้างแบบซับซ้อนในโครโมโซมที่ 8 [9] เป็นต้น รูปที่ 1.6 แสดงความผิดปกติของมือ/เท้าแบบแยกส่วน (Ectrodactyly) ที่เกิดจากการแปรผันเชิงโครงสร้างของโครโมโซมที่ 7 [10]



รูปที่ 1.4 กลุ่มอาการโพรจีเรีย (Progeria syndrome)

(*ที่มา:* Dvorsky, G. *10 Unusual Genetic Mutations in Humans*. [ONLINE] Available at: https://gizmodo.com/ [เข้าถึงออนไลน์เมื่อวันที่ 14 ก.ค. พ.ศ. 2564])

รูปที่ 1.5 ภาวะขนดก (Hypertrichosis)

(*ที่มา:* Dvorsky, G. *10 Unusual Genetic Mutations in Humans*. [ONLINE] Available at: https://gizmodo.com/ [เข้าถึงออนไลน์เมื่อวันที่ 14 ก.ค. พ.ศ. 2564])

ในปี ค.ศ. 2010 เด็กชายนิโคลาส โวลเกอร์ (Nicholas Volker) เป็นคนแรกที่รอดชีวิตจากการถอดรหัสพันธุกรรมเพื่อหาสาเหตุของโรค โวลเกอร์มีอาการลำไส้อักเสบรุนแรงโดยไม่ทราบสาเหตุและแพทย์ทำได้เพียงผ่าตัดลำไส้หลายครั้งเพื่อรักษาตามอาการจนเขาเกือบเสียชีวิต ในที่สุดคณะแพทย์โรงเรียนแพทย์วิสคอนซินจึงตัดสินใจหาลำดับเบสดีเอ็นเอของโวลเกอร์และพบการแปรผันในยีน X-linked inhibitor of apoptosis (*XIAP*) ซึ่ง

สัมพันธ์กับระบบภูมิคุ้มกันร่างกายที่ผิดปกติและเป็นสาเหตุอาการลำไส้อักเสบรุนแรง รวมทั้งได้แนวทางการรักษาที่ตรงจุด โดยแพทย์เปลี่ยนจากการผ่าตัดเป็นการรักษาด้วยภูมิคุ้มกันบำบัด (immunotherapy) เปลี่ยนถ่ายเซลล์เลือดจากรก (cord blood transplant) ทำให้รักษาชีวิตเด็กชายไว้ได้

รูปที่ 1.6 ความผิดปกติของมือ/เท้าแบบแยกส่วน (Ectrodactyly)

(*ที่มา:* Dvorsky, G. *10 Unusual Genetic Mutations in Humans*. [ONLINE] Available at: https://gizmodo.com/ [เข้าถึงออนไลน์เมื่อวันที่ 14 ก.ค. พ.ศ. 2564])

โครงการ 1000 จีโนม

โครงการ 1000 จีโนมมนุษย์เป็นโครงการแรกของโลกที่มีการหาลำดับเบสของคนจำนวนมาก โดยโครงการนี้เป็นความร่วมมือของหน่วยงานวิจัยระหว่างประเทศสหรัฐอเมริกา อังกฤษ จีน และเยอรมัน เพื่อสร้างสารบัญแฟ้ม (catalog) การแปรผันทางพันธุกรรมเพื่อใช้สนับสนุนงานวิจัยทางการแพทย์ในอนาคต โดยมีการขยายข้อมูลเพิ่มเติมจากโครงการ International HapMap โดยเป้าหมายของโครงการ 1000 จีโนม เพื่อเป็นแหล่งข้อมูลตัวแปรผันเชิงพันธุกรรมของภาวะพหุสัณฐานนิวคลีโอไทด์เดี่ยว (single nucleotide polymorphism) หรือสนิป (SNP) และการสอดแทรก (insertion) หรือการขาดหาย (deletion) ของลำดับนิวคลีโอไทด์จำนวนไม่มากที่เรียกว่าอินเดล (indel) ที่พบอย่างน้อย 1% ของกลุ่มประชากรที่ศึกษา นอกจากนี้ยังมีข้อมูลตัวแปรผันเชิงโครงสร้าง (structural variant) เป็นต้น โดยในโครงการมีการหาลำดับเบสจีโนมมนุษย์ 1000 คน ที่เก็บตัวอย่างดีเอ็นเอมาจากทั่วโลก นำมาหาลำดับเบสโดยเทคโนโลยีการหาลำดับเบสยุคใหม่ (next generation sequencing) หรือเอ็นจีเอส (NGS) แบบต่างๆ และวิเคราะห์ข้อมูลการแปรผันที่พบ โดยมีการตีพิมพ์โครงการนำร่อง [11] เฟสที่ 1 [12] และเฟสที่ 3 [13, 14] โครงการ 1000 จีโนมมนุษย์มีระยะเวลาโครงการระหว่างปี ค.ศ. 2008 ถึง 2015 โดยในปัจจุบัน (ค.ศ. 2018) แม้โครงการจะสิ้นสุดแล้ว ข้อมูลที่เป็นผลผลิตจากโครงการยังเปิดให้เข้าถึงได้โดยสาธารณะภายใต้การดูแลของศูนย์ข้อมูลที่ EMBL-EBI (The European Bioinformatics Institute) โดยได้รับเงินทุนสนับสนุนจากเวลคัม

ทรัส (Wellcome Trust) และมีโครงการ IGSR: The International Genome Sample Resource เป็นโครงการต่อเนื่อง

โครงการ 100,000 จีโนมของอังกฤษ [15] มีเป้าหมายหลักเพื่อหาลำดับเบสจีโนมของคนอังกฤษ 100,000 คน โดยเน้นผู้ป่วยของระบบบริการสุขภาพแห่งชาติ (National Health Service) หรือเอ็นเอชเอส (NHS) กลุ่มที่เป็น “โรคหายาก” (rare diseases) และสมาชิกในครอบครัว รวมถึงผู้ป่วยที่เป็นโรคมะเร็งประเภทที่พบโดยทั่วไป โครงการ 100K จีโนมเอเชียมีเป้าหมายหลักเพื่อหาลำดับเบสจีโนมของคนเชื้อชาติเอเชีย 100,000 คน เพื่อสนับสนุนความก้าวหน้าในทางการแพทย์โดยเฉพาะการแพทย์แม่นยำหรือเวชกรรมตรงเหตุ (precision medicine) ประเทศกาตาร์เป็นตัวอย่างประเทศที่มีโครงการแห่งชาติในการหาลำดับเบสจีโนมของประชากร 10,000 คน เพื่อเป็นฐานข้อมูลอ้างอิงของประเทศในการวิจัยและพัฒนาทางการแพทย์โดยเฉพาะเวชกรรมเฉพาะบุคคล (personalized medicine)

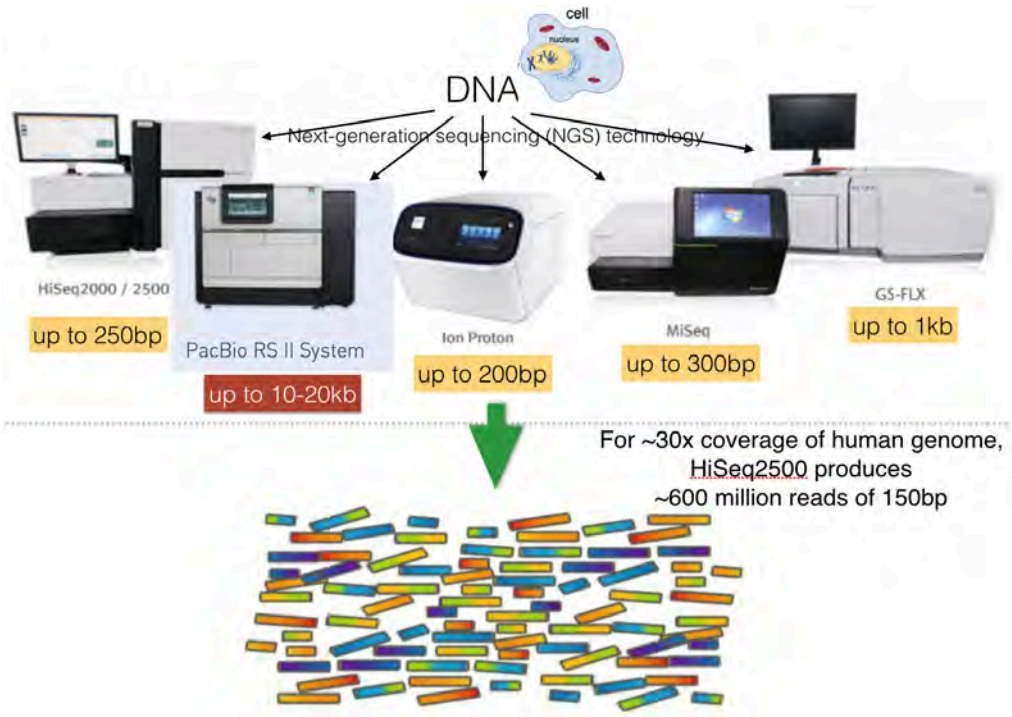
เทคโนโลยีในการหาลำดับเบสดีเอ็นเอ

เทคโนโลยีที่ใช้ในการหาลำดับเบสดีเอ็นเอมีประวัติที่ยาวนานกว่า 50 ปี [16] โดยแพลตฟอร์มหลักที่มีการใช้งานกันอย่างแพร่หลายในปัจจุบันคือแพลตฟอร์มอิลูมินา (Illumina) ด้วยราคาที่ถูกลงเป็นอย่างมากประมาณ 40,000 บาทสำหรับ 1 จีโนมมนุษย์ หรือถูกกว่านั้นถ้ามีการหาลำดับเบสของจีโนมจำนวนมาก ทั้งนี้ข้อมูลรหัสพันธุกรรมที่ได้จากแพลตฟอร์มอิลูมินามักเป็นคู่ของดีเอ็นเอสายสั้น (เรียกว่ารีด: read) ยาวประมาณ 100-150 เบสต่อเส้น (ขึ้นอยู่กับแพลตฟอร์มจำเพาะที่เลือกใช้) โดยข้อมูลที่ส่งออกจากเครื่องหาลำดับเบสมีขนาดประมาณ 90 กิกะไบต์ต่อ 1 จีโนมมนุษย์ ซึ่งใหญ่กว่าขนาดของจีโนมมาก เนื่องจากดีเอ็นเอจะถูกตัดแบบสุ่มก่อนเข้าเครื่องหาลำดับเบส จึงจำเป็นต้องตัดสายดีเอ็นเอหลายรอบเพื่อให้ได้รหัสพันธุกรรมครอบคลุมทั้งจีโนม ข้อมูล 90 กิกะไบต์ข้างต้นเป็นผลจากการตัดสายดีเอ็นเอแบบสุ่มและหาลำดับเบส 30 รอบ หรือ 30x ซึ่งสะท้อนค่า sequencing coverage หรือจำนวนรีดโดยเฉลี่ยที่สามารถนำไปเทียบหรือแมพ (map) กับจีโนมอ้างอิงในบริเวณที่มีลักษณะเบสเหมือนกันหรือใกล้เคียงกันกับรีดที่สุด เทคโนโลยีแพคไบโอ (PacBio) เป็นเทคโนโลยีที่สามารถหาลำดับเบสได้ยาวว่าอิลูมินามาก (โดยได้รีดยาวประมาณ 10k-20k) อย่างไรก็ตามแพคไบโอยังมีข้อจำกัดในเรื่องของความผิดพลาดในการอ่านลำดับเบสซึ่งอาจสูงถึง 15% ในการอ่านหนึ่งรอบ [17] ของความยาวรีดและยังมีราคาสูงมากเมื่อเทียบกับแพลตฟอร์มอิลูมินา รูปที่ 1.7 แสดงตัวอย่างเครื่องมือที่ใช้ในการหาลำดับเบส

ตัวอย่างโจทย์ทางชีวสารสนเทศ

ปัญหาการประกอบร่างจีโนมใหม่

ปัญหาการประกอบร่างจีโนมใหม่ (*de novo* genome assembly) เกิดจากเครื่องหาลำดับเบสส่วนใหญ่ผลิตข้อมูลลำดับเบสสายสั้นจำนวนมากซึ่งอาจเป็นสายคู่หรือสายเดี่ยว การได้มาซึ่งลำดับเบสที่มีความยาวในระดับโครโมโซม

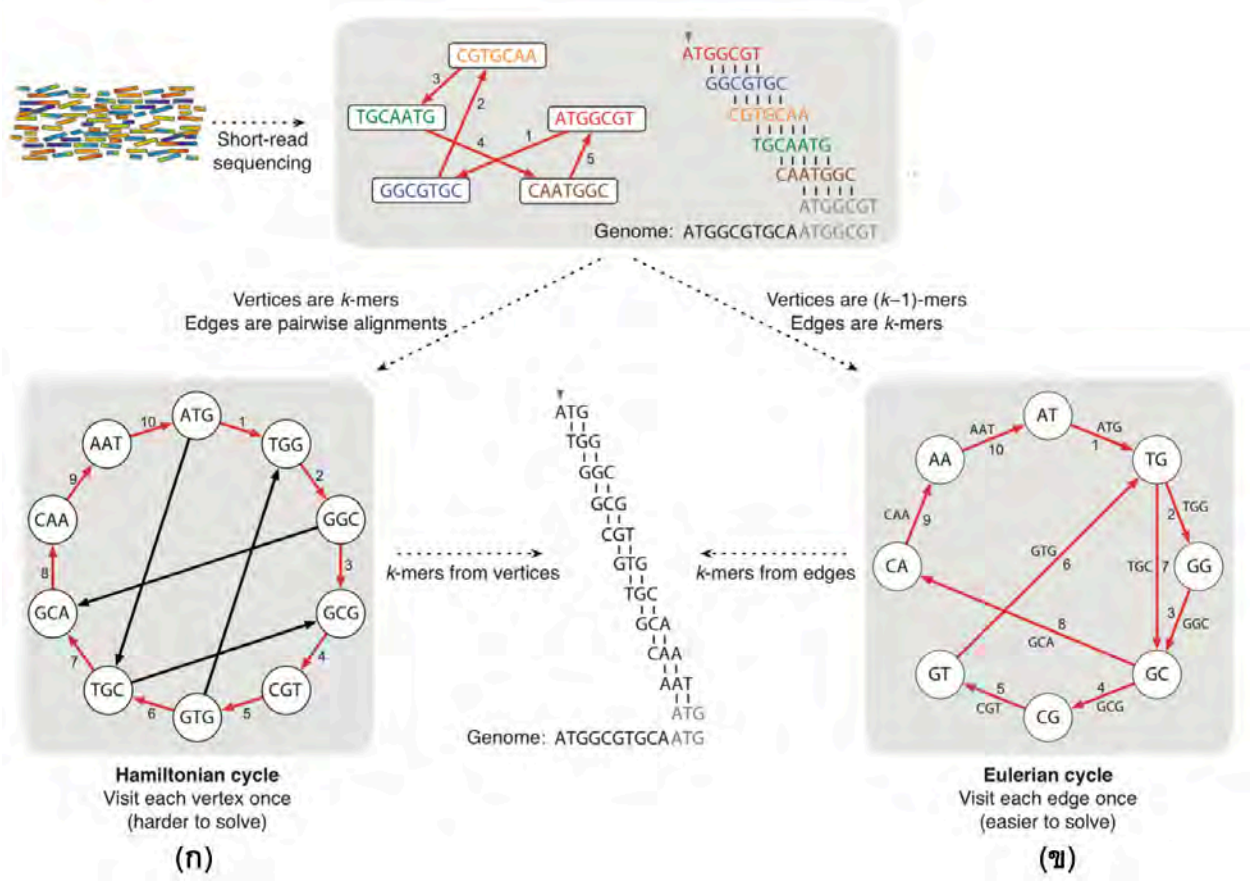


รูปที่ 1.7 แพลตฟอร์มและเครื่องมือที่ใช้ในการหาลำดับเบส

จำเป็นต้องนำดีเอ็นสายสั้นเหล่านี้มาต่อกัน คำถามเชิงอัลกอริทึมและการคำนวณคือจะต่อดีเอ็นเอสายสั้นจำนวนเกือบพันล้านรีดเข้าด้วยกันให้เป็นโครโมโซมที่ถูกต้อง โดยใช้ทรัพยากรการคำนวณอย่างมีประสิทธิภาพได้อย่างไร รูปที่ 1.8 แสดงสองแนวคิดในการต่อดีเอ็นเอสายสั้นเข้าด้วยกัน รูปที่ 1.8(ก) ใช้เส้นทางฮามิลโทเนียนเป็นตัวแทนของดีเอ็นเอสายยาวหลังการต่อ แต่ละโหนดเป็นตัวแทนรีดและเส้นเชื่อมระหว่างโหนดแสดงการเกิดความทับซ้อน (overlapping) ของลำดับเบสระหว่างรีดสองโหนด รูปที่ 1.8(ข) ใช้เส้นทางออยเลอร์เป็นตัวแทนของดีเอ็นเอสายยาวหลังการต่อ โดยออกแบบให้แต่ละโหนดเป็นตัวแทนส่วนของรีดและเส้นเชื่อมระหว่างโหนดแสดงลำดับเบสที่เป็นส่วนทับซ้อนกันของสองโหนด ตัวอย่างโปรแกรมที่พัฒนาขึ้นเพื่อแก้ปัญหาการประกอบร่างจีโนมใหม่สามารถศึกษาเพิ่มเติมได้จาก [18, 19]

ปัญหาการเทียบรีดกับจีโนมอ้างอิง

ปัญหาการเทียบรีดกับจีโนมอ้างอิง (read mapping) เชื่อมโยงกับปัญหาการประกอบร่างจีโนมข้างต้นโดยในกรณีที่มีจีโนมอ้างอิงที่เกิดจากการประกอบร่างจีโนมแล้ว ข้อมูลดีเอ็นเอสายสั้นที่ได้จากการหาลำดับเบส มักถูกนำมาเทียบกับจีโนมอ้างอิงแทนการนำไปประกอบร่างใหม่ ทั้งนี้เพื่อเป็นการลดเวลาในประมวลผลและเพิ่มความถูกต้อง โดยสามารถนำรีดทั้งหมดที่ตกอยู่ในบริเวณเดียวกันของจีโนมมาต่อกันให้ยาวขึ้น โดยทั่วไปผลการเทียบรีดกับจีโนมอ้างอิงจะถูกนำไปวิเคราะห์การแปรผันทางพันธุกรรม คำถามเชิงอัลกอริทึมและการคำนวณคือจะออกแบบอัลกอริทึมให้สามารถนำรีดที่มีความยาวประมาณ 100-150 เบสจำนวนเกือบพันล้านเส้นไปหาบริเวณที่เหมือนที่สุด



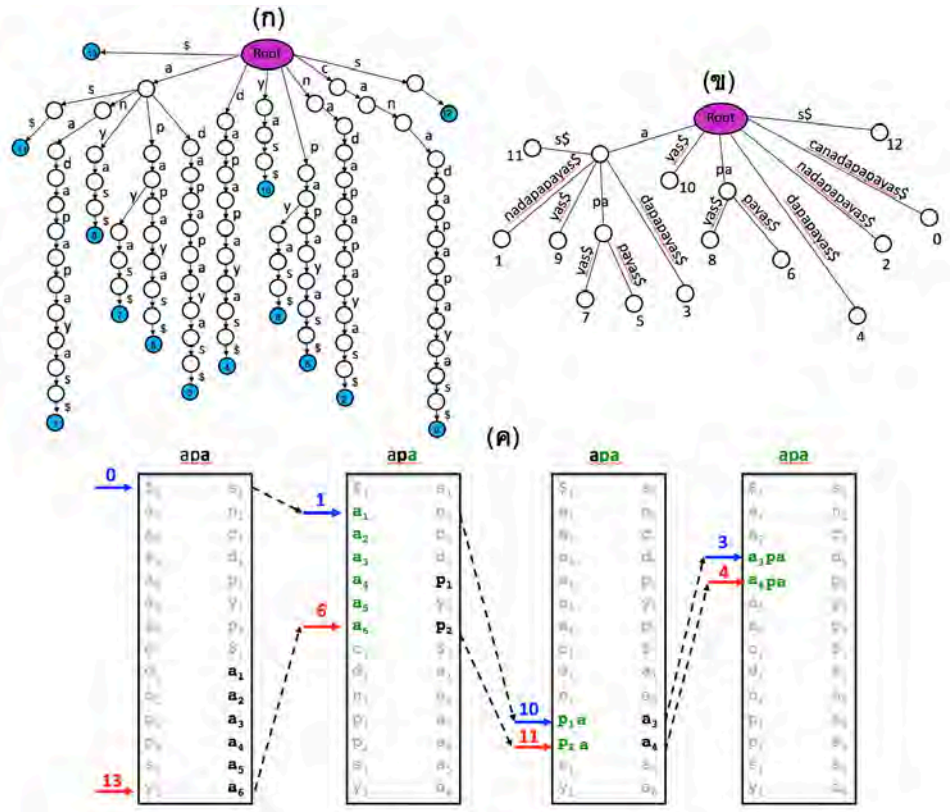
รูปที่ 1.8 การประกอบร่างจีโนมใหม่โดยจำลองปัญหาในรูปแบบกราฟและหา (ก) เส้นทางฮามิลโทเนียน (ข) เส้นทางออยเลอร์

(ที่มา: รูปที่ 3 ของ [20])

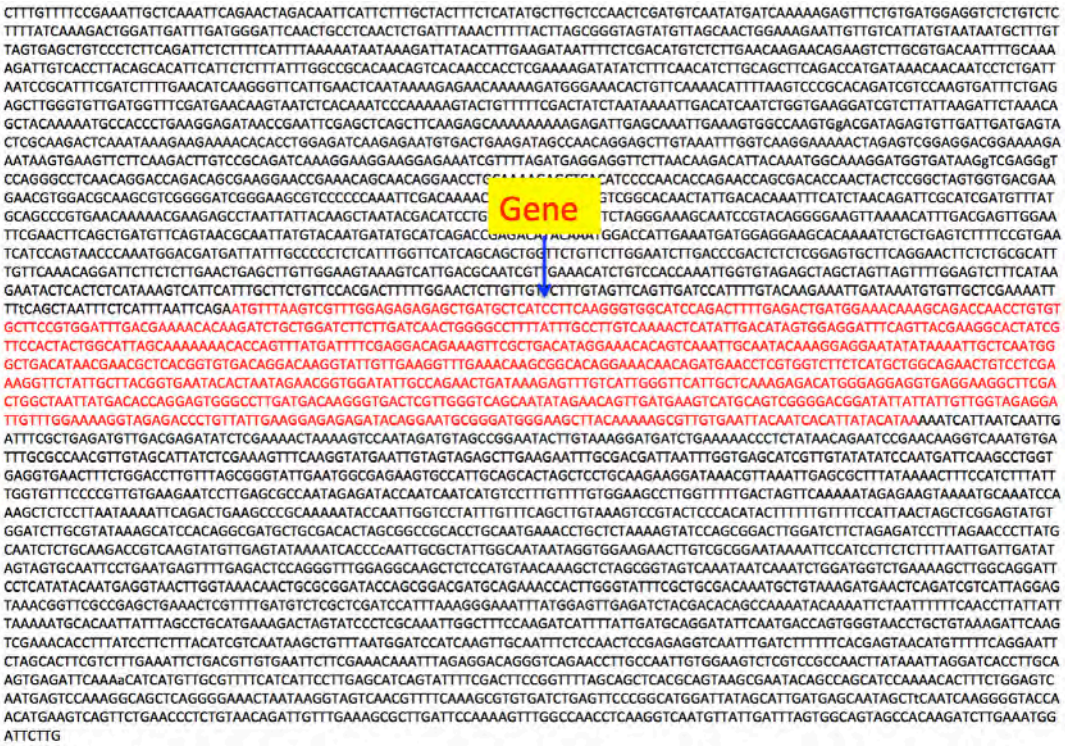
ในจีโนมได้อย่างรวดเร็วและใช้ทรัพยากรการคำนวณอย่างมีประสิทธิภาพได้อย่างไร รูปที่ 1.9 แสดงตัวอย่างโครงสร้างข้อมูลที่แตกต่างกันในการแก้ปัญหาการเทียบรีดกับจีโนมอ้างอิง โดยรูปที่ 1.9(ก) และ 1.9(ข) ใช้ซัฟฟิกซ์ทรัยและซัฟฟิกซ์ทรีตามลำดับ ส่วนรูปที่ 1.9(ค) แสดงการใช้ Burrows-Wheeler Transform (BWT) ตัวอย่างโปรแกรมที่แก้ปัญหาการเทียบรีดกับจีโนมอ้างอิงนี้สามารถศึกษาเพิ่มเติมได้จาก [21] เป็นต้น

ปัญหาการตรวจหาบริเวณที่เป็นยีนในจีโนม

ปัญหาการตรวจหาบริเวณที่เป็นยีนในจีโนมเกิดขึ้นหลังจากได้ลำดับเบสของจีโนมมาเรียบร้อยแล้ว จากรูปที่ 1.10 เบสสีแดงแสดงตัวอย่างบริเวณที่เป็นยีนที่สามารถแปลรหัสไปเป็นโปรตีน (protein-coding gene) คำถามคือเบสบริเวณใดบ้างควรเป็นสีแดงในลำดับเบสของจีโนม ปัญหาการตรวจหาบริเวณที่เป็นยีนนี้เป็นตัวอย่างที่องค์ความรู้ทางชีววิทยาช่วยทำให้สามารถออกแบบวิธีการแก้ปัญหาในเชิงคำนวณได้ดี และมีความถูกต้องมากขึ้น



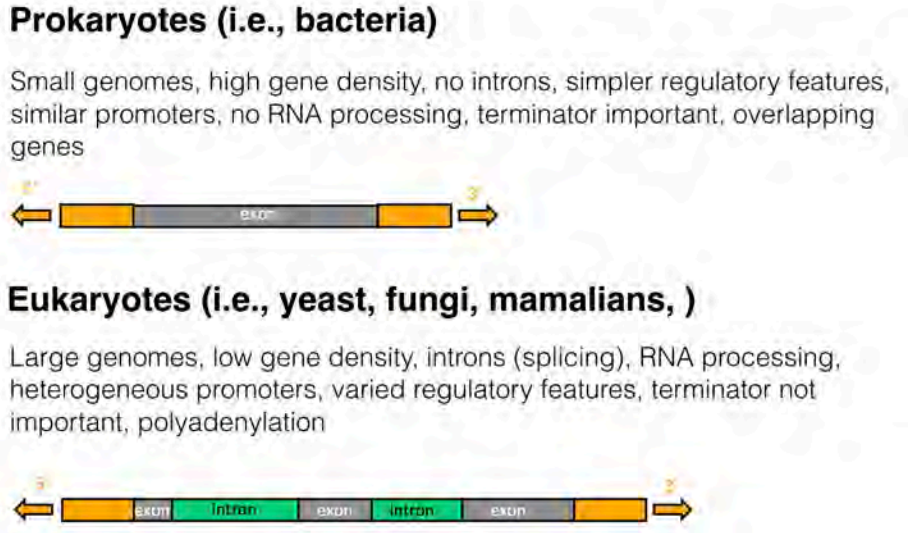
รูปที่ 1.9 โครงสร้างข้อมูลที่สามารถใช้แก้ปัญหาการเทียบรีดสายสั้นกับจีโนมอ้างอิง (ก) ซัฟฟิกซ์ทรัย (ข) ซัฟฟิกซ์ทรี (ค) Burrows-Wheeler Transform (BWT)



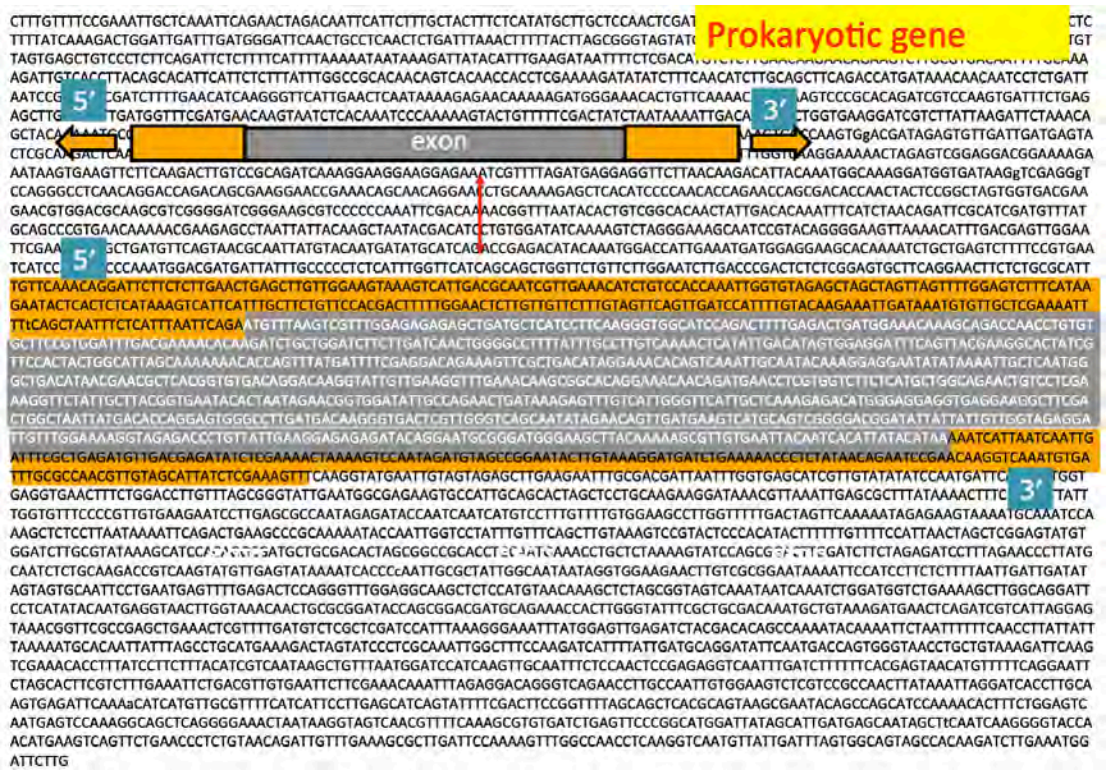
รูปที่ 1.10 ลำดับเบสส่วนที่เป็นยีนในจีโนมที่สามารถแปลรหัสไปเป็นโปรตีน

รูปที่ 1.11 แสดงลักษณะพื้นฐานที่แตกต่างกันระหว่างจีโนมและยีนในสิ่งมีชีวิตกลุ่มยูแคริโอตและโพรแคริโอต โดยจีโนมของสิ่งมีชีวิตกลุ่มโพรแคริโอตจะมีขนาดเล็ก มีความหนาแน่นของยีนคือยีนอยู่ใกล้กันมาก โครงสร้างของยีนไม่ซับซ้อนเพราะมีเพียงเฉพาะส่วนที่เป็นเอกซอน (exon) ไม่มีอินทรอน (intron) มีระบบการควบคุมการแสดงออกของยีนไม่ซับซ้อน ไม่มีการประมวลผลอาร์เอ็นเอเพิ่มเติมเพราะไม่มีการเลือกส่วนของเอกซอนไปแปลรหัสเป็นโปรตีน และยีนมีโอกาสที่จะทับซ้อนกัน สำหรับจีโนมของสิ่งมีชีวิตกลุ่มยูแคริโอตจะมีขนาดใหญ่กว่ามาก (แล้วแต่ประเภทของสิ่งมีชีวิต) มีความหนาแน่นของยีนน้อย โครงสร้างของยีนซับซ้อนกว่าเพราะมีส่วนที่เป็นอินทรอนแทรกอยู่ระหว่างเอกซอนซึ่งต้องมีกระบวนการเพิ่มเติมในการประมวลผลอาร์เอ็นเอ เช่น การตัดเชื่อมอาร์เอ็นเอ (RNA splicing) หรือการเลือกเอาเฉพาะบางเอกซอนมาต่อกันเพื่อแปลรหัสต่อไปเป็นโปรตีน และมีรูปแบบการควบคุมการแสดงออกของยีนที่หลากหลาย

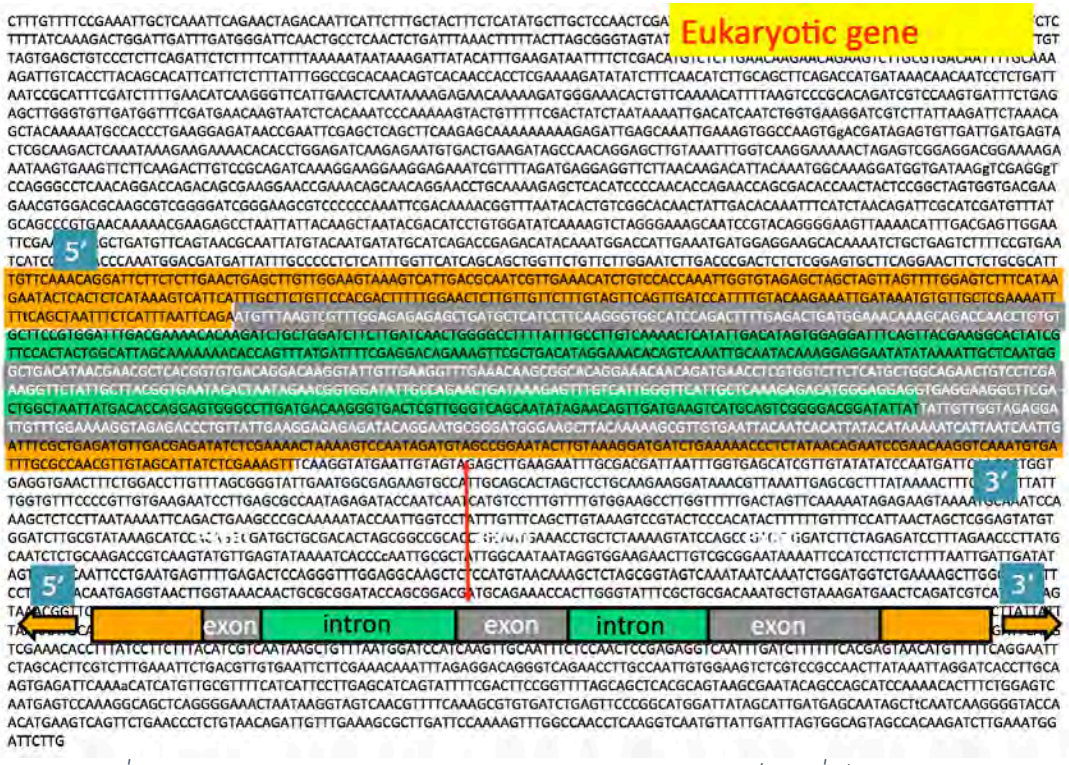
ความรู้ทางชีววิทยาข้างต้นแสดงให้เห็นว่าถ้าเทียบลักษณะโครงสร้างยีนของสิ่งมีชีวิตกลุ่มโพรแคริโอตและยูแคริโอตกับส่วนของจีโนมที่ไม่ได้ระบุประเภทของสิ่งมีชีวิตในรูปที่ 1.10 จะได้ผลตามรูปที่ 1.12 และ 1.13 ตามลำดับ สำหรับแนวคิดเชิงคำนวณที่ใช้ในการตรวจหาบริเวณที่เป็นยีนนั้นมีความหลากหลาย โดยวิธีการหลักประกอบด้วย Homology method และ *Ab initio* method วิธีการแรกใช้การเทียบเคียงจีโนมกับยีนที่มีการรายงานมาก่อนหน้าในสิ่งมีชีวิตอื่น ส่วนวิธีการหลังนั้นสามารถทำได้หลายวิธี เช่น การหายีนโดยใช้ชุดของกฎที่สร้างจากลักษณะเฉพาะทางชีววิทยา การพิจารณาองค์ประกอบของลำดับเบสในยีนที่ทราบมาก่อนหน้าเพื่อหาคุณลักษณะ (feature) ที่แตกต่างระหว่างบริเวณที่เป็นยีนและบริเวณที่ไม่ใช่ยีน การมองหารูปแบบจำเพาะในบางบริเวณของยีน เช่น ส่วนหัวของยีน ส่วนท้ายของยีน ลักษณะจำเพาะของบริเวณที่เชื่อมต่อระหว่างเอกซอนกับอินทรอน โดยอาจนำไปใช้เป็นข้อมูลเข้าของแบบจำลองการเรียนรู้ด้วยเครื่อง เช่น ต้นไม้การตัดสินใจ (decision tree), นิวรัลเน็ตเวิร์ค (neural network), และแบบจำลองมาร์คอฟซ่อนเร้น (Hidden Markov Model) เพื่อใช้จำแนกบริเวณที่เป็นยีนและที่ไม่ใช่ยีนต่อไป



รูปที่ 1.11 ความแตกต่างทางชีววิทยาของโครงสร้างยีนและจีโนมระหว่างกลุ่มโพรแคริโอตและยูแคริโอต

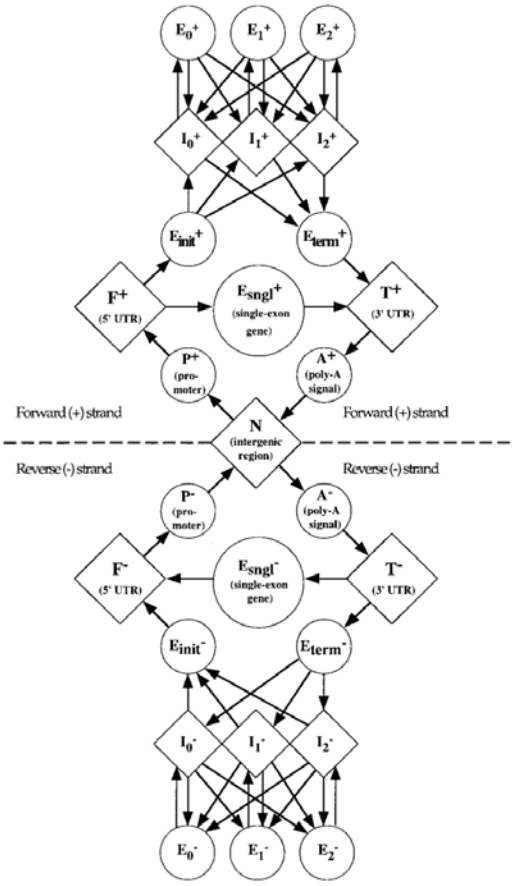


รูปที่ 1.12 โครงสร้างยีนในกลุ่มโพรแคริโอตเทียบกับส่วนของดีเอ็นเอที่เป็นยีนในจีโนม



รูปที่ 1.13 โครงสร้างยีนในกลุ่มยูแคริโอตเทียบกับส่วนของดีเอ็นเอที่เป็นยีนในจีโนม

รูปที่ 1.14 แสดงแบบจำลองมาร์คอฟซ่อนเร้น (Hidden Markov Model) ของ GENSCAN [22] ในการตรวจจับบริเวณที่เป็นยีนตีพิมพ์ในปี ค.ศ.1997

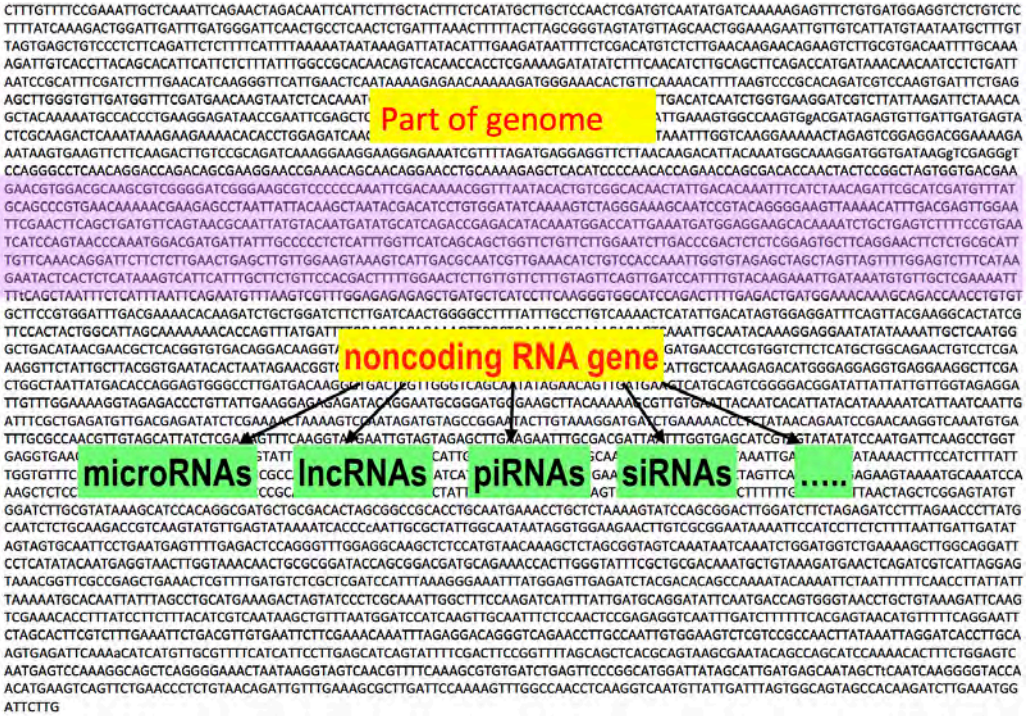


รูปที่ 1.14 แบบจำลองมาร์คอฟซ่อนเร้น (Hidden Markov Model) ของ GENSCAN [22] ในการตรวจจับบริเวณที่เป็นยีน

(*ที่มา*: *รูปที่ 3 ของ [22]*)

ปัญหาการตรวจหาบริเวณที่เป็นอาร์เอ็นเอไม่กำหนดรหัสในจีโนม

จีโนมมนุษย์มีขนาดประมาณ 3 พันล้านเบสและในปัจจุบันมีเพียง 2-3% ของลำดับเบสที่ถูกรายงานว่าเป็นยีนที่สามารถแปลรหัสต่อไปเป็นโปรตีน (protein-coding gene) บริเวณอื่นๆ ถือว่าเป็นบริเวณที่เป็นขยะ อย่างไรก็ตาม ในปัจจุบันมีรายงานว่าลำดับเบส 97% ที่เหลือ หลายบริเวณเกี่ยวข้องกับการควบคุมกระบวนการต่างๆ มีผลต่อการเกิดโรค และหลายบริเวณเป็นยีนที่สามารถถอดรหัสไปเป็นอาร์เอ็นไม่กำหนดรหัส (noncoding RNA) ซึ่งไม่มีการแปลรหัสเป็นโปรตีน แต่มีส่วนในการควบคุมการแสดงออกของยีน คำถามคือ จะสามารถตรวจหาบริเวณที่เป็นอาร์เอ็นเอไม่กำหนดรหัสเหล่านี้ (รูปที่ 1.15) ได้อย่างไร และสามารถประยุกต์ใช้แนวทางเดียวกับการตรวจหาบริเวณที่เป็นยีนได้หรือไม่ อย่างไร มีองค์ความรู้ทางชีววิทยาอะไรบ้างที่จำเป็นต้องทราบก่อนการออกแบบวิธีการทางคอมพิวเตอร์ในการตรวจหา



รูปที่ 1.15 ลำดับเบสส่วนที่เป็นยีนในจีโนมที่สามารถถอดรหัสไปเป็นอาร์เอ็นไม่กำหนดรหัส

ปัญหาการตรวจหาการแปรผันของรหัสพันธุกรรมในจีโนม

การแปรผันของรหัสพันธุกรรมสามารถแบ่งออกได้เป็น 2 กลุ่มหลัก (รูปที่ 1.16) คือ กลุ่มที่การแปรผันมีผลในลำดับเบสแต่ไม่มีผลเชิงโครงสร้าง (รูปที่ 1.16(ก)) ประกอบด้วย (1) เอสเอ็นวี (SNV; Single Nucleotide Variant) โดยในตัวอย่างนี้จีโนมอ้างอิง (reference genome) เป็นเบส G ในขณะที่ข้อมูลรหัสพันธุกรรมที่นำมาเทียบเป็นเบส A (2) อินเดล (indel) เกิดการสอดแทรกลำดับเบสขนาดสั้น (small insertion) เมื่อเทียบกับจีโนมอ้างอิง โดยในรูปมีการเพิ่มลำดับเบส GACG เข้ามา หรือ (3) การขาดหายไปของลำดับเบสขนาดสั้น (small deletion) โดย GTCA หายไป สำหรับกลุ่มที่การแปรผันมีผลในเชิงโครงสร้าง (รูปที่ 1.16(ข)) ประกอบด้วย deletion, insertion, duplication, inversion และ translocation โดย (1) deletion เกิดการหายไปของลำดับเบส โดยตัวอย่างในรูปส่วนของบริเวณ B หายไปทั้งหมด (2) duplication เกิดการซ้ำของชุดลำดับเบส เช่นมีบริเวณ C เพิ่มเข้ามาอีกชุด (3) inversion มีการกลับด้านของลำดับเบสโดยจำนวนเบสไม่เปลี่ยนแปลง ในรูป ทั้ง C และ D มีลำดับเบสที่กลับด้านเมื่อเทียบกับจีโนมอ้างอิง (4) translocation มีการเคลื่อนย้ายของลำดับเบสภายในโครโมโซมหรือระหว่างโครโมโซม ในรูป บริเวณ E, F, และ G ย้ายมาจากโครโมโซมอื่น เป็นต้น การแปรผันเหล่านี้มีผลต่อการเกิดโรคต่างๆ ตามที่ยกตัวอย่างมาก่อนหน้า สำหรับการแก้ปัญหานี้เชิงอัลกอริทึม ในขั้นต้นจำเป็นต้องทำความเข้าใจลักษณะการแปรผันแต่ละประเภทรวมทั้งลักษณะของข้อมูลเข้า เพื่อนำไปสู่การออกแบบและพัฒนาอัลกอริทึมที่มีประสิทธิภาพในการวิเคราะห์และตรวจหา ตัวอย่างอัลกอริทึมที่ใช้ในการตรวจหาการแปรผันสามารถศึกษาเพิ่มเติมได้จาก [23, 24]

รูปที่ 1.16 ประเภทการแปรผันของรหัสพันธุกรรมในจีโนม (ก) การแปรผันลำดับเบส (ข) การแปรผันเชิงโครงสร้าง

(ที่มา: รูปที่ 2.1 ของ [25])

ปัญหาการตรวจหาโมทิฟ

โมทิฟเป็นส่วนของดีเอ็นเอหรือโปรตีนที่มีรูปแบบจำเพาะ โดยมักเป็นบริเวณที่โมเลกุลจำเพาะหนึ่งๆ เข้ามาจับเพื่อควบคุมการทำงาน เช่น ดีเอ็นเอโมทิฟในส่วนหน้าของยีน (ลำดับเบสส่วนหน้า) จะเป็นบริเวณที่แฟกเตอร์ถอดรหัส (transcription factor) เข้ามาจับเพื่อควบคุมการแสดงออกของยีน

รูปที่ 1.17 ดีเอ็นเอโมทิฟ (ลำดับเบสสีแดง) รูปแบบเดียวกันที่แทรกเพิ่มในลำดับเบสส่วนหน้าของยีน 7 ยีนที่มีการแสดงออกร่วมกัน

ในรูปที่ 1.17 ลำดับเบสที่เป็นสีแดงและขีดเส้นใต้แสดงส่วนของโมทิฟในส่วนหน้าของยีน 7 ยีน ที่มีการแสดงออกในรูปแบบเดียวกัน โดยมีสมมติฐานว่ายีนที่มีการแสดงออกไปในแนวทางเดียวกันจะมีดีเอ็นเอโมทิฟเหมือนหรือคล้ายกันในส่วนหน้าของยีน เนื่องจากในรูปที่ 1.17 มีการใช้สีที่แตกต่างในลำดับเบสที่เป็นดีเอ็นเอโมทิฟ ทำให้สามารถเห็นรูปแบบที่เกิดร่วมกันในสายดีเอ็นเอ 7 เส้นชัดเจน รูปที่ 1.18 เมื่อเปลี่ยนสีโมทิฟเป็นสีเดียวกับลำดับเบสอื่นๆ จะเห็นว่าการมองหาบริเวณที่เป็นดีเอ็นเอโมทิฟทำได้ยากขึ้น นอกจากนี้ถ้าอนุญาตให้ดีเอ็นเอโมทิฟของแต่ละยีนมีลำดับเบสที่แตกต่างกันได้ เช่น ต่างกันไม่เกิน 2 เบส (รูปที่ 1.19) การออกแบบอัลกอริทึมที่มีประสิทธิภาพเพื่อหาดีเอ็นเอโมทิฟเหล่านี้จะมีความซับซ้อนมากขึ้น เมื่อเทียบกับการหาโมทิฟที่มีความเหมือนกัน 100%

รูปที่ 1.18 ส่วนหน้าของยีน 7 ยีนจากรูปที่ 1.17 แต่ไม่แสดงสีดีเอ็นเอโมทิฟ

รูปที่ 1.19 ส่วนหน้าของยีน 7 ยีนจากรูปที่ 1.17 โดยดีเอ็นเอโมทิฟแตกต่างกันบางลำดับเบส

ปัญหาการเทียบความคล้ายคลึงกันของลำดับเบสข้อมูลเข้ากับลำดับเบสในฐานข้อมูล

หลังจากได้บริเวณที่เป็นยีนในจีโนมแล้ว ข้อมูลถัดไปที่นักชีววิทยามักต้องการทราบคือ ยีนที่ทำนายได้เหล่านี้มีฟังก์ชันการทำงานใดบ้าง วิธีการหนึ่งเพื่อให้ได้มาซึ่งฟังก์ชันคือการนำลำดับเบสของยีนที่ทำนายได้ไปเทียบกับข้อมูลลำดับเบสที่อยู่ในฐานข้อมูลที่มีการระบุฟังก์ชันการทำงานมาก่อนหน้า (รูปที่ 1.20) คำถามในเชิงวิธีการคำนวณคือควรออกแบบอัลกอริทึมเพื่อเทียบความคล้ายคลึงกันของสายข้อมูลเข้าซึ่งอาจเป็นสายดีเอ็นเอหรือโปรตีนกับสายข้อมูลดีเอ็นเอหรือโปรตีนจำนวนมาก (มากกว่า 200 ล้านเส้น) ที่อยู่ในฐานข้อมูลได้ถูกต้องและรวดเร็วได้อย่างไร โปรแกรม BLAST [1] เป็นโปรแกรมที่ใช้ในการสืบค้นลำดับเบสในฐานข้อมูลที่มีการใช้งานอย่างแพร่หลายและมีจำนวนอ้างอิงในระดับต้นๆ

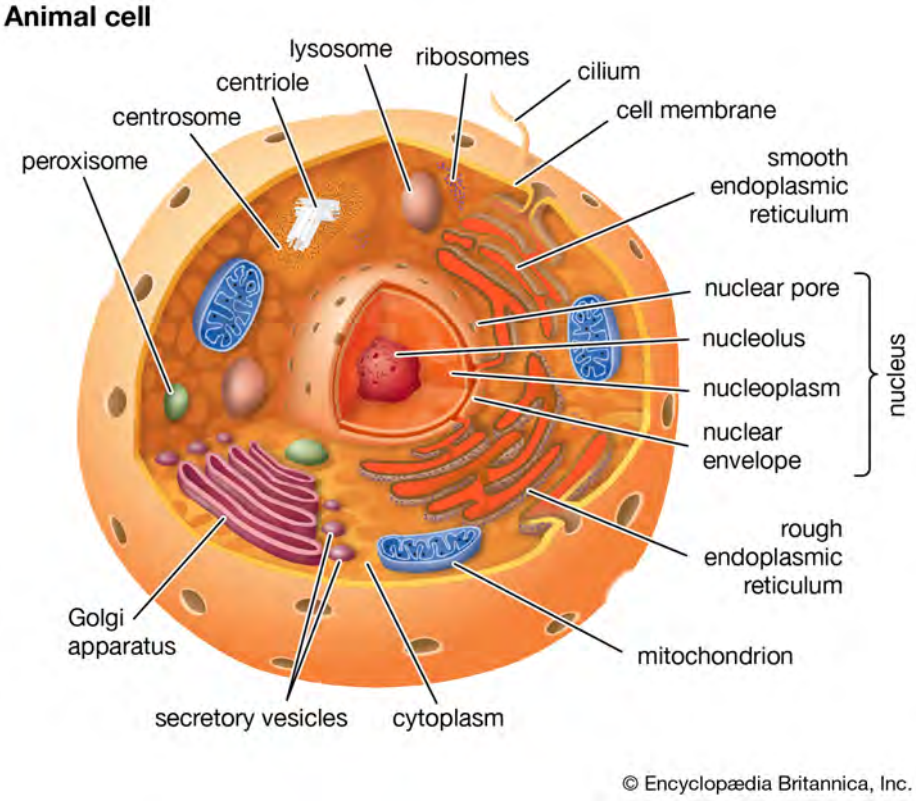
รูปที่ 1.20 การเทียบความคล้ายคลึงกันของลำดับเบสข้อมูลเข้ากับลำดับเบสในฐานข้อมูล

ความรู้พื้นฐานทางอณูชีววิทยา

เซลล์

เซลล์ (cell) เป็นองค์ประกอบพื้นฐานของสิ่งมีชีวิตทุกชนิด เซลล์ในกลุ่มยูแคริโอตที่เป็นเซลล์สัตว์ (รูปที่ 1.21) มีนิวเคลียสที่มีเยื่อหุ้มเป็นศูนย์ควบคุมกลางของเซลล์ รวมทั้งเป็นที่อยู่ของดีเอ็นเอซึ่งเก็บรหัสพันธุกรรมในรูปแบบของโครโมโซม นิวเคลียสอยู่ภายในไซโทพลาซึม (cytoplasm) ซึ่งประกอบด้วยส่วนที่เป็นของเหลวภายในเซลล์ (cytosol) และออร์แกเนลล์ (organelle) หรือโครงสร้างย่อยอื่นๆ เช่น ไลโซโซม (lysosome) ซึ่งภายในมีเอนไซม์จำนวนมาก มีหน้าที่หลักในการย่อยสารและทำลายเซลล์เมื่อเซลล์หมดอายุ ร่างแหเอนโดพลาซึม (endoplasmic

reticulum) หรืออีอาร์ (ER) มีหน้าที่หลากหลาย อีอาร์แบบขรุขระเป็นที่เกาะของไรโบโซม (ribosome) ที่มีหน้าที่สร้างโปรตีนจากรหัสพันธุกรรม อีอาร์แบบผิวเรียบทำหน้าที่สังเคราะห์ไขมัน เกี่ยวข้องกับการสร้างสเตอรอยด์ฮอร์โมนและการกำจัดพิษจากยา ไมโทคอนเดรีย (mitochondria) เป็นแหล่งผลิตพลังงานของเซลล์ และมีดีเอ็นเอเป็นของตัวเอง กอลไจ (golgi) เป็นแหล่งเก็บสารที่เซลล์สร้างขึ้นก่อนส่งออกนอกเซลล์ ร่างกายมนุษย์ประกอบด้วยเซลล์จำนวนล้านล้านเซลล์



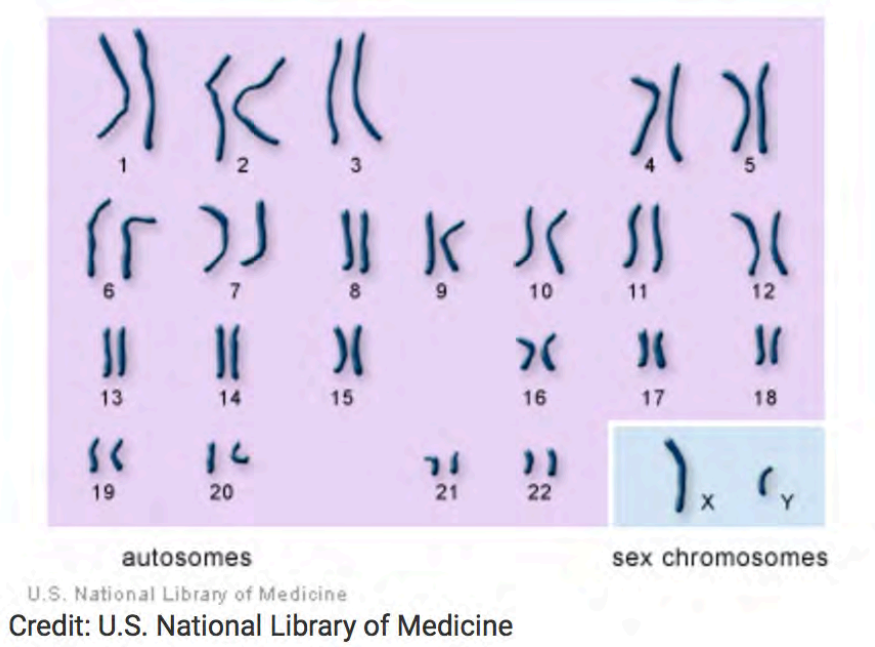
รูปที่ 1.21 องค์ประกอบพื้นฐานของเซลล์สิ่งมีชีวิตกลุ่มยูแคริโอต

(*ที่มา:* Britannica, The Editors of Encyclopedia. *Eukaryote*. [ONLINE] Available at: https://www.britannica.com [เข้าถึงออนไลน์เมื่อวันที่ 14 ก.ค. พ.ศ. 2564])

โครโมโซม

โครโมโซม (chromosome) เป็นโมเลกุลเดี่ยวของดีเอ็นเอสายยาวโดยเป็นที่เก็บข้อมูลรหัสพันธุกรรมของสิ่งมีชีวิต รหัสพันธุกรรมในโครโมโซมมีโครงสร้างที่เป็นระบบและมีบริเวณจำเพาะที่เป็นยีนเพียงส่วนน้อย บริเวณที่เป็นยีนเหล่านี้สามารถถูกแปลรหัสไปเป็นโปรตีนเพื่อทำงานในกระบวนการต่างๆ โครโมโซมของสิ่งมีชีวิตกลุ่มโพรแคริโอต (prokaryote) เช่น แบคทีเรียมีลักษณะเป็นวงแหวน (circular) อยู่ในไซโทพลาซึม (cytoplasm) ในบริเวณที่เรียกว่านิวคลีออยด์ (nucleoid) สำหรับโครโมโซมของสิ่งมีชีวิตกลุ่มยูแคริโอต (eukaryote) เช่น มนุษย์ มีลักษณะเป็นสายหรือเชิงเส้น (linear) อยู่ในนิวเคลียส (nucleus) โดยแต่ละโครโมโซมประกอบด้วยดีเอ็นเอที่พันตัวอย่าง

หนาแน่นรอบนิวเคลียร์โปรตีนที่เรียกว่าฮิสโทน (histone) มนุษย์มี 46 โครโมโซมแบ่งเป็น 22 คู่ของออโตโซม (autosome) และมีโครโมโซมเพศอีก 2 โครโมโซม (รูปที่ 1.22) ซึ่งถ้าเป็นเพศหญิงจะมีโครโมโซมเอ็กซ์ (X) 2 โครโมโซมในขณะที่เพศชายจะมีโครโมโซมเอ็กซ์ (X) และวาย (Y) อย่างละ 1 โครโมโซม ออโตโซมคือโครโมโซมหรือชุดของโครโมโซมที่ควบคุมลักษณะทางพันธุกรรมและลักษณะที่แสดงออกต่างๆ เซลล์ที่มีโครโมโซม 2 ชุด เช่น เซลล์ทั่วไปของมนุษย์ เรียกว่า ดิพลอยด์ (diploid) ส่วนเซลล์สืบพันธุ์ที่พัฒนาต่อไปเป็นเซลล์ไข่ (egg cell) หรือเซลล์อสุจิ (sperm cell) เรียกว่าเป็นแฮพลอยด์ (haploid) ซึ่งจะมีโครโมโซมเพียง 1 ชุดหรือครึ่งหนึ่งของเซลล์ดิพลอยด์ ที่มา https://www.nature.com/scitable/definition/chromosome-6



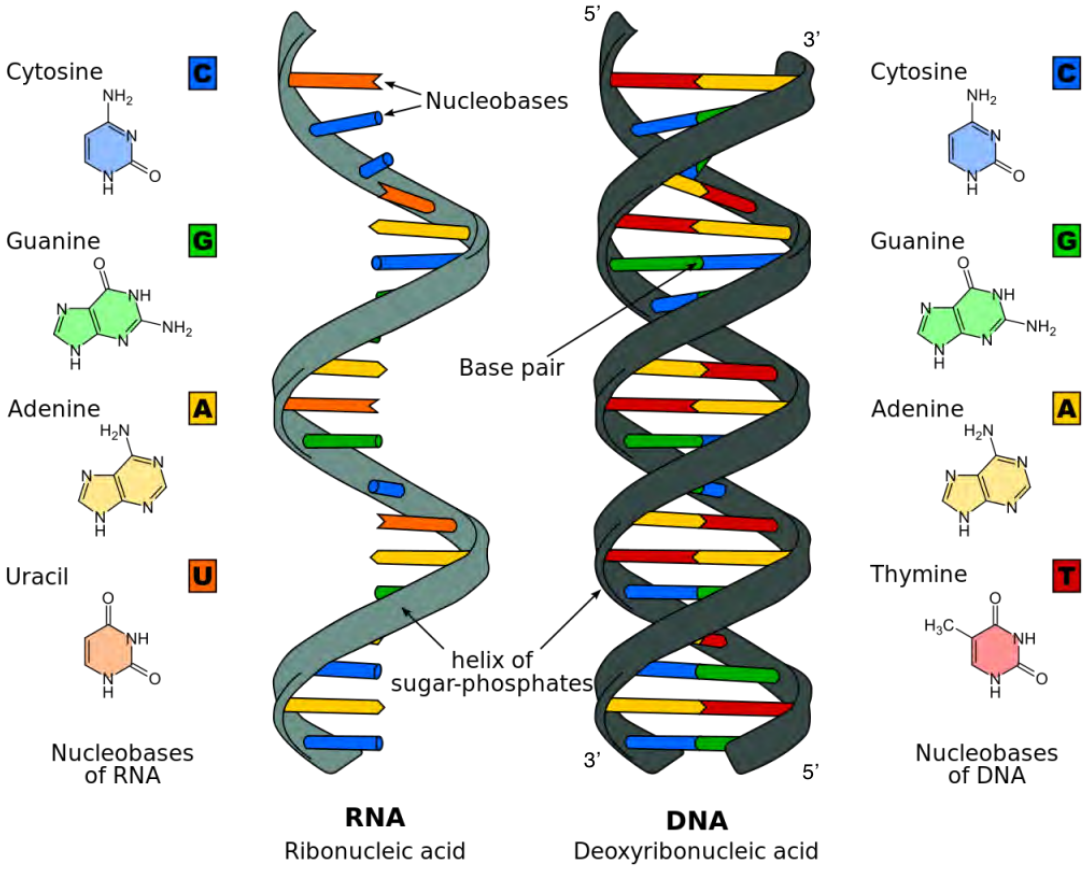
รูปที่ 1.22 โครโมโซมมนุษย์

(*ที่มา:* MedlinePlus, Genetics. *How many chromosomes do people have?*. [ONLINE] Available at: https://medlineplus.gov [เข้าถึงออนไลน์เมื่อวันที่ 14 ก.ค. พ.ศ. 2564])

ดีเอ็นเอ

ดีเอ็นเอ (DNA) [26] หรือกรดดีออกซีไรโบนิวคลีอิก (deoxyribonucleic acid) (รูปที่ 1.23) เป็นสารรหัสพันธุกรรมที่พบในสิ่งมีชีวิตทั้งในกลุ่มยูแคริโอต (eukaryote) เช่น มนุษย์ ลิง หมู ช้าง ม้า วัว ยีสต์ และเชื้อรา เป็นต้น และกลุ่มโพรแคริโอต (prokaryote) เช่น แบคทีเรียต่างๆ รูปร่างของดีเอ็นเอมีลักษณะเป็นเกลียวคู่ (double helix) เส้นแรกอ่านไปข้างหน้าจาก 5’ (five-prime) ไป 3’ (three-prime) และถูกกำหนดว่าเป็นฟอร์เวิร์ดสแตรนด์ (forward strand) หรือ สายบวก (plus strand) และอีกสายอ่านย้อนกลับจาก 3’ ไป 5’ เป็นรีเวิร์สสแตรนด์ (reverse strand) หรือสายลบ (minus strand) แต่ละสายประกอบไปด้วยนิวคลีโอไทด์ (nucleotide) ที่เรียงต่อกัน โดยแต่ละนิวคลีโอไทด์มีโครงสร้างหลักที่เป็นไปได้สี่แบบคือ ไซโทซีน (cytosine), ไทมีน (thymine), อะดีนีน

(adenine) และ กัวนีน (guanine) ในสายนิวคลีโอไทด์หนึ่งๆ ลำดับเบสเหล่านี้จะถูกแทนค่าด้วยตัวอักษร “C”, “T”, “A”, และ “G” ตามลำดับ ดีเอ็นเอสองสายที่จับกันเป็นเกลียวคู่นั้น คู่ของเบสที่จับกันมีได้สองแบบ คือไซโทซีน (C) จับกับ กัวนีน (G) และ ไทมีน (T) จับกับอะดีนีน (A)



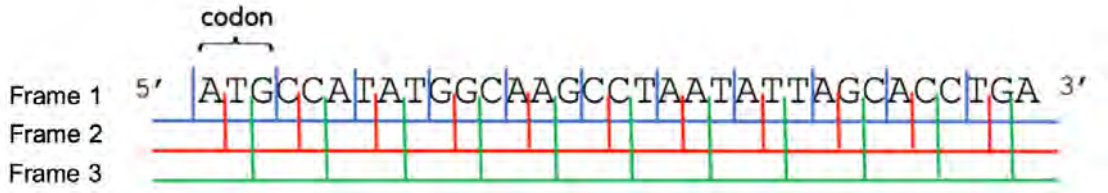
รูปที่ 1.23 ดีเอ็นเอ (DNA)

(*ที่มา:* Sponk, Public domain, via Wikimedia Commons. 2010. *Comparison of a single-stranded RNA and a double-stranded DNA.* [*เข้าถึงออนไลน์เมื่อวันที่* 11 *ก.ค. พ.ศ.* 2564])

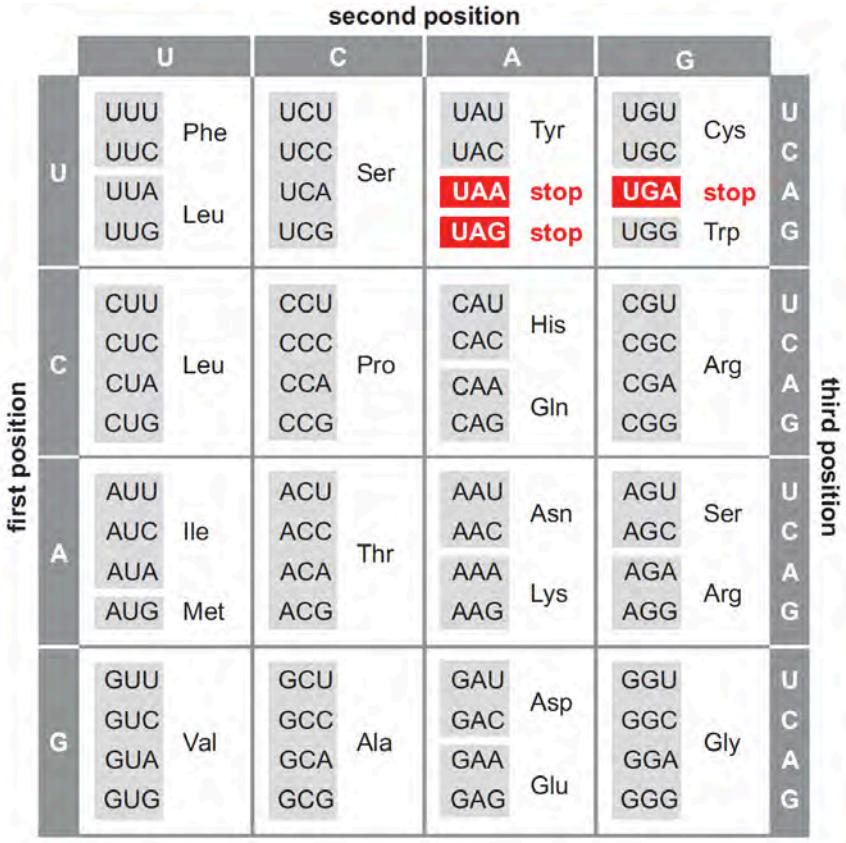
การอ่านเฟรมในลำดับนิวคลีโอไทด์

การอ่านเฟรมในลำดับนิวคลีโอไทด์ทั้งที่เป็นสายดีเอ็นเอและอาร์เอ็นเอ [27] สามารถอ่านได้ 3 เฟรม ตามรูปที่ 1.24 โดยจะอ่านทีละ 3 เบสที่อยู่ติดกันเรียกว่า 1 โคดอน (codon) จากรูป โคดอนแรกของเฟรมที่หนึ่ง คือ “ATG” และโคดอนที่สองคือ “CCA” ตามลำดับ โดยลำดับเบสระหว่างสองโคดอนภายในเฟรมจะไม่ทับซ้อนกัน สำหรับเฟรมที่สองชุดของโคดอนจะเลื่อนไปหนึ่งเบส (เกิด frame shift) ในกรณีนี้โคดอนแรกคือ “TGC” และโคดอนที่สองคือ “CAT” ตามลำดับ สำหรับเฟรมที่สามชุดของโคดอนจะเลื่อนไปสองเบสในกรณีนี้โคดอนแรกคือ “GCC” และโคดอนที่สองคือ “ATA” ตามลำดับ ดังนั้นอาร์เอ็นเอหนึ่งเส้นสามารถอ่านเฟรมได้ 3 แบบในสายบวก และอีก

3 แบบในสายลบ รวมอ่านเฟรมได้ทั้งหมด 6 แบบรูปที่ 1.25 แสดงการแปลรหัสของแต่ละโคดอนไปเป็นกรดแอมิโน



รูปที่ 1.24 การอ่านเฟรมของลำดับนิวคลีโอไทด์



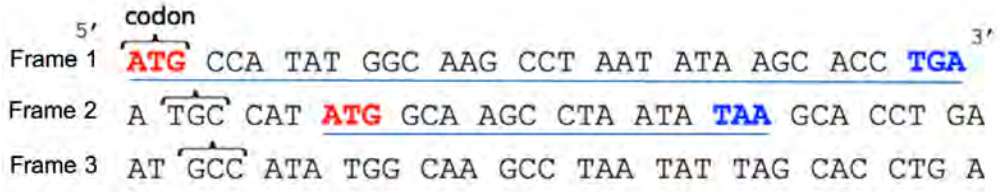
รูปที่ 1.25 ตารางการแปลงโคดอนไปเป็นกรดแอมิโน

(*ที่มา: รูปที่* 2-3 *หน้า* 38 *[26]*)

ขอบเขตการอ่านรหัส

ขอบเขตการอ่านรหัส (open reading frame) หรือโออาร์เอฟ (ORF) คือส่วนของเฟรมที่ถูกอ่านออกมาตามรูปแบบที่อธิบายไว้ในหัวข้อ *การอ่านเฟรมในลำดับนิวคลีโอไทด์* โดยโออาร์เอฟเป็นส่วนของเฟรมตั้งแต่รหัสเริ่มต้น (initiation codon หรือ start codon) หรือโคดอน “AUG” (แปลงมาจาก “ATG” ในระดับดีเอ็นเอ โดยการเปลี่ยนจากไทมีน (T) เป็นยูราซิล (U)) จนถึงรหัสหยุด (termination codon หรือ stop codon) ซึ่งสามารถเป็นโคดอน

“UAA”, “UAG” หรือ “UGA” (แปลงมาจาก “TAA”, “TAG” หรือ “TGA” ในระดับดีเอ็นเอ ตามลำดับ) โดยโออาร์เอฟนี้สามารถแปลรหัสต่อไปเป็นโปรตีนได้ รูปที่ 1.26 แสดงตัวอย่างโออาร์เอฟ 2 แบบจากการอ่านเฟรมที่ 1 และ 2 (เฟรมที่ 3 ไม่มีโออาร์เอฟ) โดยทั่วไปมักมีเฟรมเดียวที่ให้โออาร์เอฟที่ถูกต้องครบถ้วน สามารถแปลงต่อไปเป็นโปรตีนที่ทำงานได้ สำหรับเฟรมที่พบเฉพาะโคดอนปิดจะไม่สามารถแปลรหัสไปเป็นโปรตีนได้



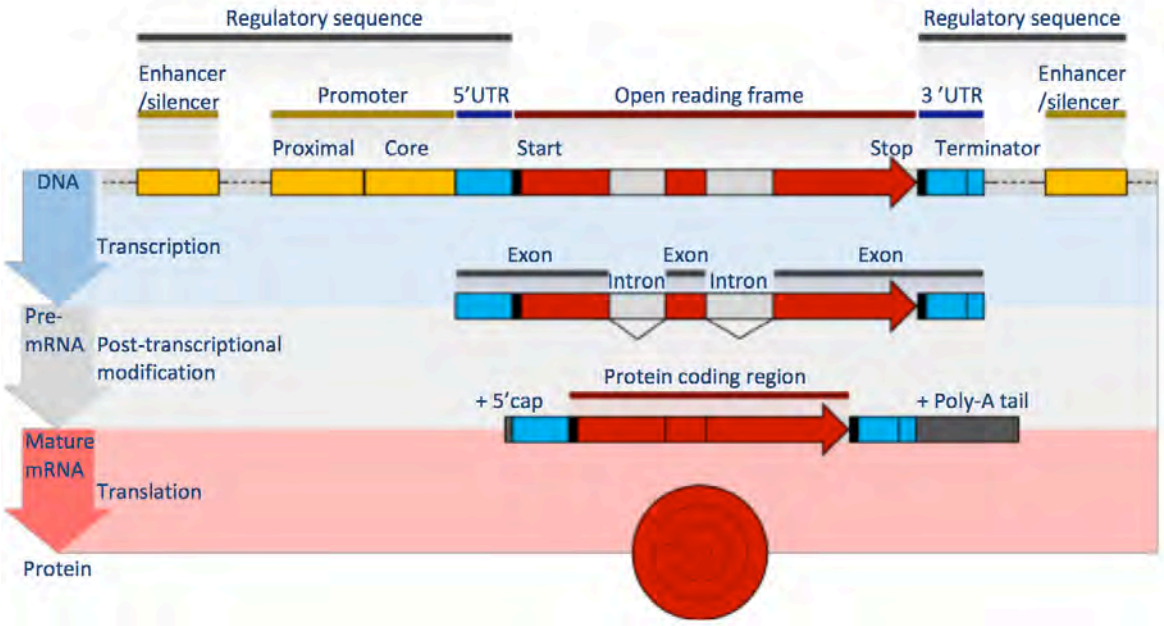
รูปที่ 1.26 โออาร์เอฟ (ชุดโคดอนที่ขีดเส้นใต้) ที่แตกต่างกันจากการอ่านเฟรมที่ 1 และ 2 ในขณะที่เฟรมที่ 3 ไม่พบโออาร์เอฟ

ยีน

ยีน (gene) [28] เป็นส่วนของดีเอ็นเอที่สามารถถอดรหัสต่อไปเป็นอาร์เอ็นเอนำรหัส (messenger RNA) หรือเอ็มอาร์เอ็นเอ (mRNA) และแปลรหัสต่อไปเป็นโปรตีน (protein) จากความเชื่อตามหลักชีววิทยาระดับโมเลกุล (central dogma of molecular biology) ยีนในสิ่งมีชีวิตกลุ่มยูแคริโอต เช่น ยีนในมนุษย์ มีโครงสร้างที่ซับซ้อนมากกว่ายีนในสิ่งมีชีวิตกลุ่มโพรแคริโอต เช่น แบคทีเรีย โดยจะมีทั้งส่วนที่เป็นเอกซอน (exon) และส่วนที่เป็นอินทรอน (intron) (รูปที่ 1.27) ในขณะที่ยีนในกลุ่มโพรแคริโอตไม่มีส่วนที่เป็นอินทรอน (รูปที่ 1.28) นอกจากนี้ยีนทั้งสองกลุ่มยังประกอบด้วยบริเวณไม่แปลรหัส (untranslated region) หรือยูทีอาร์ (UTR) ซึ่งอยู่ส่วนหน้าของยีนเรียกว่า 5’ UTR และส่วนท้ายของยีนเรียกว่า 3’ UTR ส่วนของโพรโมเตอร์ซึ่งมักอยู่ติดกับ 5’ UTR ส่วนของตัวส่งเสริม (enhancer) และไซเลนเซอร์ (silencer) ที่อยู่ห่างออกไปทั้งในส่วนของ 5’ และ 3’ UTR โพรโมเตอร์และไซเลนเซอร์เกี่ยวข้องกับการควบคุมการถอดรหัสยีนไปเป็นเอ็มอาร์เอ็นเอแรกสร้าง (pre-mRNA) ซึ่งจะถูกจัดการเพิ่มเติมโดยนำเฉพาะส่วนที่เป็นเอกซอนมาต่อกันกลายเป็นลำดับกำหนดรหัส (coding sequence) และมีการเติม 5’ แค็ป (cap) ที่ส่วนหน้าของยีนให้สามารถโน้มนำไรโบโซมเข้ามาจับ และเติมลำดับเบสอะดีนีนเป็นหางพอลิเอ (poly-A tail) ที่ส่วนปลายของยีนเพื่อเพิ่มความเสถียรของเอ็มอาร์เอ็นเอ ส่วน 5’ UTR และ 3’ UTR ควบคุมการแปลรหัสไปเป็นโปรตีน

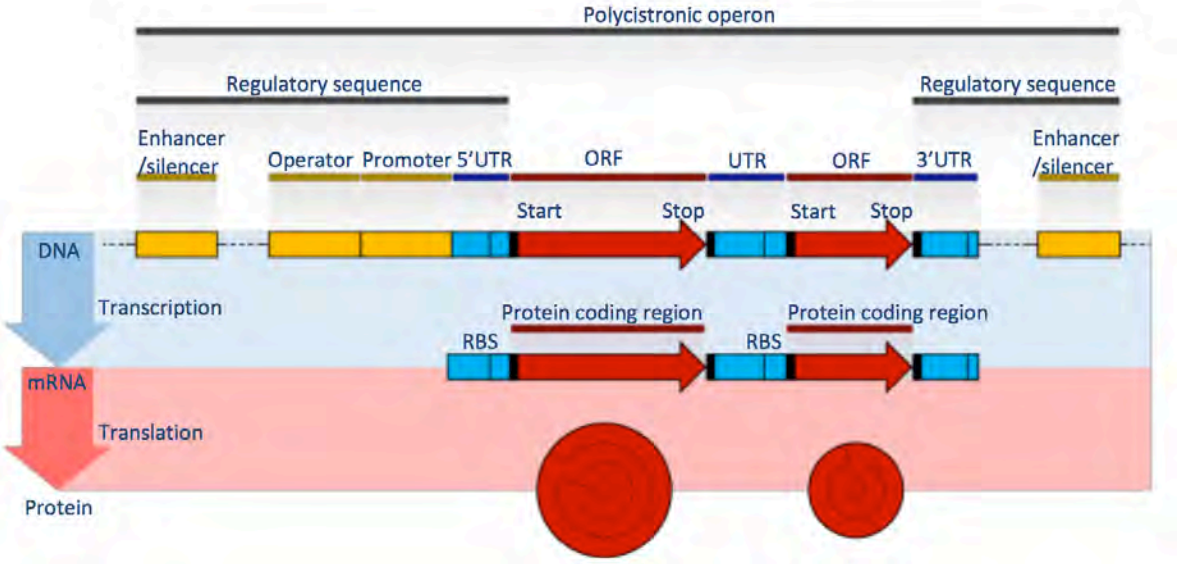
เนื่องจากดีเอ็นเอเป็นสายเกลียวคู่ ยีนอาจอยู่บนดีเอ็นเอที่เป็นสายบวกหรือลบก็ได้ โดยสายที่มียีนหนึ่งๆ อยู่เป็นสายกำหนดรหัส (coding strand) หรือ สายเซนส์ (sense strand) ของยีนนั้นๆ และดีเอ็นเออีกสายของเกลียวคู่จะเป็นสายดีเอ็นเอแม่แบบ (template strand) หรือสายแอนติเซนส์ (antisense strand) โดยในการถอดรหัสของยีนนั้นๆ ไปเป็นเอ็มอาร์เอ็นเอ อาร์เอ็นเอพอลิเมอเรส (RNA polymerase) หรือ RNAP ในรูปที่ 1.29 จะเคลื่อนที่ไปในทิศทางเดียวกันกับสายกำหนดรหัสที่ยีนนั้นอยู่ เช่น ถ้ายีนอยู่บนสายบวก อาร์เอ็นเอพอลิเมอเรส

จะเคลื่อน จาก 5’ ไป 3’ โดยการอ่านและสร้างสายนิวคลีโอไทด์ของเอ็มอาร์เอ็นเอ (เส้นสีน้ำเงิน) จากสายที่เป็นดีเอ็นเอแม่แบบซึ่งมีทิศทาง 3’ ไป 5’ จึงได้ลำดับเบสของเอ็มอาร์เอ็นเออยู่ในทิศทาง 5’ ไป 3’ ตามสายกำหนดรหัส



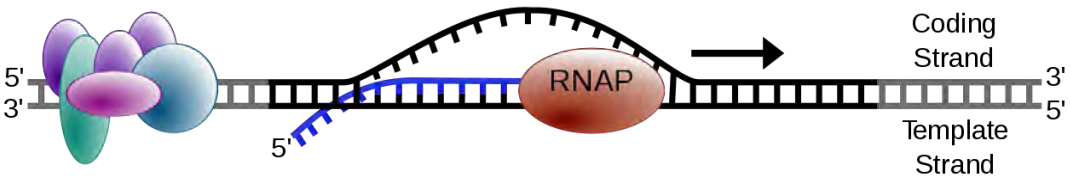
รูปที่ 1.27 โครงสร้างยีนของสิ่งมีชีวิตกลุ่มยูแคริโอต

(*ที่มา: รูปที่* 1 *ของ* [28])



รูปที่ 1.28 โครงสร้างยีนของสิ่งมีชีวิตกลุ่มโพรแคริโอต

(ที่มา: รูปที่ 2 ของ [28])

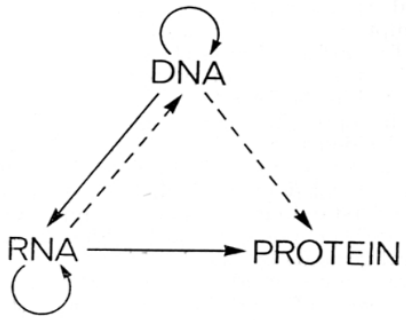


รูปที่ 1.29 การอ่านดีเอ็นเอโดยอาร์เอ็นเอพอลิเมอเรสเพื่อถอดรหัสยีนไปเป็นเอ็มอาร์เอ็นเอ

(*ที่มา:* Forluvoft, Public domain, via Wikimedia Commons. 2007. *Simple transcription elongation.* [*เข้าถึงออนไลน์เมื่อวันที่* 11 *ก.ค. พ.ศ.* 2564])

ความเชื่อตามหลักชีววิทยาระดับโมเลกุล

ฟรานซิส คริก (Francis Crick) อธิบายความเชื่อตามหลักชีววิทยาระดับโมเลกุล (central dogma of molecular biology) เป็นครั้งแรกในปี ค.ศ. 1958 [29] และนำเสนออีกครั้งในปี ค.ศ. 1970 [30] เกี่ยวกับกระบวนการถ่ายโอนข้อมูลรหัสพันธุกรรมระหว่างกรดนิวคลีอิก (nucleic acid) หรือจากกรดนิวคลีอิกไปเป็นโปรตีน แต่จะไม่มีการถ่ายโอนระหว่างโปรตีน หรือการถ่ายโอนย้อนกลับจากโปรตีนมายังกรดนิวคลีอิกดังแสดงในรูปที่ 1.30 ลูกศรเส้นทึบแสดงทิศทางหลักในการถ่ายโอนข้อมูลรหัสพันธุกรรม ลูกศรเส้นประแสดงทิศทางอื่นที่เป็นไปได้ และเส้นที่หายไปคือไม่สามารถเกิดขึ้นได้

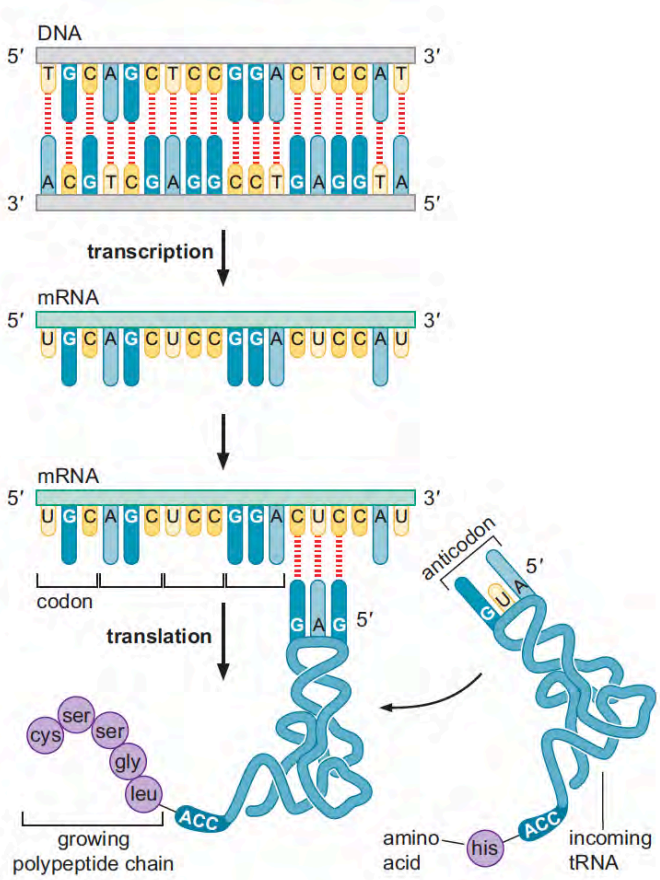


รูปที่ 1.30 การถ่ายโอนข้อมูลรหัสพันธุกรรมนำเสนอโดยฟรานซิส คริก ในปี ค.ศ. 1970

(*ที่มา*: *ภาพที่* 2 *ของ [30]*)

ความเชื่อตามหลักชีววิทยาระดับโมเลกุล [26] ถือเป็นพื้นฐานสำคัญในอณูชีววิทยาของสิ่งมีชีวิต หลักการทำงานในส่วนของการสร้างโปรตีนจากดีเอ็นเอ (รูปที่ 1.31) ประกอบด้วย 1) การถอดรหัส (transcription) จากสายดีเอ็นเอ (DNA) ไปเป็นเอ็มอาร์เอ็นเอ โดยอาร์เอ็นเอพอลิเมอเรส เบสที่เป็นไทมีน (T) เดิมจะเปลี่ยนเป็นนิวคลีโอไทด์ใหม่เรียกว่ายูราซิล (uracil) และแทนด้วยตัวอักษร “U” 2) การแปลรหัส (translation) จากเอ็มอาร์เอ็นเอไปเป็นโปรตีน โดยไรโบโซม (ribosome) ทำหน้าที่เป็นตัวจับและอ่านข้อมูลจากเอ็มอาร์เอ็นเอในทิศทางจาก 5’ ไปยัง 3’ เพื่อกำหนดกรดแอมิโนที่ต้องการ โดยมีอาร์เอ็นเอถ่ายโอน (transfer RNA) หรือทีอาร์เอ็นเอ (tRNA) ทำหน้าที่ช่วยขนย้ายกรดแอมิโนที่ต้องการนั้นมายังไรโบโซมและเชื่อมต่อกรดแอมิโนที่นำมาเข้ากับสายพอลิเพปไทด์

(polypeptide) หรือโปรตีน ทั้งนี้แต่ละสามเบสที่อยู่ติดกันในเอ็มอาร์เอ็นเอจะนับเป็นหนึ่งหน่วยโคดอน (codon) ซึ่งแต่ละโคดอนจะถูกแปลรหัสไปเป็นหนึ่งกรดแอมิโนตามรูปที่ 1.25



รูปที่ 1.31 กระบวนการถอดรหัสและแปลรหัส

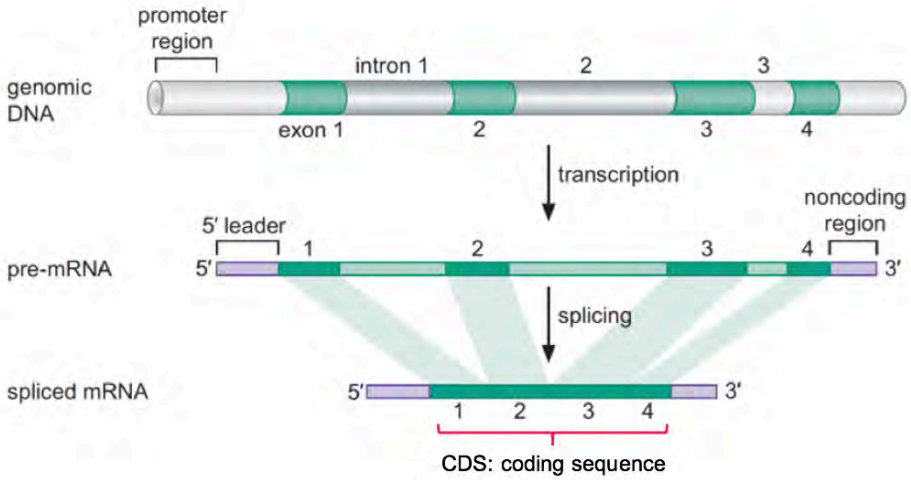
(ที่มา: รูปที่ 2-15 หน้า 36 [26])

การตัดเชื่อมอาร์เอ็นเอ

การถอดรหัสสิ่งมีชีวิตกลุ่มยูแคริโอตมีรายละเอียดเพิ่มเติมดังแสดงในรูปที่ 1.32 โดยผลการถอดรหัสขั้นต้นคือ เอ็มอาร์เอ็นเอแรกสร้าง (pre-mRNA) ที่ยังมีอินทรอนประกอบอยู่จะผ่านการตัดเชื่อมอาร์เอ็นเอ (RNA splicing) โดยเอกซอนของเอ็มอาร์เอ็นเอแรกสร้างจะถูกเลือกมาต่อกันให้เป็นเอ็มอาร์เอ็นเอที่ถูกตัดเชื่อมแล้ว (spliced mRNA) ซึ่งเป็นเอ็มอาร์เอ็นเอสมบูรณ์ (mature mRNA) มีรหัสเริ่มต้นคือ (“AUG”) และปิดท้ายด้วยรหัสหยุด (“UGA”, “UAG”, หรือ “UAA”) ที่พร้อมใช้แปลรหัสเป็นโปรตีนต่อไป

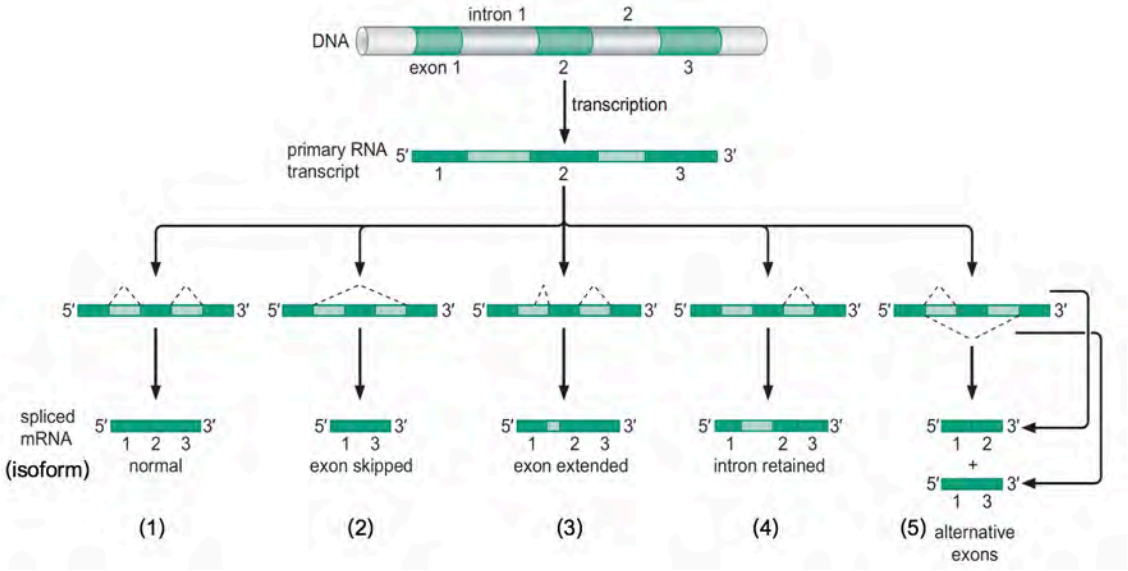
ทั้งนี้กระบวนการตัดเชื่อมอาร์เอ็นเอนี้ เอกซอนที่อยู่ในเอ็มอาร์เอ็นเอแรกสร้างอาจถูกเลือกมาต่อไม่ครบทำให้สามารถมีเอ็มอาร์เอ็นเอที่พร้อมนำไปแปลรหัสเป็นโปรตีนได้มากกว่า 1 แบบ (มากกว่า 1 ไอโซฟอร์ม) จากยีนเดียวกัน นอกจากนี้ในบางกรณีส่วนที่เป็นอินทรอนอาจยังถูกนำมาต่อเป็นส่วนของเอ็มอาร์เอ็นเอที่จะถูกนำไปแปล

รหัสดังแสดงในรูปที่ 1.33 โดยเอ็มอาร์เอ็นเอที่ถูกตัดเชื่อมแล้วรูปแบบที่ 3 และ 4 ยังมีส่วนของอินทรอนประกอบอยู่



รูปที่ 1.32 โครงสร้างพื้นฐานของยีนในกลุ่มยูแคริโอต

(*ที่มา*: *รูปที่* 14-1 *หน้า* 468 *[26] โดยเพิ่มการระบุบริเวณที่เป็น* CDS)



รูปที่ 1.33 ความหลากหลายของรูปแบบการตัดเชื่อมอาร์เอ็นเอ

(*ที่มา*: *รูปที่* 14-15 *หน้า* 484 *[26] โดยเพิ่มเลขกำกับไอโซฟอร์ม*)

รูปที่ 1.34 แสดงตัวอย่างลำดับเบสของลำดับกำหนดรหัส (conding sequence) ที่มีเฉพาะส่วนเอกซอนที่ถูกนำมาต่อกันของยีน *BRCA1* ที่เกี่ยวข้องกับมะเร็งเต้านม (ไม่รวมส่วนหน้าของรหัสเริ่มต้นและส่วนตามหลังของรหัสหยุด) โดยอยู่ในรูปแบบฟาสต้า (FASTA format) ซึ่งประกอบด้วย 2 บรรทัด บรรทัดแรกขึ้นต้นด้วยอักขระ “>” เสมอ จากนั้นตามด้วยคำอธิบายข้อมูล อาจเป็นชื่อยีน โปรตีนและรายละเอียด หรืออาจเป็นรหัสตัวเลข โดย

ไม่มีข้อจำกัดในส่วนนี้ บรรทัดที่สองเป็นลำดับเบสของลำดับกำหนดรหัส รูปที่ 1.35 แสดงตัวอย่างลำดับกรดแอมิโนที่ถูกแปลรหัสมาจากลำดับกำหนดรหัสของยีน *BRCA1* โดยอยู่ในรูปแบบฟาสต้า (FASTA format) เช่นกัน

```
>ENA|U64805|U64805.1 Homo sapiens Brca1-delta11b (Brca1) mRNA
ATGGATTTATCTGCTCTTCGCGTTGAAGAAGTACAAAATGTCATTAATGCTATGCAGAAA
ATCTTAGAGTGTCCCATCTGTCTGGAGTTGATCAAGGAACCTGTCTCCACAAAGTGTGAC
CACATATTTTGCAAATTTTGCATGCTGAAACTTCTCAACCAGAAGAAAGGGCCTTCACAG
TGTCCTTTATGTAAGAATGATATAACCAAAAGGAGCCTACAAGAAAGTACGAGATTTAGT
CAACTTGTTGAAGAGCTATTGAAAATCATTTGTGCTTTTCAGCTTGACACAGGTTTGGAG
TATGCAAACAGCTATAATTTTGCAAAAAAGGAAAATAACTCTCCTGAACATCTAAAAGAT
GAAGTTTCTATCATCCAAAGTATGGGCTACAGAAACCGTGCCAAAAGACTTCTACAGAGT
GAACCCGAAAATCCTTCCTTGCAGGAAACCAGTCTCAGTGTCCAACTCTCTAACCTTGGA
ACTGTGAGAACTCTGAGGACAAAGCAGCGGATACAACCTCAAAAGACGTCTGTCTACATT
GAATTGGGATCTGATTCTTCTGAAGATACCGTTAATAAGGCAACTTATTGCAGTGTGGGA
GATCAAGAATTGTTACAAATCACCCCTCAAGGAACCAGGGATGAAATCAGTTTGGATTCT
GCAAAAAAGGCTGCTTGTGAATTTTCTGAGACGGATGTAACAAATACTGAACATCGTCAA
CCCAGTAATAATGATTTGAACACCACTGAGAAGCGTGTAGCTGAGAGGCATCCAGAAAAG
TATCAGGGTGAAGCAGCATCTGGGTGTGAGAGTGAAACAAGCGTCTCTGAAGACTGCTCA
GGGCTATCCTCTCAGAGTGACATTTTAACCACTCAGCAGAGGGATACCATGCAACATAAC
CTGATAAAGCTCCAGCAGGAAATGGCTGAACTAGAAGCTGTGTTAGAACAGCATGGGAGC
CAGCCTCCTAACAGCTACCCTTCCATCATAAGTGACTCCTCTGCCCTTGAGGACCTGCGA
AATCCAGAACAAAGCACATCAGAAAAAGTATTAACTTCACAGAAAAGTAGTGAATACCCT
ATAAGCCAGAATCCAGAAGGCCTTTCTGCTGACAAGTTTGAGGTGTCTGCAGATAGTTCT
ACCAGTAAAAATAAAGAACCAGGAGTGGAAAGGTCATCCCCCTTCTAAATGCCCATCATTA
GATGATAGGTGGTACATGCACAGTTGCTCTGGGAGTCTTCAGAATAGAAACTACCCATCT
CAAGAGGAGCTCATTAAGGTTGTTGATGTGGAGGAGCAACAGCTGGAAGAGTCTGGGCCA
CACGATTTGACGGAAACATCTTACTTGCCAAGGCAAGATCTAGAGGGAACCCCTTACCTG
GAATCTGGAATCAGCCTCTTCTCTGATGACCCTGAATCTGATCCTTCTGAAGACAGAGCC
CCAGAGTCAGCTCGTGTTGGCAACATACCATCTTCAACCTCTGCATTGAAAGTTCCCCAA
TTGAAAGTTGCAGAATCTGCCCAGGGTCCAGCTGCTGCTCATACTACTGATACTGCTGGG
TATAATGCAATGGAAGAAAGTGTGAGCAGGGAGAAGCCAGAATTGACAGCTTCAACAGAA
AGGGTCAACAAAAGAATGTCCATGGTGGTGTCTGGCCTGACCCCAGAAGAATTTATGCTC
GTGTACAAGTTTGCCAGAAAACACCACATCACTTTAACTAATCTAATTACTGAAGAGACT
ACTCATGTTGTTATGAAAACAGATGCTGAGTTTGTGTGTGAACGGACACTGAAATATTTT
CTAGGAATTGCGGGAGGAAAATGGGTAGTTAGCTATTTCTGGGTGACCCAGTCTATTAAA
GAAAGAAAAATGCTGAATGAGCATGATTTTGAAGTCAGAGGAGATGTGGTCAATGGAAGA
AACCACCAAGGTCCAAAGCGAGCAAGAGAATCCCAGGACAGAAAGATCTTCAGGGGGCTA
GAAATCTGTTGCTATGGGCCCTTCACCAACATGCCCACAGATCAACTGGAATGGATGGTA
CAGCTGTGTGGTGCTTCTGTGGTGAAGGAGCTTTCATCATTCACCCTTGGCACAGGTGTC
CACCCAATTGTGGTTGTGCAGCCAGATGCCTGGACAGAGGACAATGGCTTCCATGCAATT
GGGCAGATGTGTGAGGCACCTGTGGTGACCCGAGAGTGGGTGTTGGACAGTGTAGCACTC
TACCAGTGCCAGGAGCTGGACACCTACCTGATACCCCAGATCCCCCACAGCCACTACTGA
```

รูปที่ 1.34 ลำดับกำหนดรหัสยีน BRCA1 ในรูปแบบฟาสต้า

(ที่มา: https://www.ebi.ac.uk/ena/browser/api/fasta/U64805.1?lineLimit=1000)

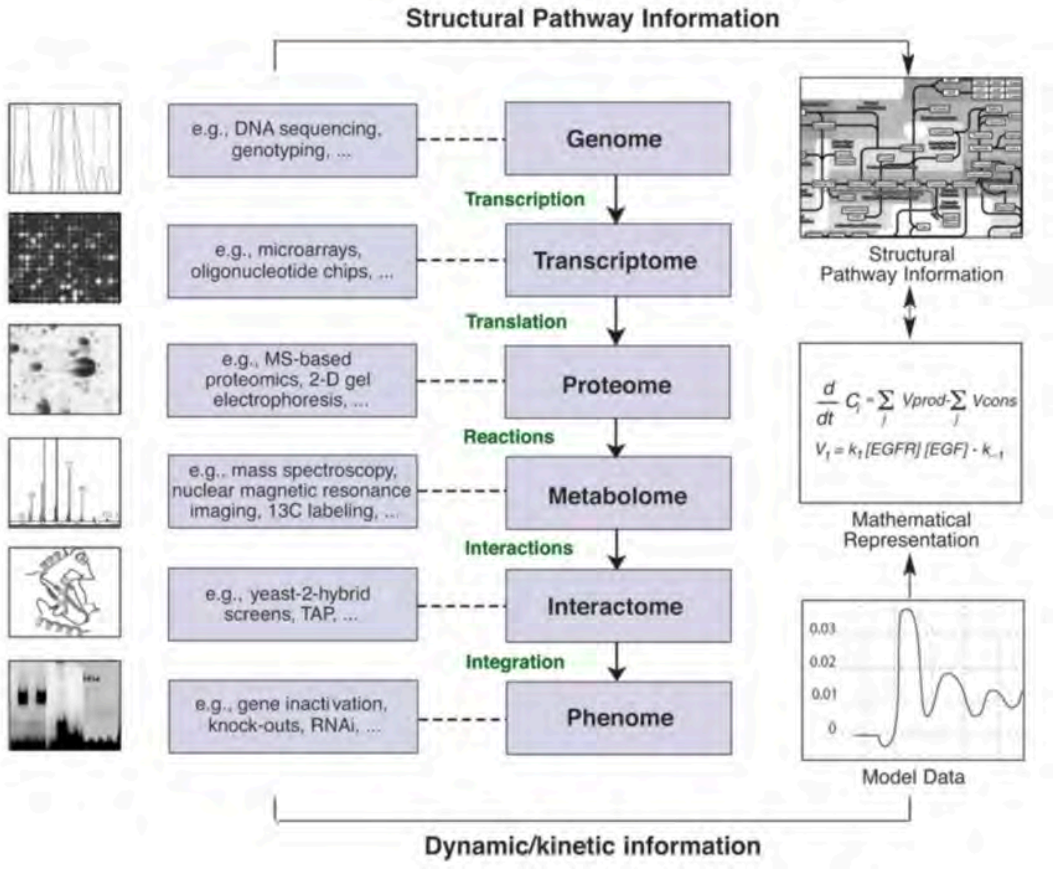
```
>sp|P38398|BRCA1_HUMAN Breast cancer type 1 susceptibility pr
MDLSALRVEEVQNVINAMQKILECPICLELIKEPVSTKCDHIFCKFCMLKLLNQKKGPSQ
CPLCKNDITKRSLQESTRFSQLVEELLKIICAFQLDTGLEYANSYNFAKKENNSPEHLKD
EVSIIQSMGYRNRAKRLLQSEPENPSLQETSLSVQLSNLGTVRTLRTKQRIQPQKTSVYI
ELGSDSSEDTVNKATYCSVGDQELLQITPQGTRDEISLDSAKKAACEFSETDVTNTEHHQ
PSNNDLNTTEKRAAERHPEKYQGSSVSNLHVEPCGTNTHASSLQHENSSLLLTKDRMNVE
KAEFCNKSKQPGLARSQHNRWAGSKETCNDRRTPSTEKKVDLNADPLCERKEWNKQKLPC
SENPRDTEDVPWITLNSSIQKVNEWFSRSDELLGSDDSHDGESESNAKVADVLDVLNEVD
EYSGSSEKIDLLASDPHEALICKSERVHSKSVESNIEDKIFGKTYRKKASLPNLSHVTEN
LIIGAFVTEPQIIQERPLTNKLKRKRRPTSGLHPEDFIKKADLAVQKTPEMINQGTNQTE
QNGQVMNITNSGHENKTKGDSIQNEKNPNPIESLEKESAFKTKAEPISSSISNMELELNI
HNSKAPKKNRLRRKSSTRHIHALELVVSRNLSPPNCTELQIDSCSSSEEIKKKKYNQMPV
RHSRNLQLMEGKEPATGAKKSNKPNEQTSKRHDSDTFPELKLTNAPGSFTKCSNTSELKE
FVNPSLPREEKEEKLETVKVSNNAEDPKDLMLSGERVLQTERSVESSSISLVPGTDYGTQ
ESISLLEVSTLGKAKTEPNKCVSQCAAFENPKGLIHGCSKDNRNDTEGFKYPLGHEVNHS
RETSIEMEESELDAQYLQNTFKVSKRQSFAPFSNPGNAEEECATFSAHSGSLKKQSPKVT
FECEQKEENQGKNESNIKPVQTVNITAGFPVVGQKDKPVDNAKCSIKGGSRFCLSSQFRG
NETGLITPNKHGLLQNPYRIPPLFPIKSFVKTKCKKNLLEENFEEHSMSPEREMGNENIP
STVSTISRNNIRENVFKEASSSNINEVGSSTNEVGSSINEIGSSDENIQAELGRNRGPKL
NAMLRLGVLQPEVYKQSLPGSNCKHPEIKKQEYEEVVQTVNTDFSPYLISDNLEQPMGSS
HASQVCSETPDDLLDDGEIKEDTSFAENDIKESSAVFSKSVQKGELSRSPSPFTHTHLAQ
GYRRGAKKLESSEENLSSEDEELPCFQHLLFGKVNNIPSQSTRHSTVATECLSKNTEENL
LSLKNSLNDCSNQVILAKASQEHHLSEETKCSASLFSSQCSELEDLTANTNTQDPFLIGS
SKQMRHQSESQGVGLSDKELVSDDEERGTGLEENNQEEQSMDSNLGEAASGCESETSVSE
DCSGLSSQSDILTTQQRDTMQHNLIKLQQEMAELEAVLEQHGSQPSNSYPSIISDSSALE
DLRNPEQSTSEKAVLTSQKSSEYPISQNPEGLSADKFEVSADSSTSKNKEPGVERSSPSK
CPSLDDRWYMHSCSGSLQNRNYPSQEELIKVVDVEEQQLEESGPHDLTETSYLPRQDLEG
TPYLESGISLFSDDPESDPSEDRAPESARVGNIPSSTSALKVPQLKVAESAQSPAAAHTT
DTAGYNAMEESVSREKPELTASTERVNKRMSMVVSGLTPEEFMLVYKFARKHHITLTNLI
TEETTHVVMKTDAEFVCERTLKYFLGIAGGKWVVSYFWVTQSIKERKMLNEHDFEVRGDV
VNGRNHQGPKRARESQDRKIFRGLEICCYGPFTNMPTDQLEWMVQLCGASVVKELSSFTL
GTGVHPIVVVQPDAWTEDNGFHAIGQMCEAPVVTREWVLDSVALYQCQELDTYLIPQIPH
SHY
```

รูปที่ 1.35 ลำดับกรดแอมิโนที่แปลรหัสจากลำดับกำหนดรหัสยีน BRCA1 ในรูปแบบฟาสต้า

(ที่มา: https://www.uniprot.org/uniprot/P38398.fasta)

เทคโนโลยีโอมิกส์

เทคโนโลยีโอมิกส์ (omics technology) [31] (รูปที่ 1.36) เป็นการศึกษารายละเอียดในองค์รวมเกี่ยวกับโมเลกุลต่างๆ ที่ประกอบขึ้นเป็นเซลล์ เนื้อเยื่อ อวัยวะ และสิ่งมีชีวิต โดยประกอบด้วยเทคโนโลยีจีโนมิกส์ (genomics) ใช้ในการศึกษาจีโนม (genome) หรือรหัสพันธุกรรมทั้งหมดในโครโมโซมพื้นฐาน 1 ชุด ศึกษาเอกโซม (exome) หรือรหัสพันธุกรรมทั้งหมดเฉพาะส่วนที่เป็นเอกซอน เทคโนโลยีทรานสคริปโทมิกส์ (transcriptomics) ใช้ในการศึกษาทรานสคริปโทม (transcriptome) หรือปริมาณอาร์เอ็มเอนำรหัส (messenger RNA) ของยีนทั้งหมดที่แสดงออก เทคโนโลยีโพรทีโอมิกส์ (proteomics) ใช้ในการศึกษาโปรตีโอม (proteome) หรือโปรตีนทั้งหมดที่แสดงออก และเทคโนโลยีเมแทบอลอมิกส์ (metabolomics) ใช้ในการศึกษาเมแทบอโลม (metabolome) หรือเมแทบอไลต์ (metabolite) ทั้งหมดที่แสดงออก โดยทรานสคริปโทม โปรตีโอม และเมแทบอโลม มีความจำเพาะกับประเภทของเซลล์ เนื้อเยื่อ และเงื่อนไขในการตรวจวัด และมีความเป็นพลวัตของปริมาณที่ตรวจวัดได้ตามเวลาที่เปลี่ยนไป

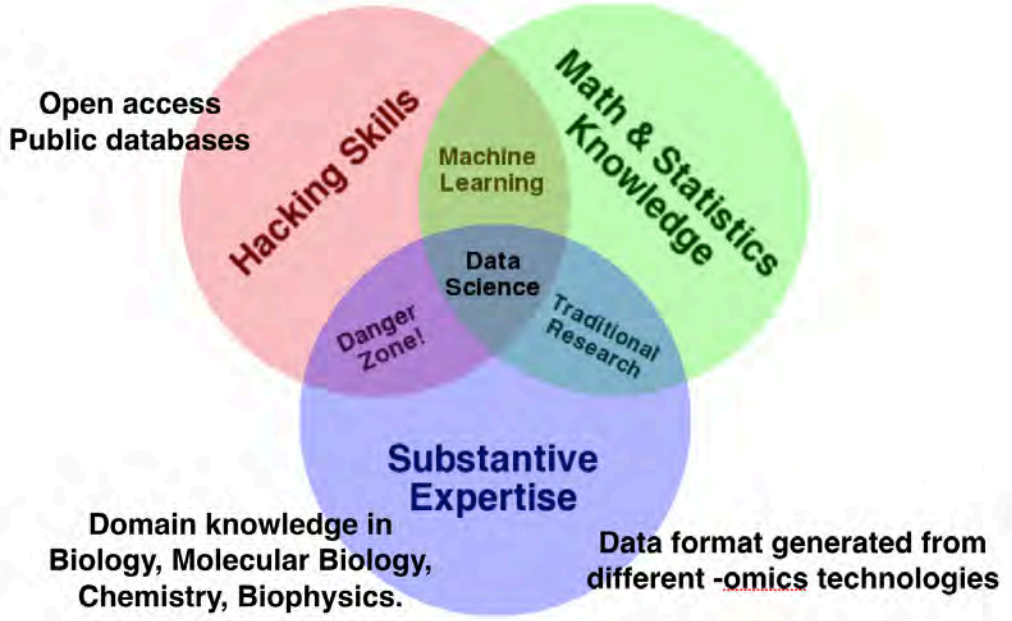


รูปที่ 1.36 เทคโนโลยีโอมิกส์

(*ที่มา: รูปที่ 2 ของ [32]*)

วิทยาศาสตร์ข้อมูลทางชีววิทยา

รูปที่ 1.37 แสดงแผนภาพเวนน์ที่นำเสนอโดย ดรูว์ คอนเวย์ (Drew Conway) เกี่ยวกับองค์ความรู้และทักษะที่จำเป็นในการศึกษาและทำงานทางด้านวิทยาศาสตร์ข้อมูล ซึ่งประกอบด้วย 3 ส่วนหลักคือ (1) องค์ความรู้ทางคณิตศาสตร์และสถิติ (2) ความสามารถในการได้มาซึ่งข้อมูลและการจัดการข้อมูลให้อยู่ในรูปแบบที่เหมาะสม และ (3) ความเข้าใจในข้อมูลที่จำเพาะกับโดเมน (domain) เช่น ข้อมูลการตลาด ข้อมูลความเชื่อมโยงของเครือข่ายออนไลน์ ข้อมูลผลข้างเคียงของยาแต่ละชนิด โดยเข้าใจและสามารถอธิบายความหมายได้ ทั้งนี้งานทางด้านวิทยาศาสตร์ข้อมูลทางชีววิทยาสามารถอธิบายโดยใช้แผนภาพเวนน์ของคอนเวย์เช่นกัน โดยต้องการ (1) องค์ความรู้ทางคณิตศาสตร์และสถิติ (2) ความสามารถในการได้มาซึ่งข้อมูลและการจัดการข้อมูลให้อยู่ในรูปแบบที่เหมาะสม เช่น ทราบแหล่งข้อมูล ความน่าเชื่อถือของแต่ละฐานข้อมูล ช่องทางในการเข้าถึงข้อมูล เช่น สามารถดาวน์โหลดข้อมูลได้ทั้งชุดหรือเข้าถึงได้เป็นรายการผ่าน API (Application Programming Interface) และ (3) ความเข้าใจกระบวนการและข้อมูลทางชีววิทยาและอณูชีววิทยาที่เกี่ยวข้องกับโจทย์วิจัยหรือปัญหาที่ต้องการหาคำตอบ ลักษณะของข้อมูลที่เกิดจากเทคโนโลยีโอมิกส์ต่างๆ และความหมายของข้อมูล เป็นต้น

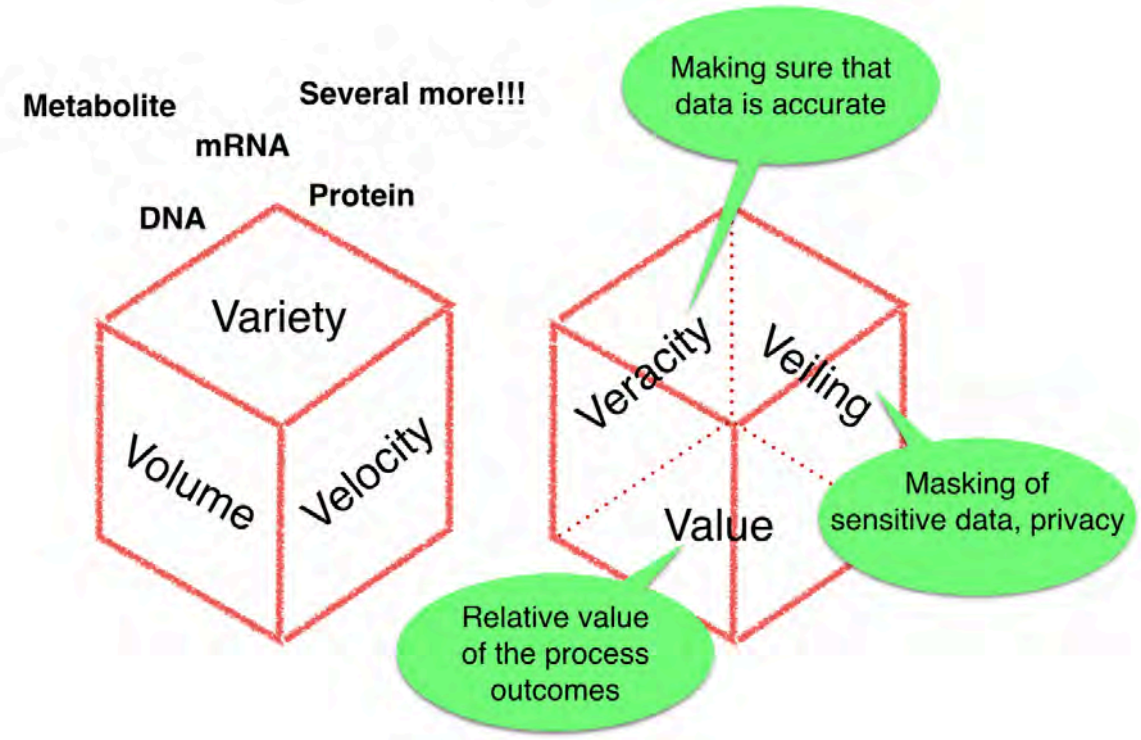


รูปที่ 1.37 แผนภาพเวนน์ของดรูว์ คอนเวย์แสดงองค์ความรู้ที่จำเป็นต่อการทำงานด้านวิทยาศาสตร์ข้อมูล (ที่มา: Convey, D. *The Data Science Venn Diagram*. [ONLINE] Available at: http://drewconway.com [เข้าถึงออนไลน์เมื่อวันที่ 11 ก.ค. พ.ศ. 2564]) *โดยปรับเพิ่มเติมตัวอย่างที่เกี่ยวข้องกับข้อมูลทางชีววิทยา*

ข้อมูลมหัตกับชีวสารสนเทศ

ข้อมูลมหัตหรือบิกดาต้า (big data) มีลักษณะหลักที่ต้องพิจารณา 3 ประการ (รูปที่ 1.38) คือ ขนาดของข้อมูล (volume) ความหลากหลายของข้อมูล (variety) และอัตราการเกิดข้อมูลใหม่ของข้อมูล (velocity) ตัวอย่างข้อมูล

ขนาดใหญ่ เช่น ข้อมูลโพสของเฟซบุ๊ก (Facebook) โดยข้อมูลมีลักษณะที่หลากหลายไม่จำกัดเฉพาะตัวอักษรและประโยค แต่รวมถึงวีดีโอ คลิปเสียง และข้อมูลเหล่านี้มีจำนวนเพิ่มขึ้นอย่างมากในทุกวันจากทั่วโลก ในกรณีของข้อมูลโอมิกส์ขนาดข้อมูลโดยเฉพาะข้อมูลจีโนม (genome) และเอกโซม (exome) มีจำนวนเพิ่มขึ้นอย่างรวดเร็ว (รูปที่ 1.39) ด้วยราคาของเทคโนโลยีการหาลำดับเบสที่ถูกลงอย่างมากเมื่อเทียบกับในอดีต (รูปที่ 1.40) ในเชิงความหลากหลายของข้อมูลนอกจากข้อมูลรหัสพันธุกรรมแล้ว ยังมีข้อมูลจากเทคโนโลยีโอมิกส์อื่นๆ เช่น ข้อมูลการแสดงออกของยีนผ่านเทคโนโลยีการหาลำดับเบสของอาร์เอ็นเอทั้งหมด (RNA sequencing; RNA-seq) ที่มีข้อมูลลักษณะเดียวกับลำดับเบสดีเอ็นเอ ข้อมูลไมโครอาร์เรย์ที่มีลักษณะเป็นตารางสองมิติโดยแต่ละบรรทัดแสดงข้อมูลของหนึ่งยีนและแต่ละคอลัมน์แสดงปริมาณการแสดงออกของยีนนั้นตามเงื่อนไขการทดลองจำเพาะ ข้อมูลการแสดงออกของโปรตีนจากเทคนิคแมสสเปกโทรเมตรี (mass spectrometry) ที่อยู่ในรูปแบบของยอด (peak) จำนวนมากในกราฟ ข้อมูลการแสดงออกของเมแทบอไลต์ เป็นต้น นอกจากองค์ประกอบหลัก 3 ส่วนข้างต้นแล้ว ระยะหลังมีการเพิ่มเติมการพิจารณาข้อมูลในอีก 3 มิติ คือ ความถูกต้องของข้อมูล (veracity) คุณค่าของข้อมูล (value) และการรักษาสิทธิส่วนบุคคลของข้อมูล (veilling) ข้อมูลรหัสพันธุกรรมเป็นตัวอย่างสำคัญที่จำเป็นต้องพิจารณามิติเหล่านี้ด้วย



รูปที่ 1.38 องค์ประกอบ 3 วี (Vs) ของข้อมูลขนาดใหญ่ในบริบทของข้อมูลทางชีวสารสนเทศ

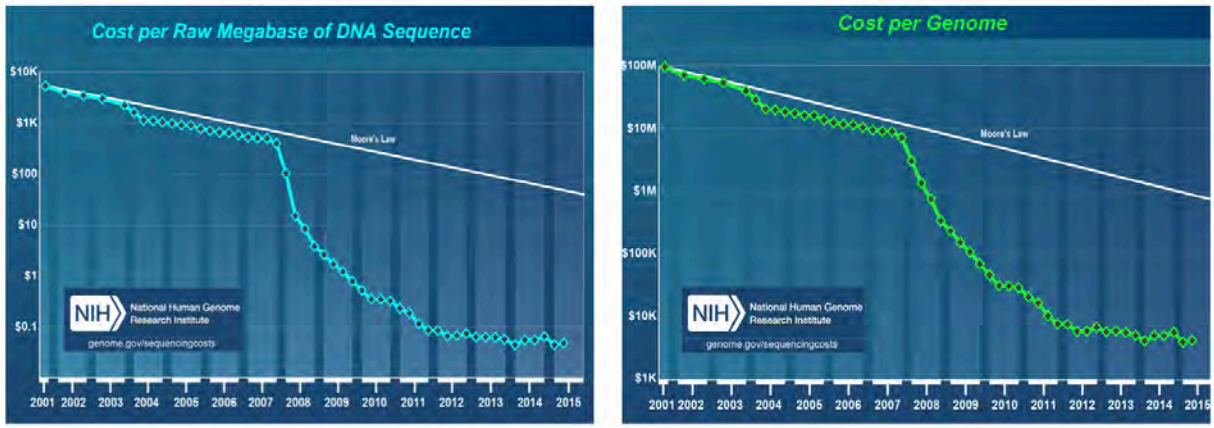
รูปที่ 1.39 แสดงจำนวนเบสและลำดับเบสที่เพิ่มขึ้นอย่างก้าวกระโดดในฐานข้อมูล GenBank ในช่วง 25 ปีที่ผ่านมา โดยเฉพาะหลังการตีพิมพ์จีโนมมนุษย์เป็นครั้งแรกในปี ค.ศ. 2001 รูปที่ 1.40 แสดงราคาต่อเบสและต่อจีโนมที่ลดลงอย่างมากในช่วง 10 ปีที่ผ่านมา ซึ่งอัตราที่ลดลงนี้ลดลงมากกว่ากฎของมัวร์ (Moore's law)

GENBANK AND WGS STATISTICS

		GenBank		WGS	
Release	Date	Bases	Sequences	Bases	Sequences
3	Dec 1982	680338	606		
14	Nov 1983	2274029	2427		
20	May 1984	3002088	3665		
24	Sep 1984	3323270	4135		
25	Oct 1984	3368765	4175		
129	Apr 2002	19072679701	16769983	692266338	172768
130	Jun 2002	20648748345	17471130	3267608441	397502
131	Aug 2002	22616937182	18197119	3848375582	427771
132	Oct 2002	26525934656	19808101	3892435593	434224
176	Feb 2010	112326229652	116461672	163991858015	57134273
177	Apr 2010	114348888771	119112251	165536009514	58361599
178	Jun 2010	115624497715	120604423	167725292032	58592700
179	Aug 2010	117476523128	122941883	169253846128	58994334
180	Oct 2010	118551641086	125764384	175339059129	59397637
181	Dec 2010	122082812719	129902276	177385297156	59608311
228	Oct 2018	279668290132	209656636	3444172142207	722438528
229	Dec 2018	285688542186	211281415	3656719423096	773773190
230	Feb 2019	303709510632	212260377	4164513961679	945019312
231	Apr 2019	321680566570	212775414	4421986382065	993732214
232	Jun 2019	329835282370	213383758	4847677297950	1022913321

รูปที่ 1.39 จำนวนเบสและลำดับเบสที่เพิ่มขึ้นในฐานข้อมูล GenBank

(ที่มา: http://www.ncbi.nlm.nih.gov/genbank/statistics โดยเลือกเพียงบางส่วน)



รูปที่ 1.40 ค่าใช้จ่ายต่อเบสและจีโนมที่ลดลงเป็นอย่างมากตั้งแต่ปี ค.ศ. 2007

(ที่มา: http://www.genome.gov/sequencingcosts/)

จีโนมิกส์บนคลาวด์

ปริมาณข้อมูลจีโนมเพิ่มขึ้นอย่างมากด้วยราคาที่ถูกลง บริษัทให้บริการคลาวด์อย่างกูเกิ้ลและแอมะซอนมีบริการกูเกิ้ลจีโนมิกส์ (Google Genomics) (https://cloud.google.com/genomics/) และ Genomics in the Cloud (https://aws.amazon.com/health/genomics/) ตามลำดับ โดยทั้งสองบริษัท มีไปปไลน์พื้นฐานพร้อมใช้สำหรับวิเคราะห์ข้อมูลจีโนม มีสำเนาข้อมูลหลักจากฐานข้อมูลสาธารณะ เช่น ข้อมูลจากโครงการ 1000 จีโนม ข้อมูลจีโนมอ้างอิง ข้อมูลจาก The Cancer Genome Atlas (TCGA) เป็นต้น เพื่อสนับสนุนการวิเคราะห์ข้อมูลจีโนม เอกโซม ได้โดยสะดวก ในกรณีของกูเกิ้ลจีโนมิกส์ ผู้ใช้สามารถวิเคราะห์การแปรผันในรหัสพันธุกรรมโดยใช้ BigQuery นอกจากกูเกิ้ลและแอมะซอนซึ่งนักวิจัยสามารถใช้โครงสร้างพื้นฐานเพื่อการวิเคราะห์และประมวลผลข้อมูลจีโนมมิกส์แล้ว บริษัทอิลูมินามีบริการอิลูมินาเบสสเปซ (Illumina BaseSapce: https://basespace.illumina.com/home/index) ซึ่งเป็นคลาวด์ของบริษัทอิลูมินาโดยลูกค้าของบริษัทสามารถเข้าถึงและวิเคราะห์ข้อมูลรหัสพันธุกรรมที่ได้จากการหาลำดับเบสโดยบริษัท นอกจากนี้ยังมีบริการจากบริษัท Seven Bridges Genomics (https://www.sevenbridges.com/) ที่ได้รับเงินสนับสนุนจากสถาบันมะเร็งแห่งชาติ (National Cancer Institute; NCI) ภายใต้สถาบันสุขภาพแห่งชาติ (National Institutes of Health) หรือเอ็นไอเอช (NIH) ในการจัดเตรียมโครงสร้างพื้นฐานบนคลาวด์ชื่อ CGC หรือ Cancer Genomics Cloud [33] (http://www.cancergenomicscloud.org) เพื่อใช้วิเคราะห์ข้อมูลจีโนมิกส์ของโรคมะเร็งโดยมีข้อมูลหลักจาก TCGA (The Cancer Genome Atlas: https://cancergenome.nih.gov) ภายใต้เอ็นไอเอช โดยทาง CGC มีการเตรียมไปป์ไลน์พื้นฐานเพื่อการวิเคราะห์ข้อมูล ผู้ใช้สามารถอัพโหลดข้อมูลเพิ่มเติมเพื่อวิเคราะห์ร่วมกับข้อมูลจาก TCGA ได้ นอกจากนี้ยังมี Canadian Genomics Cloud (https://genomicscloud.ca) ที่เป็นแพลตฟอร์มสาธารณะเพื่อการวิเคราะห์ข้อมูลจีโนมิกส์และข้อมูลเชิงคลีนิกสำหรับนักวิทยาศาสตร์และนักวิจัยของประเทศแคนาดา เป็นต้น

ตัวอย่างฐานข้อมูลสาธารณะ

เอ็นซีบีไอ

เอ็นซีบีไอ (NCBI; National Center for Biotechnology Information) (https://www.ncbi.nlm.nih.gov) ก่อตั้งเมื่อวันที่ 4 พฤศจิกายน ค.ศ. 1988 (พ.ศ. 2531) โดยเป็นส่วนหนึ่งของหอสมุดแพทย์แห่งชาติอเมริกา (National Library of Medicine) หรือเอ็นแอลเอ็ม (NLM) ภายใต้เอ็นไอเอช (NIH) เพื่อเป็นแหล่งข้อมูลสารสนเทศสนับสนุนงานวิจัยและพัฒนาทางด้านการแพทย์และเทคโนโลยีชีวภาพ เอ็นซีบีไอเป็นฐานข้อมูลสาธารณะขนาดใหญ่ที่มีการอ้างอิงเป็นระดับต้นๆ ของโลก ประกอบด้วยฐานข้อมูลจำเพาะที่สำคัญหลายฐาน เช่น ฐานข้อมูลนิวคลีโอไทด์ (Nucleotide) เก็บลำดับเบสของสายดีเอ็นเอ ฐานข้อมูลโปรตีน (Protein) เก็บลำดับกรดแอมิโน ฐานข้อมูลจีโนม (Genome) เก็บลำดับเบสดีเอ็นเอของจีโนมสิ่งมีชีวิตต่างๆ ฐานข้อมูลการแสดงออกของยีน (GEO DataSets, GEO profiles) ฐานข้อมูลสนิป (dbSNP) ฐานข้อมูลการแปรผันในกลุ่มเบส (dbVar) เช่น การเกิดส่วนของดีเอ็นเอชุดซ้ำ

เกิดการสอดแทรกหรือการขาดหายไปของลำดับเบสในโครโมโซม เกิดการกลับด้านของลำดับเบส โดยข้อมูลในฐานข้อมูลเหล่านี้รวบรวมจากงานตีพิมพ์ในวารสารวิชาการที่จัดเก็บในฐานข้อมูลผับเมด (PubMed) ซึ่งมีมากกว่า 27 ล้านรายการ (เข้าถึงออนไลน์เมื่อวันที่ 30 กันยายน พ.ศ. 2560) โดยผับเมดเป็นฐานข้อมูลหลัก ฐานข้อมูลหนึ่งภายใต้เอ็นซีบีไอ

ยูนิพรอต

ยูนิพรอต (UniProt) (https://www.uniprot.org) เป็นฐานข้อมูลอ้างอิงหลักเกี่ยวกับข้อมูลโปรตีนอีกแหล่งนอกเหนือจากฐานข้อมูลโปรตีนที่เอ็นซีบีไอ โดยเก็บลำดับกรดแอมิโนและฟังก์ชันของสายโปรตีนของสิ่งมีชีวิตต่างๆ ภายในฐานข้อมูลมีการแบ่งข้อมูลออกเป็นสองส่วนหลัก คือ UniProt/Swiss-Prot และ UniProt/TrEMBL โดย UniProt/Swiss-Prot มีข้อมูลโปรตีนทั้งสิ้น 555,594 รายการ (เข้าถึงออนไลน์เมื่อวันที่ 30 กันยายน พ.ศ. 2560) โดยเป็นข้อมูลที่มีการตรวจสอบด้วยมือและผ่านการทวนสอบแล้ว ส่วน UniProt/TrEMBL มีข้อมูล 90,050,711 รายการ ซึ่งมีจำนวนมากกว่า Swiss-Prot แต่ข้อมูลส่วนใหญ่ยังไม่ผ่านการตรวจสอบด้วยมือและการทวนสอบ

สารานุกรมขององค์ประกอบดีเอ็นเอ

สารานุกรมขององค์ประกอบดีเอ็นเอ (ENCODE; Encyclopedia of DNA Elements) ถูกสร้างขึ้นด้วยความร่วมมือของกลุ่มสถาบัน (consortium) โดยได้รับเงินทุนสนับสนุนจาก National Human Genome Research Institute (NHGRI) เป้าหมายหลักของโครงการ คือการสร้างองค์ความรู้เชิงลึกเกี่ยวกับฟังก์ชันขององค์ประกอบต่างๆ ในจีโนมมนุษย์ องค์ประกอบที่มีการทำงานในระดับอาร์เอ็นเอและโปรตีน และองค์ประกอบที่เกี่ยวข้องกับการควบคุมการแสดงออกของยีน ซึ่งข้อมูลเหล่านี้เปิดให้เข้าถึงโดยสาธารณะที่ https://www.encodeproject.org

ตัวอย่างฐานข้อมูลเปิดอื่นๆ

นอกจากตัวอย่างฐานข้อมูลหลักข้างต้น ยังมีฐานข้อมูลเปิดอีกมากมายซึ่งมีข้อมูลแตกต่างกันไป อาทิ ฐานข้อมูลจำเพาะกับสิ่งมีชีวิต เช่น ฐานข้อมูลจีโนมมนุษย์ที่ University of California, Santa Cruz (UCSC) (https://genome.ucsc.edu) ฐานข้อมูล GENCODE (https://www.gencodegenes.org) เก็บข้อมูลลำดับนิวคลีโอไทด์ ลำดับกรดแอมิโน รวมทั้งคำอธิบายประกอบ (annotation) ฟังก์ชันของยีน โปรตีนของมนุษย์และหนู (*Mus musculus*) ฐานข้อมูล TAIR (https://www.arabidopsis.org) เก็บข้อมูลรหัสพันธุกรรมและฟังก์ชันของยีนของ *Arabidopsis thaliana* ซึ่งเป็นพืชใบเลี้ยงคู่ต้นแบบในการหาลำดับเบส ฐานข้อมูลโปรตีนดาต้าแบงค์ (Protein Data Bank; RCSB PDB) (https://www.rcsb.org) เก็บข้อมูลโครงสร้าง 3 มิติของโปรตีนที่มีการทดลองจากห้องปฏิบัติการ ฐานข้อมูลกลุ่มโปรตีนหรือพีแฟม (Pfam: http://pfam.xfam.org) เก็บข้อมูลการจัดกลุ่มโปรตีนตามชุดของโปรตีนโดเมน (protein domain) ที่มีร่วมกัน โดยแต่ละโปรตีนโดเมนจะเป็นส่วนของสายโปรตีนที่มักมีฟังก์ชันการทำงานจำเพาะ ฐานข้อมูลกลุ่มอาร์เอ็นเอหรืออาร์แฟม (http://rfam.xfam.org) เก็บข้อมูลการจัดกลุ่ม

อาร์เอ็นเอตามการปรากฏร่วมกันของโครงสร้างหรือส่วนของโครงสร้าง 2 มิติ (secondary structure) ที่มักมีผลต่อฟังก์ชันการทำงานจำเพาะ ฐานข้อมูล KEGG Pathway เก็บข้อมูลพาธ์เวย์หรือชีววิถีของสิ่งมีชีวิตต่างๆ (https://www.genome.jp/kegg/pathway.html)

แบบฝึกหัดบทที่ 1

1. เขียนโปรแกรมเพื่อแก้ปัญหาโจทย์ที่โรซาลินด์ (http://rosalind.info) ดังต่อไปนี้
    1) GenBank Introduction (http://rosalind.info/problems/gbk/)
    2) Data Formats (http://rosalind.info/problems/frmt/)
    3) FASTQ Format Introduction (http://rosalind.info/problems/tfsq/)
    4) Read Quality Distribution (http://rosalind.info/problems/phre/)
    5) Read Filtration by Quality (http://rosalind.info/problems/filt/)
    6) Introduction to Protein Databases (http://rosalind.info/problems/dbpr/)
    7) Complementing a Strand of DNA (http://rosalind.info/problems/rvco/)
2. จากข้อ 1.6 ของโรซาลินด์ข้างต้น จงเขียนฟังก์ชัน `translate()` เอง โดยใช้ตารางโคดอนในรูปที่ 1.5
3. ไฟล์ในรูปแบบฟาสต้า (FASTA) มีลักษณะอย่างไร จงอธิบาย
4. ไฟล์ในรูปแบบฟาสคิว (FASTQ) มีลักษณะอย่างไร จงอธิบาย
5. จงยกตัวอย่างฐานข้อมูลสาธารณะ พร้อมตัวอย่างข้อมูลที่อยู่ในฐานข้อมูลเหล่านั้น

ภาคผนวกบทที่ 1

รูปแบบไฟล์ที่เก็บรหัสพันธุกรรมจากเครื่องหาลำดับเบส

ข้อมูลทางชีววิทยาที่ได้จากเทคโนโยลีโอมิกส์ต่างๆ รวมทั้งผลของการวิเคราะห์ข้อมูลเบื้องต้นมักอยู่ในรูปแบบแฟ้มข้อความ (text file) ที่มีโครงสร้างจำเพาะ สำหรับข้อมูลลำดับเบสของนิวคลีโอไทด์ที่อ่านได้จากเครื่องหาลำดับเบสจะอยู่ในรูปแบบฟาสคิว (FASTQ) และเมื่อทำการประกอบร่างจีโนมและได้ผลลัพธ์เป็นดีเอ็นสายยาวแล้ว มักบันทึกในรูปแบบฟาสต้า (FASTA)

FASTQ

ไฟล์ฟาสคิว (FASTQ) เก็บลำดับเบสของสายดีเอ็นเอที่ได้จากการหาลำดับเบส โดยความยาวของสายดีเอ็นเอแต่ละเส้นขึ้นอยู่กับเทคโนโลยีที่ใช้ในการหาลำดับเบส ตัวอย่างเช่น Illumina HiSeq มีความยาวของสายดีเอ็นเอระหว่าง 50 ถึง 250 เบส ขึ้นอยู่กับชุดคิท (kit) ในห้องปฏิบัติการที่ใช้ในการเตรียมข้อมูลก่อนการหาลำดับเบส รูปที่ 1.41 แสดงตัวอย่างโครงสร้างข้อมูลในไฟล์ฟาสคิว โดยดีเอ็นเอแต่ละเส้นใช้ 4 บรรทัดในการแสดงข้อมูล บรรทัดแรก

แสดงรหัสสายดีเอ็นเอขึ้นต้นด้วยเครื่องหมายแอด (@) เสมอ บรรทัดที่สองแสดงลำดับเบสของดีเอ็นที่ได้จากเครื่องหาลำดับเบส บรรทัดที่สามออกแบบไว้ให้เพิ่มเติมข้อมูลได้ในอนาคตขึ้นต้นด้วยเครื่องหมายบวก (+) เสมอ บรรทัดสุดท้ายเป็นลำดับของอักขระแสดงคุณภาพของแต่ละเบสที่อ่านได้จากเครื่องหาลำดับเบส

บรรทัดที่ 1 **@**รหัสสายดีเอ็นเอ และข้อมูลอื่นๆ เกี่ยวกับสายดีเอ็นเอนั้น
บรรทัดที่ 2 ลำดับเบสของสายดีเอ็นเอที่อ่านจากเครื่องหาลำดับเบส
บรรทัดที่ 3 **+** อาจมีข้อมูลเพิ่มเติมหรือปล่อยว่างไว้
บรรทัดที่ 4 อักขระแสดงคุณภาพของแต่ละเบสในสายดีเอ็นเอในบรรทัดที่ 2

รูปที่ 1.41 โครงสร้างข้อมูลในไฟล์ฟาสคิว

ตัวอย่างข้อมูลในรูปที่ 1.42 ชื่อของสายดีเอ็นเอคือ `HWI-ST797:281:D198UACXX:5:1101:1945:20491:N:0:GGCTAC` ลำดับเบสที่อ่านได้จากเครื่องหาลำดับเบสคือ `NTTATCCTCCACACAATTCCTTTCACTTTAGACAAAGAGATTTGTATTGCTCAGAAGCAGAGAATCTAGGTTTCTGTGGAATCTATTGGAGTTAGAAGGTA` โดยแต่ละตำแหน่งมีอักษรหลักที่เป็นไปได้ 4 ตัว คือ `A`, `T`, `C`, และ `G` ซึ่งเป็นตัวแทนนิวคลีโอไทด์ อะดีนีน (adenine), ไทมีน (thymine), ไซโทซีน (cytosine) และ กัวนีน (guanine) ตามลำดับ สำหรับอักษร N แสดงถึงความไม่แน่ใจของเครื่องหาลำดับเบสว่าเป็นนิวคลีโอไทด์ใด คุณภาพของแต่ละลำดับเบสที่อ่านได้คือ `#1:DDDDDHHHFHJJGJIEHHEHHGGHJGHIGHHIIJJJFHIJJHGIAFFGIJIIII@BFBFG@CCHGJIDCGIIDCAEHHDEFFBEEE>;ACCC@ACB;;` โดยเรียงตามลำดับนิวคลีโอไทด์ในบรรทัดที่ 2 เบสต่อเบส หมายเหตุ บรรทัดที่ 2 และ 4 ในรูปที่ 1.42 ใช้ตัวอักษรเอียงเพื่อให้เห็นความแตกต่างระหว่างบรรทัดชัดเจนขึ้น

```
@HWI-ST797:281:D198UACXX:5:1101:1945:20491:N:0:GGCTAC
NTTATCCTCCACACAATTCCTTTCACTTTAGACAAAGAGATTTGTATTGCTCAGAAGCAGAGAATCTAGGTTTCTGTGGAATCTATTGGAGTTAGAAGGTA
+
#1:DDDDDHHHFHJJGJIEHHEHHGGHJGHIGHHIIJJJFHIJJHGIAFFGIJIIII@BFBFG@CCHGJIDCGIIDCAEHHDEFFBEEE>;ACCC@ACB;;
```

รูปที่ 1.42 ตัวอย่างข้อมูลไฟล์ฟาสคิว

สำหรับอักขระแสดงคุณภาพในบรรทัดที่ 4 ตัวอักขระ “!” แสดงคุณภาพต่ำสุด และอักขระ “~” แสดงคุณภาพสูงสุดตามลำดับต่อไปนี้

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

โดยอักขระเหล่านี้สัมพันธ์กับ Phred quality score หรือ q score (คำศัพท์ถัดไป) ตามรูปที่ 1.43 (*ที่มา:* https://en.wikipedia.org/wiki/FASTQ_format#Quality)

Phred quality score

อีวิงก์ (Ewing) และกรีน (Green) [34] ได้พัฒนาอัลกอริทึมที่ใช้ในการอ่านลำดับเบสโดยอัตโนมัติจากเครื่องหาลำดับเบสโดยมีการให้คะแนนกับแต่ละเบสที่อ่านได้ตามค่า q ในสมการลอการิทึมต่อไปนี้

$$q = -10 \times \log_{10}(p)$$

โดยที่ p คือค่าประมาณของความน่าจะเป็นที่อาจเกิดความผิดพลาดในการอ่านเบสนั้นๆ ตัวอย่าง เช่นถ้าเบสนั้นมีค่า p เป็น 1/1000 หมายความว่าโอกาสที่จะเกิดความผิดพลาดในการอ่านนั้นเป็น 1 ใน 1000 ซึ่งจะได้ค่า q เป็น 30 (Q30) ค่า q นี้เรียกว่า q score หรือ Phred quality score ถ้าค่า p เป็น 1/100 ค่า q score จะเป็น 20 เป็นต้น ค่า q score กับค่าความน่าจะเป็น p นี้แปรผกผันระหว่างกัน

ค่า Phred quality score ถูกใช้เป็นมาตรวัดอย่างแพร่หลายในการประเมินความถูกต้องของแพลตฟอร์มที่ใช้ในการหาลำดับเบส เช่น อาจเทียบว่าสองแพลตฟอร์มมีร้อยละของเบสที่มีค่า q score >= 30 ของรีด 1 และรีด 2 (ในกรณีของ paired-end reads) เป็นเท่าใด และมีค่าเฉลี่ยของทั้งสองรีดรวมกันเท่าใด เป็นต้น

ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger

Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	33 !	11	0.07943	44 ,	22	0.00631	55 7	33	0.00050	66 B
1	0.79433	34 "	12	0.06310	45 -	23	0.00501	56 8	34	0.00040	67 C
2	0.63096	35 #	13	0.05012	46 .	24	0.00398	57 9	35	0.00032	68 D
3	0.50119	36 $	14	0.03981	47 /	25	0.00316	58 :	36	0.00025	69 E
4	0.39811	37 %	15	0.03162	48 0	26	0.00251	59 ;	37	0.00020	70 F
5	0.31623	38 &	16	0.02512	49 1	27	0.00200	60 <	38	0.00016	71 G
6	0.25119	39 '	17	0.01995	50 2	28	0.00158	61 =	39	0.00013	72 H
7	0.19953	40 (	18	0.01585	51 3	29	0.00126	62 >	40	0.00010	73 I
8	0.15849	41)	19	0.01259	52 4	30	0.00100	63 ?	41	0.00008	74 J
9	0.12589	42 *	20	0.01000	53 5	31	0.00079	64 @	42	0.00006	75 K
10	0.10000	43 +	21	0.00794	54 6	32	0.00063	65 A			

รูปที่ 1.43 ชุดอักขระที่แสดงค่า Phred quality score

(ที่มา: Edgar, R. *Quality (Phred) scores*. [ONLINE] Available at: https://drive5.com [เข้าถึงออนไลน์เมื่อวันที่ 11 ก.ค. พ.ศ. 2564])

FASTA

ไฟล์ฟาสต้า (FASTA) ใช้เก็บลำดับเบสของสายดีเอ็นเอ อาร์เอ็นเอ หรือลำดับกรดแอมิโนของสายโปรตีน โดยดีเอ็นเอ อาร์เอ็นเอ หรือโปรตีนแต่ละสายใช้ 2 บรรทัดในการแสดงข้อมูลดังรูปที่ 1.44 บรรทัดแรกขึ้นต้นด้วยเครื่องหมายมากกว่า (>) เสมอ และตามด้วยคำอธิบาย เช่น รหัสของลำดับสายดีเอ็นเอ อาร์เอ็นเอ หรือโปรตีน เป็นต้น บรรทัดที่สองแสดงลำดับเบสของสายดีเอ็นเอ อาร์เอ็นเอ หรือลำดับกรดแอมิโนของสายโปรตีน

บรรทัดที่ 1 >รหัสสายดีเอ็นเอ อาร์เอ็นเอ หรือโปรตีน

บรรทัดที่ 2 ลำดับเบสของสายดีเอ็นเอ อาร์เอ็นเอ หรือลำดับกรดแอมิโนของสายโปรตีน

รูปที่ 1.44 โครงสร้างข้อมูลไฟล์ฟาสต้า

รูปที่ 1.45 แสดงตัวอย่างข้อมูลในไฟล์ฟาสต้าโดย `ENA|U64805|U64805.1 Homo sapiens Brca1-delta11b (Brca1) mRNA, complete cds.` แสดงชื่อของสายอาร์เอ็นเอโดยมีการระบุว่าเป็นของมนุษย์ (*Homo sapiens*) เป็นอาร์เอ็นเอประเภทเอ็มอาร์เอ็นเอ (mRNA) และเป็นส่วนของลำดับกำหนดรหัส (coding sequence; cds) ที่สมบูรณ์ สามารถถูกแปลรหัสเป็นโปรตีนดังแสดงในรูป ที่ 1.46 สำหรับลำดับเบสในบรรทัดถัดๆ มา ถือเป็นส่วนหนึ่งของบรรทัดที่สองที่เก็บลำดับเบสอาร์เอ็นเอตามรหัสชื่อที่ระบุในบรรทัดแรก

```
>ENA|U64805|U64805.1 Homo sapiens Brca1-delta11b (Brca1) mRNA,
complete cds.
ATGGATTTATCTGCTCTTCGCGTTGAAGAAGTACAAAATGTCATTAATGCTATGCAGAAA
ATCTTAGAGTGTCCCATCTGTCTGGAGTTGATCAAGGAACCTGTCTCCACAAAGTGTGAC
CACATATTTTGCAAATTTTGCATGCTGAAACTTCTCAACCAGAAGAAAGGGCCTTCACAG
TGTCCTTTATGTAAGAATGATATAACCAAAAGGAGCCTACAAGAAAGTACGAGATTTAGT
CAACTTGTTGAAGAGCTATTGAAAATCATTTGTGCTTTTCAGCTTGACACAGGTTTGGAG
TATGCAAACAGCTATAATTTTGCAAAAAAGGAAAATAACTCTCCTGAACATCTAAAAGAT
…
```

รูปที่ 1.45 ตัวอย่างข้อมูลไฟล์ฟาสต้าเก็บลำดับเบสของสายอาร์เอ็นเอ

```
>ENA|U64805|U64805.1 Homo sapiens Brca1-delta11b (Brca1)
protein
MDLSALRVEEVQNVINAMQKILECPICLELIKEPVSTKCDHIFCKFCMLKLLNQKKGPSQCPL
CKNDITKRSLQESTRFSQLVEELLKIICAFQLDTGLEYANSYNFAKKENNSPEHLKD
…
```

รูปที่ 1.46 ตัวอย่างข้อมูลไฟล์ฟาสต้าเก็บลำดับกรดแอมิโนของสายโปรตีนที่มีกรดแอมิโนเมไทโอนีน (methionine; M) หรือรหัสเริ่มต้นที่เป็นไปได้ 3 ตำแหน่ง

บทที่ 2 การประกอบร่างจีโนมใหม่ (*De novo* genome assembly)

วัตถุประสงค์

- เพื่อให้นิสิตเข้าใจกระบวนการและเทคโนโลยีที่เกี่ยวข้องกับการหาลำดับเบสจีโนม
- เพื่อให้นิสิตคุ้นเคยกับตัวอย่างข้อมูลตั้งต้นที่ได้จากการหาลำดับเบสจีโนมและเข้าใจการทำงานของอัลกอริทึมพื้นฐานที่ใช้ในการประกอบร่างจีโนม
- เพื่อให้นิสิตได้เห็นตัวอย่างงานวิจัยและผลงานวิจัย รวมทั้งตัวอย่างโปรแกรมที่ใช้ในการประกอบร่างจีโนม
- เพื่อให้นิสิตได้เห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทาย รวมทั้งงานวิจัยอื่นๆ ที่เกี่ยวข้อง

ผลลัพธ์ที่คาดหวัง

- นิสิตสามารถอธิบายความแตกต่างรวมทั้งข้อดีข้อเสียระหว่างแพลตฟอร์มที่ใช้ในการหาลำดับเบสจีโนมได้
- นิสิตเข้าใจคุณลักษณะของข้อมูลตั้งต้นที่ได้จากการหาลำดับเบสจีโนม
- นิสิตสามารถอธิบายการทำงานของอัลกอริทึมหลักที่ใช้ในการประกอบร่างจีโนมได้
- นิสิตสามารถเขียนโปรแกรมที่ใช้ในการประกอบร่างจีโนมอย่างง่ายได้
- นิสิตสามารถยกตัวอย่างโปรแกรมประกอบร่างจีโนมที่มีการใช้งานกันอย่างแพร่หลายได้
- นิสิตสามารถยกตัวอย่างความท้าทายในการประกอบร่างจีโนม สามารถนำเสนอแนวทางในการพัฒนาวิธีการทางคอมพิวเตอร์เพื่อแก้ปัญหาเหล่านี้ รวมทั้งสามารถประยุกต์องค์ความรู้จากบทเรียนเพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้

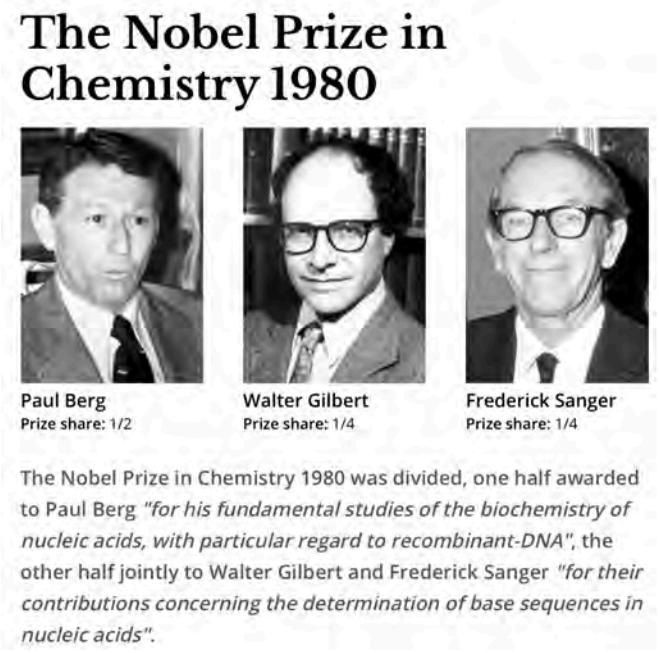
เนื้อหาโดยสรุป

เทคโนโลยีการหาลำดับเบสยุคใหม่ (next generation sequencing) หรือ เอ็นจีเอส (NGS) ลักษณะของข้อมูลและปริมาณข้อมูลที่ได้จากการหาลำดับเบสจีโนม โจทย์ทางชีวสารสนเทศ การประกอบร่างจีโนมใหม่ (*de novo* genome assembly) หรือการประกอบร่างจีโนมแบบไม่มีจีโนมอ้างอิงโดยวิธีการทางคอมพิวเตอร์ เพื่อต่อดีเอ็นเอสายสั้นยาวประมาณ 100-150 เบสจำนวนมากที่ได้จากเทคโนโลยีการหาลำดับเบสให้ยาวขึ้นจนเป็นสายโครโมโซมที่ถูกต้อง ตัวอย่างอัลกอริทึมและโครงสร้างข้อมูลที่เกี่ยวข้องกับการประกอบร่างจีโนมใหม่ เช่น ปัญหาการ

สร้างสายอักขระต้นฉบับจากชุดของสายอักขระย่อย (string reconstruction problem) กราฟแสดงความคาบเกี่ยว (overlap graph) ปัญหาการหาเส้นทางฮามิลโทเนียน (Hamiltonian path problem) กราฟ de Bruijn ปัญหาการหาเส้นทางออยเลอร์ (Eulerian path problem) การประกอบร่างจีโนมใหม่จากชุดของดีเอ็นเอเส้นสั้นคู่ และตัวอย่างโปรแกรมที่มีการใช้งานกันอย่างแพร่หลาย

บทที่ 2 การประกอบร่างจีโนมใหม่ (*De novo* genome assembly)

การได้มาซึ่งรหัสพันธุกรรมในระดับจีโนมเป็นจุดเริ่มต้นในการทำความเข้าใจกระบวนการต่างๆ ทางชีววิทยาและอณูชีววิทยาของสิ่งมีชีวิต ในปี ค.ศ. 1977 วอลเตอร์ กิลเบิร์ต (Walter Gilbert) และ เฟรดเดอริก แซงเกอร์ (Frederick Sanger) ต่างพัฒนาวิธีการหาลำดับเบสดีเอ็นเอ และในปี ค.ศ. 1980 ทั้งสองท่านได้รับรางวัลโนเบลในสาขาเคมีร่วมกัน (รูปที่ 2.1)



รูปที่ 2.1 วอลเตอร์ กิลเบิร์ต (Walter Gilbert) และ เฟรดเดอริก แซงเกอร์ (Frederick Sanger) ได้รับรางวัลโนเบลในปี ค.ศ. 1980 สาขาเคมีเรื่องการหาลำดับเบสในสายของกรดนิวคลีอิก

อย่างไรก็ตามนักวิทยาศาสตร์ยังใช้เวลาอีกร่วม 20 ปี กับงบประมาณอีกกว่า 3 พันล้านยูเอสดอลลาร์ ในการหาลำดับเบสจีโนมมนุษย์จนได้โครงร่างแรกในปี ค.ศ. 2001 [2] ในปี ค.ศ. 1990 ฟรานซิส คอลลิน (Francis Collins) ในฐานะผู้อำนวยการเอ็นไอเอชเป็นผู้นำโครงการจีโนมมนุษย์ (Human Genome Project) ซึ่งเป็นโครงการเปิดต่อสาธารณะโดยมีเป้าหมายในการถอดรหัสจีโนมมนุษย์ให้สำเร็จภายในปี ค.ศ. 2005 และในปี ค.ศ. 1997 เครก เวนเตอร์ (Craig Venter) ได้ก่อตั้งบริษัทเอกชนภายใต้ชื่อ Celera Genomics โดยมีเป้าหมายเดียวกัน

หลังประสบความสำเร็จในการหาลำดับเบสจีโนมมนุษย์ฉบับโครงร่างในปี ค.ศ. 2001 จีโนมของสิ่งมีชีวิตอื่นๆ ในกลุ่มยูแคริโอตได้ถูกหาลำดับเบสออกมาอย่างต่อเนื่อง ทั้งจีโนมหนู *Mus musculus* [35] และ *Rattus norvegicus* [36], สุนัข (*Canis lupus familiaris*) [37], ชิมแปนซี (*Pan troglodytes*) [38], ลิงวอก (*Macaca*

mulatta) [39], ม้า (*Equus caballus*) [40], โอพอสซัม (*Didelphimorphia*) [41], วัว (*Bos taurus*) [42] โดยในสมัยแรกนั้นการหาลำดับเบสจีโนมใช้การหาลำดับแบบแซงเกอร์ (Sanger sequencing) ซึ่งมีข้อจำกัดสำคัญคือจำนวนเบสที่อ่านได้ในหนึ่งหน่วยเวลาน้อยกว่าการหาลำดับเบสยุคใหม่ (next generation sequencing) หรือเอ็นจีเอส (NGS) มาก ทำให้ใช้เวลานานกว่ามากในการหาลำดับเบสจีโนม ปัจจุบันด้วยเทคโนโลยีเอ็นจีเอส จึงมีการหาลำดับเบสจีโนมของสิ่งมีชีวิตอื่นๆ อีกมากมาย เช่น จีโนมหมีแพนด้า (*Ailuropoda melanoleura*) [43], จีโนมงูหลามพม่า (*Python molurus bivittatus*) [44] นอกจากการหาลำดับเบสจีโนมมนุษย์และจีโนมสัตว์หลายชนิดยังมีการหาลำดับเบสจีโนมพืชจำนวนมาก เช่น ข้าวสายพันธุ์จาโปนิกา (*Oryza sativa L. ssp. japonica*) [45] และสายพันธุ์อินดิกา (*Oryza sativa L. ssp. indica*), ยางพารา *(Hevea brasiliensis*) [46, 47], ปาล์มน้ำมัน (*Elaeis guineensis*) [48] และทุเรียน (*Durio zibethinus*) [49] เป็นต้น องค์ความรู้จากการวิเคราะห์จีโนมของสิ่งมีชีวิตเหล่านี้สามารถนำไปประยุกต์ใช้ในการวินิจฉัยและวิจัยทางการแพทย์ การปรับปรุงพันธุ์พืช และเทคโนโลยีชีวภาพ

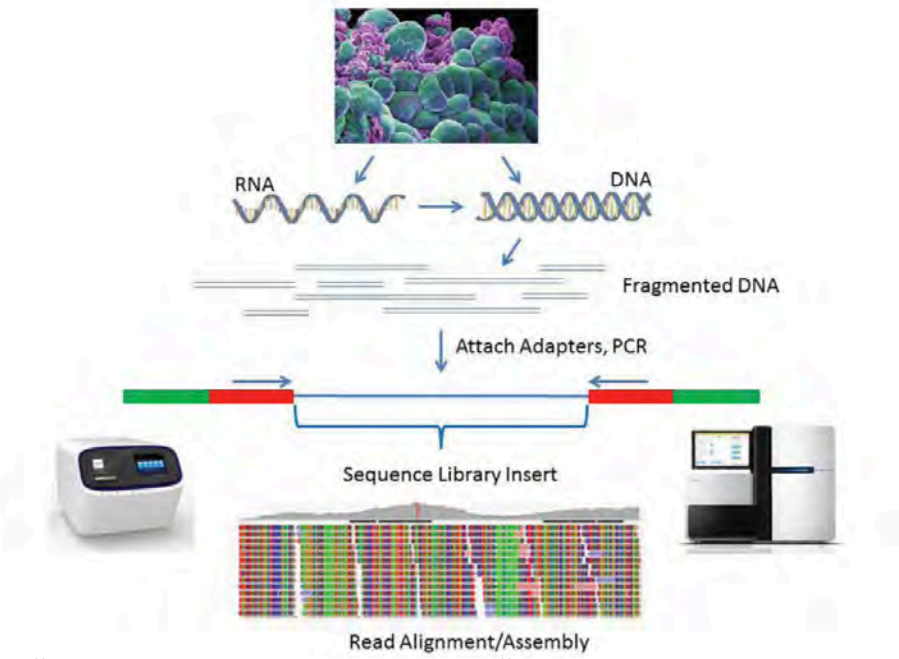
ความก้าวหน้าของเทคโนโลยีการหาลำดับเบส

ช่วงปลายทศวรรษของคริสต์ศตวรรษที่ 2000 ตลาดเทคโนโลยีและเครื่องมือหาลำดับเบสมีการขยายตัวอย่างมาก บริษัทอิลูมินา (Illumina) สามารถลดราคาการหาลำดับเบสจีโนมมนุษย์จาก 3 พันล้านยูเอสดอลลาร์ เหลือประมาณ 1 พันดอลลาร์ โดยใช้เทคโนโลยีการหาลำดับเบสยุคใหม่ (next generation sequencing) หรือเอ็นจีเอส (NGS) ในขณะที่บริษัทคอมพลีต จีโนมิกส์ (Complete Genomics) ก่อตั้งโรงงานจีโนมิกส์ในซิลิคอล วัลลีย์ (Silicon Valley) โดยให้บริการหาลำดับเบสหลายร้อยจีโนมต่อเดือน สถาบันจีโนมของปักกิ่ง (Beijing Genome Institute; BGI) ได้สั่งเครื่องหาลำดับเบสมาใช้ในสถาบันหลายร้อยเครื่องและกลายเป็นศูนย์หาลำดับเบสที่ใหญ่ที่สุดในโลก ปี ค.ศ. 2010 มีโครงการหาลำดับเบสจีโนมของสัตว์มีกระดูกสันหลัง 10,000 ชนิด ปี ค.ศ. 2015 ประเทศอังกฤษได้เริ่มโครงการหาลำดับเบสจีโนมของชาวอังกฤษ 100,000 คนผ่านโครงการจีโนมิกส์อิงแลนด์ (Genomics England) [50] และประเทศกาต้าร์เริ่มโครงการหาลำดับเบสจีโนมของชาวกาต้าร์ 10,000 คนผ่านโครงการ (Qatar Genome Program) (https://qatargenome.org.qa/) ในขณะที่บริษัทอิลูมินามีเป้าหมายลดราคาในการหาลำดับเบสจีโนมลงให้เหลือ 100 ยูเอสดอลลาร์ต่อคน ทั้งนี้ข้อมูลรหัสพันธุกรรมที่ได้จากเทคโนโลยีเอ็นจีเอสนั้นเป็นข้อมูลลำดับเบสสายสั้น คำถามคือจะออกแบบและพัฒนาวิธีการทางคอมพิวเตอร์อย่างไร ให้สามารถต่อดีเอ็นเสายสั้นจำนวนมากให้เป็นลำดับเบสของชุดโครโมโซมที่เป็นตัวแทนจีโนมที่สมบูรณ์

การเตรียมดีเอ็นเอเพื่อหาลำดับเบส

ขั้นตอนพื้นฐานในการหาลำดับเบส [51] (รูปที่ 2.2) ประกอบด้วยการสกัดดีเอ็นเอจากเนื้อเยื่อหรือเซลล์ตัวอย่าง ในกรณีการหาลำดับเบสอาร์เอ็นเอ อาร์เอ็นเอจะถูกแปลงให้เป็นดีเอ็นเอคู่สม (complementary DNA) หรือซีดีเอ็นเอ (cDNA) ที่ถูกสังเคราะห์ย้อนกลับจากเอ็มอาร์เอ็นเอโดยเอ็นไซม์รีเวิร์สทรานสคริปเทส (reverse transcrip-

tase) จากนั้นดีเอ็นเอจะถูกทำให้เป็นเส้นสั้น (fragmented DNA) และทำให้เป็นไลบรารีโดยการนำดีเอ็นเอสายสั้นเหล่านี้ไปเชื่อมต่อกับตัวปรับ (adapter) โดยวิธีการไลเกชัน (ligation) ซึ่งตัวปรับเหล่านี้ถูกออกแบบให้มีลำดับเบสที่มีความจำเพาะกับแพลตฟอร์มที่ใช้ในการหาลำดับเบส เช่น ตัวปรับที่สามารถเชื่อมต่อได้กับโฟล-เซลล์ (flow-cell) ของแพลตฟอร์มอิลูมินา (Illumina) หรือบีดส์ (bead) ของแพลตฟอร์ม Ion Torrent เป็นต้น หลังจากต่อกับตัวปรับแล้ว ขั้นตอนถัดไปคือการเพิ่มจำนวนดีเอ็นเอในไลบรารีซึ่งมีวิธีการแตกต่างกันไปตามแพลตฟอร์ม จากนั้นจึงนำไปทำการอ่านลำดับเบสตามวิธีการจำเพาะของแต่ละแพลตฟอร์มต่อไป



รูปที่ 2.2 ขั้นตอนการเตรียมไลบรารีของดีเอ็นเอสายสั้นเพื่อหาลำดับเบสโดยเทคโนโลยีเอ็นจีเอส

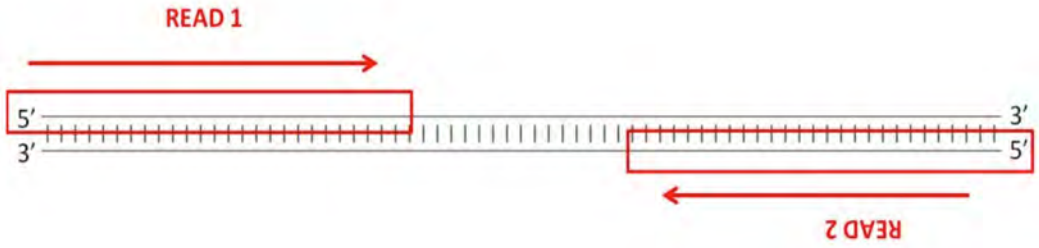
(*ที่มา*: *รูปที่* 1 *ของ [51]*)

เทคโนโลยีการหาลำดับเบส

หลังความสำเร็จในการหาลำดับเบสจีโนมมนุษย์ในปี ค.ศ. 2001 [2] เทคโนโลยีการหาลำดับเบสยุคใหม่ หรือเอ็นจีเอสมีการพัฒนาอย่างก้าวกระโดด มีการเพิ่มความสามารถ เช่น เพิ่มจำนวนของเบสที่สามารถอ่านได้ต่อหนึ่งหน่วยเวลา เพิ่มความยาวของลำดับเบสที่สามารถอ่านได้ หาลำดับเบสได้ทั้งสายเดี่ยว (single-end sequencing) และสายคู่ (paired-end sequencing) โดยการหาลำดับเบสสายเดี่ยว ดีเอ็นเอเกลียวคู่จะถูกอ่านลำดับเบสออกมาเป็นสายรหัสพันธุกรรมจำนวนมากในทิศทางเดียวคือทิศทางจาก 5’ ไป 3’ ในขณะที่การหาลำดับเบสสายคู่ ดีเอ็นเอจะถูกอ่านจากทั้งสองทิศทางคืออ่านไปข้างหน้าในสายบวกและอ่านย้อนกลับในสายลบ (รูปที่ 2.3)

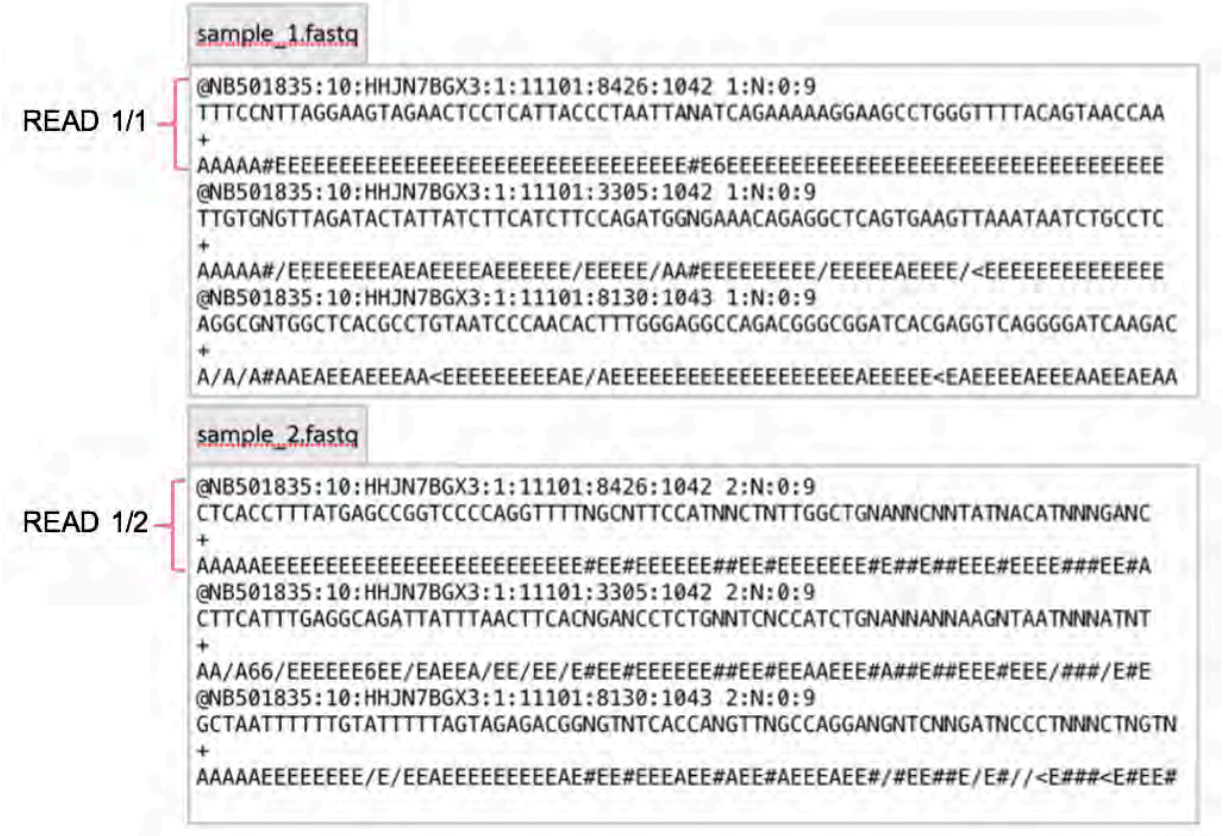
สำหรับแพลตฟอร์มอิลูมินาข้อมูลรหัสพันธุกรรมของดีเอ็นสายคู่ (paired-end) อยู่ในรูปแบบไฟล์ฟาสคิว (FASTQ) 2 ไฟล์ คือ [ชื่อไฟล์]_1.fastq และ [ชื่อไฟล์]_2.fastq โดยไฟล์ที่ลงท้ายด้วย _1.fastq เก็บข้อมูลสายดี

เอ็นเอหรือรีดที่อ่านได้จากสายบวก (READ 1 ในรูปที่ 2.3) และ ไฟล์ที่ลงท้ายด้วย _2.fastq เก็บข้อมูลสายดีเอ็นเอหรือรีดที่อ่านได้จากสายลบ (READ 2 ในรูปที่ 2.3) ดังตัวอย่างลำดับเบสในรูปที่ 2.4



รูปที่ 2.3 การอ่านลำดับเบสดีเอ็นเอในกรณีสายคู่ (paired-end sequencing)

(ที่มา: Minikel, E.V. 2012. *Forward and reverse reads in paired-end sequencing*. [ONLINE] Available at: http://www.cureffi.org [เข้าถึงออนไลน์เมื่อวันที่ 11 ก.ค. พ.ศ. 2564])



รูปที่ 2.4 ตัวอย่างลำดับเบสในไฟล์ฟาสคิวของการหาลำดับเบสแบบสายคู่โดยรีดเดียวกันที่อ่านไปข้างหน้า (READ 1/1) และอ่านย้อนกลับ (READ 1/2) มีชื่อเดียวกันในทั้งสองไฟล์

Goodwin, S et al. [17] ได้แบ่งเทคโนโลยีที่ใช้ในการหาลำดับเบสออกเป็นกลุ่ม ตามตารางในรูปที่ 2.5 ประกอบด้วย

1. การหาลำดับเบสแบบสายสั้น (short-read NGS) สามารถแบ่งออกเป็น 3 กลุ่มคือ

1.1 Sequencing by ligation (SBL) ตัวอย่างแพลตฟอร์ม เช่น SOLiD และ Complete Genomics ความยาวของสายรหัสพันธุกรรมที่อ่านได้จากแพลตฟอร์มในกลุ่มนี้อยู่ระหว่าง 50-100 คู่เบส (base pair; bp) และหาลำดับเบสได้ทั้งสายเดี่ยว (single-end reads) และสายคู่ (paired-end reads)

1.2 Sequencing by synthesis (SBS): CRT (cyclic reversible termination) ตัวอย่างแพลตฟอร์ม เช่น อิลูมินา (Illumina) และ Qiagen ความยาวของสายรหัสพันธุกรรมที่อ่านได้จากแพลตฟอร์มในกลุ่มนี้อยู่ระหว่าง 36-300 คู่เบส และหาลำดับเบสได้ทั้งสายเดี่ยวและสายคู่เช่นกัน

1.3 Sequencing by synthesis (SBS): SNA (single-nucleotide addition) ตัวอย่างแพลตฟอร์ม เช่น 454 และ Ion Torrent ความยาวของสายรหัสพันธุกรรมที่อ่านได้จากแพลตฟอร์ม 454 อยู่ระหว่าง 400-1,000 คู่เบส และหาลำดับเบสได้ทั้งสายเดี่ยวและสายคู่ สำหรับแพลตฟอร์ม Ion Torrent ความยาวของสายรหัสพันธุกรรมที่อ่านได้อยู่ระหว่าง 200 หรือ 400 คู่เบส และเป็นสายเดี่ยวเท่านั้น

เปรียบเทียบระหว่างแพลตฟอร์มเหล่านี้ เทคนิค Sequencing by Ligation ที่ใช้โดยแพลตฟอร์ม SOLiD และ Complete Genomics ให้ความถูกต้องสูงถึงประมาณ 99.99% อย่างไรก็ตาม มีรายงานว่าสายดีเอ็นเอที่อ่านได้ขาดตัวแปรผัน (variant) จริงไปบางส่วน ในขณะที่ตัวแปรผันลวงกลับปรากฏในบางลำดับเบส นอกจากนี้ยังมีหลักฐานว่าแพลตฟอร์มในกลุ่มนี้ถอดรหัสได้ไม่แม่นยำในบริเวณสายดีเอ็นเอที่เป็น AT-rich คือมีนิวคลีโอไทด์อะดีนีนติดกับไทมีนซ้ำๆ รวมทั้งบริเวณที่เป็น GC-rich ที่มีนิวคลีโอไทด์กัวนีนและไซโทซีนอยู่ติดกันหลายๆ คู่ นอกจากนี้ข้อจำกัดหลักของแพลตฟอร์มกลุ่มนี้คือ อ่านสายรหัสพันธุกรรมได้สั้น แพลตฟอร์ม SOLiD อ่านได้ความยาวสูงสุด 75 คู่เบส และแพลตฟอร์ม Complete Genomics อ่านได้ความยาวระหว่าง 28-100 คู่เบส ทำให้ไม่สามารถนำข้อมูลที่ได้ไปวิเคราะห์ลักษณะการผันแปรเชิงโครงสร้าง (structural variation) ของสายดีเอ็นได้ ปัจจุบันแพลตฟอร์มอิลูมินาที่หาลำดับเบสแบบสายคู่มีการใช้งานมากที่สุด และมีรุ่นทางการตลาดที่หลากหลายในเชิงจำนวนรหัสพันธุกรรมที่สามารถอ่านได้ต่อหนึ่งหน่วยเวลา ที่ความถูกต้องมากกว่าหรือเท่ากับ 99.5% สำหรับแพลตฟอร์ม 454 และ Ion Torrent อ่านสายรหัสพันธุกรรมได้ยาวกว่าสองกลุ่มข้างต้นที่ความยาวเฉลี่ย 700 และ 400 คู่เบส ตามลำดับ ซึ่งมีประโยชน์จำเพาะกับการวิเคราะห์ข้อมูลดีเอ็นเอในบริเวณที่มีความซับซ้อนหรือมีจำนวนซ้ำของนิวคลีโอไทด์จำนวนมาก อย่างไรก็ตามทั้งสองเทคโนโลยีนี้ใช้หลักการ SNA (single-nucleotide addition) ทำให้มีโอกาสเกิดข้อผิดพลาดมากกว่าแพลตฟอร์มในสองกลุ่มแรกในการอ่านลำดับเบสในบริเวณที่เกิดอินเดล (การมีชุดของลำดับเบสสอดแทรกหรือขาดหายไปสายสั้น) โดยบริษัท Roche ได้หยุดการผลิตแพลตฟอร์ม 454 ไปในปี ค.ศ. 2016

2. การหาลำดับเบสแบบสายยาว (Long-read NGS)

ด้วยความรู้ที่เพิ่มขึ้นจากการหาลำดับเบสจีโนมของสิ่งมีชีวิตต่างๆ โดยเฉพาะจีโนมในกลุ่มยูแคริโอตพบว่าจีโนมของสิ่งมีชีวิตหลายชนิดรวมทั้งจีโนมมนุษย์มีความซับซ้อน เนื่องจากประกอบด้วยบริเวณที่มีลำดับเบสหรือคู่เบสที่เกิดการซ้ำจำนวนมาก การขาดหายไป การสอดแทรกเข้ามา และการกลับด้านของลำดับเบส รวมทั้งการย้ายบริเวณภายในโครโมโซมเดียวกันหรือต่างโครโมโซมกัน หรือเกิดลักษณะเหล่านี้มากกว่าหนึ่งลักษณะร่วมกัน ข้อมูลรหัสพันธุกรรมที่ได้จากเทคโนโลยีการหาลำดับเบสแบบสายสั้น (short reads) จำนวนมากไม่เพียงพอต่อการนำมาวิเคราะห์จีโนมในบริเวณที่มีความซับซ้อนข้างต้น จำเป็นต้องมีเทคโนโลยีที่สามารถหาลำดับเบสสายยาว (Long-read sequencing) เพื่อให้สามารถวิเคราะห์การแปรผันเชิงโครงสร้าง รวมทั้งการเกิดการซ้ำในลักษณะต่างๆ ได้ถูกต้องมากขึ้น โดยเทคโนโลยีการหาลำดับเบสสายยาวที่มีอยู่ในปัจจุบัน สามารถแบ่งออกได้เป็น 2 กลุ่ม ดังต่อไปนี้

2.1 Single-molecule real-time sequencing ตัวอย่างแพลตฟอร์ม เช่น Pacific Biosciences และ Oxford Nanopore Technologies (ONT) โดยความยาวของสายรหัสพันธุกรรมที่อ่านได้จากแพลตฟอร์ม PacBio Sequel อยู่ระหว่าง 8-12 กิโลเบส (Kilo bases; Kb) และประมาณ 20 กิโลเบสในกรณีของ PacBio RS II ในขณะที่ Oxford Nanopore MK 1 MinION สามารถอ่านได้ยาวถึง 200 กิโลเบส

2.2 Synthetic long-read sequencing ตัวอย่างแพลตฟอร์ม เช่น Illumina และ 10X Genomics แพลตฟอร์มกลุ่มนี้อาศัยฐานเทคโนโลยีจากการหาลำดับเบสสายสั้นที่มีอยู่ โดยเพิ่มบาร์โค้ดเพื่อให้สามารถแยกได้ว่ารหัสพันธุกรรมที่อ่านได้นั้นอยู่ในสายดีเอ็นเอตั้งต้นเดียวกัน จากนั้นสร้างรหัสพันธุกรรมสายยาวโดยใช้วิธีการทางคอมพิวเตอร์ ความยาวของสายรหัสพันธุกรรมที่สามารถสร้างได้จากทั้งสองแพลตฟอร์มอยู่ที่ประมาณ 100 กิโลเบส

เทคโนโลยีการหาลำดับเบสสายยาวช่วยให้การประกอบร่างจีโนม (genome assembly) สำหรับสิ่งมีชีวิตที่ยังไม่เคยมีการหาลำดับเบสจีโนมมาก่อนทำได้ง่ายขึ้น รวมทั้งช่วยให้การศึกษาการแปรผันเชิงโครงสร้างมีความถูกต้องครอบคลุมมากขึ้น ข้อจำกัดของทั้งสองแพลตฟอร์มนี้ คือ ยังมีความผิดพลาดของการอ่านค่าข้อมูลสูงมากโดยอาจสูงถึง 15% ในการอ่าน 1 รอบ [17] ซึ่งความผิดพลาดหลักอยู่ในบริเวณที่เป็นอินเดล อย่างไรก็ตามความผิดพลาดจากการอ่านนี้เกิดขึ้นกระจายแบบสุ่ม ดังนั้นการหาลำดับเบสโดยเพิ่มรอบการอ่านสำเนาของดีเอ็นเอเพื่อให้ได้จำนวนสายที่อ่านได้จากเครื่องในบริเวณหนึ่งๆ หลายเส้น (high coverage) จะช่วยลดความผิดพลาด เนื่องจากสามารถใช้เส้นเหล่านี้ในการทวนสอบการอ่านตำแหน่งต่างๆ ให้ถูกต้องขึ้น

Platform	Read length (bp)	Throughput	Reads	Runtime	Error profile	Instrument cost (US$)	Cost per Gb (US$, approx.)
Sequencing by ligation							
SOLiD 5500 Wildfire	50 (SE)	80 Gb	~700 M*	6 d*	≤0.1%, AT bias‡	NA§	$130‡
	75 (SE)	120 Gb					
	50 (SE)*	160 Gb*					
SOLiD 5500xl	50 (SE)	160 Gb	~1.4 B*	10 d*	≤0.1%, AT bias‡	$251,000‡	$70‡
	75 (SE)	240 Gb					
	50 (SE)*	320 Gb*					
BGISEQ-500 FCS[155]	50–100 (SE/PE)*	8–40 Gb*	NA‖	24 h*	≤0.1%, AT bias‡	$250 (REF. 155)	NA‖
BGISEQ-500 FCL[155]	50–100 (SE/PE)*	40–200 Gb*	NA‖	24 h*	≤0.1%, AT bias‡	$250,000 (REF. 155)	NA‖
Sequencing by synthesis: CRT							
Illumina MiniSeq Mid output	150 (SE)*	2.1–2.4 Gb*	14–16 M*	17 h*	<1%, substitution‡	$50,000 (REF. 118)	$200–300 (REF. 118)
Illumina MiniSeq High output	75 (SE)	1.6–1.8 Gb	22–25 M (SE)*	7 h	<1%, substitution‡	$50,000 (REF. 118)	$200–300 (REF. 118)
	75 (PE)	3.3–3.7 Gb	44–50 M (PE)*	13 h			
	150 (PE)*	6.6–7.5 Gb*		24 h*			
Illumina MiSeq v2	36 (SE)	540–610 Mb	12–15 M (SE)	4 h	0.1%, substitution‡	$99,000‡	~$1,000
	25 (PE)	750–850 Mb	24–30 M (PE)*	5.5 h			$996
	150 (PE)	4.5–5.1 Gb		24 h			$212
	250 (PE)*	7.5–8.5 Gb*		39 h*			$142‡
Illumina MiSeq v3	75 (PE)	3.3–3.8 Gb	44–50 M (PE)*	21–56 h*	0.1%, substitution‡	$99,000‡	$250
	300 (PE)*	13.2–15 Gb*					$110‡
Illumina NextSeq 500/550 Mid output	75 (PE)	16–20 Gb	Up to 260 M (PE)*	15 h	<1%, substitution‡	$250‡	$42
	150 (PE)*	32–40 Gb*		26 h*			$40‡
Illumina NextSeq 500/550 High output	75 (SE)	25–30 Gb	400 M (SE)*	11 h	<1%, substitution‡	$250‡	$43
	75 (PE)	50–60 Gb	800 M (PE)*	18 h			$41
	150 (PE)*	100–120 Gb*		29 h*			$33‡
Illumina HiSeq2500 v2 Rapid run	36 (SE)	9–11Gb	300 M (SE)*	7 h	0.1%, substitution‡	$690‡	$230
	50 (PE)	25–30 Gb	600 M (PE)*	16 h			$90
	100 (PE)	50–60 Gb		27 h			$52
	150 (PE)	75–90 Gb		40 h			$45
	250 (PE)*	125–150 Gb*		60 h*			$40‡
Illumina HiSeq2500 v3	36 (SE)	47–52 Gb	1.5 B (SE)	2 d	0.1%, substitution‡	$690‡	$180
	50 (PE)	135–150 Gb	3 B (PE)*	5.5 d			$78
	100 (PE)*	270–300 Gb		11 d*			$45‡
Illumina HiSeq2500 v4	36 (SE)	64–72 Gb	2 B (SE)	29 h	0.1%, substitution‡	$690‡	$150
	50 (PE)	180–200 Gb	4 B (PE)*	2.5 d			$58
	100 (PE)	360–400 Gb		5 d			$45
	125 (PE)*	450–500 Gb*		6 d*			$30‡
Illumina HiSeq3000/4000	50 (SE)	105–125 Gb	2.5 B (SE)*	1–3.5 d*	0.1%, substitution‡	$740/$900 (REF. 156)	$50
	75 (PE)	325–375 Gb					$31
	150 (PE)*	650–750 Gb*					$22 (REF. 157)

รูปที่ 2.5 ตารางเปรียบเทียบเทคโนโลยีหาลำดับเบสจีโนม

(ที่มา: ตารางที่ 1 ของ [17])

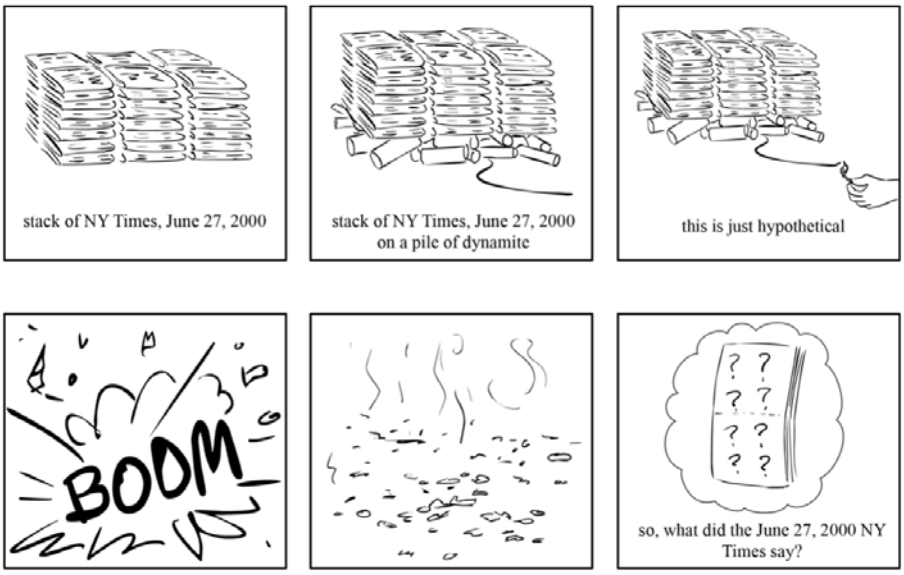
Platform	Read length (bp)	Throughput	Reads	Runtime	Error profile	Instrument cost (US$)	Cost per Gb (US$, approx.)
Sequencing by synthesis: SNA (cont.)							
Illumina HiSeq X	150 (PE)*	800–900 Gb per flow cell*	2.6–3 B (PE)*	<3 d*	0.1%, substitution‡	$1,000‡,¶	$7.0‡
Qiagen GeneReader	NA‖	12 genes; 1,250 mutations[22]	NA‖	Several days[22]	Similar to other SBS systems[22]	NA‖	$400–$600 per panel[22]
Sequencing by synthesis: SNA							
454 GS Junior	Up to 600; 400 average (SE, PE)*	35 Mb*	~0.1 M*	10 h*	1%, indel‡	NA§	$40,000‡
454 GS Junior+	Up to 1,000; 700 average (SE, PE)*	70 Mb*	~0.1 M*	18 h*	1%, indel‡	$108,000‡	$19,500‡
454 GS FLX Titanium XLR70	Up to 600; 450 mode (SE, PE)*	450 Mb*	~1 M*	10 h*	1%, indel‡	NA§	$15,500‡
454 GS FLX Titanium XL+	Up to 1,000; 700 mode (SE, PE)*	700 Mb*	~1 M*	23 h*	1%, indel‡	$450,000‡	$9,500‡
Ion PGM 314	200 (SE)	30–50	400,000–550,000*	23 h	1%, indel‡	$49‡	$25–3,500‡
	400 (SE)	60–100 Mb*		3.7 h*			
Ion PGM 316	200 (SE)	300–500 Mb	2–3 M*	3 h	1%, indel‡	$49‡	$700–1,000‡
	400 (SE)*	600 Mb–1 Gb*		4.9 h*			
Ion PGM 318	200 (SE)	600 Mb–1 Gb	4–5.5 M*	4 h	1%, indel‡	$49‡	$450–800‡
	400 (SE)*	1–2 Gb*		7.3 h*			
Ion Proton	Up to 200 (SE)	Up to 10 Gb*	60–80 M*	2–4 h*	1%, indel‡	$224‡	$80‡
Ion S5 520	200 (SE)	600 Mb–1 Gb	3–5 M*	2.5 h	1%, indel‡	$65 (REF. 158)	$2,400*
	400 (SE)*	1.2–2 Gb*		4 h*			$1,200*
Ion S5 530	200 (SE)	3–4 Gb	15–20 M*	2.5 h	1%, indel‡	$65 (REF. 158)	$950*
	400 (SE)*	6–8 Gb*		4 h*			$475*
Ion S5 540	200 (SE)*	10–15 Gb*	60–80 M*	2.5 h*	1%, indel‡	$65 (REF. 158)	$300*
Single-molecule real-time long reads							
Pacific BioSciences RS II	~20 Kb	500 Mb–1 Gb*	~55,000*	4 h*	13% single pass, ≤1% circular consensus read, indel‡	$695‡	$1,000‡
Pacific Biosciences Sequel	8–12 Kb[69]	3.5–7 Gb*	~350,000*	0.5–6 h*	NA‖	$350 (REF. 69)	NA‖
Oxford Nanopore MK 1 MinION	Up to 200 Kb[159]	Up to 1.5 Gb[159]	>100,000 (REF. 159)	Up to 48 h[160]	~12%, indel[159]	$1,000*	$750*
Oxford Nanopore PromethION	NA‖	Up to 4 Tb*	NA‖	NA‖	NA‖	$75*	NA‖
Synthetic long reads							
Illumina Synthetic Long-Read	~100 Kb synthetic length*	See HiSeq 2500	See HiSeq 2500	See HiSeq 2500	See HiSeq 2500 (possible barcoding and partitioning errors)	No additional instrument required	~$1,000*
10X Genomics	Up to 100 Kb synthetic length*	See HiSeq 2500	See HiSeq 2500	See HiSeq 2500	See HiSeq 2500 (possible barcoding and partitioning errors)	$75 (REFS 72,161)	See HiSeq 2500 +$500 per sample[161]

รูปที่ 2.6 ตารางเปรียบเทียบเทคโนโลยีหาลำดับเบสจีโนม (ต่อ)

(*ที่มา: ตารางที่* 1 (*ต่อ*) *ของ* [17])

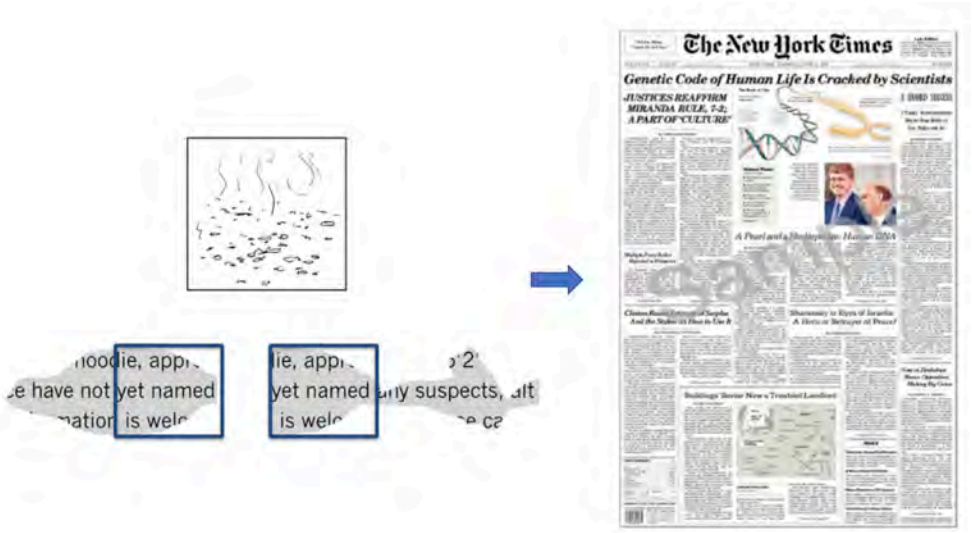
การจำลองเหตุการณ์ระเบิดของหนังสือพิมพ์

สมมติว่าหนังสือพิมพ์ฉบับเดียวกันกองอยู่หนึ่งตั้งและเกิดระเบิดขึ้น โดยหนังสือพิมพ์ที่กองอยู่ฉีกขาดกระเด็นเป็นชิ้นเล็กชิ้นน้อยตามรูปที่ 2.7 คำถามคือ เราจะต่อชิ้นส่วนหนังสือพิมพ์ที่ฉีกขาดจากแรงระเบิดนี้กลับมาเป็นหนังสือพิมพ์ต้นฉบับได้อย่างไร ปัญหาการต่อชิ้นส่วนหนังสือพิมพ์นี้เป็นปัญหาการหาชิ้นส่วนที่ทับซ้อน (overlap) และนำมาต่อเข้าด้วยกันเพื่อให้ได้ต้นฉบับเดิม ดังแสดงในรูปที่ 2.8



รูปที่ 2.7 การจำลองเหตุการณ์ระเบิดของหนังสือพิมพ์

(*ที่มา*: *รูปที่* 3.1 *ของ [52]*)

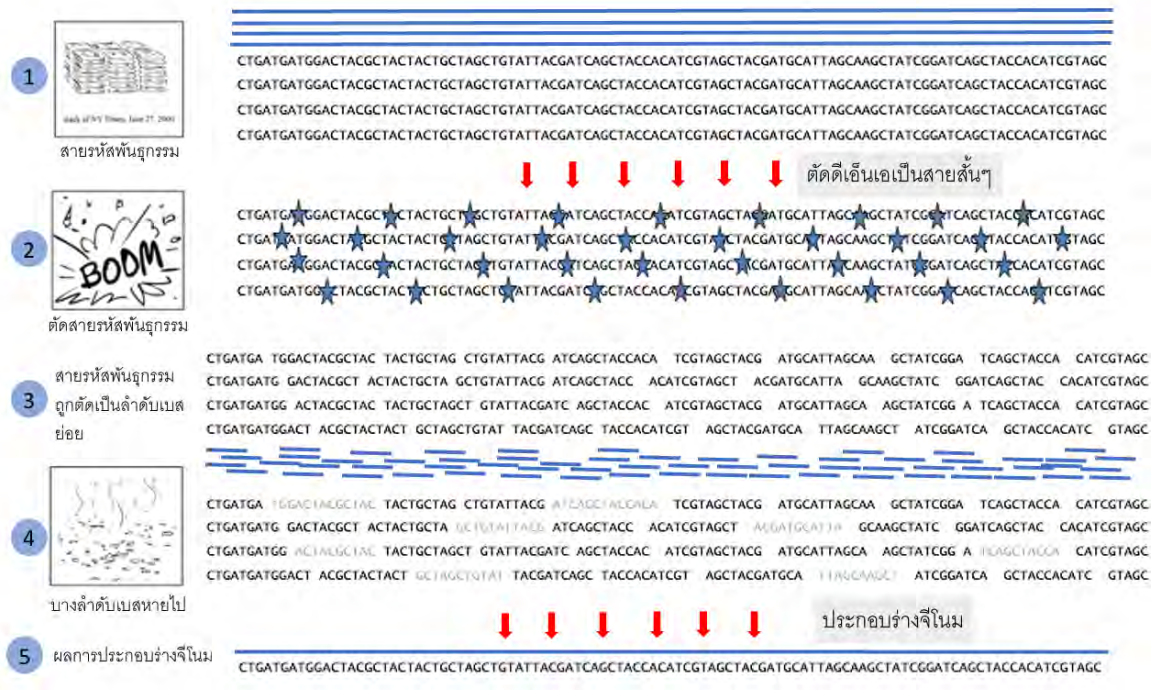


รูปที่ 2.8 การประกอบร่างชิ้นส่วนหนังสือพิมพ์จากการระเบิดเพื่อให้ได้ต้นฉบับเดิม

(*ที่มา*: *ปรับจากรูปที่* 3.2 *ของ [52] โดยเพิ่มตัวอย่างต้นฉบับทางขวา*)

ปัญหาการต่อชิ้นส่วนหนังสือพิมพ์นี้สามารถเปรียบเทียบได้ใกล้เคียงกับปัญหาการประกอบร่างจีโนม ในรูปที่ 2.9 สายดีเอ็นเอเปรียบเสมือนหนังสือพิมพ์ 1 ฉบับและการทำสำเนาดีเอ็นเอหลายๆ สำเนาเปรียบได้กับมีตั้งของหนังสือพิมพ์ฉบับเดียวกัน ในการหาลำดับเบส สายดีเอ็นเอเหล่านี้จะถูกนำมาตัดแบบสุ่มออกเป็นสายสั้น เพื่อให้เครื่องหาลำดับเบสแบบสายสั้นสามารถอ่านรหัสพันธุกรรมได้ การตัดดีเอ็นเอออกเป็นสายสั้นอาจทำได้หลายวิธี เช่น ใช้เอ็นไซม์ คลื่นอัลตราโซนิค ไนโตรเจนที่ถูกบีบอัด หรือแรงดันอากาศ โดยสายดีเอ็นเอจะถูกบังคับให้เคลื่อนผ่านรูที่มีขนาดเล็กมาก ดีเอ็นเอสายสั้นที่ได้จากการตัดด้วยวิธีการต่างๆ เหล่านี้มีประสิทธิภาพต่างกัน ตามที่มีการประเมินไว้ใน [53] การตัดดีเอ็นเอเป็นสายสั้นเทียบได้กับการระเบิดกองหนังสือพิมพ์ ซึ่งรหัสพันธุกรรมสายสั้นบางส่วนจะหายไป เหมือนกับชิ้นส่วนหนังสือพิมพ์ถูกไฟไหม้ไปนั่นเอง ปัญหาการประกอบร่างจีโนมมีข้อมูลเข้าหลักเป็นดีเอ็นเอสายสั้นจำนวนมากและผลลัพธ์ที่คาดหวัง คือ รหัสพันธุกรรมสายยาวที่เป็นผลจากการต่อดีเอ็นเอสายสั้นเหล่านี้ ซึ่งคาดหวังว่าจะเป็นตัวแทนดีเอ็นเอต้นฉบับที่มีความถูกต้อง ดังแสดงในรูปที่ 2.9 (ขั้นตอนที่ 5) คำถามคือ ทำไมการหาลำดับเบสจีโนมถึงต้องเตรียมสายดีเอ็นเอหลายสำเนา

ความท้าทายของปัญหาการประกอบร่างจีโนมเกิดจากเทคโนโลยีการหาลำดับเบสจีโนมในปัจจุบันไม่สามารถอ่านลำดับรหัสพันธุกรรมได้ครบทั้งโครโมโซม ในกรณีของแพลตฟอร์มอิลูมินาที่มีการใช้งานกันอย่างแพร่หลายในปัจจุบัน ความยาวของลำดับเบสอยู่ที่ 100-150 นิวคลีโอไทด์ ปัญหาการประกอบร่างจีโนมไม่ใช่ปัญหาการต่อจิ๊กซอว์ แต่เป็นปัญหาหาส่วนที่ทับซ้อน (overlap) ระหว่างดีเอ็นเอสายสั้นเหล่านี้



รูปที่ 2.9 เทียบเคียงปัญหาการประกอบร่างจีโนมกับการประกอบชิ้นส่วนหนังสือพิมพ์

(ที่มา: แก้ไขเพิ่มเติมข้อมูลจากรูปที่ 3.1 ของ [52])

ปัญหาการประกอบสายอักขระ

k-mer (อ่านว่า เค-เมอร์) แสดงลำดับเบสส่วนย่อย (substring/fragment) ของสายอักขระต้นแบบ โดย k คือค่าจำนวนเต็มที่บอกจำนวนของเบสในลำดับเบสส่วนย่อย เช่น ถ้ามีลำดับเบสต้นฉบับเป็น

GATTCAACGCTTAGCTT

และมี k=3 ชุด 3-mer ของลำดับเบสต้นฉบับนี้ประกอบด้วย **GAT**, **ATT**, **TTC**, **TCA**, ..., **CTT** (โดยมีการขยับลำดับเบสไปทีละ 1 เบสเพื่อสร้าง 3-mer ถัดไป นิยามปัญหาการประกอบสายอักขระ (String Reconstruction Problem) แสดงดังต่อไปนี้

นิยามปัญหาที่ 2.1 ปัญหาการประกอบสายอักขระ

ปัญหาการประกอบสายอักขระ (String Reconstruction Problem) สร้างสายอักขระต้นฉบับจากลำดับเบสที่เป็นส่วนย่อยของสายอักขระขนาด k-mers จำนวนมาก	
ข้อมูลเข้า	ชุดของลำดับเบสย่อยขนาด k
ผลลัพธ์	สายอักขระต้นฉบับที่ประกอบด้วยชุดของลำดับเบสย่อยขนาด k ทั้งหมด

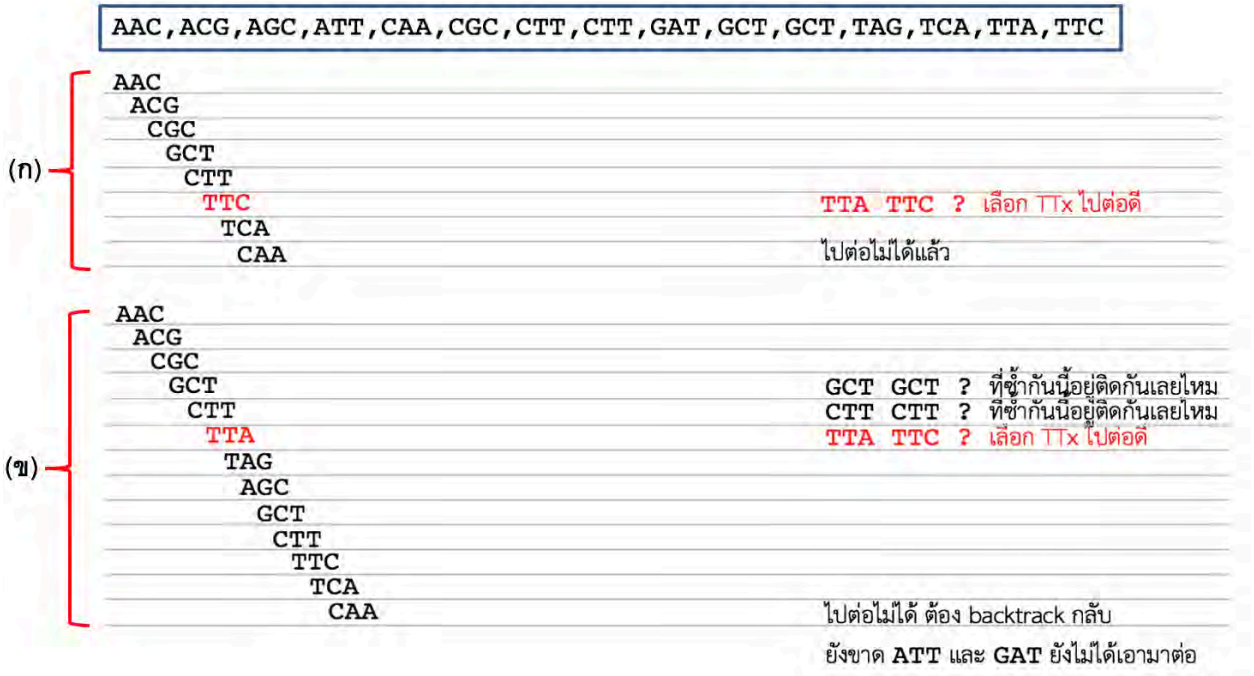
วิธีการแก้ปัญหาการประกอบสายอักขระอย่างง่าย (Naïve approach)

ขั้นตอน	สิ่งที่ทำ
1.	เรียงชุดของ k-mer ที่เป็นข้อมูลเข้าตามลำดับตัวอักษรในพจนานุกรม ตัวอย่างเช่น ข้อมูลเข้า 3-mer **GAT, ATT, TTC, TCA, CAA, AAC, ACG, CGC, GCT, CTT, TTA, TAG, AGC, GCT, CTT** จะถูกเรียงลำดับใหม่เป็น **AAC, ACG, AGC, ATT, CAA, CGC, CTT, CTT, GAT, GCT, GCT, TAG, TCA, TTA, TTC**
2.	นำ k-mer ซ้ายสุดมาเป็นลำดับเบสตั้งต้นของต้นฉบับ ได้ลำดับเบสตั้งต้นของต้นฉบับเป็น **AAC**
3.	หาลำดับเบส k - n เบสจากขวามาซ้ายของลำดับเบสตั้งต้นของต้นฉบับนี้ อย่างเช่น k - 1 ของ **AAC** จะได้ **AC** ตรวจสอบว่ามี k-mer อื่นใดบ้างที่ขึ้นต้นด้วย **AC** จะได้ชุด k-mers ลำดับที่ 2 ที่เข้าเงื่อนไข
4.	นำ k-mer ที่สองมาต่อกับลำดับเบสตั้งต้นของต้นฉบับ ได้เป็น **AACG** และทำซ้ำข้อ 3. จนกว่าชุด k-mers ทั้งหมดจะถูกใช้

ฝึกหัด	จากตัวอย่าง k-mers และคำอธิบายเกี่ยวกับขั้นตอนการประกอบสายอักขระอย่างง่าย จงเขียนแผนภาพแสดงการเชื่อมต่อระหว่าง k-mers

วิธีการแก้ปัญหาการประกอบสายอักขระอย่างง่ายข้างต้นมีข้อจำกัดคือในบางกรณีอาจพบเงื่อนไขที่ไม่สามารถประกอบทุก k-mers เข้าด้วยกันได้ ดังตัวอย่างชุด k-mers ข้างต้น ในขั้นตอนที่ต้องเลือกระหว่าง **TTA** และ **TTC**

ถ้าเลือก **TTC** มาใช้ k-mer ถัดไปที่ต้องนำมาประกอบคือ **TCA** และตามด้วย **CAA** จากนั้นจะไม่สามารถนำ k-mers ที่เหลือมาต่อได้และยังไม่ได้สายอักขระต้นฉบับที่ถูกต้องครบถ้วนดังแสดงในรูปที่ 2.10(ก) ต้องมีการย้อนการทำงานกลับ (backtrack) โดยไปเลือก **TTA** แทน **TTC** ซึ่งในกรณีนี้ k-mer ถัดไปที่ต้องนำมาประกอบคือ **TAG** ตามด้วย **AGC** และสามารถประกอบชุด k-mers ที่เหลืออยู่จนถึง **CAA** ได้ดังรูปที่ 2.10(ข) ซึ่งได้สายยาวขึ้นแต่ยังขาด **ATT** และ **GAT** ที่ยังไม่ได้นำมาประกอบ และต้องย้อนการทำงานกลับเช่นกัน



*รูปที่ 2.10 ข้อจำกัดของการประกอบสายอักขระโดยวิธีการอย่างง่าย (ก) k-mer **TTC** ถูกเลือกใช้และไม่สามารถประกอบสายอักขระได้สมบูรณ์ (ข) k-mer **TTA** ถูกเลือกใช้แทนและสามารถประกอบสายอักขระได้ยาวขึ้นแต่ไม่สมบูรณ์*

ความซับซ้อนเพิ่มเติมจากการมี k-mer เดียวกันหลายซ้ำภายในชุด

การประกอบสายอักขระต้นฉบับจากชุดของ k-mers ข้างต้นจะมีความซับซ้อนมากขึ้น ถ้า k-mer เช่น **CTT** มีสำเนาเป็นจำนวนซ้ำมากขึ้น ทำให้เรามีหลายเส้นทางที่สามารถนำมาต่อกับ **CTT** ได้ รวมทั้งมีความเป็นไปได้ที่ k-mer หลายซ้ำนี้อยู่ในตำแหน่งที่หลากหลายตัวอย่าง เช่น อยู่ติดกัน อยู่ไม่ติดกัน โดยมี k-mer อื่นแทรกอยู่ 1 k-mer หรือมี k-mer อื่นแทรกอยู่ 2 k-mers เป็นต้น พิจารณาตัวอย่างต่อไปนี้จากชุดสายอักขระเลขฐานสอง (binary string) ขนาด 4-mer ซึ่งเกิดจากการเลือกใส่แบบสุ่มในแต่ละตำแหน่งด้วยเลข 0 หรือ 1 มีรูปแบบที่เป็นไปได้อยู่ทั้งหมด 16 แบบ ดังต่อไปนี้

0000 0001 0010 0011 0100 0101 0110 0111

1000 1001 1010 1011 1100 1101 1110 1111

คำถามคือค่าความน่าจะเป็น (probability) ในการปรากฏสายอักขระย่อย “01” อย่างน้อย 1 ครั้งในแต่ละสายอักขระ 4-mer เป็นเท่าใด คำตอบคือ 11/16 โดยพบสายอักขระย่อย “01” ในสายอักขระ 4-mer ที่ถูกขีดเส้นใต้กำกับไว้ดังต่อไปนี้

0000 0001 0010 0011 0100 0101 0110 0111

1000 1001 1010 1011 1100 1101 1110 1111

ถ้ามีคำถามเดียวกันแต่เปลี่ยนสายอักขระย่อยที่สนใจเป็น “11” จะได้ค่าความน่าจะเป็นเท่ากับ 8/16 หรือ 1/2 ในสายอักขระ 4-mer ที่ถูกขีดเส้นใต้กำกับไว้ต่อไปนี้

0000 0001 0010 0011 0100 0101 0110 0111

1000 1001 1010 1011 1100 1101 1110 1111

คำถามคือทำไมค่าความน่าจะเป็นของการพบสายอักขระย่อย “11” น้อยกว่าการพบสายอักขระย่อย “01” ในสายอักขระ 4-mer ทั้งหมดที่เกิดจากการสร้างขึ้นมาแบบสุ่ม และถ้าเปลี่ยนเงื่อนไขในการหาค่าความน่าจะเป็นใหม่ โดยเปลี่ยนเป็นหาค่าความน่าจะเป็นในการปรากฏสายอักขระย่อย “01” อย่างน้อยสองครั้งในแต่ละ 4-mer จะได้ค่าความน่าจะเป็นเท่ากับ 1/16 ซึ่งพบใน 4-mer “0101” เท่านั้น ตามที่ถูกขีดเส้นใต้ไว้ดังต่อไปนี้

0000 0001 0010 0011 0100 0101 0110 0111

1000 1001 1010 1011 1100 1101 1110 1111

และถ้าเปลี่ยนสายอักขระย่อยที่สนใจเป็น “11” โดยใช้เงื่อนไขเดียวกันจะได้ค่าความน่าจะเป็นเท่ากับ 3/16 ดังที่แสดงไว้ต่อไปนี้

0000 0001 0010 0011 0100 0101 0110 0111

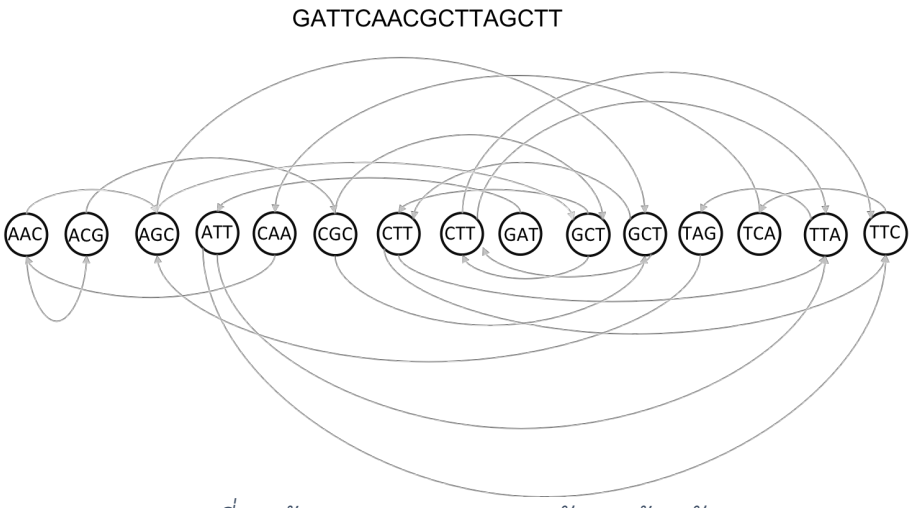
1000 1001 1010 1011 1100 1101 1110 1111

จากตัวอย่างจะเห็นว่าแต่ละ 4-mer ข้างต้นมีโอกาสในการพบสายอักขระย่อยที่มีรูปแบบจำเพาะหนึ่งๆ ไม่เท่ากัน โดยทั่วไปปรากฏการณ์นี้เรียกว่า overlapping word paradox โดยหมายถึงการพิจารณาสายอักขระย่อยโดยอนุญาตให้เกิดการคาบเกี่ยว (overlap) กันได้ ตามตัวอย่างข้างต้นปรากฏการณ์นี้มีผลต่อจำนวนที่ปรากฏของสายอักขระย่อยบางรูปแบบเช่น รูปแบบซ้ำเดิม “11” ในขณะที่รูปแบบอื่นๆ จะไม่มีผลกระทบ ดังตัวอย่างสุดท้ายข้างต้นที่ 0111 และ 1110 ถูกนับด้วยว่าพบสายอักขระย่อย “11” อย่างน้อยสองครั้ง ซึ่งปรากฏการณ์นี้ทำให้การคำนวณค่าความน่าจะเป็นในการปรากฏรูปแบบจำเพาะหนึ่งๆ มีความซับซ้อนมากขึ้นเนื่องจากขึ้นอยู่กับรูปแบบจำเพาะของสายอักขระย่อย และจำนวนที่คาดว่าจะพบสายอักขระย่อยนั้นใน k-mer

วิธีการแก้ปัญหาการประกอบสายอักขระโดยใช้เส้นทางฮามิลโทเนียน

แนวทางที่สองในการสร้างสายอักขระต้นฉบับจากชุดของลำดับเบสที่เป็นส่วนย่อยของสายอักขระขนาด k-mers คือการสร้างกราฟแสดงความคาบเกี่ยว (overlap graph) (ปัญหาที่ 2.2) แล้วหาเส้นทางฮามิลโทเนียน (Hamiltonian path) (ปัญหาที่ 2.3) โดยกราฟแสดงความคาบเกี่ยว (รูปที่ 2.11) แสดงการเชื่อมต่อกันระหว่าง k-

mers โดยแต่ละโหนดในกราฟคือแต่ละ k-mer และเส้นเชื่อมที่มีทิศทาง (directed edge) จากโหนด A ไปโหนด B (A→B) แสดงความสัมพันธ์ว่า k-mer ของโหนด A และ B นั้นมีรูปแบบที่มีความคาบเกี่ยวกัน สามารถเอามาประกอบกันได้ โดยอธิบายผ่านฟังก์ชัน `OVERLAP(Patterns)` ซึ่งจะมีเส้นเชื่อมระหว่าง A และ B ก็ต่อเมื่อ `SUFFIX(Pattern) = PREFIX(Pattern')` โดย `Pattern` คือรูปแบบจำเพาะของโหนด A และ `Pattern'` เป็นรูปแบบจำเพาะของโหนด B ตามลำดับ



*รูปที่ 2.11 กราฟแสดงความคาบเกี่ยวสร้างจาก 3-mer ของสายอักขระต้นฉบับ **GATTCAACGCTTAGCTT***

นิยามปัญหาที่ 2.2 ปัญหาการสร้างกราฟแสดงความคาบเกี่ยวระหว่างโหนด

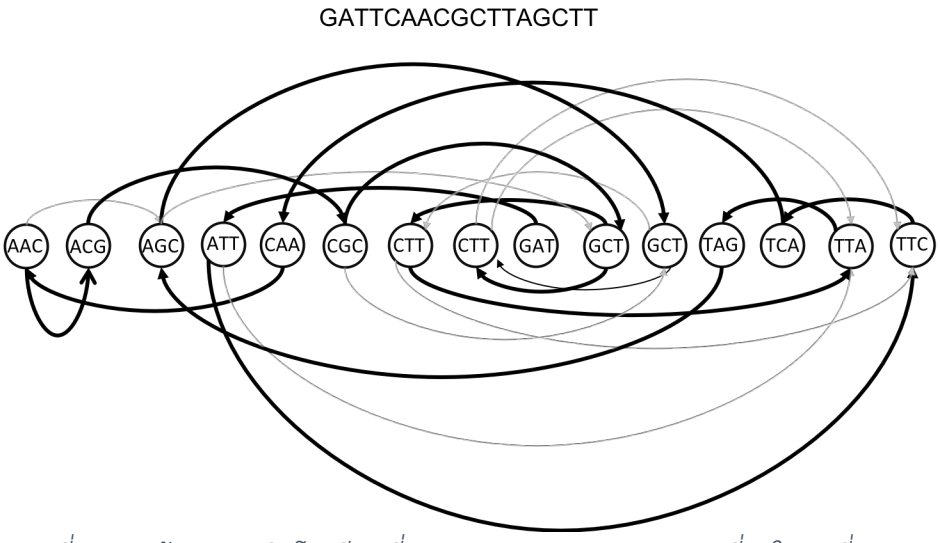
ปัญหาการสร้างกราฟแสดงความคาบเกี่ยวระหว่างโหนด (Overlap Graph Problem)	
ข้อมูลเข้า	ชุดของสายอักขระย่อยที่มีขนาด k-mer
ผลลัพธ์	กราฟแสดงความคาบเกี่ยวระหว่างโหนด โดยผ่านฟังก์ชัน `OVERLAP(Patterns)`

กราฟแสดงความคาบเกี่ยวระหว่างโหนดจะเป็นข้อมูลเข้าของปัญหาที่ 2.3 และผลลัพธ์ที่คาดหวัง คือ เส้นทางฮามิลโทเนียน ดังตัวอย่างเส้นสีดำในรูปที่ 2.12 ซึ่งแสดงเส้นทางการประกอบกันของ k-mer ทุกโหนดเพื่อให้ได้เป็นสายอักขระต้นฉบับ

นิยามปัญหาที่ 2.3 ปัญหาการหาเส้นทางฮามิลโทเนียน

ปัญหาการหาเส้นทางฮามิลโทเนียน (Hamiltonian Path Problem)	
ข้อมูลเข้า	กราฟแสดงความคาบเกี่ยวระหว่างโหนด
ผลลัพธ์	เส้นทางฮามิลโทเนียนที่มีการเดินผ่านทุก*โหนด*เพียงครั้งเดียว

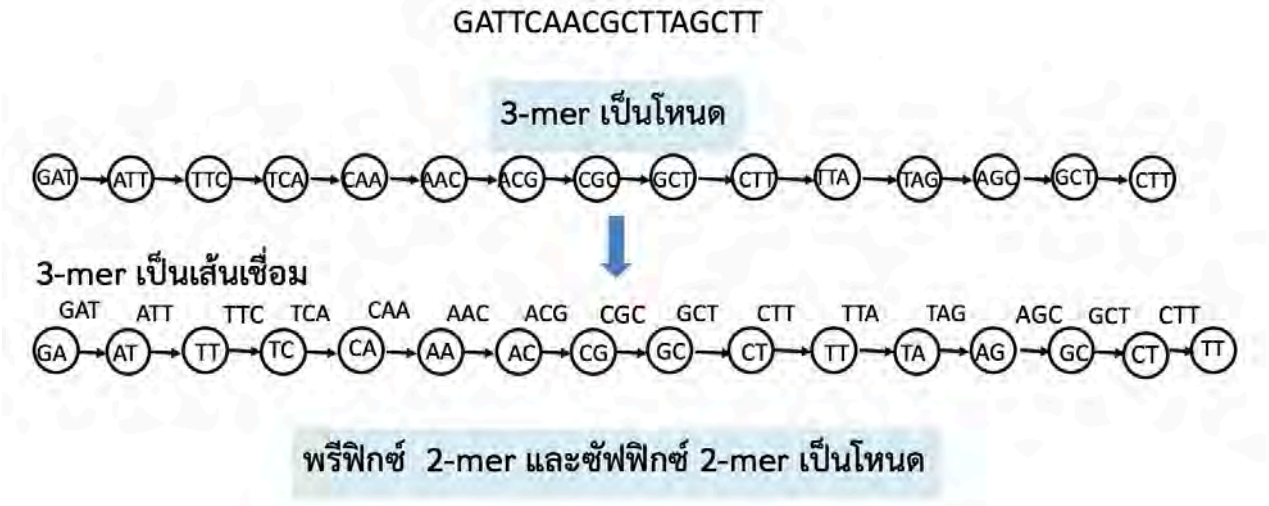
ความซับซ้อนของการหาเส้นทางฮามิลโนเนียนในกราฟเป็น NP-complete



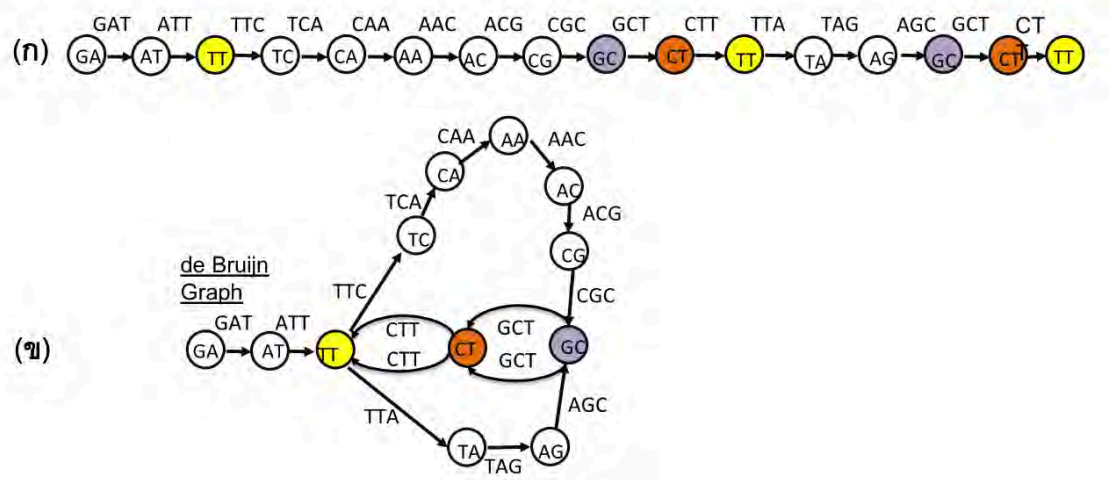
รูปที่ 2.12 เส้นทางฮามิลโทเนียนที่หาจากกราฟแสดงความคาบเกี่ยวในรูปที่ 2.11

วิธีการแก้ปัญหาการประกอบสายอักขระโดยใช้เส้นทางออยเลอร์

แนวทางที่สามในการสร้างสายอักขระต้นฉบับจากชุดของลำดับเบสที่เป็นส่วนย่อยของสายอักขระขนาด k-mer คือการสร้างกราฟ de Bruijn (ปัญหาที่ 2.4) แล้วหาเส้นทางออยเลอร์ (ปัญหาที่ 2.5) กราฟ de Bruijn แสดงการเชื่อมต่อระหว่าง k-mers โดยแต่ละโหนดในกราฟคือ ส่วนของ *k-mer* ในส่วนที่เป็น `PREFIX(k-mer)` และ `SUFFIX(k-mer)` ขนาด k – n ตามลำดับ และเส้นเชื่อมระหว่าง `PREFIX` และ `SUFFIX` โหนดของ k-mer เดียวกัน แสดงลำดับเบสของ k-mer นั้นๆ (รูปที่ 2.13) ตัวอย่างโหนดและเส้นเชื่อมของกราฟ de Bruijn ในรูปที่ 2.14(ก) แสดงโหนดที่มีลำดับเบส k – n mer ซ้ำกัน ในขณะที่รูปที่ 2.14(ข) แสดงกราฟ de Bruijn หลังการรวมโหนดที่มี k – n mer ซ้ำเข้าด้วยกัน



รูปที่ 2.13 โหนดและเส้นเชื่อมในกราฟ de Bruijn



*รูปที่ 2.14 กราฟ de Bruijn ของสายอักขระต้นฉบับ **GATTCAACGCTTAGCTT** (ก) โหนดและเส้นเชื่อมก่อนการรวมโหนด k – n mer ที่ซ้ำกัน (ข) โหนดและเส้นเชื่อมหลังรวมโหนดที่ซ้ำแล้ว*

นิยามปัญหาที่ 2.4 ปัญหาการสร้างกราฟ de Bruijn

ปัญหาการสร้างกราฟ de Bruijn	
ข้อมูลเข้า	ชุดของสายอักขระย่อยที่มีขนาด k-mer
ผลลัพธ์	กราฟ de Bruijn

วิธีการสร้างกราฟ de Bruijn

ขั้นตอน	สิ่งที่ทำ
1.	สำหรับแต่ละ k-mer จากชุดของ k-mer ที่เป็นข้อมูลเข้า ให้ทำการแบ่งแต่ละ k-mer นั้นออกเป็น `PREFIX(k-mer)` และ `SUFFIX(k-mer)` ตามลำดับ และลากเส้นเชื่อมต่อระหว่างสองโหนด
2.	รวมข้อมูลการเชื่อมต่อของโหนดที่ได้จากข้อ 1. ที่มีลำดับเบสภายในโหนดเป็นแบบเดียวกันเข้าด้วยกัน

กราฟ de Bruijn จะเป็นข้อมูลเข้าของปัญหาที่ 2.5 และผลลัพธ์ที่คาดหวังคือเส้นทางออยเลอร์ (Euler path) ซึ่งแสดงเส้นทางที่ผ่านทุกเส้นเชื่อมเพื่อให้ได้เป็นสายอักขระต้นฉบับ

ทฤษฎีบทของออยเลอร์ (Euler's theorem) ประยุกต์ใช้กับกราฟมีทิศทาง (directed graph) โดยกราฟมีคุณสมบัติ Eulerian ถ้ากราฟนั้นสมดุล (balance) *และ* มีคุณสมบัติ strongly connected คือทุกโหนดในกราฟต้องสามารถเชื่อมถึงกันได้โดยเส้นทางใดเส้นทางหนึ่ง กราฟจะสมดุลถ้าทุกโหนดในกราฟมีความสมดุลซึ่งหมายถึง

in-degree (เส้นชี้เข้าโหนด) และ out-degree (เส้นชี้ออกจากโหนด) ของแต่ละโหนดนั้นจะต้องเท่ากัน $IN(v_i) = OUT(v_i)$

นิยามปัญหาที่ 2.5 ปัญหาการหาเส้นทางออยเลอร์

ปัญหาการหาเส้นทางออยเลอร์ (Euler path problem)	
ข้อมูลเข้า	กราฟ de Bruijn
ผลลัพธ์	เส้นทางที่จะมีการเดินผ่านทุก *เส้นเชื่อม* เพียงครั้งเดียว (ถ้ามีเส้นทางนั้นอยู่)

วิธีการแก้ปัญหาการหาเส้นทางออยเลอร์แบบกลับมาจุดตั้งต้น

ขั้นตอน	สิ่งที่ทำ
1.	ตั้งค่าเริ่มต้นเป็นเส้นทางที่ยังไม่มีเส้นเชื่อมใดๆ
2.	ถ้ากราฟยังมีเส้นเชื่อมที่ไม่ยังถูกเดินผ่านให้ทำข้อ 3. ถ้าไม่ใช่ให้หยุดการทำงานและส่งออกเส้นทางออยเลอร์
3.	เลือกโหนดที่มีเส้นเชื่อมที่ยังไม่ถูกเดินผ่านมา 1 โหนด หาเส้นทางเดินจากจุดตั้งต้นนี้โดยไม่เดินผ่านเส้นเชื่อมซ้ำเดิมและสามารถกลับมาที่จุดเริ่มต้น เมื่อเดินถึงจุดตั้งต้นที่เลือกนี้ ให้เพิ่มเส้นทางนี้ใน 1. และกลับไป 2.

กราฟ de Bruijn และกราฟแสดงความคาบเกี่ยว

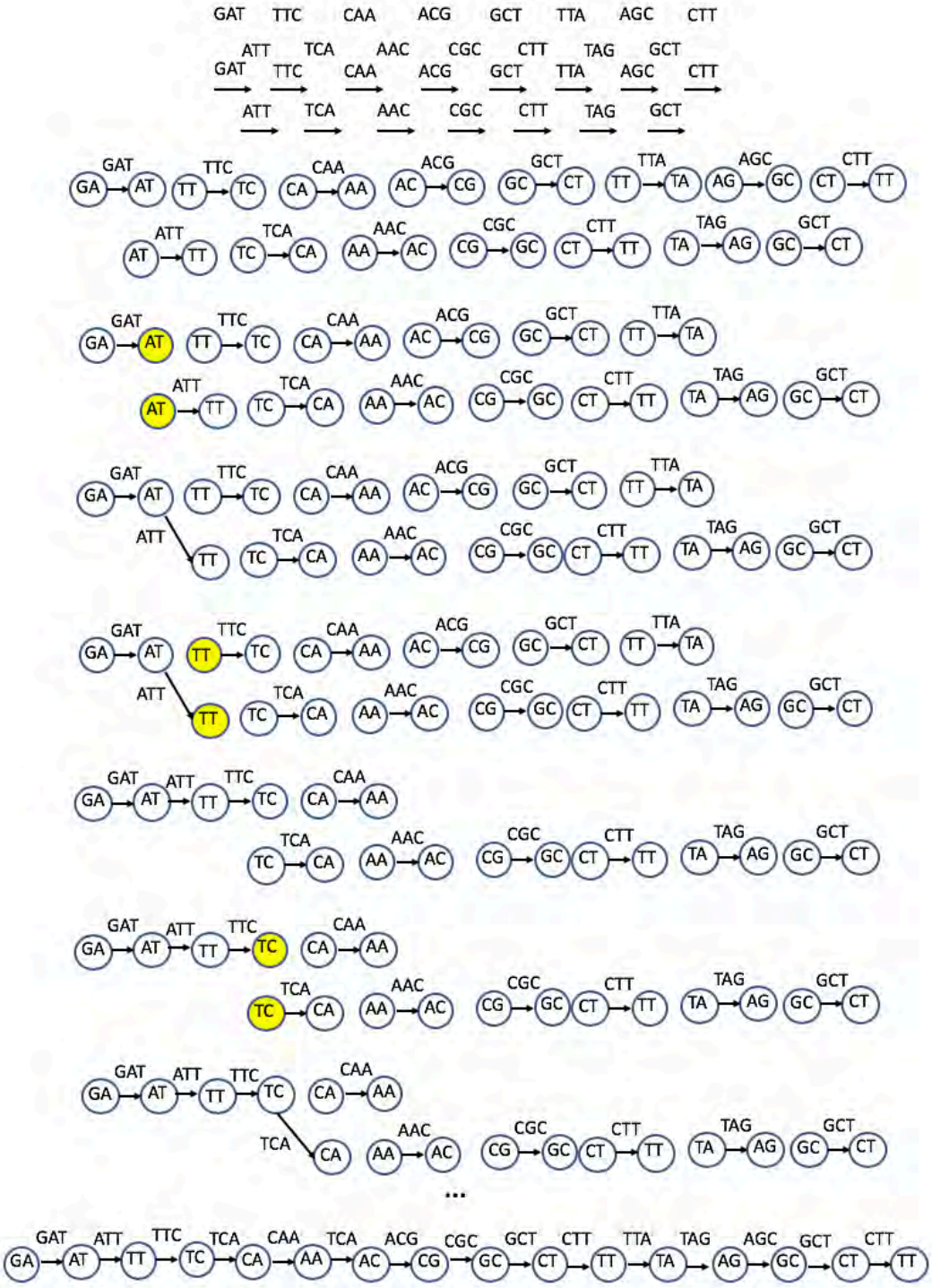
แนวทางแก้ปัญหาการประกอบร่างจีโนมจากชุดของ k-mer สามารถทำได้โดยการหาเส้นทางฮามิลโทเนียนจากกราฟแสดงความคาบเกี่ยวโดยต้องเดินผ่านทุกโหนดในกราฟเพียงครั้งเดียวหรือการหาเส้นทางออยเลอร์จากกราฟ de Bruijn โดยต้องเดินผ่านทุกเส้นเชื่อมในกราฟเพียงครั้งเดียว คำถามคือเราควรเลือกวิธีไหน

อัลกอริทึมที่ใช้ในการหาเส้นทางออยเลอร์จากกราฟ de Bruijn ใช้เวลาในการทำงานเป็นระดับพหุนาม (polynomial) ในขณะที่ไม่มีอัลกอริทึมใดที่สามารถหาเส้นทางฮามิลโทเนียลภายในกรอบเวลาที่เป็นพหุนาม การประกอบร่างรหัสพันธุกรรมดีเอ็นเอใน 20 ปีย้อนหลัง มีการพยายามใช้กราฟแสดงความคาบเกี่ยวในการประกอบร่างจีโนมมนุษย์ในเวอร์ชันแรกๆ ต่อมาจึงมีการนำกราฟ de Bruijn เข้ามาจำลองการเชื่อมต่อของดีเอ็นเอสายสั้น โดยมีข้อมูลบนเส้นเชื่อมโหนดแทนการอยู่ในโหนดแบบเดิม อัลกอริทึมที่ถูกออกแบบและพัฒนาในสมัยหลังจึงมีการใช้กราฟ de Bruijn เป็นหลักในการประกอบร่างจีโนม

การสร้างกราฟ de Bruijn จากชุดของดีเอ็นเอสายสั้น

ตัวอย่างข้างต้นอธิบายลักษณะของกราฟ de Bruijn โดยสร้างกราฟจากจีโนมที่ทราบลำดับเบสอยู่แล้ว อย่างไรก็ตามในการทำงานจริงเราไม่ทราบลำดับเบสของจีโนม คำถามคือ เราจะสร้างลำดับเบสของจีโนมจากชุดของดีเอ็น

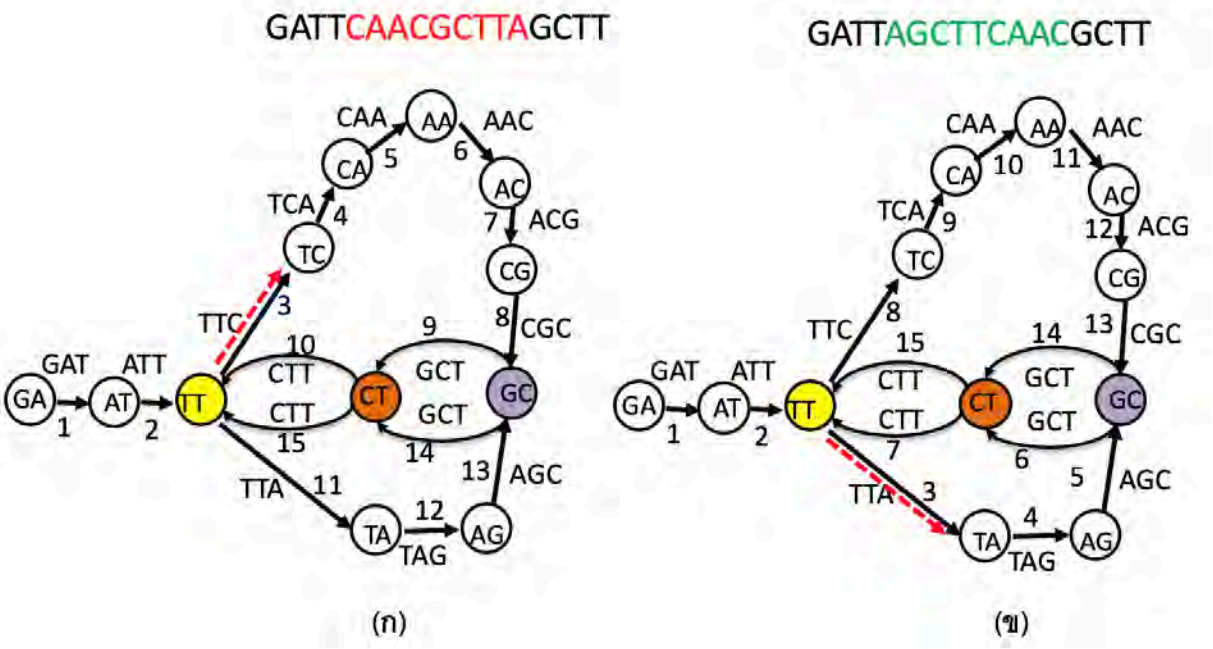
เอสายสั้นได้อย่างไร รูปที่ 2.15 แสดงการสร้างชุดของโหนดและเส้นเชื่อมในรูปแบบกราฟ de Bruijn โดยหลังจากเชื่อมโยงทุกโหนดให้เป็นกราฟเดียวกันแล้ว โหนดซ้ำจะถูกนำมารวมกันและสร้างเป็นกราฟ de Bruijn สุดท้ายในรูปที่ 2.14



รูปที่ 2.15 ขั้นตอนการสร้างกราฟ de Bruijn จากชุดของดีเอ็นเอสายสั้น

การประกอบร่างจีโนมโดยใช้ดีเอ็นเอสายคู่

กราฟ de Bruijn หนึ่งๆ สามารถมีเส้นทางออยเลอร์ได้มากกว่า 1 เส้นทาง ดังตัวอย่างในรูปที่ 2.16 แต่ในจีโนมต้นฉบับมีลำดับเบสเป็น **GATTCAACGCTTAGCTT** ซึ่งแสดงโดยเส้นทางในรูปที่ 2.16(ก) เพื่อเป็นการลดความคลุมเครือของการประกอบร่างจีโนมนี้ วิธีแรกที่เป็นไปได้คือการเพิ่มความยาวของ k-mer อย่างไรก็ตามการเพิ่มความยาวของ k-mer มีสมมติฐานว่าเครื่องหาลำดับเบสสามารถอ่านลำดับเบสสายยาวได้ ซึ่งในปัจจุบัน กลางปี พ.ศ. 2561 (ค.ศ. 2018) การหาลำดับเบสสายยาวยังมีราคาสูงมากและมีความผิดพลาดในการอ่านมากเมื่อเทียบกับการหาลำดับเบสสายสั้นโดยเทคโนโลยีเอ็นจีเอส แพลตฟอร์มอิลูมินา [17]

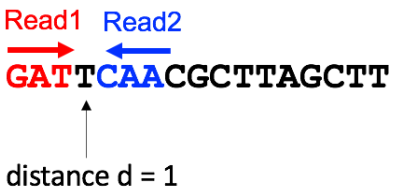


*รูปที่ 2.16 เส้นทางออยเลอร์สองเส้นทางในกราฟ de Bruijn เดียวกันที่สร้างจากสายอักขระต้นฉบับ **GATTCAACGCTTAGCTT** (ก) เส้นทางที่แสดงสายอักขระต้นฉบับ (ข) เส้นทางที่ไม่ตรงกับสายอักขระต้นฉบับ โดยเส้นประสีแดงแสดงทางแยกจากเส้นทางเริ่มต้น **GATT** เดียวกัน*

อีกแนวทางที่เป็นไปได้คือการออกแบบการทดลองให้เครื่องหาลำดับเบสอ่านรหัสพันธุกรรมแบบสายคู่ได้ ซึ่งปัจจุบันเทคโนโลยีการหาลำดับเบสสายสั้นอย่างอิลูมินาเน้นการอ่านรหัสพันธุกรรมแบบสายคู่ ถ้ากำหนด (*k*,*d*)-mer เป็น ($a_1 \ldots a_k | b_1, \ldots b_k$) โดย *k* คือจำนวนเบส และ *d* คือระยะห่างระหว่างคู่ของ k-mer ตามการวัดค่า *d* ในรูปที่ 2.17 ถ้า $d = 1$ พรีฟิกซ์ (prefix) และ ซัฟฟิกซ์ (suffix) ของ k-mer สายคู่สามารถกำหนดได้ดังต่อไปนี้

$$\mathrm{PREFIX}((a_1 \ldots a_k | b_1, \ldots b_k)) = (a_1 \ldots a_{k-1} | b_1, \ldots b_{k-1})$$
$$\mathrm{SUFFIX}((a_1 \ldots a_k | b_1, \ldots b_k)) = (a_2 \ldots a_k | b_2, \ldots b_k)$$

เช่น PREFIX((GAT|CAA))=(GA|CA) และ SUFFIX((GAT|CAA))=(AT|AA) ซึ่งซัฟฟิกซ์ของ k-mer นี้จะเป็นพรีฟิกซ์ของ k-mer ถัดไปคือ PREFIX((ATT|AAC))=(AT|AA)



รูปที่ 2.17 การนับระยะห่างระหว่างคู่ของสายดีเอ็นเอ

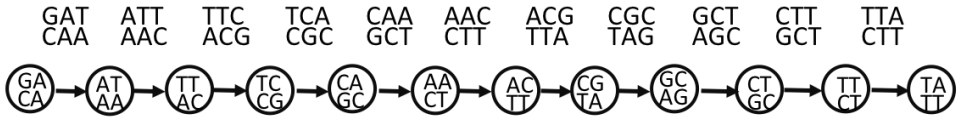
รูปที่ 2.18 แสดงตัวอย่างลำดับเบสสายคู่เทียบกับจีโนมต้นฉบับโดยในตัวอย่างนี้แต่ละสายของคู่ยาว 3-mer และแต่ละคู่ห่างกัน 1 เบส รูปที่ 2.19 แสดงเส้นทางกราฟที่เชื่อมต่อ 3-mer สายคู่ที่สร้างจากสายอักขระต้นฉบับ **GATTCAACGCTTAGCTT** โดยโหนดและเส้นเชื่อมอยู่ในรูปแบบของกราฟ de Bruijn

PAIRCOMPOSITION$_{3,1}$(GATTCAACGCTTAGCTT)

```
GAT CAA
 ATT AAC
  TTC ACG
   TCA CGC
    CAA GCT
     AAC CTT
      ACG TTA
       CGC TAG
        GCT AGC
         CTT GCT
          TTA CTT
```

3-mer ชุดซ้าย	GAT	ATT	TTC	TCA	CAA	AAC	ACG	CGC	GCT	CTT	TTA
3-mer ชุดขวา	CAA	AAC	ACG	CGC	GCT	CTT	TTA	TAG	AGC	GCT	CTT

*รูปที่ 2.18 การสร้างคู่ของ 3-mer ที่ห่างกัน 1 เบสจากสายอักขระต้นฉบับ **GATTCAACGCTTAGCTT***



*รูปที่ 2.19 เส้นทางกราฟ (path graph) ที่เชื่อมต่อ 3-mer สายคู่ที่สร้างจากสายอักขระต้นฉบับ **GATTCAACGCTTAGCTT***

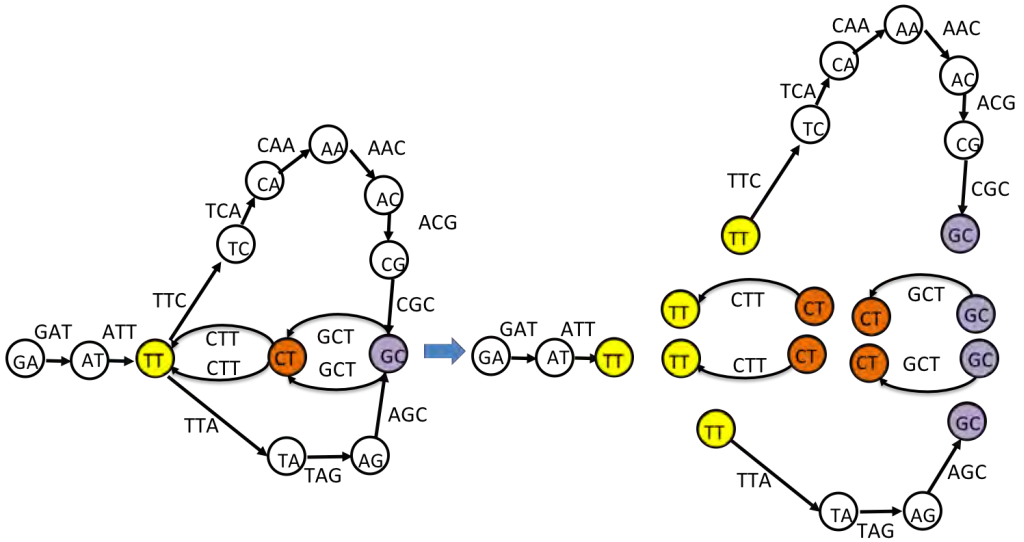
k-mer สายคู่จากเส้นทางกราฟในรูปที่ 2.19 สามารถสร้างเป็นกราฟ de Bruijn สุดท้ายโดยการรวมโหนดที่มีลำดับเบสภายในโหนดเป็นแบบเดียวกันซึ่งในกรณีนี้ไม่มี ดังนั้นกราฟ de Bruijn สุดท้ายจะเป็นกราฟเดียวกับรูปที่ 2.19 และมีเส้นทางออยเลอร์เดียวที่เป็นไปได้คือ **GATTCAACGCTTAGCTT** จะเห็นว่ากราฟ de Bruijn ที่สร้างจาก k-mer สายคู่สามารถช่วยลดความคลุมเครือในการเลือกเส้นทางเดินได้เมื่อเทียบกับกราฟที่สร้างจาก k-mer สายเดี่ยวในรูปที่ 2.16

บทส่งท้าย

ในขณะที่บทเรียนข้างต้นสามารถนำมาประยุกต์ใช้ในการแก้ปัญหาการประกอบร่างจีโนม (genome assembly) การออกแบบอัลกอริทึมเพื่อประกอบร่างจีโนมจากข้อมูลจริงที่ได้จากเครื่องหาลำดับเบสจำเป็นต้องพิจารณาปัจจัยที่เกี่ยวข้องเพิ่มเติมนอกเหนือจากโอกาสในการเกิดความผิดพลาดในการอ่านข้อมูลของเครื่องหาลำดับเบสเอง เช่น

(1) *การหาค่า k ที่เหมาะสมในการแบ่งข้อมูลลำดับเบสที่อ่านได้ซึ่งเรียกว่ารีด (read)* ออกเป็นลำดับเบสย่อยขนาด k-mer เพื่อให้แต่ละ k-mer มีโอกาสถูกต้องทุกลำดับเบสมากขึ้น เพื่อเป็นการเพิ่มความครอบคลุม (coverage) และความถูกต้องของลำดับเบสที่อ่านได้จากจีโนมในบริเวณหนึ่งๆ เปรียบได้กับการมีชิ้นส่วนของหนังสือพิมพ์ที่ใกล้เคียงกันหลายๆ ชิ้นโดยอาจจะเป็นชิ้นเล็กๆ แต่ถูกต้องเพื่อตรวจสอบซึ่งกันและกัน อย่างไรก็ตาม k ที่มีขนาดเล็กเกินไปจะทำให้เกิดทางเลือกมากขึ้นในกราฟ de Bruijn ทำให้การหาเส้นทางออยเลอร์มีความซับซ้อนมากขึ้น

(2) *การแตกจีโนมออกเป็นส่วนๆ เรียกว่าคอนทิก (contig)* เพื่อแก้ปัญหาเครื่องหาลำดับเบสไม่สามารถอ่านรหัสพันธุกรรมในบางบริเวณของจีโนมได้ เช่น บริเวณนั้นมีลำดับเบสซ้ำ (repeat) จำนวนมาก ทำให้เครื่องอ่านได้ไม่ดี ข้อมูลจึงหายไปในบริเวณเหล่านั้น เกิดช่องว่าง (gap) ของข้อมูล ทำให้เส้นเชื่อมระหว่างโหนดในกราฟ de Bruijn บางส่วนหายไป และทำให้ไม่สามารถหาเส้นทางออยเลอร์ที่สมบูรณ์ได้ โดยแต่ละคอนทิกจะถูกแสดงโดย maximal non-branching path ดังแสดงในรูปที่ 2.20

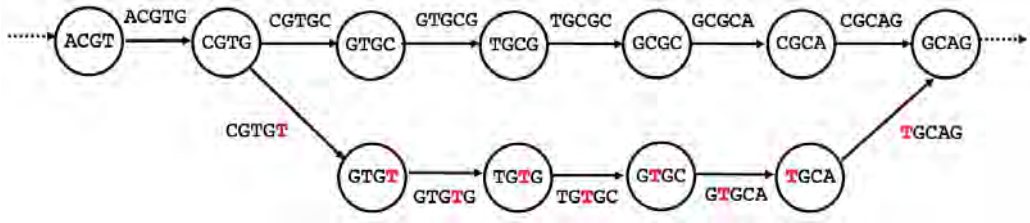


*รูปที่ 2.20 กราฟ de Bruijn ที่ถูกแตกออกเป็น 7 maximal non-branching paths ซึ่งถูกแสดงโดย **GATT**, **CTT**, **CTT**, **GCT**, **GCT**, **TTCAACGC** และ **TTAGC***

ตัวอย่างผลงานวิจัยของผู้เขียนตำรานี้ในโครงการหาลำดับเบสจีโนมของเชื้อรา *Ophiocordyceps polyrhachis-furcata* [54] โดยแพลตฟอร์ม 454 (Roche 454 GS FLX) ซึ่งเป็นเครื่องหาลำดับเบสสายสั้นแต่ยาวกว่าแพลต

ฟอร์มอิลูมินา ผลการประกอบร่างจีโนมแบบ *de novo* assembly (คือประกอบร่างจีโนมใหม่โดยไม่มีการเทียบกับจีโนมอ้างอิง เนื่องจากเป็นเชื้อที่ไม่เคยมีการหาลำดับเบสจีโนมมาก่อน) โดยใช้โปรแกรม Newbler v2.8 ได้ลำดับเบสในรูปแบบคอนทิก (contig) จำนวน 4,419 คอนทิก และเพื่อให้สามารถเชื่อมต่อคอนทิกเหล่านี้ให้ยาวขึ้น ได้มีการหาลำดับเบสเพิ่มเติมโดยใช้แพลตฟอร์มอิลูมินาที่ให้ข้อมูลเป็น ไลบรารี mate pairs โดยได้ผลข้อมูลเป็นรหัสพันธุกรรมสายคู่ที่มีช่องว่างระหว่างคู่ของสายขนาด 3, 6 และ 8 Kb (กิโลเบส) และนำรหัสพันธุกรรมสายคู่เหล่านี้มาประกอบร่างเข้ากับคอนทิกที่มีอยู่ก่อนหน้าโดยใช้โปรแกรมสร้างสแคฟโฟลด์ (scaffold) ชื่อ SSPACE 2.0 [55] ได้ผลเป็น 418 สแคฟโฟลด์ โดยแต่ละสแคฟโฟลด์ประกอบด้วยชุดของคอนทิกที่คั่นด้วยช่องว่าง หรือ gap และ มีคู่ของรีด (คู่ของ mate pair) ที่สายที่หนึ่งอยู่ในคอนทิกแรกและคู่ของมันตกอยู่ในคอนทิกที่สอง โดย 59 สแคฟโฟลด์มีความยาวมากกว่า 1 Kb และมีค่า N50 เท่ากับ 3.3 ล้านเบส ทั้งนี้ N50 คือค่ามัธยฐาน (median) ของความยาวคอนทิกหรือสแคฟโฟลด์ ใช้ในการประเมินคุณภาพของจีโนมที่ประกอบร่างได้ โดยถ้า N50 มีค่ามากแสดงว่าจีโนมที่ประกอบร่างขึ้นมานี้มีคุณภาพดีถึงดีมาก ทั้งนี้จากการเปรียบเทียบค่า N50 ของเชื้อรานี้กับราอื่นๆ ที่มีการหาลำดับเบสจีโนมมาก่อนถือว่าจีโนมที่ได้มีคุณภาพดีมาก เชื้อรา *Ophiocordyceps polyrhachis-furcata* มีขนาดของจีโนมโดยประมาณ 43 Mb (43 ล้านเบส) ในขณะที่จีโนมมนุษย์มีขนาดประมาณ 3 พันล้านเบส (3 Gb)

(3) *การจัดการเรื่องบับเบิ้ล (bubble)* ในกราฟ ที่เป็นผลจากการเกิดความผิดพลาดในการอ่านรีด ตัวอย่างเช่น ลำดับเบส **ACGTGTGCAG** ตรงตำแหน่งนิวคลีโอไทด์ **T** ถูกอ่านมาไม่ถูกต้องโดยเปลี่ยนจาก **C** เป็น **T** รูปที่ 2.21 เกิดบับเบิ้ลเนื่องจากมีเส้นทางแยกที่ประกอบจากชุด 5-mers ที่ไม่ถูกต้องเนื่องจากมีเบส **T** นี้เป็นส่วนประกอบ **CGTGT**, **GTGTG**, **TGTGC**, **GTGCA** และ **TGCAG** และมีโหนดปลายที่กลับมาร่วมเส้นทางกันเพราะเป็นโหนดที่มีลำดับเบสเดียวกัน โดยเส้นทางเหล่านี้จะต้องสั้นกว่าค่าขีดแบ่ง (threshold) ที่กำหนด



รูปที่ 2.21 การเกิดบับเบิ้ลในกราฟ de Bruijn จากลำดับเบสที่อ่านผิดโดยตำแหน่งที่เป็น C ถูกอ่านเป็น T

โปรแกรมประกอบร่างจีโนม (genome assembler) ส่วนใหญ่จะทำการกำจัดบับเบิ้ลเหล่านี้ออกไปซึ่งมีโอกาสที่เส้นทางที่ถูกต้องจะถูกกำจัดออกไปด้วยทำให้การประกอบร่างจีโนมมีข้อผิดพลาด นอกจากนี้ข้อมูลลำดับเบสซ้ำแต่ไม่ซ้ำทั้งหมด เช่น เกิดการแปรผันในบางลำดับเบสก็ทำให้เกิดบับเบิ้ลทับซ้อนกันได้ ซึ่งการกำจัดบับเบิ้ลโดยการเลือกเส้นทางใดเส้นทางหนึ่งจะทำให้เกิดข้อผิดพลาดในการประกอบร่างจีโนม เช่นกัน ดังนั้นโปรแกรมประกอบร่างจีโนมสมัยหลังจึงมีความพยายามในการแยกความแตกต่างระหว่างการเปลี่ยนแปลงของลำดับเบสที่เกิด

จากการอ่านข้อมูลผิดพลาดจากเครื่องหาลำดับเบสกับการเปลี่ยนแปลงของลำดับเบสที่เกิดจากการแปรผันที่เกิดขึ้นจริงของจีโนม

นอกจากนี้สมมติฐานในการออกแบบอัลกอรีทึมที่นำเสนอในบทเรียนนี้ไม่เป็นจริงสำหรับข้อมูลจริงที่อ่านได้จากเครื่องหาลำดับเบส เช่น ระยะห่างระหว่างลำดับเบสสายคู่ (paired-end read) มีค่าคงที่ รีดที่อ่านได้จากเครื่องหาลำดับเบสครอบคลุมทุกบริเวณในจีโนม เราทราบรูปแบบที่เป็นไปได้ทั้งหมดของ k-mer ในการหาลำดับเบสจีโนมของสิ่งมีชีวิตหนึ่งๆ ในทางปฏิบัติระยะห่างระหว่างลำดับเบสสายคู่จะถูกระบุเป็นช่วงของค่าแทนค่าคงที่และอาจแตกต่างกันไปตามการเตรียมดีเอ็นเอโดยห้องปฏิบัติการ

ตัวอย่างโปรแกรมประกอบร่างจีโนมที่มีการใช้งานอย่างแพร่หลาย

โปรแกรมที่มีการใช้งานอย่างแพร่หลายในการประกอบร่างจีโนมของสิ่งมีชีวิตกลุ่มยูแคริโอต เช่น SOAPdenovo2 [56, 57], ALLPATHS [58], Velvet [59], ABySS [60] ใช้กราฟ de Bruijn เป็นฐาน บทปริทัศน์ (review) ของนาคาราชาน นิราชาน (Nagarajan Nirajan) และคณะ [61] ได้ทำการเปรียบเทียบข้อดีข้อเสียของโปรแกรมที่ใช้ประกอบร่างจีโนม รวมทั้งการประยุกต์ใช้โปรแกรมเหล่านี้กับข้อมูลรหัสพันธุกรรมในระดับอาร์เอ็นเอ และข้อมูลรหัสพันธุกรรมที่หาลำดับเบสจากกลุ่มเชื้อในการศึกษาเมทาจีโนมิกส์ (metagenomics) เช่น กลุ่มของเชื้อที่เก็บจากสิ่งแวดล้อม กลุ่มของเชื้อที่เก็บจากลำไส้มนุษย์ กลุ่มเชื้อที่เก็บจากช่องปาก เป็นต้น

แบบฝึกหัดบทที่ 2

เขียนโปรแกรมเพื่อแก้ปัญหาที่เกี่ยวข้องกับการประกอบร่างจีโนมโดยใช้โจทย์ที่โรซาลินด์ (http://rosalind.info) ดังต่อไปนี้

1) Construct the De Bruijn Graph of a Collection of k-mers (http://rosalind.info/problems/ba3e/)
2) Find an Eulerian Path in a Graph (http://rosalind.info/problems/ba3g/)
3) Reconstruct a String From its Paired Composition (http://rosalind.info/problems/ba3j/)

ภาคผนวกบทที่ 2

WGS และ WES

ดับเบิลยูจีเอส (WGS) หรือการหาลำดับเบสทั้งจีโนม (whole genome sequencing) คือการหาลำดับเบสซึ่งครอบคลุมทั้งส่วนที่เป็นยีนที่สามารถแปลรหัสไปเป็นโปรตีนและส่วนอื่นทั้งหมดในจีโนม ในขณะที่ดับเบิลยูอีเอส (WES) หรือการหาลำดับเบสเอกโซม (whole exome sequencing) คือการหาลำดับเบสเฉพาะส่วนที่เป็นเอกซอนทั้งหมดในจีโนมซึ่งครอบคลุมเฉพาะส่วนที่เป็นยีนที่สามารถแปลรหัสต่อไปเป็นโปรตีนซึ่งครอบคลุมรหัสพันธุกรรมเพียง

ประมาณ 1-2% ของจีโนม แพลตฟอร์มหลักที่ใช้ในการหาลำดับเบสเอกโซมคือ NimbleGen, Agilent และ Illumina [62] โดยข้อมูลรหัสพันธุกรรมเอกโซมที่ได้จะถูกนำไปเทียบกับจีโนมอ้างอิงเพื่อวิเคราะห์หาการแปรผันลักษณะต่างๆ เช่น การแปรผันของนิวคลีโอไทด์เดี่ยวที่เรียกว่าเอสเอ็นวี (SNV; single nucleotide variant) และการสอดแทรกหรือขาดหายไปของลำดับเบสสายสั้นที่รวมเรียกว่าอินเดล (indel) เป็นต้น

บทที่ 3 การเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิง (Short read mapping to reference genome)

วัตถุประสงค์

- เพื่อให้นิสิตได้เห็นขั้นตอนหลัก (ขั้นตอนแรก) ในการวิเคราะห์ข้อมูลรหัสพันธุกรรมของจีโนมหนึ่งๆ เทียบกับจีโนมอ้างอิง
- เพื่อให้นิสิตคุ้นเคยกับข้อมูลที่เกี่ยวข้องและเข้าใจการพัฒนาวิธีการหาสายอักขระย่อยในสายอักขระหลักแบบเหมือนทั้งเส้นโดยเน้นความเร็วและการใช้หน่วยความจำที่มีประสิทธิภาพ
- เพื่อให้นิสิตได้เห็นตัวอย่างงานวิจัยและผลงานวิจัย รวมทั้งตัวอย่างโปรแกรมที่ใช้ในการหาสายอักขระย่อยในสายอักขระหลัก
- เพื่อให้นิสิตได้เห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทาย รวมถึงงานวิจัยอื่นๆ ที่เกี่ยวข้อง

ผลลัพธ์ที่คาดหวัง

- นิสิตสามารถอธิบายความแตกต่างรวมทั้งข้อดีข้อเสียระหว่างแพลตฟอร์มที่ใช้ในการหาลำดับเบสจีโนมได้
- นิสิตเข้าใจคุณลักษณะของข้อมูลตั้งต้นที่ได้จากการหาลำดับเบสจีโนม
- นิสิตสามารถอธิบายการทำงานของอัลกอริทึมหลักที่ใช้ในการหาสายอักขระย่อยในสายอักขระหลักได้
- นิสิตสามารถเขียนโปรแกรมที่ใช้ในการหาสายอักขระย่อยในสายอักขระหลักอย่างง่ายได้
- นิสิตสามารถยกตัวอย่างโปรแกรมที่ใช้ในการเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิงที่มีการใช้งานอย่างแพร่หลายได้
- นิสิตสามารถยกตัวอย่างความท้าทายที่ยังมีอยู่และสามารถนำเสนอแนวทางในการพัฒนาวิธีการแก้ปัญหาเหล่านี้ได้ รวมทั้งสามารถประยุกต์องค์ความรู้จากบทเรียนเพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้

เนื้อหาโดยสรุป

ในบทก่อนหน้า เราพูดถึงการหาลำดับเบสจีโนมและการประกอบร่างจีโนมจากดีเอ็นเอสายสั้นจำนวนมาก โดยการประกอบร่างจีโนมนี้เน้นการทำแบบ *de novo* คือสร้างจีโนมขึ้นมาใหม่จากการประกอบดีเอ็นเอสายสั้นเข้าด้วยกันโดยไม่มีจีโนมอ้างอิง ในบทนี้มีความแตกต่างจากบทที่ 2 โดยมีสมมติฐานว่ามีข้อมูลจีโนมอ้างอิงอยู่แล้ว ตัวอย่าง

เช่น จีโนมมนุษย์ โจทย์คือถ้ามีดีเอ็นเอสายสั้นจำนวนมากจากการหาลำดับเบส จะนำดีเอ็นเอสายสั้นเหล่านี้ไปเทียบกับจีโนมอ้างอิงอย่างไรให้เร็ว มีความถูกต้องสูง และใช้ทรัพยากรในการคำนวณต่ำ ผลการเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิง (short read mapping/alignment) นอกจากจะเป็นทางเลือกในการประกอบร่างจีโนมโดยมีสมมติฐานว่ามีจีโนมอ้างอิงแล้ว ยังสามารถใช้เป็นข้อมูลพื้นฐานในการวิเคราะห์การแปรผันของดีเอ็นเอในลักษณะต่างๆ เช่น เอสเอ็นวี (SNV; single nucleotide variant) หรือตัวแปรผันที่เกิดจากการเปลี่ยนแปลงนิวคลีโอไทด์เดี่ยว ซีเอ็นวี (CNV; copy number variation) หรือการแปรผันในจำนวนซ้ำของลำดับเบส เอสวี (SV; structural variation) หรือการแปรผันในเชิงโครงสร้างของจีโนมในลักษณะอื่น เช่น เกิดการกลับด้านของลำดับเบส (inversion) ในรหัสพันธุกรรมของตัวอย่างเทียบกับจีโนมอ้างอิง การแปรผันเมื่อเทียบระหว่างจีโนมของบุคคลภายในครอบครัว เช่น พ่อ แม่ ลูก (Trio) การแปรผันในระดับกลุ่มประชากร การแปรผันระหว่างจีโนมของเซลล์ปกติและเซลล์มะเร็ง ตัวอย่างโครง สร้างข้อมูลที่ใช้เทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิง เช่น ทรัย (trie) ซัฟฟิกซ์ทรี (suffix tree) ซัฟฟิกซ์อาร์เรย์ (suffix array) แนวคิดในเรื่องการบีบอัดสายอักขระ (string compression) และ Burrows-Wheeler Transform (BWT) ตัวอย่างโปรแกรมที่มีการใช้งานกันอย่างแพร่หลายและงานวิจัยที่เกี่ยวข้อง

บทที่ 3 การเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิง (Short read mapping to reference genome)

ประมาณ 1% ของทารกแรกเกิดมีปัญหาความบกพร่องด้านสติปัญญา (mental retardation) ซึ่งความบกพร่องนี้สามารถเกิดจากความผิดปกติทางพันธุกรรมที่หลากหลาย จึงยังไม่ทราบสาเหตุที่แน่ชัด ตัวอย่างเช่น ผู้ป่วยโรค Ohdo syndrome ไม่สามารถแสดงออกทางสีหน้าได้ (expressionless) หรือมีใบหน้าที่เหมือนใส่หน้ากาก (mask-like) ในปี ค.ศ. 2011 นักชีววิทยาได้ทราบสาเหตุของโรค Ohdo syndrome จากข้อมูลการกลายพันธุ์ (mutation) ในตำแหน่งต่างๆ ของรหัสพันธุกรรมที่เกิดร่วมกันในผู้ป่วย โดยสาเหตุหลักของโรคเกิดจากการกลายพันธุ์ในระดับดีเอ็นเอในบริเวณที่ทำให้การแปลรหัสไปเป็นโปรตีนหยุดเร็วกว่าปกติทำให้ได้โปรตีนที่ไม่สมบูรณ์ [63] อีกตัวอย่างคือกรณีเด็กชายนิโคลาส โวลเกอร์ (Nicholas Volker) ที่มีอาการลำไส้อักเสบอย่างรุนแรง โดยแพทย์ไม่ทราบสาเหตุของอาการและโรค ทำได้เพียงรักษาตามอาการโดยการผ่าตัดลำไส้หลายครั้งจนกระทั่งโรงเรียนแพทย์ของวิสคอนซิน (Medical college of Wisconsin) ตัดสินใจหาลำดับเบสดีเอ็นเอของนิโคลาสและทราบสาเหตุของอาการลำไส้อักเสบรุนแรงซึ่งเกิดจากการกลายพันธุ์ของยีน *XIAP* (X-linked inhibitor of apoptosis) ซึ่งนำไปสู่ความผิดปกติของระบบภูมิคุ้มกัน หลังทราบสาเหตุแล้วแพทย์ใช้วิธีภูมิคุ้มกันบำบัด (immunotherapy) ในการรักษา ซึ่งทำให้สามารถรักษาชีวิตของโวลเกอร์ไว้ได้ การหาบริเวณที่เกิดการแปรผัน (variation) ในจีโนมของบุคคลหนึ่งๆ เทียบกับจีโนมอ้างอิง และการแปรผันที่อาจเป็นการกลาย (mutation) คือมีผลต่อการเกิดโรคหรือความรุนแรงของโรค สามารถนำไปสู่การวินิจฉัยและเลือกวิธีการรักษาที่ตรงจุด ด้วยเทคโนโลยีการหาลำดับเบสที่มีอยู่การหาการแปรผันในจีโนมทำได้โดยการนำข้อมูลรหัสพันธุกรรมในรูปแบบดีเอ็นเอสายสั้น (รีด) จำนวนมากของผู้ป่วยไปเทียบกับจีโนมอ้างอิงและตรวจหาบริเวณที่มีความแตกต่างกัน ซึ่งการแปรผันมีได้หลายรูปแบบ เช่น การแปรผันของนิวคลีโอไทด์เดี่ยวเรียกว่าเอสเอ็นวี (SNV; single nucleotide variant) การแปรผันโดยมีชุดของลำดับเบสแบบสั้นแทรกสอดหรือหายไปเมื่อเทียบกับจีโนมอ้างอิงเรียกว่าอินเดล (indel) การแปรผันโดยมีชุดของลำดับเบสกลับด้านกับจีโนมอ้างอิง (inversion) การแปรผันโดยมีชุดของลำดับเบสย้ายจากโครโมโซมหนึ่งไปยังอีกโครโมโซมหนึ่ง (translocation) การแปรผันของจำนวนชุดดีเอ็นเอที่แตกต่างจากจีโนมอ้างอิงเรียกว่าซีเอ็นวี (CNV; copy number variation) เป็นต้น ซึ่งในบทเรียนนี้จะเน้นการหาการแปรผันของนิวคลีโอไทด์เดี่ยว โดยเริ่มจากการแนะนำอัลกอริทึมต่างๆ ที่ถูกออกแบบและพัฒนาเพื่อใช้ในการค้นหาสายอักขระย่อยซึ่งเป็นตัวแทนของดีเอ็นสายสั้น (รีด) ในสายอักขระหลักซึ่งเป็นตัวแทนของจีโนมอ้างอิง โดยเน้นการหาบริเวณในสายอักขระหลักที่ลำดับเบสเหมือนกับสายอักขระย่อยทั้งสาย (exact match) ส่วนท้ายบทนำเสนอตัวอย่างวิธีการหาเอสเอ็นวีโดยวิธีการหาสายอักขระย่อยในสายอักขระหลักแบบประมาณ

ปัญหาการเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิง

การเทียบดีเอ็นเอสายสั้นจำนวนมากกับจีโนมอ้างอิงเป็นตัวอย่างปัญหาการหาชุดของสายอักขระย่อยในสายอักขระหลักซึ่งถูกนิยามในนิยามปัญหาที่ 3.1 ต่อไปนี้

นิยามปัญหาที่ 3.1 ปัญหาการหาชุดของสายอักขระย่อยในสายอักขระหลัก

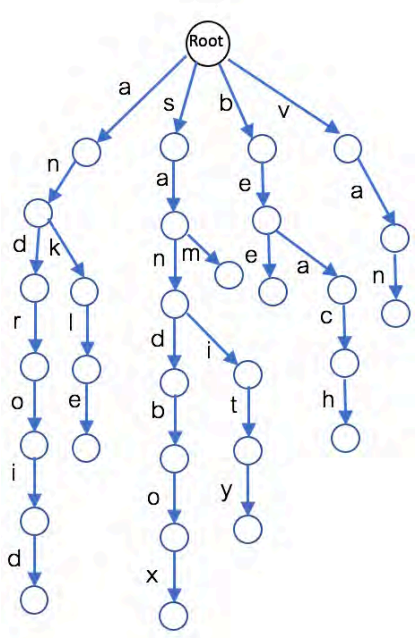
ปัญหาการหาชุดของสายอักขระย่อยในสายอักขระหลัก (Multiple Pattern Matching Problem) หาตำแหน่งที่พบทั้งหมดของแต่ละสายอักขระย่อยในสายอักขระหลัก	
ข้อมูลเข้า	สายอักขระหลักและชุดของสายอักขระย่อย
ผลลัพธ์	ตำแหน่งเริ่มต้นทั้งหมดในสายอักขระหลักที่พบสายอักขระย่อยแต่ละรูปแบบ

วิธีการแก้ปัญหาการหาชุดของสายอักขระย่อยในสายอักขระหลักแบบทำทุกรูปแบบ

หลักการพื้นฐานของแนวทางทำทุกรูปแบบ (brute force) สำหรับการหาชุดของสายอักขระย่อยในสายอักขระหลักทำได้โดยนำแต่ละสายอักขระย่อยมาเทียบหาบริเวณที่เหมือนที่สุดในสายอักขระหลัก โดยแต่ละสายอักขระย่อยเป็นอิสระต่อกัน ถ้าความยาวของสายอักขระหลักคือ |Text| และความยาวของแต่ละสายอักขระย่อยเท่ากับ |Pattern| เวลาที่ใช้ในการหาสายอักขระย่อยนั้นในสายอักขระหลักจะเท่ากับ $O(|Text| * |Pattern|)$ และถ้าความยาวรวมของสายอักขระย่อยทั้งหมดเท่ากับ |Patterns| เวลาที่ใช้ในการหาชุดของสายอักขระย่อยในสายอักขระหลักจะเท่ากับ $O(|Text| * |Patterns|)$ ถ้านำวิธีการทำทุกรูปแบบนี้ไปใช้หาว่าแต่ละรีดที่อ่านได้จากเครื่องหาลำดับเบสอยู่ตรงไหนในจีโนมอ้างอิงจะใช้เวลานานมาก โดยความยาวรวมของทุกรีดหรือ |Patterns| มีขนาดประมาณ 1 TB (terabyte) หรือประมาณหนึ่งล้านล้านอักขระในขณะที่จีโนมอ้างอิงมีความยาวประมาณ 3 GB (gigabyte) หรือประมาณสามพันล้านอักขระ

วิธีการแก้ปัญหาการหาชุดของสายอักขระย่อยในสายอักขระหลักโดยใช้ทรัย

แนวทางที่สองในการหาชุดของสายอักขระย่อยในสายอักขระหลัก อัลกอริทึมถูกออกแบบให้อ่านสายอักขระหลักเพียงครั้งเดียว ในขณะที่วิธีการทำทุกรูปแบบข้างต้นจำนวนครั้งในการอ่านสายอักขระหลักจะเท่ากับจำนวนสายอักขระย่อย เปรียบได้กับการนำแต่ละสายอักขระย่อยใส่รถยนต์ส่วนตัวแล้ววิ่งไปตามถนนสายอักขระหลักโดยแต่ละสายอักขระย่อยมีรถยนต์เป็นของตัวเอง ในขณะที่วิธีการที่สองนี้เปรียบได้กับการขนชุดสายอักขระย่อยทั้งหมดด้วยรถโดยสารคันเดียวแล้วใช้รถโดยสารนี้วิ่งไปตามถนนสายอักขระหลักเพียงครั้งเดียว โดยโครงสร้างข้อมูลที่นำมาใช้บรรจุชุดของสายอักขระย่อยทั้งหมดเข้าด้วยกันนี้เรียกว่าทรัย (Trie) ดังแสดงในรูปที่ 3.1 วิธีการสร้างทรัยจากชุดของสายอักขระย่อยแสดงไว้ในรหัสเทียม (pseudo code) ที่ 3.1 `TrieConstruction()`



รูปที่ 3.1 ทรัยที่มีชุดของสายอักขระย่อยประกอบด้วย “and”, “ankle”, “android”, “sand”, “sandbox”, “sanity”, “sam”, “bee”, “beach” และ “van”

นิยามปัญหาที่ 3.2 ปัญหาการสร้างทรัยจากชุดของสายอักขระย่อย

ปัญหาการสร้างทรัยจากชุดของสายอักขระย่อย	
ข้อมูลเข้า	ชุดของสายอักขระย่อย (สตริงย่อย)
ผลลัพธ์	ทรัยของชุดสายอักขระย่อย

วิธีการสร้างทรัยจากชุดของสายอักขระย่อย

รหัสเทียมที่ 3.1 TrieConstruction()

```
TrieConstruction(Patterns)
    Trie <- กราฟที่มีโหนดเดียวคือรูทโหนด
    for แต่ละสตริงย่อย pattern ใน Patterns
        currentNode <- root ของ Trie
        for แต่ละตัวอักษร c ใน pattern
            if มีเส้นเชื่อมที่มีค่าเป็น c จาก currentNode ไปโหนด n
                เปลี่ยน currentNode ไปที่โหนด n
            else
                เพิ่มโหนดใหม่ใน Trie
                เพิ่มเส้นเชื่อมชี้จาก currentNode มาที่โหนดใหม่นี้
                    และใส่ค่าเส้นเชื่อมนี้เป็น c
                เปลี่ยน currentNode มาที่โหนดใหม่นี้
    ส่งกลับ Trie
```

หยุดคิด	ทรัยที่สร้างขึ้นนี้นำไปใช้ค้นหาตำแหน่งของสายอักขระย่อยในสายอักขระหลักอย่างไร

ทรัยที่สร้างขึ้นสามารถนำไปหาสายอักขระย่อยในสายอักขระหลักได้ดังแสดงในรหัสเทียมที่ 3.2 `Prefix TrieMatching()` โดยมีหลักการทำงานคือเริ่มอ่านจากอักขระแรกของสายอักขระหลักเทียบกับค่าของแต่ละเส้นเชื่อมจากโหนดราก (โหนดตั้งต้นในทรัย) ถ้าพบทำการอ่านอักขระถัดไปจากสายอักขระหลัก พร้อมทั้งเปลี่ยนค่าโหนดตั้งต้นในทรัยเป็นโหนดที่ถูกชี้โดยเส้นเชื่อมที่มีค่าตรงกับอักขระที่อ่านมาได้ก่อนหน้า และวนกลับไปทดสอบว่ามีเส้นเชื่อมจากโหนดตั้งต้นใหม่นี้ที่มีค่าตรงกับอักขระล่าสุดที่อ่านได้จากสายอักขระหลักหรือไม่ วนซ้ำการทำงานนี้และส่งกลับค่าเส้นทางจากโหนดรากถึงโหนดใบ (สายอักขระย่อยที่พบ) ถ้าสามารถหาเส้นทางนั้นได้

วิธีการหาชุดของสายอักขระย่อยในสายอักขระหลักโดยวิธีการ prefix trie matching

รหัสเทียมที่ 3.2 PrefixTrieMatching()

```
PrefixTrieMatching(Text, Trie)
    symbol <- อักขระแรกของสายสตริงหลัก (Text)
    v <- รูทของ Trie
    while True
        if v เป็นโหนดใบของ Trie
            ส่งกลับ สายสตริงย่อยซึ่งเป็นเส้นทางจากรูทมาที่โหนดใบนี้
        else if มีเส้นเชื่อมระหว่างโหนด v ไปยัง w ที่แสดงอักขระเดียวกับ symbol
            symbol <- อักขระถัดไปในสายสตริงหลัก (Text)
            v <- w เลื่อนโหนดตั้งต้นใน Trie จาก v เป็น w
        else
            output "ไม่พบ"
            return
```

เนื่องจาก `PrefixTrieMatching()` จะตรวจสอบสายอักขระหลักโดยเริ่มอ่านจากอักขระแรกเสมอ ดังนั้นเพื่อให้สามารถนำ `PrefixTrieMatching()`มาใช้ในการหารูปแบบของสายอักขระย่อยทั้งหมดที่พบในสายอักขระหลักก็สามารถทำได้โดยนำ `PrefixTrieMatching()`มากราดตรวจ (scan) สายอักขระหลัก โดยสายอักขระหลักที่เป็นข้อมูลเข้าของ `PrefixTrieMatching()`จะมีขนาดสั้นลงในแต่ละรอบของการทำงาน โดยอักขระทางซ้ายมือสุดจะถูกตัดออกไป ดังรหัสเทียมที่ 3.3

รหัสเทียมที่ 3.3 `TrieMatching()`

```
TrieMatching(Text, Trie)
    while ยังมีอักขระในสายสตริงหลัก (Text)
        PrefixTrieMatching(Text, Trie)
        Text <- สายสตริงหลักตัดอักขระซ้ายสุดออกไป
```

เวลาที่ใช้ในการหาชุดของสายอักขระย่อยในสายอักขระหลักโดยใช้ทรัยแบ่งออกเป็น 2 ส่วน ส่วนแรกเป็นเวลาในการสร้างทรัยซึ่งเท่ากับ $O(|Patterns|)$ ตามความยาวรวมของสายอักขระย่อยทั้งหมด ส่วนที่สองเป็นเวลาในการใช้ทรัยเพื่อค้นหาสายอักขระย่อยทั้งหมดในสายอักขระหลักผ่าน `TrieMatching()` ซึ่งใช้เวลา $O(|Text| * |Longest\ pattern|)$ โดย |Text| คือความยาวของสายอักขระหลักและ |Longest pattern| คือความยาวของสายอักขระย่อยที่ยาวที่สุด ในขณะที่เวลาที่ใช้ในกรณีของการทำทุกรูปแบบเท่ากับ $O(|Text| * |Patterns|)$

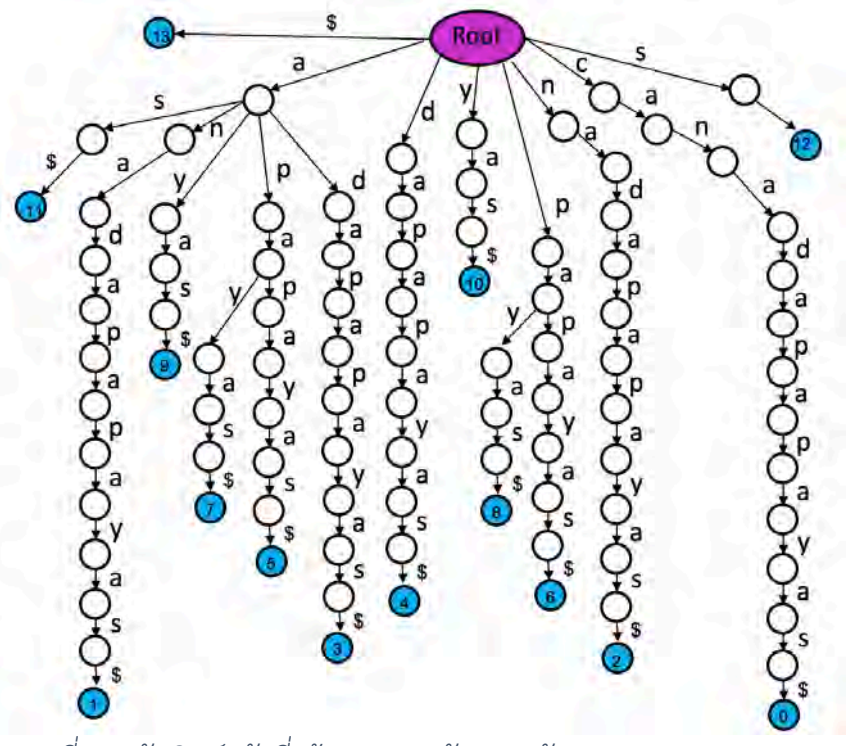
หยุดคิด	การใช้ทรัยเพื่อหาชุดของสายอักขระย่อยในสายอักขระหลักทำงานได้เร็วกว่าวิธีการทำทุกรูปแบบมาก คำถามคือการใช้ทรัยมีข้อจำกัดอะไรบ้างหรือไม่

ถึงแม้การใช้ทรัยทำให้การหาสายอักขระย่อยในสายอักขระหลักทำได้เร็วขึ้นมาก แต่ก็ต้องใช้หน่วยความจำเป็นจำนวนมากในการเก็บโครงสร้างข้อมูลทรัย การสร้างทรัยจากข้อมูลรีดทั้งหมดที่ได้จากการหาลำดับเบสจีโนมมนุษย์อาจต้องใช้หน่วยความจำถึง 1 TB

วิธีการแก้ปัญหาการหาชุดของสายอักขระย่อยในสายอักขระหลักโดยใช้ซัฟฟิกซ์ทรัย

เนื่องจากการสร้างทรัยจากชุดของสายอักขระย่อยหรือรีดทั้งหมดที่อ่านได้จากเครื่องหาลำดับเบสจะได้ทรัยที่มีขนาดใหญ่มากซึ่งต้องใช้หน่วยความจำจำนวนมาก จึงมีแนวความคิดในการสร้างทรัยจากชุดของซัฟฟิกซ์ทั้งหมดที่เป็นไปได้ของข้อมูลสายอักขระหลักแทน และเมื่อต้องการหาชุดของสายอักขระย่อยในสายอักขระหลักจะนำสายอักขระสายย่อยเหล่านี้มาเทียบกับซัฟฟิกซ์ทรัย (suffix trie) ของสายอักขระหลักแทน รูปที่ 3.2 แสดงตัวอย่างของซัฟฟิกซ์ทรัยที่สร้างจากสายอักขระหลัก “canadapapayas$” ซึ่งมีการใส่โหนดใบปิดท้ายแต่ละซัฟฟิกซ์ในทรัยเพื่อเป็นดัชนีของตัวอักขระแรกของซัฟฟิกซ์นั้นๆ ซึ่งในกรณีนี้มีทั้งหมด 13 ซัฟฟิกซ์ ตัวอย่างเช่น ซัฟฟิกซ์ “canadapapayas$” ตัวอักขระแรกคือตัวอักษร c มีดัชนีที่ 0 ซัฟฟิกซ์ “anadapapayas$” ตัวอักขระแรกคือตัว a มีดัชนีที่ 1 การหาสายอักขระย่อยอย่าง “yas” จะสามารถเทียบลำดับอักขระจนพบโหนดใบ มีค่าเป็น 10 ซึ่งหมายถึงพบคำว่า “yas” ในสายอักขระหลักโดยตัวอักษรแรกของคำ (y) อยู่ตำแหน่งที่ 10 (ใช้ดัชนีเริ่มที่ 0) ถ้าค้นหาคำว่า “apa” จะพบทางแยกเป็นสองเส้นทาง ซึ่งหมายถึงพบ “apa” 2 ตำแหน่งในสายอักขระหลักคือตำแหน่งที่ 5 และ 7 ตามลำดับ

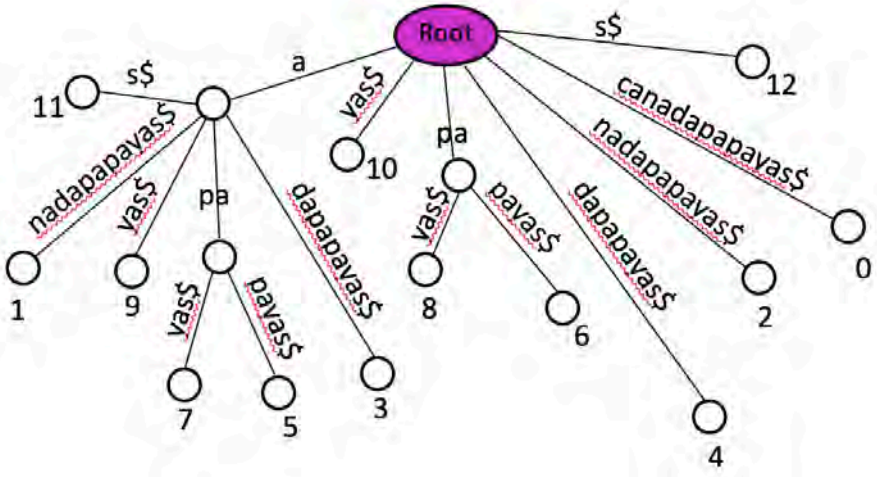
การสร้างสายอักขระหลักเป็นซัฟฟิกซ์ทรัยแทนการสร้างทรัยจากสายอักขระย่อยทั้งหมด จะได้จำนวนซัฟฟิกซ์ทั้งหมดเท่ากับความยาวของสายอักขระหลัก (ในตัวอย่างสายอักขระหลัก canadapapayas$ มีทั้งหมด 14 ซัฟฟิกซ์โดยรวมสายอักขระว่างด้วย) และถ้าสายอักขระหลักมีขนาดเท่ากับ |Text| จำนวนโหนดที่แทนซัฟฟิกซ์ทั้งหมดจะเท่ากับ $|Text| * (|Text|-1)/2 = O(|Text|^2)$



รูปที่ 3.2 ซัฟฟิกซ์ทรัยที่สร้างจากสายอักขระหลัก “canadapapayas$”

วิธีการแก้ปัญหาการหาชุดของสายอักขระย่อยในสายอักขระหลักโดยใช้ซัฟฟิกซ์ทรี

ซัฟฟิกซ์ทรี (suffix tree) เป็นโครงสร้างข้อมูลที่ลดจำนวนโหนดในซัฟฟิกซ์ทรัยลง โดยเส้นเชื่อมทั้งหมดที่ไม่มีทางแยกจะถูกรวมเข้าด้วยกันกลายเป็นเส้นเชื่อมเดียว และมีค่าของเส้นเป็นชุดของอักขระที่ต่อกันแทน การหาสายอักขระย่อยในสายอักขระหลักใช้วิธีการเดียวกับการใช้ซัฟฟิกซ์ทรัย รูปที่ 3.3 แสดงตัวอย่างซัฟฟิกซ์ทรีที่สร้างจากซัฟฟิกซ์ทรัยในรูปก่อนหน้า



รูปที่ 3.3 ซัฟฟิกซ์ทรีที่สร้างจากสายอักขระหลัก “canadapapayas$”

ในขณะที่ซัฟฟิกซ์ทรัยอาจมีจำนวนโหนดในทรัยเป็นกำลังสอง (quadratic) ของความยาวของสายอักขระหลัก ในกรณีของซัฟฟิกซ์ทรีจำนวนของโหนดในทรีมีมากที่สุดเท่ากับ $2 * |\text{Text}|$ (มาจากจำนวนโหนดใบเท่ากับ |Text| และโหนดภายในอีกอย่างมากจำนวน |Text| โหนด) ดังนั้นจะใช้หน่วยความจำอย่างมาก $O(|\text{Text}|)$ อย่างไรก็ตาม การเก็บซัฟฟิกซ์ทรีตามตัวอย่างข้างต้น ใช้หน่วยความจำไม่ต่างจากการเก็บซัฟฟิกซ์ทรัย เพราะยังต้องเก็บค่าของเส้นเชื่อมตามจำนวนไบต์ของอักขระที่นำมาต่อกัน ในทางปฏิบัติเราไม่จำเป็นต้องเก็บสายอักขระย่อยที่เป็นค่าของแต่ละเส้นเชื่อมไว้แต่สามารถเก็บตัวชี้ (pointer) ของตำแหน่งเริ่มต้นและความยาวของสายอักขระย่อยแทน นอกจากนี้ยังสามารถสร้างซัฟฟิกซ์ทรีได้โดยตรง ใช้เวลาเป็นสมการเส้นตรง โดยไม่ต้องสร้างจากซัฟฟิกซ์ทรัยอีกที

ถึงแม้ซัฟฟิกซ์ทรีจะใช้หน่วยความจำลดลงมากจาก $O(|\text{Text}|^2)$ เป็น $O(|\text{Text}|)$ โดยเฉลี่ย เมื่อเทียบกับซัฟฟิกซ์ทรัย ในทางปฏิบัติซัฟฟิกซ์ทรียังต้องการหน่วยความจำประมาณ 20 เท่าของ |Text| ดังนั้นถ้าต้องการสร้างซัฟฟิกซ์ทรีของจีโนมมนุษย์ (สายอักขระหลักที่แทนจีโนมอ้างอิง) ขนาด 3 GB จะต้องใช้หน่วยความ จำประมาณ 60 GB ซึ่งน้อยลงมากเมื่อเทียบกับการสร้างทรัยจากรีดทั้งหมดที่ได้จากการหาลำดับเบสของจีโนมซึ่งใช้หน่วยความจำประมาณ 1 TB ก่อนศึกษาแนวทางเพิ่มเติมเพื่อลดหน่วยความจำที่ต้องใช้ลงไปอีก ขอให้ลองแก้ปัญหาต่อไปนี้โดยใช้ซัฟฟิกซ์ทรี

ฝึกหัด

ปัญหาการหาสายอักขระย่อยซ้ำที่ยาวที่สุด (Longest Repeat Problem)

หาสายอักขระย่อยเหมือนที่ยาวที่สุดและพบในสายอักขระหลักมากกว่า 1 ตำแหน่ง

ข้อมูลเข้า	สายอักขระหลัก
ผลลัพธ์	สายอักขระย่อยเหมือนที่ยาวที่สุดและพบในสายอักขระหลักมากกว่า 1 ตำแหน่ง

ฝึกหัด

ปัญหาการหาสายอักขระย่อยเหมือนที่ยาวที่สุดและพบในสายอักขระหลักทั้งสองเส้น

(Longest Shared Substring Problem)

หาสายอักขระย่อยเหมือนที่ยาวที่สุดและพบในสายอักขระหลักทั้งสองเส้น

ข้อมูลเข้า	สายอักขระหลัก 2 เส้น
ผลลัพธ์	สายอักขระย่อยเหมือนที่ยาวที่สุดและพบในสายอักขระหลักทั้งสองเส้น

ฝึกหัด	
ปัญหาการหาสายอักขระย่อยสั้นที่สุดที่พบในสายอักขระหลักเส้นเดียว (Shortest Non-Shared Substring Problem) หาสายอักขระย่อยสั้นที่สุดและพบในสายอักขระหลักเส้นใดเส้นหนึ่งเท่านั้น	
ข้อมูลเข้า	สายอักขระหลัก 2 เส้น
ผลลัพธ์	สายอักขระย่อยสั้นที่สุดที่พบในสายอักขระหลักเส้นใดเส้นหนึ่งเท่านั้น

วิธีการแก้ปัญหาการหาชุดของสายอักขระย่อยในสายอักขระหลักโดยใช้ซัฟฟิกซ์อาร์เรย์

วิธีการสร้างซัฟฟิกซ์อาร์เรย์ จะต้องทำการเรียงลำดับซัฟฟิกซ์ทั้งหมดที่มีอยู่ (รูปที่ 3.2) ตามลำดับตัวอักษรในพจนานุกรมดังในรูปที่ 3.4 โดยถือว่าอักขระ $ เป็นอักขระแรกของอักขระที่เป็นตัวอักษร ซัฟฟิกซ์อาร์เรย์จะเก็บรายการ (list) ของดัชนีตัวอักษรแรกของแต่ละซัฟฟิกซ์ที่มีการเรียงลำดับแล้ว ดังแสดงในบรรทัดต่อไปนี้

SUFFIXARRAY(“canadapapayas$”) = [13, 3, 1, 5, 7, 11, 9, 0, 4, 2, 6, 8, 12, 10]

ทั้งนี้ซัฟฟิกซ์อาร์เรย์สามารถสร้างโดยง่ายหลังจากได้ชุดของซัฟฟิกซ์ที่มีการเรียงลำดับแล้ว โดยอัลกอริทึมเรียงข้อมูลที่เร็วที่สุดมีการเปรียบเทียบค่าระหว่างข้อมูล $O(n \log n)$ ครั้ง ซึ่งหมายถึงต้องมีการเปรียบเทียบ $O(|Text| \log |Text|)$ ครั้ง อย่างไรก็ตามยังมีอัลกอริทึมที่สามารถสร้างซัฟฟิกซ์อาร์เรย์โดยใช้เวลาในสมการเชิงเส้นและต้องการหน่วยความจำประมาณ 1/5 เท่าของหน่วยความจำที่ใช้ในการเก็บซัฟฟิกซ์ทรี ทำให้ใช้หน่วยความจำประมาณ 12 GB สำหรับเก็บซัฟฟิกซ์อาร์เรย์ของจีโนมมนุษย์

The Burrows-Wheeler Transform

การใช้ซัฟฟิกซ์อาร์เรย์ลดความต้องการใช้หน่วยความจำลงไปเป็นอย่างมากและเป็น state of the art ในการทำ pattern matching จนกระทั่งต้นคริสตศักราช 2000 เกิดคำถามว่ามีโครงสร้างข้อมูลอื่นอีกไหมที่สามารถเข้ารหัสสายอักขระหลัก (Text) โดยใช้หน่วยความจำที่มีขนาดใกล้เคียงกับขนาดของสายอักขระหลัก ก่อนจะตอบปัญหานี้ลองพิจารณาการบีบอัดสายอักขระ (text compression) โดยใช้การเข้ารหัสแบบ run-length encoding การเข้ารหัสโดยวิธีการนี้แทนที่ลำดับของสายอักขระที่เป็นอักขระเดียวกันเรียกว่ารัน (run) ด้วยจำนวนซ้ำที่พบอักขระนั้นๆ เช่น **TTTTTGGGAAAACCCCCCA** จะถูกแทนที่ด้วย **5T3G4A6C1A** เป็นต้น การเข้ารหัสแบบ run-length นี้จะมีประสิทธิภาพถ้าในสายอักขระนั้นมีชุดของอักขระซ้ำขนาดยาวจำนวนมาก อย่างไรก็ตามในกรณีของจีโนมมนุษย์ไม่ได้มีชุดอักขระซ้ำจำนวนมาก แต่มีความซ้ำของชุดอักขระเป็นชุดๆ เรียกว่า รีพีท (repeat) แทรกในสายอักขระหลักหรืออยู่ต่อเนื่องกันไป ถ้ามีวิธีการที่สามารถแปลงจากรีพีทเหล่านี้ไปเป็นชุดของรันก่อนและค่อยเข้ารหัสแบบ run-length อีกทีก็น่าจะเป็นแนวทางที่ดี

ซัฟฟิกซ์ทั้งหมดที่มีการเรียงแล้ว	ตำแหน่งเริ่มต้นของซัฟฟิกซ์นั้นๆ
$	13
adapapayas$	3
anadapapayas$	1
apapayas$	5
apayas$	7
as$	11
ayas$	9
canadapapayas$	0
dapapayas$	4
nadapapayas$	2
papayas$	6
payas$	8
s$	12
yas$	10

รูปที่ 3.4 รายการของซัฟฟิกซ์ทั้งหมดของสายอักขระหลัก "canadapapayas$" ที่มีการเรียงลำดับตามตัวอักษร (โดยถือว่า $ เป็นอักษรลำดับแรก) และตำแหน่งเริ่มต้นของซัฟฟิกซ์นั้นๆ ที่พบในสายอักขระหลัก

ถ้าสร้างรันโดยนำอักขระในสายอักขระหลัก เช่น **TACGTAACGATACGAT** มาเรียงลำดับตามตัวอักษรได้เป็น **AAAAACCCGGGTTTT** ซึ่งเข้ารหัสเป็น **5A3C3G4T** วิธีการนี้จะสามารถแสดงจีโนมมนุษย์ที่มีขนาด 3 GB โดยใช้ตัวเลขเพียง 4 ตัว

หยุดคิด	วิธีการเข้ารหัสจีโนมข้างต้นมีข้อผิดพลาดอย่างไร

การนำอักขระทั้งหมดมาเรียงลำดับตามตัวอักษรจากนั้นนับจำนวนซ้ำ ใช้ไม่ได้กับการบีบอัดข้อมูลเพราะว่าสายอักขระที่แตกต่างกัน เช่น **GCATCATGCAT** และ **ACTGACTACTG** ที่มีชุดและจำนวนของอักขระเท่ากัน แต่มีลำดับของตัวอักษรแตกต่างกันจะถูกเรียงลำดับเป็น **AAACCCGGTTT** เหมือนกันและบีบอัดออกมาเป็นสายอักขระเดียวกันคือ **3A3C2G3T** เป็นต้น ซึ่งไม่สามารถแตก (decompress) สายอักขระที่ถูกบีบอัดนี้ออกมาเป็นสายอักขระต้นฉบับที่ถูกต้องได้ เนื่องจากไม่สามารถระบุตำแหน่งที่ถูกต้องของแต่ละอักขระได้

การสร้าง Burrows-Wheeler Transform

ไมเคิล เบอร์โรวส์ (Michael Burrows) และเดวิด วีลเลอร์ (David Wheeler) นำเสนออีกวิธีการในการแปลงรีพีทต่างๆ ให้เป็นรันในปี ค.ศ. 1994 วิธีการนี้เริ่มจากการสร้างลำดับของอักขระในสายอักขระหลักโดยการทำ cyclic rotation อักขระตัวขวาสุดจะถูกนำไปเพิ่มไว้ที่ตำแหน่งซ้ายสุดและเลื่อนสายอักขระที่เหลือไปทางขวา 1 ตำแหน่ง

ทำอย่างนี้ไปเรื่อยๆ จนกว่าการหมุนสายอักขระนี้จะได้ผลกลับมาเป็นสายอักขระหลักตั้งต้น ขั้นที่สองทำการเรียงลำดับสายอักขระที่เกิดจากการหมุนรอบละ 1 ตัวอักขระนี้ตามลำดับพจนานุกรม ได้เป็นเมทริกซ์ของเบอร์โรวส์-วีลเลอร์ `M(Text)` ดังแสดงในรูปที่ 3.5 โดยตัวอย่าง Text ในรูปนี้คือ “canadapapayas$” และได้ Burrows-Wheeler Transform หรือ `BWT(Text)` เป็น “sncdpyp$aaaaaa” (คอลัมน์ขวาสุดของเมทริกซ์)

ผลการหมุนสายอักขระหลักทีละ 1 อักขระ	ผลการเรียงสายอักขระที่ได้ทางซ้ายตามตัวอักษร M("canadapapaya$")
canadapapayas$	$canadapapayas
$canadapapayas	adapapayas$can
s$canadapapaya	anadapapayas$c
as$canadapapay	apapayas$canad
yas$canadapapa	apayas$canadap
ayas$canadapap	as$canadapapay
payas$canadapa	ayas$canadapap
apayas$canadap	canadapapayas$
papayas$canada	dapapayas$cana
apapayas$canad	nadapapayas$ca
dapapayas$cana	papayas$canada
adapapayas$can	payas$canadapa
nadapapayas$ca	s$canadapapaya
anadapapayas$c	yas$canadapapa
(ก)	(ข)

รูปที่ 3.5 (ก) ผลการหมุนสายอักขระหลัก “canadapapayas$” (ข) เมทริกซ์เบอร์โรวส์-วีลเลอร์ที่เป็นผลจากการเรียงสายอักขระที่ผ่านการหมุน โดยคอลัมน์ขวาสุด คือ Burrows-Wheeler Transform

นิยามปัญหาที่ 3.3 ปัญหาการสร้าง Burrows-Wheeler Transform

ปัญหาการสร้าง Burrows-Wheeler Transform สร้าง Burrows-Wheeler Transform จากสายอักขระ	
ข้อมูลเข้า	สายอักขระ
ผลลัพธ์	BWT(สายอักขระ)

หยุดคิด	รูปที่ 3.5 แสดงตัวอย่างการหา `BWT(Text)` จาก `M(Text)` คำถามคือเราสามารถสร้าง `BWT(Text)` โดยใช้หน่วยความจำน้อยลง ถ้ามีข้อมูลเข้าเป็น Text และ `SUFFIXARRAY(Text)` ได้หรือไม่

ความสัมพันธ์ระหว่างรีพีทและรัน

จากรูปที่ 3.5 ข้างต้น **BWT("canadapapayas$") = "sncdpyp$aaaaaa"** จะสังเกตเห็นว่ามีรันเช่น "aaaaaa" เกิดขึ้น คำถามคือทำไม Burrows-Wheeler Transform ถึงมีรันนี้เกิดขึ้น ลองจินตนาการว่าถ้าเราเอาคำทั้งหมดจากผลงานตีพิมพ์ในปี ค.ศ. 1958 ของวัตสันและคริกเกี่ยวกับดีเอ็นเอสายคู่มาสร้างเมทริกซ์ของเบอร์โรวส์-วีลเลอร์ จะพบว่าคำว่า "and" บ่อย ซึ่งถ้าพิจารณาผลการหมุนสายอักขระทั้งหมดในเมทริกซ์ของเบอร์โรวส์-วีลเลอร์ที่มีการเรียงลำดับสายอักขระจะพบว่าทุกบรรทัดของสายอักขระที่ขึ้นต้นด้วย "nd" คอลัมน์สุดท้ายของบรรทัดเดียวกันมักเป็นตัวอักษร "a" และบรรทัดเหล่านี้จะอยู่กันเป็นกลุ่ม ดังแสดงในรูปที่ 3.6 สายอักขระย่อยอย่าง "apa" ในสายอักขระหลัก "canadapapayas$" เทียบได้กับคำว่า "and" ในตัวอย่างนี้ ซึ่งก็เป็นคำอธิบายตัวอักษร "a" 2 ใน 6 ตัว ในรีพีท "aaaaaa" ของ **BWT("canadapapayas$") = "sncdpyp$aaaaaa"** การประยุกต์ใช้ Burrows-Wheeler Transform กับจีโนมทำให้สามารถแปลงรีพีทต่างๆ ให้อยู่ในรูปแบบของรัน และสามารถใช้วิธีการเข้ารหัสอย่าง run-length เพื่อบีบอัดข้อมูล BWT เพิ่มเติมได้

```
nd Corey (1).  They kindly made their manuscript availa ..... a
nd criticism, especially on interatomic distances.  We  ..... a
nd cytosine.  The sequence of bases on a single chain d ..... a
nd experimentally (3,4) that the ratio of the amounts o ..... u
nd for this reason we shall not comment on it.  We wish ..... a
nd guanine (purine) with cytosine (pyrimidine).  In oth ..... a
nd ideas of Dr.  M. H. F. Wilkins, Dr.  R. E. Franklin  ..... a
nd its water content is rather high.  At lower water co ..... a
nd pyrimidine bases.  The planes of the bases are perpe ..... a
nd stereochemical arguments.  It has not escaped our no ..... a
nd that only specific pairs of bases can bond together  ..... u
nd the atoms near it is close to Furberg's 'standard co ..... a
nd the bases on the inside, linked together by hydrogen ..... a
nd the bases on the outside.  In our opinion, this stru ..... a
nd the other a pyrimidine for bonding to occur.  The hy ..... a
nd the phosphates on the outside.  The configuration of ..... a
nd the ration of guanine to cytosine, are always very c ..... a
nd the same axis (see diagram).  We have made the usual ..... u
nd their co-workers at King's College, London.  One of  ..... a
```

รูปที่ 3.6 ส่วนของ M(Text) ที่ถูกเลือกออกมาจากคำทั้งหมดที่ได้จากผลงานตีพิมพ์ของวัตสันและคริกเกี่ยวกับดีเอ็นเอสายคู่ในปี ค.ศ. 1958

(ที่มา: รูปที่ 9.9 ของ [52])

การแปลง Burrows-Wheeler Transform กลับเป็นสายอักขระตั้งต้น

การบีบอัดข้อมูลจะไม่มีประโยชน์ถ้าไม่สามารถแปลงข้อมูลที่ถูกบีบอัดเป็นข้อมูลต้นฉบับที่ถูกต้องได้ อย่างไรก็ตามสายอักขระต้นฉบับที่ถูกต้องสามารถแปลงกลับจาก BWT ได้ เช่น **BWT(Text)= "ard$rcaaaabb"** ในเมทริกซ์ของเบอร์โรวส์-วีลเลอร์ที่ทราบเฉพาะคอลัมน์ซ้ายสุดและขวาสุดดังต่อไปนี้

อักขระคอลัมน์ขวาสุดเรียงจากบนลงล่างคือ BWT ส่วนคอลัมน์ซ้ายสุดคืออักขระตัวแรกของแต่ละสายอักขระที่ผ่านการหมุน 1 ตัวอักษรและมีการเรียงลำดับแล้ว คำถามคืออักขระตัวแรกสุดของสายอักขระต้นฉบับคืออักขระใด ถ้าพิจารณาจากเมทริกซ์นี้คำตอบคือตัวอักษร “a” ซึ่งมาจากตัวอักษรซ้ายสุดของบรรทัดที่ 4 ที่มี “$” เป็นตัวอักษรทางขวาสุด

หยุดคิด	อักขระตัวถัดไปของสายอักขระตั้งต้นข้างต้นคืออักขระใด

จากวิธีคิดข้างต้นพบว่าตัวอักขระถัดไปของสายอักขระต้นฉบับอาจเป็น “b”, “c”, หรือ “d” ก็ได้ ซึ่งเป็นตัวอักขระซ้ายสุดในบรรทัดที่ 7, 9, และ 10 ตามลำดับ เพราะทั้งสามบรรทัดมี “a” เป็นตัวอักขระขวาสุด โดยจะเป็น “b”, “c” หรือ “d” ขึ้นอยู่กับว่า “a” ตัวซ้ายสุดในบรรทัดที่ 4 นั้นเป็น a ตัวไหนใน BWT “ard$rcaaaabb” เช่น ถ้า “a” นั้นเป็น “a” ตัวที่ 7 อักขระตัวที่สองในสายอักขระต้นฉบับจะเป็น “b” ถ้า “a” นั้นเป็นตัวที่ 9 อักขระตัวที่สองในสายอักขระต้นฉบับจะเป็น “c” และถ้า “a” เป็นตัวที่ 10 อักขระตัวที่สองในสายอักขระต้นฉบับจะเป็น “d” เป็นต้น คำถามคือเราทราบได้อย่างไรว่า “a” ที่เป็นตัวซ้ายสุดจากบรรทัดที่ 4 นี้ เป็น “a” ตัวไหนใน BWT

คุณสมบัติ First-Last

เพื่อให้สามารถตัดสินได้ว่าอักขระที่เกิดซ้ำนั้นเป็นตัวไหน จึงเพิ่มการบอกตำแหน่งที่ปรากฏของอักขระนั้นๆ ในคอลัมน์ซ้ายสุดจากบนลงล่าง ตามตัวอย่างต่อไปนี้ (แสดงเฉพาะตำแหน่งของอักขระ “a”) พิจารณา a_1 ที่เป็นอักขระซ้ายสุดในบรรทัดที่ 2 จากเฉลยของสายอักขระต้นฉบับที่มีอยู่ “$a_1$dapapayas$can” ถ้ามีการหมุนจนได้สายอักขระต้นฉบับจะได้ “cana$_1$dapapayas$” ซึ่ง a_1 อยู่ในลำดับที่ 2 ของ “a” หกตัวที่ปรากฏในสายอักขระต้นฉบับ

$canadapapayas
$a_1$dapapanas$can
$a_2$nadapapanas$c
$a_3$papanas$canad
$a_4$panas$canadap
a_5s$canadapapay
$a_6$yas$canadapap
canadapapayas$
dapapanas$cana
nadapapanas$ca
panas$canadapa
papanas$canada
s$canadapapaya
yas$canadapapa

พิจารณาต่อว่า a_1 ที่อยู่คอลัมน์ซ้ายสุดในบรรทัดที่ 2 นี้เป็น “a” ตัวไหนในคอลัมน์ขวาสุด ถ้ามีการหมุน a_1 ในบรรทัดนี้ไปทางขวา 1 ตำแหน่ง จะได้สายอักขระใหม่คือ “dapapayas$cana$_1$” ซึ่งตรงกับบรรทัดที่ 9 ดังตัวอย่างต่อไปนี้

$canadapapayas
$a_1$dapapanas$can
$a_2$nadapapanas$c
$a_3$papanas$canad
$a_4$panas$canadap
a_5s$canadapapay
$a_6$yas$canadapap
canadapapayas$
dapapanas$cana$a_1$
nadapapanas$ca
panas$canadapa
papanas$canada
s$canadapapaya
yas$canadapapa

ฝึกหัด	ลองหาดูว่า “a” อีก 5 ตัว (a_2, a_3, a_4, a_5, a_6) อยู่ที่บรรทัดไหนบ้างในคอลัมน์ขวาสุด

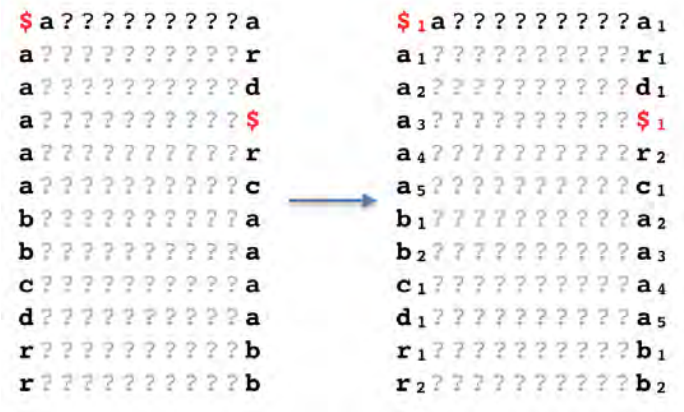
จากการลองหาตำแหน่งของ a_2, a_3, a_4, a_5, a_6 จะพบลำดับของอักษร “a” เหล่านี้อยู่ในลำดับเดียวกับลำดับที่อยู่ในคอลัมน์ซ้ายสุดดังแสดงในรูปด้านล่างนี้ โดยสามารถสรุปคุณสมบัติ First-Last ได้ว่าอักขระใดๆ ในคอลัมน์ซ้ายสุดที่อยู่ในลำดับ k ของสายอักขระต้นฉบับ อักขระตัวเดียวกันนั้นเมื่ออยู่ในคอลัมน์ขวาสุดจะอยู่ในลำดับที่ k ของสายอักขระต้นฉบับเช่นกัน

เพื่อเป็นการอธิบายว่าทำไมคุณสมบัติ First-Last ข้างต้นถึงเป็นจริง พิจารณาเมทริกซ์ย่อยต่อไปนี้โดยเน้นเฉพาะบรรทัดที่มี a_i อยู่ในคอลัมน์ซ้ายสุดในเมทริกซ์ย่อยทางซ้ายมือ ถ้าลองหมุน a_i ในแต่ละบรรทัดไปต่อท้ายสายอักขระในบรรทัดเดียวกันตามที่แสดงในเมทริกซ์ย่อยทางขวามือ จะพบว่าไม่มีผลกระทบกับลำดับของสายอักขระในแต่ละบรรทัด ซึ่งลักษณะนี้เป็นจริงสำหรับทุกอักขระและทุกสายอักขระ

การประยุกต์ใช้คุณสมบัติ First-Last ในการแปลง Burrows-Wheeler Transform กลับเป็นสายอักขระตั้งต้น

จาก `BWT(Text)=“ard$rcaaaabb”` ในเมทริกซ์ของเบอร์โรวส์-วีลเลอร์ที่ทราบเฉพาะคอลัมน์ซ้ายสุดและขวาสุด ดังเมทริกซ์ทางซ้ายของรูปต่อไปนี้ เมื่อมีการใส่ดัชนีของแต่ละอักขระโดยอาศัยคุณสมบัติของ First-Last จะได้เมทริกซ์ทางขวา ซึ่งการหาอักขระตัวแรกของสายอักขระต้นฉบับสามารถหาได้จากอักขระที่อยู่ในคอลัมน์ซ้ายสุดในบรรทัดที่มีอักขระในคอลัมน์ขวาสุดเป็น “$” ซึ่งในที่นี้คืออักษร “a” ตามที่ได้อธิบายมาก่อนหน้า สำหรับอักขระถัดไปในตำแหน่งที่สองจะเป็นตัวอักษร “b” (b_2) โดยอาศัยดัชนีจากคุณสมบัติ First-Last เนื่องจาก

“a” ในตำแหน่งแรกคือ a_3 และอักขระถัดไปหาได้จากอักขระที่อยู่ในคอลัมน์ซ้ายสุดของบรรทัดที่มี a_3 เป็นอักขระที่อยู่ในคอลัมน์ขวาสุด จากนั้นอักขระถัดไปจะเป็น “r” (r_2), “a” (a_4), “c” (c_1), “a” (a_5) ตามลำดับ



ฝึกหัด	ลองสร้างสายอักขระต้นฉบับจาก BWT “enwvpeoseu$llt”

วิธีการหาชุดของสายอักขระย่อยในสายอักขระหลักโดยใช้ Burrows-Wheeler Transform

จาก BWT ที่มีข้อมูลเฉพาะคอลัมน์แรกและคอลัมน์สุดท้ายและมีดัชนีของแต่ละอักขระแล้ว เราสามารถหาสายอักขระย่อยในสายอักขระหลักโดยตรวจดูแต่ละอักขระในสายอักขระย่อยเรียงลำดับจากขวามาซ้าย ดังแสดงในรูปที่ 3.7 โดยให้เลขที่บรรทัดเริ่มจาก 0 สมมติว่าถ้าต้องการหาสายอักขระย่อย “apa” ในสายอักขระต้นฉบับ อักขระ “a” ทางขวาสุดใน “apa” จะถูกนำมาดูในคอลัมน์ซ้ายสุดว่ามีบรรทัดใดบ้างที่เป็น “a” และดูว่าอักขระตัวก่อนหน้ามันคือ “p” หรือไม่โดยดูจากคอลัมน์ขวาสุดในบรรทัดเดียวกัน พบว่ามี 2 บรรทัดคือบรรทัดที่ 4 และ 6 ที่เข้าเงื่อนไข ซึ่งคือ p_1, และ p_2 ตามลำดับ ถ้านำ p_1 และ p_2 นี้มาตรวจสอบต่อว่าอยู่บรรทัดใดบ้างในคอลัมน์ซ้ายสุดและดูว่าคอลัมน์ทางขวาสุดคือ “a” หรือไม่ พบว่าทั้ง 2 บรรทัดคือบรรทัดที่ 10 และ 11 เข้าเงื่อนไข นำ a_3 และ a_4 มาตรวจสอบต่อว่าอยู่บรรทัดใดบ้างในคอลัมน์ซ้ายสุด และเนื่องจากไม่มีอักขระในสายอักขระย่อยแล้ว หมายความว่าพบสายอักขระย่อย “apa” 2 ตำแหน่งในสายอักขระต้นฉบับ

<u>หมายเหตุ</u>

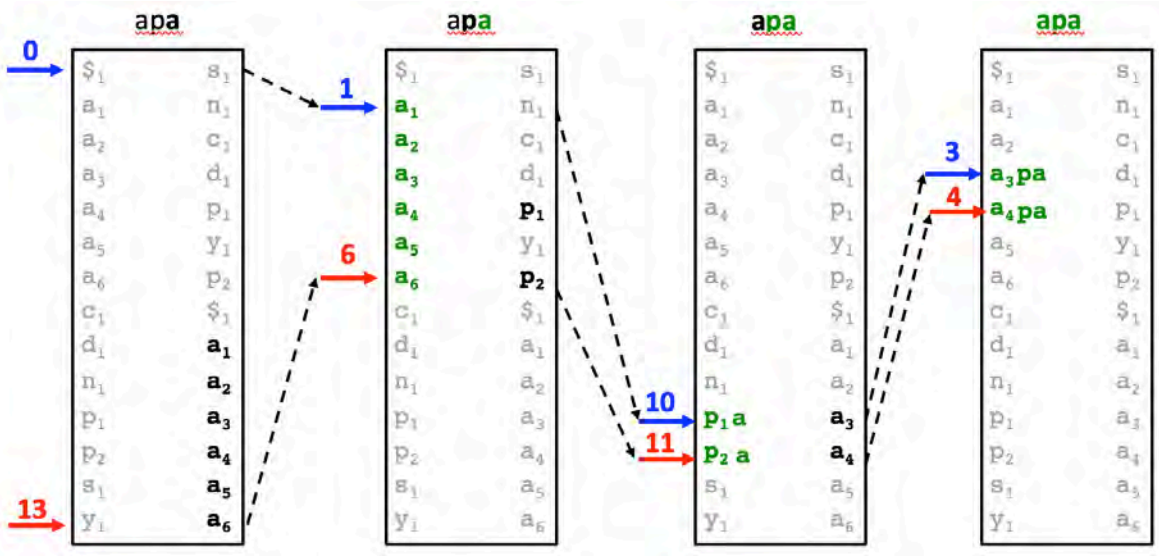
เริ่มไล่อักขระจากคอลัมน์ซ้ายสุด อักขระในคอลัมน์ขวาสุดในบรรทัดเดียวกันคืออักขระที่มาก่อนหน้า

เริ่มไล่อักขระจากคอลัมน์ขวาสุด อักขระในคอลัมน์ซ้ายสุดในบรรทัดเดียวกันคืออักขระที่ตามมา

การหาบรรทัดในคอลัมน์ซ้ายสุดของอักขระทางขวาสุด

จากคำอธิบายข้างต้นในการหาสายอักขระย่อยโดยไล่ดูทีละอักขระในสายอักขระย่อยจากขวามาซ้าย ในขั้นตอนเดินย้อนกลับแต่ละรอบนี้จะต้องมีการเปลี่ยนชุดของบรรทัดทางซ้ายที่ต้องพิจารณาในรอบนั้นๆ ดังแสดงในรูปที่

3.7 ต่อไปนี้ โดยในรอบแรกชุดของบรรทัดทางซ้ายที่ต้องพิจารณาคือบรรทัดที่ 1-6 ในรอบที่สองบรรทัดที่ 10-11 และรอบที่สามบรรทัดที่ 3-4 ซึ่งจะเห็นว่าสามารถใช้ตัวชี้เพียง 2 ตัวคือ *top* กับ *bottom* เช่น ในรอบแรก *top* = 1 และ *bottom* = 6 ในรอบที่สอง *top* = 10 และ *bottom* = 11 เพื่อให้การเปลี่ยนชุดของบรรทัดที่ต้องพิจารณาทำได้รวดเร็วในแต่ละรอบ สามารถเพิ่มฟังก์ชัน `LASTTOFIRST(i)` ทำหน้าที่หาเลขที่บรรทัดทางซ้ายของตัวอักขระที่ถูกระบุโดยเลขที่บรรทัดทางขวา จากตัวอย่างในรูปที่ 3.7 `LASTTOFIRST(4)` จะได้ค่าส่งกลับเป็น 10 และ `LASTTOFIRST(6)` จะได้ค่าส่งกลับเป็น 11 เป็นต้น รหัสเทียมที่ 3.4 `BWMatching()` แสดงโค้ดที่ใช้ในการหาจำนวนตำแหน่งที่พบสายอักขระย่อยที่ต้องการหาในสายอักขระต้นฉบับที่อยู่ในรูปของ BWT



รูปที่ 3.7 การเปลี่ยนค่าของตัวชี้ top และ bottom ของบรรทัดที่ต้องพิจารณาในแต่ละรอบ

รหัสเทียมที่ 3.4 BWMatching()

```
BWMatching(FirstColumn, LastColumn, Pattern, LASTTOFIRST)
    top <- 0
    bottom <- จำนวนบรรทัดทั้งหมด
    while top <= bottom
        if ยังมีอักขระในสตริงย่อย Pattern
            symbol <- อักขระทางขวาสุดของสตริงย่อย
            นำอักขระขวาสุดนี้ออกจากสตริงย่อย
            if มีตำแหน่งอักขระใน LastColumn ในช่วงระหว่าง top ถึง bottom ที่เป็นตัวเดียวกับ symbol
                topIndex <- ตำแหน่งแรกในช่วง top ถึง bottom ที่เป็นตัวเดียวกับ symbol
                bottomIndex <- ตำแหน่งสุดท้ายในช่วง top ถึง bottom ที่เป็นตัวเดียวกับ symbol
                top <- LASTTOFIRST(topIndex)
                bottom <- LASTTOFIRST(bottomIndex)
            else
                ส่งกลับ ค่า 0
        else
            ส่งกลับ ค่า bottom-top+1 ซึ่งคือค่าจำนวนตำแหน่งในสตริงหลักที่พบสายสตริงย่อย
```

วิธีการหาชุดของสายอักขระย่อยในสายอักขระหลักโดยไม่ต้องเหมือนกันทั้งสาย

วิธีการที่อธิบายมาก่อนหน้าเน้นการหาสายอักขระย่อยในสายอักขระหลักที่มีประสิทธิภาพและใช้หน่วยความจำน้อยที่สุด โดยสายอักขระย่อยทั้งสายจะต้องเหมือนกับบางส่วนของสายอักขระหลัก ในหัวข้อนี้เป็นการประยุกต์ใช้วิธีการข้างต้นในการหาสายอักขระย่อยในสายอักขระหลักโดยอาจมีบางอักขระที่ต่างกัน

นิยามปัญหาที่ 3.4 ปัญหาการหาชุดของสายอักขระย่อยในสายอักขระหลักแบบประมาณ

ปัญหาการหาชุดของสายอักขระย่อยในสายอักขระหลักแบบประมาณ (Multiple Approximate Pattern Matching Problem) หาตำแหน่งทั้งหมดบนสายอักขระหลักที่พบสายอักขระย่อยโดยอาจมีอักขระบางส่วนแตกต่างกัน	
ข้อมูลเข้า	สายอักขระหลัก ชุดของสายอักขระย่อย และค่าจำนวนเต็ม d
ผลลัพธ์	ตำแหน่งเริ่มต้นทั้งหมดในสายอักขระหลักที่พบสายอักขระย่อยแต่ละรูปแบบโดยสายอักขระย่อยเหล่านั้นสามารถต่างกับสายอักขระหลักได้ d อักขระ

จากสายอักขระสองสายต่อไปนี้ เราสามารถหาว่าสายอักขระย่อยที่ต่างจากสายอักขระหลัก 1 อักขระ โดยการแบ่งสายอักขระย่อยออกเป็นสองส่วน และนำทั้งสองส่วนนั้นมาหาในสายอักขระหลักแบบต้องเหมือนทั้งเส้น (exact match) ถ้าทั้งสองส่วนเหมือนกับสายอักขระหลักทั้งเส้นให้ตรวจสอบเพิ่มเติมว่าสายอักขระย่อยทั้งสายมี 1 อักขระที่ต่างจากสายอักขระหลักหรือไม่

สายอักขระย่อย **acttggct**

สายอักขระหลัก **....ggcacactaggctcc....**

วิธีการนี้สามารถนำไปประยุกต์ใช้กับจำนวนความต่างมากกว่า 1 (d>1) ได้ โดยถ้าสายอักขระย่อยต่างจากสายอักขระหลักอย่างมาก d อักขระ หมายความว่า ทั้งสายอักขระย่อยและสายอักขระหลักจะมีส่วนของสายอักขระที่เหมือนกันยาวที่สุด k อักขระ (k-mer) 1 สาย ตัวอย่างเช่น ถ้าเรามีสายอักขระย่อยยาว 20 อักขระและค่า d = 3 สายอักขระย่อยนี้จะถูกแบ่งออกเป็น d+1 ส่วน โดยมีความยาวของแต่ละส่วนเป็น 20/(3+1) = 5 ซึ่งเราสามารถนำทั้งสี่ส่วนไปหาในสายอักขระหลักแบบเหมือนทั้งเส้น เช่น

สายอักขระย่อย acttaggctcgggataatcc

สายอักขระหลัก ....actaagtctcgggataagcc....

สิ่งที่สังเกตได้นี้มีประโยชน์เพราะสามารถนำแต่ละส่วนย่อยนั้นไปเทียบหากับสายอักขระหลักแบบเหมือนทั้งเส้นโดยใช้วิธีการอย่างซัฟฟิกซ์ทรีหรือซัฟฟิกซ์อาร์เรย์ได้ ถ้าสายอักขระย่อยมีความยาว 23 อักขระและต่างจากสายอักขระหลัก 3 อักขระ คำถามคือสายอักขระย่อยจะมีส่วนของสายยาวที่สุดขนาด 5 หรือ 6 อักขระที่เหมือนกับสายอักขระหลัก

ในคำอธิบายข้างต้นยังไม่ได้กล่าวถึงวิธีการหาค่า k โดยค่า k นี้มีทฤษฎีอธิบายดังต่อไปนี้

ทฤษฎี ถ้าในส่วนที่เหมือนกัน n อักขระระหว่างสายอักขระสองสาย มีอย่างมาก d อักขระที่ต่างกัน สายอักขระสองสายนั้นจะต้องมีส่วนของสายขนาด k อักขระ (k-mer) ร่วมกัน โดย k = $\lfloor n/(d+1) \rfloor$

พิสูจน์ แบ่งสายอักขระแรกออกเป็น d+1 ส่วน โดยที่ d ส่วนแรกมีความยาวเท่ากับ k และ ส่วนสุดท้ายมีความยาวอย่างน้อย k จากคำถามข้างต้น เมื่อกำหนด d = 3 จะแบ่ง 23 อักขระได้เป็น d + 1 = 4 ส่วน ซึ่ง 3 ส่วนแรกยาว $\lfloor 23/(3+1) \rfloor = \lfloor 23/4 \rfloor = 5$ อักขระและส่วนสุดท้ายยาว 8 อักขระ ดังต่อไปนี้

acttaggctcgggataatccgga

ถ้ากระจายตำแหน่งที่ไม่เหมือนจำนวร 3 ตำแหน่งลงไปในสายอักขระส่วนต่างๆ ข้างต้น จะพบว่าตำแหน่งที่ไม่เหมือนเหล่านี้จะมีผลกระทบกับสายอักขระ 3 ส่วน ซึ่งจะเหลืออีกส่วนที่ยาวอย่างน้อย k ที่ไม่ถูกกระทบ ดังนั้นส่วนย่อยที่ยาว k นี้จะเหมือนกันระหว่างสายอักขระสองสาย

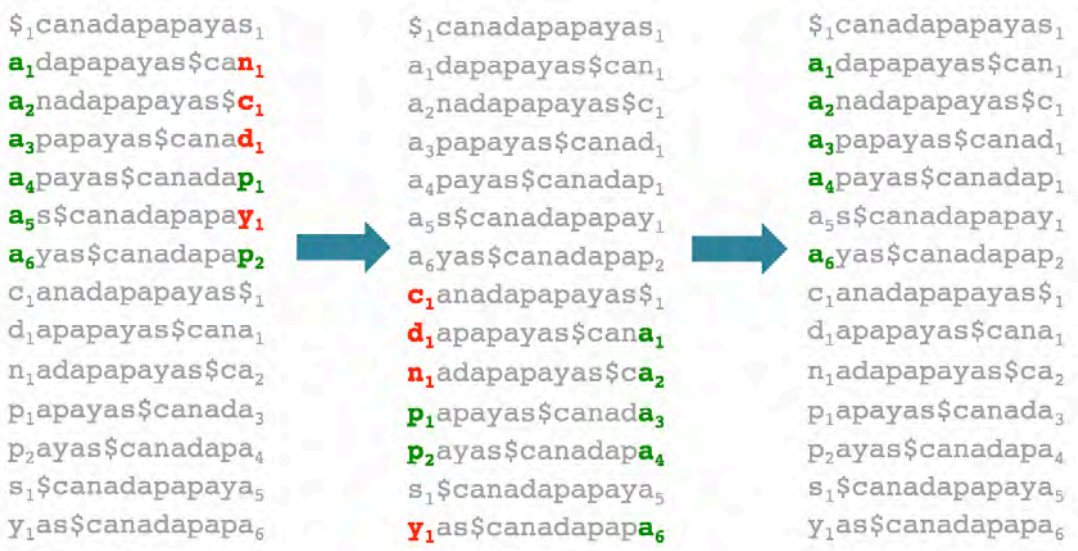
ดังนั้นอัลกอริทึมที่ใช้ในการหาสายอักขระย่อยที่มีความยาว n อักขระ ในสายอักขระหลัก โดยสามารถมีอักขระที่ต่างกันได้ไม่เกิน d มีขั้นตอนหลักประกอบด้วย แบ่งสายอักขระย่อยยาว n ออกเป็น d+1 ส่วน โดย d ส่วนแรกยาว k = $\lfloor n/(d+1) \rfloor$ เรียกว่าสีด (seed) หรือชิ้นส่วนเริ่มต้น นำสีดเหล่านี้ไปหาในสายอักขระหลักและตรวจสอบดูว่าสีดใดบ้างที่ทั้งเส้นเหมือนกับส่วนของสายอักขระหลัก เมื่อได้สีดเหล่านี้แล้วขั้นถัดไปจะทำการขยายสีดทั้งด้านซ้ายและขวาเพื่อดูว่าสายอักขระย่อยทั้งเส้นที่เกิดจากการขยายสีดหนึ่งๆ นี้ ยังต่างจากสายอักขระหลักไม่เกิน d อักขระหรือไม่

วิธีการหาชุดของสายอักขระย่อยในสายอักขระหลักแบบประมาณโดยใช้ Burrows-Wheeler Transform

การประยุกต์ใช้แนวทางของเบอร์โรวส์-วีลเลอร์ กับการหาชุดของสายอักขระย่อยในสายอักขระหลักแบบประมาณนั้น ขั้นตอนการเทียบทีละอักขระของสายอักขระย่อยกับคอลัมน์ซ้ายสุดถ้าไม่ตรงจะไม่หยุดการทำงาน ถ้าจำนวนอักขระที่ไม่ตรงนั้นยังไม่เกิน d อักขระ BWT ทางซ้ายมือ (รูปที่ 3.8(ซ้าย)) แสดงการพบอักขระ “a” ตัวขวาสุดของสายอักขระย่อยในคอลัมน์ซ้ายสุด 6 บรรทัด ซึ่งตัวอักขระก่อนหน้า “a” ใน 6 บรรทัดนี้มี 4 บรรทัดเป็น “n”, “c”, “d”, “y” ซึ่งไม่ตรงกับอักขระ “p” อย่างไรก็ตามถ้า d = 1 หมายความว่ายังเดินย้อนไปต่อได้ โดยจะเก็บไว้ทั้ง 6 บรรทัด และอักขระในคอลัมน์ขวาสุด 6 บรรทัดนี้จะถูกนำมาตรวจดูว่าอยู่บรรทัดไหนในคอลัมน์ซ้ายสุดของ BWT กลาง (รูปที่ 3.8(กลาง)) และตรวจต่อไปที่คอลัมน์ขวาสุดของ BWT กลางเพื่อกลายเป็นคอลัมน์ซ้ายสุดของ BWT ทางขวา (รูปที่ 3.8(ขวา)) ซึ่งพบว่ามีทั้งสิ้น 5 ตำแหน่ง คือ “ada”, “ana”, “aya” และอีกสอง “apa” ในสายอักขระหลักที่เหมือนกับสายอักขระย่อยแบบประมาณคือมีอักขระตัวกลางที่แตกต่างกัน 1 อักขระ

ในทางปฏิบัติจะไม่อนุญาตให้เกิดอักขระที่ไม่ตรงกับสายอักขระหลักตั้งแต่ต้นๆของการอ่านจากขวามาซ้าย เนื่องจากจำนวนบรรทัดที่ต้องตรวจสอบจะเพิ่มขึ้นมาก แนวทางหนึ่งที่เป็นไปได้คือให้กำหนดเงื่อนไขเพิ่มเติมว่าซัฟฟิกซ์จำนวน x อักขระจะต้องเหมือนกับสายอักขระหลักก่อนที่จะเดินย้อนต่อ และอนุญาตอักขระที่ต่างกัน

หลังจากนั้น นอกจากนี้ขนาดของ d ยังมีผลต่อเวลาที่ใช้ในการหาเนื่องจากเพิ่มจำนวนการเดินย้อนมากขึ้นเพราะมีรูปแบบที่เป็นไปได้มากขึ้นมาก ในทางปฏิบัติค่า d มักจะไม่เกิน 3



รูปที่ 3.8 การใช้ Burrows-Wheeler Transform กับการหาสายอักขระย่อย “apa” ในสายอักขระหลักแบบประมาณ

บทส่งท้าย

วิธีการและอัลกอริทึมที่ใช้ในการเทียบ (align หรือ map) หรือค้นหาชุดของสายอักขระย่อย เช่น รีดหรือดีเอ็นเอสายสั้นจำนวนมากในสายอักขระหลัก เช่น จีโนมอ้างอิงของมนุษย์ เป็นจุดเริ่มต้นของการหาการแปรผันของดีเอ็นเอของบุคคลหนึ่งเทียบกับจีโนมอ้างอิง เทียบกันระหว่างจีโนมของบุคคลในครอบครัว หรือเทียบกันระหว่างจีโนมของกลุ่มประชากร ซึ่งการแปรผันมีได้หลากหลายรูปแบบเช่น การแปรผันในลำดับเบสเดี่ยวที่บริเวณหนึ่งๆ ที่เรียกว่าเอสเอ็นวี หรือการแปรผันเชิงโครงสร้าง เช่น มีชุดของรีพีทที่แตกต่างกันระหว่างจีโนม โดยวิธีการค้นหาชุดของสายอักขระย่อยในสายอักขระหลักแบบประมาณเป็นตัวอย่างของวิธีการหาเอสเอ็นวี นอกจากการเทียบดีเอ็นเอสายสั้นจำนวนมากกับจีโนมอ้างอิงตามที่อธิบายในบทเรียนนี้แล้ว การค้นหาสายอักขระย่อยในสายอักขระหลักยังสามารถนำไปประยุกต์ใช้ในการตอบโจทย์ทางชีวการแพทย์อื่นๆ เช่น

- วิเคราะห์การแสดงออกของยีนในระดับทรานสคริปโทม (transcriptome) โดยการเทียบสายอาร์เอ็นเอในรูปแบบซีดีเอ็นเอจำนวนมาก (ที่ได้จากเทคโนโลยีเอ็นจีเอส (NGS) กลุ่ม RNA-seq อ่านว่าอาร์เอ็นเอซีค) กับจีโนมอ้างอิง เพื่อใช้เป็นข้อมูลในการวิเคราะห์ปริมาณการแสดงออกของยีน (gene expression) ทั้งจีโนม ในเงื่อนไขจำเพาะต่างๆ เช่น การใช้ยาที่แตกต่าง การแสดงออกของยีนตามเวลาที่เพิ่มขึ้น
- วิเคราะห์การแปรผันในสายดีเอ็นเอเทียบกับจีโนมอ้างอิงโดยเน้นเฉพาะบริเวณที่เป็นเอกซอนทั้งหมดของจีโนมซึ่งเป็นผลของการทำดับเบิลยูอีเอส (WES) หรือ whole exome sequencing แทนที่จะเป็นดับเบิล

ยูจีเอส (WGS) หรือ whole genome sequencing ทั้งนี้เพื่อมุ่งเป้าการวินิจฉัยการแปรผันของดีเอ็นเอเฉพาะบริเวณที่เป็นยีนที่สามารถแปลรหัสต่อไปเป็นโปรตีน ซึ่งส่วนใหญ่รู้ฟังก์ชันและหรือความสัม พันธ์กับการเกิดโรค เหมาะกับการวินิจฉัยทางคลินิก รวมทั้งจำนวนรีดที่ได้ทั้งหมดมีจำนวนน้อยกว่ารีดที่ได้จากการหาลำดับเบสทั้งจีโนมมาก

- การประยุกต์ใช้ซัฟฟิกซ์ทรีในการเทียบ ลำดับซ้ำเรียงต่อเนื่องแบบสั้น (short tandem repeat) หรือ เอสทีอาร์ (STR) กับจีโนมอ้างอิงในงานนิติเวชศาสตร์ ที่มีการใช้เทคโนโลยีเอ็นจีเอส ในการหาลำดับเบสในบริเวณที่เป็นเอสทีอาร์ โดยจะได้ข้อมูลในรายละเอียดเพิ่มขึ้นในการพิสูจน์อัตลักษณ์และเปรียบเทียบหาความสัมพันธ์ระหว่างบุคคล

นอกจากการประยุกต์ใช้อัลกอริทึมต่างๆ ตามที่อธิบายในบทเรียนนี้แล้ว ยังมีการวิจัยและพัฒนาเพิ่มเติมในเชิงของการหาสายอักขระย่อยในสายอักขระหลักที่มีความจำเพาะ เช่น การวิเคราะห์ข้อมูลในสาขา epigenomics และเมทาจีโนมิกส์ [64] โดยในกรณี epigenomics ต้องมีการพิจารณาเพิ่มเติมการเติมหมู่เมทิล (methylation) ให้กับนิวคลีโอไทด์ซึ่งหมายถึงอาจมีการพิจารณาเพิ่มอักขระจาก “A”, “T”, “C”, และ “G” ที่เป็นตัวแทนของกรดนิวคลีโอไทด์พื้นฐานให้มีอักขระที่แสดงถึงนิวคลีโอไทด์ที่มีการเติมหมู่เมทิล หรือการจัดการโอกาสที่จะเกิดลำดับเบสที่แตกต่างมากขึ้น ในขณะที่งานวิจัยในสาขาเมทาจีโนมิกส์ (metagenomics) หรือบาง ครั้งเรียกว่า environmental genomics หรือ community genomics [65, 66] จะทำการหาลำดับเบสจากตัวอย่างที่เก็บมาจากสิ่งแวดล้อม เช่น จากธารน้ำร้อน จากบ่อย่อยไขมันของโรงงานอุตสาหกรรม จากลำไส้กุ้ง จากลำไส้ ผิวหนัง หรือช่องปากมนุษย์ โดยดีเอ็นเอที่หาลำดับเบสมาได้นั้นจะประกอบด้วยรหัสพันธุกรรมของเชื้อต่างๆ หลายชนิดอยู่รวมกัน (ซึ่งมักไม่สามารถทำการเพาะเลี้ยงเชื้อเหล่านี้ในห้องทดลองได้) โดยความคาดหวังหลักจากการวิจัยคือสามารถระบุได้ว่าตัวอย่างที่เก็บมานั้นประกอบด้วยเชื้อชนิดใดบ้าง ปริมาณเท่าใด และหรืออยู่ในส่วนไหนของวิวัฒนาการชาติพันธุ์ (phylogeny) ดังนั้นการออกแบบและพัฒนาวิธีการเทียบดีเอ็นเอกับจีโนมอ้างอิงในงานเชิงเมทาจีโนมิกส์ต้องคำนึงถึงชุดของจีโนมอ้างอิงที่ปัจจุบันมีจีโนมของเชื้อต่างๆ อยู่มากมาย ซึ่งอาจนำไปสู่ความคลุมเครือในการตัดสินว่ารหัสพันธุกรรมเหล่านั้นเป็นของเชื้อใดบ้าง โดยเฉพาะถ้ามีบริเวณที่มีลำดับเบสที่คล้ายคลึงกันหรือเหมือนกันระหว่างจีโนมของเชื้อมากกว่าหนึ่งชนิด นอกจากนี้เรนเนิร์ทและคณะ [64] ได้กล่าวถึงการนำเสนอวิธีการประกอบร่างจีโนมใหม่ (*de novo* assembly) หรือการประกอบร่างจีโนมแบบวิธีการผสมที่ใช้การเทียบดีเอ็นสายสั้นกับจีโนมอ้างอิงก่อน และดีเอ็นสายสั้นที่ตกอยู่ในบริเวณเดียวกันทั้งหมดที่คาบเกี่ยวกันจะถูกนำไปสร้างกราฟแสดงความคาบเกี่ยว (overlap graph) เพื่อทำการประกอบร่างจีโนมเฉพาะบริเวณ ให้มีความถูกต้องมากขึ้นต่อไป ในเชิงของปัญหาที่มีอยู่ คนัท เรนเนิร์ท (Knut Reinert) และคณะ [64] ได้อธิบายถึงความซับซ้อนที่เกิดจากรีพีทในการเทียบสายดีเอ็นเอกับจีโนมอ้างอิง โดยจีโนมมนุษย์มีรีพีทเป็นส่วนประกอบประมาณ 50% [2] ในขณะที่จีโนมข้าวโพดมีรีพีทเป็นส่วนประกอบมากกว่า 80% [67] ซึ่งเป็นการยากในการระบุที่มาของ

สายดีเอ็นเอที่อ่านได้ว่าเป็นของรีพีทไหนในกลุ่มของรีพีทที่เป็นประเภทเดียวกัน โดยทั่วไปถ้าผลของการเทียบดีเอ็นเอสายสั้นมีความเหมือนกับหลายบริเวณ โปรแกรมที่ทำหน้าที่เทียบนี้จะรายงานผลเป็นความมั่นใจต่ำ ในทางตรงกันข้าม ถ้าบริเวณในจีโนมอ้างอิงที่เหมือนกับดีเอ็นเอสายสั้นมีบริเวณเดียวโปรแกรมตัวเทียบจะให้ค่าความมั่นใจสูง นอกจากรีพีทแล้วอีกประเด็นที่ถูกกล่าวไว้คือโดยพื้นฐานจีโนมมนุษย์แต่ละคนมีความเหมือนกันประมาณ 99.8% ซึ่งหมายความว่าจะมีเบสที่แตกต่างกันโดยปกติอยู่แล้วและลำดับเบสที่แตกต่างกันนี้มีการกระจายตัวที่ไม่สม่ำเสมอในลักษณะการแปรผันโดยรวม [68] และการแปรผันแบบอินเดลโดยเฉพาะ [69] ซึ่งอาจมีผลต่อการแปลผลการเทียบสายดีเอ็นเอกับจีโนมอ้างอิง เนื่องจากมีบางบริเวณในจีโนมอ้างอิงที่มีจำนวนรีดมาตกน้อยหรือไม่มีเลย ซึ่งอาจเป็นผลของการเกิดการขาดหาย (deletion) ไปของบริเวณในจีโนมที่อ่านมาได้ หรือในบริเวณนั้นของจีโนมมนุษย์แต่ละคนมีความแปรผันมากจนไม่มีดีเอ็นสายสั้นจากจีโนมมนุษย์คนไหนที่เหมือนกับจีโนมอ้างอิงมากพอ โดยแนวทางในการแก้ปัญหานี้ประกอบด้วยการสร้างจีโนมอ้างอิงให้กับกลุ่มประชากรของตัวเอง ในปัจจุบันมีแนวทางนี้เกิดขึ้นมากเนื่องจากราคาในการหาลำดับเบสจีโนมถูกลงมาก ในขณะที่คุณภาพของรหัสพันธุกรรมที่ได้ดีขึ้นมาก เช่น จีโนมอ้างอิงของคนเกาหลีที่ตีพิมพ์ในนิตยสารเนเจอร์ปี ค.ศ. 2016 [70] หรือการสร้างจีโนมอ้างอิงจากบุคคลภายในครอบครัว [71] เป็นต้น อีกแนวทางหนึ่งคือการเปลี่ยนลำดับเบสในจีโนมอ้างอิงในตำแหน่งต่างๆ ที่พบว่ามีการแปรผันสูง เช่น ตำแหน่งที่อาจพบเบส “T” หรือ “C” ก็สามารถเข้ารหัสเป็นตัวอักษรใหม่เช่น “Y” ตามมาตรฐาน International Union of Pure and Applied Chemistry (IUPAC) ซึ่งใช้ระบุว่าอาจเป็นเบส “T” หรือ “C” ก็ได้ ในกรณีนี้อัลกอริทึมหรือโปรแกรมต้องการไฟล์ข้อมูลความหมายของรหัสเพิ่มเติมเหล่านี้ นอกจาก นี้ อีกแนวทางหนึ่งที่น่าสนใจคือการนำกราฟมาแสดงองค์ประกอบของลำดับเบสในจีโนมแทนการใช้สายอักขระ เนื่องจากกราฟเป็นโครงสร้างข้อมูลที่สามารถแสดงการเกิดการแปรผันได้อย่างเป็นธรรมชาติ แต่ละโหนดสามารถมีเส้นเชื่อมในลักษณะทางแยกและสามารถกลับมารวมกันได้ในลำดับเบสถัดๆ ไป เกิดเป็นบับเบิ้ลในเส้นทางเดินภายในกราฟ โดยงานวิจัยมีทั้งแนวทางเน้นการสร้างกราฟที่ใช้หน่วยความจำอย่างมีประสิทธิภาพ สามารถสืบค้นข้อมูลลำดับเบสในตำแหน่งต่างๆ ได้อย่างรวดเร็ว [72, 73] และมีการพัฒนาในรูปแบบเครื่องมือทางชีว สารสนเทศที่ช่วยในการสร้างกราฟจากลำดับเบสจีโนม [74] และเครื่องมือที่สามารถสร้างกราฟและนำรีดมาเทียบกับกราฟได้ [75] โครงการ vg (https://github.com/vgteam/vg) พัฒนาเครื่องมือในการสร้าง variant graph จากไฟล์วีซีเอฟ (VCF: variant call format) และสามารถนำสายอักขระของรีดมาเทียบกับกราฟที่สร้างขึ้นได้ บริษัทเอกชนอย่าง SevenBridges (https://www.sevenbridges.com/graph/) นำกราฟมาเป็นโครงสร้างข้อมูลพื้นฐานในการแสดงจีโนมและหาการแปรผันประเภทต่างๆ อเล็กซานเดอร์ ดิลธีย์ (Alexander Dilthey) และคณะ [76] นำเสนอจีโนมอ้างอิงในรูปแบบกราฟที่สร้างจากกลุ่มประชากรโดยนำมาประยุกต์ใช้กับบริเวณ MHC (major histocom-patibility complex) ในโครโมโซมที่ 6 รวมทั้งนำข้อมูลอื่นๆ เข้ามาร่วมวิเคราะห์ เช่น ข้อมูลลำดับเบสที่ทราบแน่ชัดของแอลลีลต่างๆ ของ HLA (human leukocyte antigen) และข้อมูล 87,640 สนิปจากโครงการจีโนมมนุษย์ 1000 จีโนม โดยได้แสดงให้เห็นว่าการใช้จีโนมกราฟช่วยเพิ่มความถูกต้องในการระบุบริเวณจำเพาะบนจีโนมได้

เรนเนิร์ทและคณะ [64] คาดการณ์ว่าการใช้ฮาร์ดแวร์ เช่น หน่วยประมวลผลกราฟิก (graphic processing unit) หรือจีพียู (GPU) จะเป็นแนวทางสำคัญในการเพิ่มประสิทธิภาพการเทียบชุดของสายอักขระย่อยกับสายอักขระหลักแบบประมาณ สำหรับแนวทางในเชิงข้อมูล คือการเพิ่มความถูกต้องของการวิเคราะห์ข้อมูลในบริเวณที่เป็นรีพีท

ตัวอย่างโปรแกรมที่มีการใช้งานอย่างแพร่หลาย

โปรแกรมที่มีการใช้งานอย่างแพร่หลายโดยใช้ Burrows-Wheeler Transform (BWT) และซัฟฟิกซ์อาร์เรย์เป็นฐาน เช่น โปรแกรม BWA [77, 78], Bowtie [79], Bowtie2 [80] ทั้งนี้ CUSHAW [81] ใช้แนวทางเดียวกันแต่พัฒนาให้เป็นการทำงานแบบขนานโดยใช้คูดา (CUDA; compute unified device architecture) ในขณะที่ SOAP2 [82] และ SOAP3 [83] ปรับแต่ง BWT ให้เป็นสองทิศทางทั้งสายบวกและลบซึ่งทำให้สามารถค้นหาดีเอ็นเอในจีโนมอ้างอิงทั้งสองทิศทางได้พร้อมกัน รายละเอียดเพิ่มเติมเกี่ยวกับการเทียบดีเอ็นสายสั้นกับจีโนมอ้างอิงและแอปพลิเคชันที่เกี่ยวข้องสามารถศึกษาได้จากบทปริทัศน์ [21, 64, 84, 85]

แบบฝึกหัดบทที่ 3

เขียนโปรแกรมเพื่อแก้ปัญหาการเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิงโดยใช้โจทย์ที่โรซาลินด์ต่อไปนี้

1) Construct the Suffix Array of a String (http://rosalind.info/problems/ba9g/)
2) Construct Burrows-Wheeler Transform of a String (http://rosalind.info/problems/ba9i/)
3) Reconstruct a string from its Burrows-Wheeler Transform (http://rosalind.info/problems/ba9j/)
4) Implement BWMatching (http://rosalind.info/problems/ba9l/)

ภาคผนวกบทที่ 3

แอลลีล (allele)

แอลลีล (allele) คือรูปแบบที่เป็นไปได้ที่แตกต่างกันของยีนเดียวกัน บางยีนอาจมีหลายแอลลีล มนุษย์เป็นตัวอย่างสิ่งมีชีวิตที่เรียกว่าดิพลอยด์ (diploid) ซึ่งแต่ละยีนจะมีสองแอลลีล รับมาจากพ่อและแม่อย่างละหนึ่งแอลลีล แต่ละคู่ของแอลลีลนี้เป็นตัวแทนจีโนไทป์ (genotype) ของยีนนั้นๆ โดยจีโนไทป์จะเป็นแบบฮอโมไซกัส (homozygous) ถ้าทั้งสองแอลลีลเหมือนกัน และเป็นเฮเทอโรไซกัส (heterozygous) ถ้าสองแอลลีลต่างกัน แอลลีลเหล่านี้ส่งผลต่อลักษณะที่ปรากฏ (phenotype) นอกจากนี้บางแอลลีลอาจเป็นแอลลีลเด่น (dominant allele) หรือแอลลีลด้อย

(recessive) ถ้าที่โลคัสหนึ่งๆ เป็นเฮเทอโรไซกัส โดยมีแอลลีลเด่นและแอลลีลด้อยอย่างละหนึ่งแอลลีล ลักษณะที่แสดงออกจะเป็นแอลลีลเด่น
ที่มา https://www.nature.com/scitable/definition/allele-48

สนิป (SNP)

สนิป (SNP; single nucleotide polymorphism) หรือภาวะพหุสัณฐานนิวคลีโอไทด์เดี่ยว เป็นลักษณะการแปรผันของลำดับเบสเดี่ยวในจีโนมของประชากร โดยถ้ามีจำนวนประชากรมากกว่า 1% เกิดการแปรผันของนิวคลีโอไทด์เดี่ยวในตำแหน่งเดียวกัน การแปรผันในตำแหน่งนั้นๆ เรียกว่าสนิป ถ้าสนิปเกิดในยีน ยีนนั้นจะมีมากกว่าหนึ่งแอลลีล ซึ่งในกรณีนี้อาจมีผลต่อการแปลรหัสยีนนั้นๆ ไปเป็นโปรตีน ทั้งนี้สนิปพบทั้งในบริเวณที่เป็นยีนและไม่ใช่ยีน และสนิปบางตำแหน่งอาจส่งผลหรือเกี่ยวข้องกับการเกิดโรค
ที่มา https://www.nature.com/scitable/definition/snp-295/

รูปแบบไฟล์ที่เกี่ยวข้อง

SAM/BAM

รูปแบบไฟล์แซม (SAM: Sequence Alignment/Map) [86] ใช้ในการแสดงผลการเทียบหรือแมพรีดที่อ่านได้กับจีโนมอ้างอิง ข้อมูลในไฟล์เป็นข้อความ (text) ที่มีรูปแบบจำเพาะโดยแต่ละคอลัมน์คั่นด้วยแท็บ (tab) และประกอบด้วย 2 ส่วน คือ ส่วนหัว (ซึ่งอาจจะมีหรือไม่มีก็ได้ ถ้ามี ทุกบรรทัดของส่วนหัวจะขึ้นต้นด้วยอักขระ “@” และต้องมาก่อนส่วนที่เป็นผล) และส่วนที่เป็นผลของการเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิง โดยส่วนที่เป็นผลนี้ประกอบด้วย 11 คอลัมน์หลัก เพื่อแสดงข้อมูลที่สำคัญ อาทิ FLAG ซึ่งประกอบด้วยชุดของ bit wise flags โดยแต่ละบิตมีความหมายจำเพาะ เช่น บิต 0x4 หมายถึง segment นั้นไม่สามารถ map ได้กับจีโนมอ้างอิง บิต 0x10 หมายถึง SEQ นั้นเป็น reverse complemented, MAPQ เป็นค่าคุณภาพในการแมพรีด ซึ่งมีค่าเท่ากับ $-10\log_{10}\Pr\{mapping\ position\ is\ wrong\}$ โดยจะปัดเป็นค่าจำนวนเต็มที่ใกล้ที่สุด และถ้ามีค่าเป็น 255 หมายความว่าไม่มีค่าคุณภาพในการแมพ CIGAR string เป็นสายของอักขระโดยที่แต่ละอักขระมีความหมายจำเพาะแตกต่างกันไป เช่น “M” หมายถึงรีดแมพได้กับจีโนมอ้างอิง “I” หมายถึงเกิดการสอดแทรก (insertion) ส่วนนี้ในจีโนมอ้างอิง “D” หมายถึงมีการขาดหาย (deletion) จากจีโนมอ้างอิง “S” หมายถึงเกิด soft clipping เป็นต้น

ไฟล์รูปแบบแบม (BAM) เป็นไฟล์แสดงผลของการเทียบหรือแมพรีด เช่นเดียวกับไฟล์แซมแต่อยู่ในรูปแบบไบนารีและถูกบีบอัดในรูปแบบ BGZF ซึ่งสามารถบีบอัดได้ดีและยังอนุญาตให้สามารถเข้าถึงไฟล์แบมในแต่ละส่วนได้โดยตรงผ่านชุดของดัชนีการสืบค้นที่สร้างมาพร้อมกัน สำหรับรายละเอียดเพิ่มเติมของรูปแบบไฟล์

ทั้งแซมและแบมสามารถศึกษาได้จาก https://samtools.github.io/hts-specs/SAMv1.pdf ทั้งนี้การจัดการไฟล์ในรูปแบบแซมและแบมสามารถทำได้โดยใช้โปรแกรม SAMtools (http://samtools.sourceforge.net/)

บทที่ 4 การหาบริเวณที่ควบคุมการแสดงออกของยีน (Regulatory motif finding)

วัตถุประสงค์

- เพื่อให้นิสิตเห็นความเชื่อมโยงของการวัดการแสดงออกของยีนกับการหาบริเวณที่ควบคุมการแสดงออกของยีน
- เพื่อให้นิสิตคุ้นเคยกับตัวอย่างข้อมูลตั้งต้นที่เกี่ยวข้องและเข้าใจการทำงานของอัลกอริทึมพื้นฐานที่ใช้ในการหาบริเวณที่ควบคุมการแสดงออกของยีน
- เพื่อให้นิสิตเห็นตัวอย่างงานวิจัยและผลงานวิจัยรวมทั้งตัวอย่างโปรแกรมที่ใช้ในการหาบริเวณที่ควบคุมการแสดงออกของยีน
- เพื่อให้นิสิตเห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทายงานวิจัยอื่นๆ ที่เกี่ยวข้อง รวมทั้งเห็นแนวทางอื่นๆ ที่สามารถนำมาใช้ในการหาบริเวณที่ควบคุมการแสดงออกของยีนได้เช่นกัน

ผลลัพธ์ที่คาดหวัง

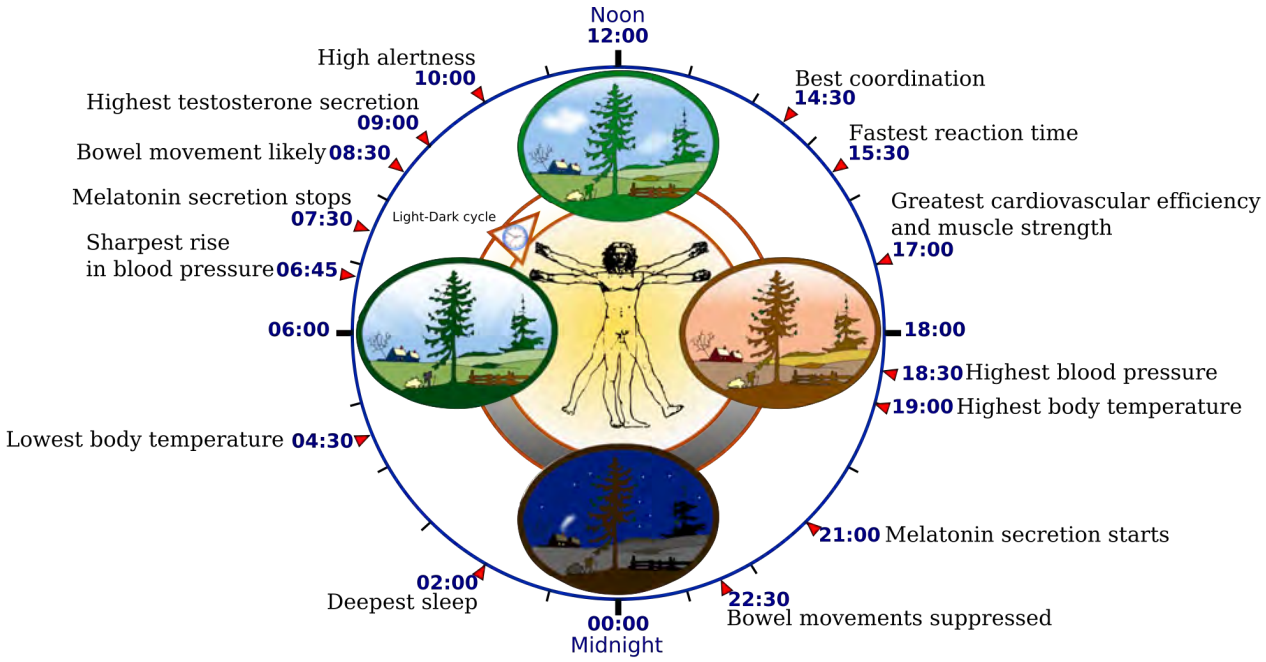
- นิสิตสามารถอธิบายความเชื่อมโยงของการวัดการแสดงออกของยีนกับการหาบริเวณที่ควบคุมการแสดงออกของยีน
- นิสิตเข้าใจคุณลักษณะของข้อมูลตั้งต้นที่ใช้ในการหาบริเวณที่ควบคุมการแสดงออกของยีน
- นิสิตสามารถอธิบายการทำงานของอัลกอริทึมหลักที่ใช้หาบริเวณที่ควบคุมการแสดงออกของยีน
- นิสิตสามารถเขียนโปรแกรมเพื่อหาบริเวณที่ควบคุมการแสดงออกของยีนได้
- นิสิตสามารถยกตัวอย่างการหาบริเวณที่ควบคุมการแสดงออกของยีนที่มีการใช้งานกันอย่างแพร่หลายได้
- นิสิตสามารถยกตัวอย่างความท้าทายที่ยังมีอยู่ในการหาบริเวณที่ควบคุมการแสดงออกของยีนและสามารถนำเสนอแนวทางในการพัฒนาวิธีการแก้ปัญหาเหล่านี้ รวมทั้งสามารถประยุกต์องค์ความรู้จากบทเรียนเพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้
- นิสิตสามารถยกตัวอย่างแนวทางอื่นๆ ในการแก้ปัญหาการหาบริเวณที่ควบคุมการแสดงออกของยีนได้

เนื้อหาโดยสรุป

การวัดการแสดงออกของยีนและความเชื่อมโยงของการวัดการแสดงออกกับการหาบริเวณที่ควบคุมการแสดงออก โดยมีสมมติฐานว่าชุดของยีนที่มีรูปแบบของการแสดงออกแบบเดียวกันจะมีบริเวณในโพรโมเตอร์ (promoter) ที่มีรูปแบบของลำดับเบสร่วมกันเรียกว่าโมทิฟควบคุม (regulatory motif) ซึ่งเป็นบริเวณที่ตัวควบคุมการแสดงออก เช่น แฟกเตอร์ถอดรหัส (transcription factor: TF) มาจับ รูปแบบของข้อมูลเข้า โจทย์ทางชีวสารสนเทศ อัลกอริทึมพื้นฐานในการหาโมทิฟควบคุม เช่น โพรไฟล์เมทริกซ์ (profile matrix) คอนเซ็นซัสสตริง (consensus string) ปัญหามีเดียนสตริง (median string problem) การค้นหาโมทิฟแบบละโมบ (greedy motif search) การค้นหาโมทิฟแบบสุ่ม (randomized motif search) และการสุ่มเลือกแบบกิบส์ (Gibbs sampling) การแสดงผลโมทิฟผ่าน sequence logo และโปรแกรมที่มีการใช้งานกันอย่างแพร่หลาย โจทย์อื่นๆ ที่เกี่ยวข้อง รวมทั้งแนวทางการหาโมทิฟโดยวิธีการเรียนรู้ของเครื่อง

บทที่ 4 การหาบริเวณที่ควบคุมการแสดงออกของยีน (Regulatory motif finding)

การดำเนินชีวิตในแต่ละวันของสิ่งมีชีวิตทั้งสัตว์และพืชถูกควบคุมโดยนาฬิกาเวลาภายในหรือนาฬิกาชีวิตซึ่งเรียกว่านาฬิกาเซอร์คาเดียน (circadian clock) การเกิดอาการเจ็ทแลก (jet lag) ตอนเดินทางข้ามทวีปเป็นตัวอย่างที่แสดงว่านาฬิกาเซอร์คาเดียนไม่เคยหยุดทำงาน รูปที่ 4.1 แสดงนาฬิกาเซอร์คาเดียนของมนุษย์ อีกตัวอย่างที่ยืนยันการทำงานของนาฬิกาเซอร์คาเดียนคือการทดลองกักบริเวณหนูแรทและมนุษย์ที่เป็นอาสาสมัครไว้ในที่หลบภัยที่มีแต่ความมืดตลอดเวลา พบว่ากระบวนการทำงานพื้นฐานยังเป็นรอบปกติประมาณ 24 ชั่วโมงของการดำเนินชีวิตในแต่ละวัน อย่างไรก็ตามนาฬิกาเซอร์คาเดียนสามารถทำงานผิดพลาดได้ซึ่งในกรณีนี้จะทำให้เกิดกลุ่มอาการนอนหลับผิดเวลา (delayed sleep-phase syndrome: DSPS)



รูปที่ 4.1 นาฬิกาเซอร์คาเดียนในมนุษย์

(*ที่มา:* YassineMrabet, Public domain, via Wikimedia Commons. 2007. *Overview of biological circadian clock in humans.* [*เข้าถึงออนไลน์เมื่อวันที่* 11 *ก.ค. พ.ศ.* 2564])

จากผลการทดลองและอาการเจ็ทแลกที่เกิดขึ้นข้างต้นคำถาม คือนาฬิกาเซอร์คาเดียนมีผลต่อกระบวนการของสิ่งมีชีวิตในระดับอณูชีววิทยาหรือชีวโมเลกุลอย่างไร ยีนนาฬิกา (clock gene) มีการทำงานและเกี่ยวข้องกับการแสดงออกของยีนและโปรตีนในแต่ละช่วงเวลาของวันอย่างไร หรือสามารถอธิบายเหตุผลช่วงเวลาของการเกิด

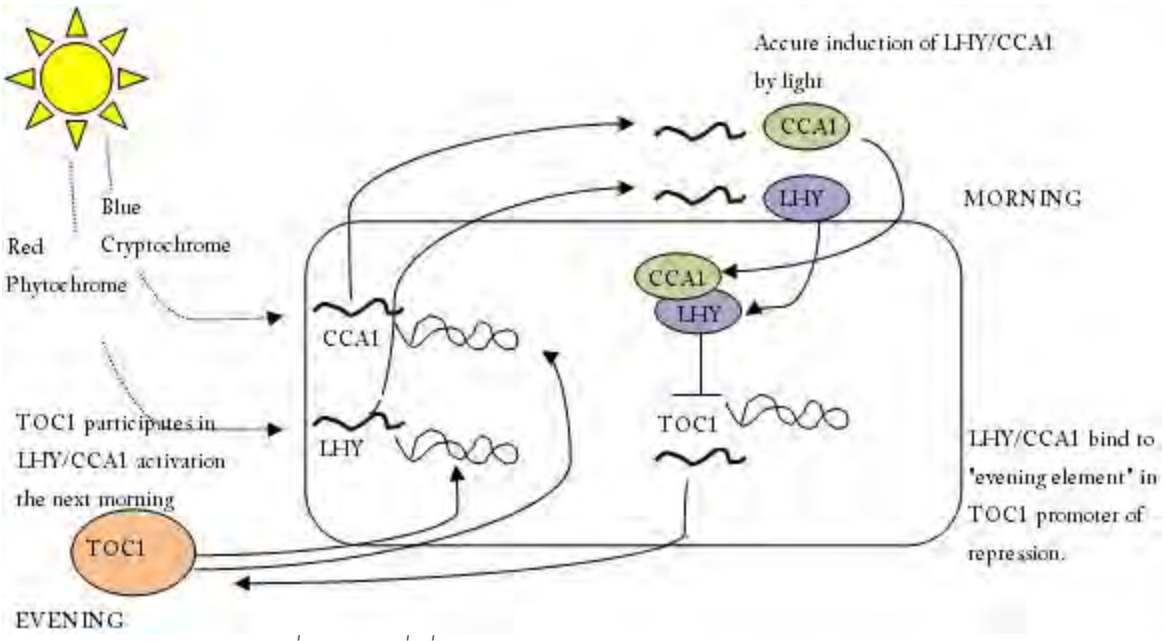
หัวใจล้มเหลวที่มักเกิดในช่วงเช้า การเกิดอาการภูมิแพ้มักเกิดตอนกลางคืน และสามารถระบุยีนที่เกี่ยวข้องกับการทำงานที่ผิดพลาดของนาฬิกาชีวิตและทำให้เกิดโรค DSPS ได้อย่างไร

ในช่วงต้น ค.ศ. 1970 รอน โคนอปกา (Ron Konopka) และ ซีมัวร์ เบนเซอร์ (Seymour Benzer) ได้ทำการศึกษาแมลงวันที่มีการกลายพันธุ์โดยมีรูปแบบของนาฬิกาเซอร์คาเดียนผิดปกติและค้นพบยีนเดี่ยวที่ให้ทำเกิดความผิดปกตินั้น [87] หลังจากนั้นอีก 20 ปี นักชีววิทยาได้ค้นพบยีนในลักษณะเดียวกันในสัตว์เลี้ยงลูกด้วยนม (mammalian) ซึ่งเป็นเพียงข้อมูลแรกของจิกซอว์ภาพใหญ่ ปัจจุบันมีการค้นพบยีนที่เกี่ยวข้องกับนาฬิกาเซอร์คาเดียนอีกหลายยีน ตัวอย่าง เช่น *TIMELESS*, *CLOCK* ที่ทำหน้าที่ควบคุมและประสานการทำงานของยีนอื่นๆอีกหลายร้อยยีนรวมทั้งมีความอนุรักษ์ของยีนเหล่านี้ในเชิงวิวัฒนาการ (evolutionary conservation) ระหว่างสปีชีส์

ในกรณีของพืชนาฬิกาเซอร์คาเดียนมีความสำคัญมากเพราะหมายถึงการอยู่รอด มีการศึกษาว่ามียีนใดบ้างที่มีการแสดงออกหรือทำงานในช่วงเช้าตอนพระอาทิตย์ขึ้นและช่วงเย็นหลังพระอาทิตย์ตกดิน นักชีววิทยามีการประมาณว่ามีมากกว่า 1000 ยีนมีความเกี่ยวข้องกับนาฬิกาชีวิตในพืช โดยยีนเหล่านี้อยู่ในกระบวนการสังเคราะห์แสง (photosynthesis) และการออกดอก (flowering) เป็นต้น คำถามคือยีนเหล่านี้ทราบได้อย่างไรว่าเป็นช่วงไหนของวันและควรเป็นเวลาที่ยีนนั้นๆ ควรแสดงออกและทำงาน ในการศึกษาทดลอง พบว่าเซลล์ของพืชทุกๆ เซลล์เป็นอิสระต่อกันในการตรวจสอบช่วงเวลาของวันและมียีนหลัก 3 ยีน คือ *LHY* (*LATE ELONGATED HYPOCOTYL*), *CCA1* (*CIRCADIAN CLOCK ASSOCIATED 1*) และ *TOC1* (*TIMING OF CAB EXPRESSION 1*) โดยแฟกเตอร์ถอดรหัส TOC1 ควบคุมการแสดงออกของยีน *LHY* และ *CCA1* ในขณะที่แฟกเตอร์ถอดรหัส LHY และ CCA1 สามารถยับยั้งการแสดงออกของยีน *TOC1* ได้ เกิดเป็นลูปเรียกว่า negative feedback loop โดยในตอนเช้าแสงแดดจะกระตุ้นการแสดงออกของยีน *LHY* และ *CCA1* ซึ่งไปยับยั้งการแสดงออกของยีน *TOC1* เมื่อตกเย็นไม่มีแสงแดดการแสดงออกของยีน *LHY* และ *CCA1* ลดลงทำให้การแสดงออกของยีน *TOC1* มีมากขึ้นและมีมากที่สุดในช่วงกลางคืน โดยมีหน้าที่เพิ่มการแสดงออกของยีน *LHY* และ *CCA1* ซึ่งมีเส้นทางย้อนกลับในการควบคุมการแสดงออกของยีน *TOC1* อีกที (รูปที่ 4.2) โปรตีนของทั้งสามยีนคือ *LHY*, *CCA1*, และ *TOC1* เป็นแฟกเตอร์ถอดรหัส (transcription factor: TF) มีหน้าที่ควบคุมการทำงานของยีนอื่นๆ โดยจะไปจับกับตำแหน่งจำเพาะในยีนเป้าหมาย ตำแหน่งนี้เรียกว่าโมทิฟควบคุม (regulatory motif) หรือเรียกสั้นๆ ว่าโมทิฟ หรือบริเวณจับของแฟกเตอร์ถอดรหัส (transcription factor binding site: TFBS) ซึ่งโดยทั่วไปจะอยู่บริเวณส่วนหน้า (upstream region) ของยีน โดยครอบคลุมประมาณ 600-1,000 เบส รวมบริเวณที่เป็นโพรโมเตอร์ โดยยีนเป้าหมายหลายยีนที่มีแฟกเตอร์ถอดรหัส CCA1 มาจับมีโมทิฟ **AAAAAATCT** ร่วมกัน

การหาโมทิฟข้างต้นมีความซับซ้อนเพิ่มขึ้นในกรณีที่ยีนเป้าหมายแต่ละยีนของแฟกเตอร์ถอดรหัสเดียวกันมีความต่างกันในบางเบสของบริเวณจับ ตัวอย่างเช่น CCA1 อาจจับได้กับโมทิฟ **AAGAACTCT** นอกจากนี้ถ้าไม่ทราบมาก่อนว่าโมทิฟของแฟกเตอร์ถอดรหัสหนึ่งๆ มีรูปแบบอย่างไร จะหาโมทิฟในชุดของสายดีเอ็นเอส่วนหน้าของยีนเป้าหมายได้อย่างไร บทเรียนนี้อธิบายอัลกอริทึมพื้นฐานที่ใช้ในการหาโมทิฟ (motif finding) โดยพยายาม

หารูปแบบจำเพาะหรือข้อความสั้นๆ (โมทิฟ) ที่ซ่อนอยู่ และพบอยู่ร่วมกันในชุดของสายดีเอ็นเอส่วนหน้าของยีนเป้าหมาย



รูปที่ 4.2 ยีนที่เกี่ยวข้องกับนาฬิกาเซอร์คาเดียนในพืช

(*ที่มา:* Bjholm, Public domain, via Wikimedia Commons. 2010. *Overview of plant circadian with LHY/CCA1 and TOC1.* [*เข้าถึงออนไลน์เมื่อวันที่* 11 *ก.ค. พ.ศ.* 2564])

ความซับซ้อนของการหาโมทิฟ

การหา evening element

ในปี ค.ศ. 2000 สตีฟ เคย์ (Steve Kay) ใช้ดีเอ็นเออาร์เรย์ในการตรวจสอบการแสดงออกของยีนในพืชใบเลี้ยงคู่ *Arabidopsis thaliana* ในช่วงเวลาต่างๆ ของวัน จากนั้นเคย์ได้ทำการสกัดข้อมูลเฉพาะส่วนที่เป็นลำดับเบสดีเอ็นเอส่วนหน้าของประมาณ 500 ยีนที่มีการแสดงออกในลักษณะที่สัมพันธ์กับวงรอบเวลาเซอร์คาร์เดียน และทำการหารูปแบบลำดับเบสจำเพาะที่พบบ่อยเป็นพิเศษในลำดับเบสดีเอ็นเอส่วนหน้าของ 500 ยีนนี้ โดยเคย์พบว่าลำดับเบส **AAAATATCT** ถูกพบบ่อยเป็นพิเศษโดยพบทั้งสิ้น 46 ครั้ง

ฝึกหัด	จงแสดงการหาจำนวนการปรากฏของ 9-mer ในสายดีเอ็นที่สร้างขึ้นแบบสุ่มจำนวน 500 เส้น โดยแต่ละเส้นมีความยาว 1,000 นิวคลีโอไทด์

เคย์เรียกโมทิฟที่พบนี้ว่า evening element และได้ทำการทดลองเพิ่มเติมเพื่อยืนยันว่าโมทิฟนี้มีผลต่อการแสดงออกของยีนที่เกี่ยวข้องกับนาฬิกาเซอร์คาร์เดียนใน *Arabidopsis thaliana* โดยทำการเปลี่ยนลำดับเบสใน

บริเวณที่เป็น evening element ของหนึ่งยีนและวัดการแสดงออก ซึ่งพบว่าไม่มีการแสดงออกในลักษณะที่เกี่ยวข้องกับนาฬิกาเซอร์คาร์เดียนอีกต่อไป รูปที่ 4.3 แสดงโมทิฟของยีน *CCA1* โดยมีที่มาจากฐานข้อมูล JASPAR 2018

รูปที่ 4.3 โมทิฟของแฟกเตอร์ถอดรหัส CCA1

(*ที่มา*: http://jaspar.genereg.net/matrix/MA0972.1/)

การหา evening element ในยีนเป้าหมายของพืชไม่ซับซ้อน เนื่องจากรูปแบบของโมทิฟค่อนข้างมีความอนุรักษ์ (conserved) ระหว่างโมทิฟที่พบในลำดับเบสส่วนหน้าของแต่ละยีน อย่างไรก็ตามถ้าพิจารณาบริเวณของยีนเป้าหมายที่ถูกจับโดยแฟกเตอร์ถอดรหัส HOXA5 (Homeobox Protein Hox-A5) ซึ่งมีหน้าที่ควบคุมพัฒนาการของเซลล์ที่เกี่ยวข้องกับแนวแกนหน้าหลัง (anterior-posterior axis) พบว่าโมทิฟของ HOXA5 ขนาด 8-mer (รูปที่ 4.4) มีความอนุรักษ์ในแต่ละคอลัมน์มากน้อยแตกต่างกันไป

รูปที่ 4.4 โมทิฟของแฟกเตอร์ถอดรหัส HOXA5 ที่พบในยีนเป้าหมายโดยตัวอักษรใหญ่ในแต่ละคอลัมน์ระบุเบสที่พบถี่สุดในคอลัมน์

(*สร้างจากข้อมูลของ* http://jaspar.genereg.net/matrix/MA*0158.1/*)

เป้าหมายของบทนี้คือการแปลงตัวอย่างโจทย์ทางชีววิทยาข้างต้นให้อยู่ในรูปแบบที่สามารถแก้ปัญหาได้ด้วยคอมพิวเตอร์ พิจารณาสายดีเอ็นเอ 10 เส้นที่ถูกจำลองขึ้นต่อไปนี้ ถ้าแต่ละเส้นมีการแทรกลำดับเบสแบบ

เดียวกันคือ aaaaaaaaggggggg ขนาด 15-mer เข้าไปโดยการสุ่มตำแหน่ง สายดีเอ็นเอที่จำลองขึ้นนี้เปรียบได้กับลำดับเบสส่วนหน้าของ 10 ยีนที่มีการแสดงออกร่วมกัน และลำดับเบส 15-mer ที่แทรกเข้าไปเทียบได้กับโมทิฟของยีนเหล่านี้

```
1  atgaccgggatactgataaaaaaaaggggggcgtacacattagataaacgtatgaagtacgttagactcgggcccgccdcg
2  acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacagaaacttttccgaataaaaaaaggggggga
3  tgagtatccctgggatgacttaaaaaaaaggggggtgctctcccgatttttgaatatgtaggatcattcgccagggtccga
4  gctgagaattggatgaaaaaaaaggggggtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
5  tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaataaaaaaaaggggggcttatag
6  gtcaatcatgttcttgtgaatggatttaaaaaaaaggggggggaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
7  cggttttggcccttgttagaggcccccgtaaaaaaaaggggggggcaattatgagagagctaatctatcgcgtgcgtgttcat
8  aacttgagttaaaaaaaaggggggggctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
9  ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcataaaaaaaaggggggggaccgaaagggaag
10 ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttaaaaaaaaggggggga
```

หยุดคิด	จากดีเอ็นเอที่จำลองขึ้น 10 เส้นข้างต้น เราจะหาลำดับเบส 15-mer ที่สุ่มแทรกเข้าไปในแต่ละเส้นได้อย่างไร

การแก้ปัญหานี้ทำได้โดยนำสายดีเอ็นที่จำลองขึ้นทั้ง 10 เส้นมาต่อกันแล้วหาคำที่มีความถี่มากสุด ซึ่งได้ผลลัพธ์ดังต่อไปนี้ (ตัวอักษรใหญ่แสดง 15-mer ที่สุ่มแทรกเข้าไปในดีเอ็นเอแต่ละเส้น)

```
1  atgaccgggatactgatAAAAAAAAGGGGGGGcgtacacattagataaacgtatgaagtacgttagactcgggcccgccdcg
2  acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacagaaacttttccgaatAAAAAAAAGGGGGGGa
3  tgagtatccctgggatgacttAAAAAAAAGGGGGGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccga
4  gctgagaattggatgAAAAAAAAGGGGGGGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
5  tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAAAAAAAAGGGGGGGcttatag
6  gtcaatcatgttcttgtgaatggatttAAAAAAAAGGGGGGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
7  cggttttggcccttgttagaggcccccgtAAAAAAAAGGGGGGGcaattatgagagagctaatctatcgcgtgcgtgttcat
8  aacttgagttAAAAAAAAGGGGGGGctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
9  ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatAAAAAAAAGGGGGGGaccgaaagggaag
10 ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttAAAAAAAAGGGGGGGa
```

เนื่องจากดีเอ็นเอที่จำลองขึ้นมานี้แต่ละเบสถูกสร้างขึ้นแบบสุ่ม จึงมีโอกาสน้อยที่ 15-mer เดียวกัน จะปรากฏบ่อยกว่าปกติ อย่างไรก็ตามถ้าลำดับเบส 15-mer ที่แทรกเข้าไปนี้สามารถแตกต่างกันได้ 4 เบสดังตัวอย่างต่อไปนี้ วิธีการหาโมทิฟโดยหาความถี่ของคำจะไม่สามารถนำมาประยุกต์ใช้ได้

```
1  atgaccgggatactgatAgAAgAAAGGttGGGggcgtacacattagataaacgtatgaagtacgttagactcggcgccgccg
2  acccctatttttttgagcagatttagtgacctggaaaaaaaatttgagtacaaaacttttccgaatacAATtAAAAcGGcGGGa
3  tgagtatccctgggatgacttAAAAtAAtGGaGtGGtgctctcccgatttttgaatatgtaggatcattcgccagggtccga
4  gctgagaattggatgcAAAAAAAGGGattGtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
5  tccctttttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaatAtAAtAAAGGaaGGGcttatag
6  gtcaatcatgttcttgtgaatggatttAAcAAtAAGGGctGGgaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
7  cggttttggcccttgttagaggcccccgtAtAAAcAAGGaGGGccaattatgagagagctaatctatcgcgtgcgtgttcat
8  aacttgagttAAAAAAtAGGGaGccctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
9  ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcatActAAAAAGGaGcGGaccgaaagggaag
10 ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttActAAAAAGGaGcGGa
```

การหาโมทิฟโดยวิธีการค้นหาทุกรูปแบบ

ถ้ามีข้อมูลเข้าประกอบด้วยชุดของสายดีเอ็นเอ และ *(k,d)-motif* ที่ต้องการหา โดย k คือความยาวของโมทิฟ และ d คือจำนวนของเบสที่แตกต่างกันได้มากสุดระหว่างโมทิฟ ปัญหาการหาโมทิฟถูกนิยามดังต่อไปนี้

นิยามปัญหาที่ 4.1 ปัญหาการหาโมทิฟที่แทรกอยู่ในสายดีเอ็นเอ

ปัญหาการหาโมทิฟที่แทรกอยู่ในสายดีเอ็นเอ (Implanted Motif Problem) หาโมทิฟ (k,d) ทั้งหมดที่อยู่ในสายดีเอ็นเอโดยที่ k คือความยาวโมทิฟและ d คือจำนวนเบสที่แตกต่างกันได้ระหว่างโมทิฟ	
ข้อมูลเข้า	ชุดของสายดีเอ็นเอ *Dna* และจำนวนเต็ม k และ d
ผลลัพธ์	ชุดของโมทิฟ (k,d) ที่พบในสายดีเอ็นเอ

การหาโมทิฟโดยวิธีการค้นหาทุกรูปแบบ (brute force search หรือ exhaustive search) เป็นการแก้ปัญหาโดยการตรวจสอบคำตอบที่เป็นไปได้ทั้งหมดว่าคำตอบใดบ้างถูกต้อง โดยอัลกอริทึมในกลุ่มนี้ไม่มีการออกแบบมากนักและสามารถหาคำตอบที่ถูกต้องได้ แต่ปัญหาหลักคือเวลาที่ต้องใช้ในการตรวจสอบคำตอบโดยเฉพาะในกรณีที่มีจำนวนคำตอบที่เป็นไปได้มากมาย สำหรับการหาโมทิฟโดยวิธีการค้นหาทุกรูปแบบนี้เราสามารถสร้างคำตอบที่เป็นไปได้ทั้งหมดจาก k-mer ใดๆ ในชุดของสายดีเอ็นเอ ทำการตรวจสอบว่าแต่ละคำตอบที่เป็นไปได้นี้ปรากฏในดีเอ็นเอทุกสายหรือไม่ โดยสามารถต่างจาก k-mer ที่สุ่มเลือกมาไม่เกิน d เบส ถ้าใช่เก็บคำตอบที่เป็นไปได้คำตอบนี้เข้าเป็นสมาชิกของชุดโมทิฟ (k,d) ที่เป็นคำตอบสุดท้าย ตามตัวอย่างรหัสเทียมที่ 4.1 `Motif Enumeration()` ต่อไปนี้

รหัสเทียมที่ 4.1 MotifEnumeration

```
MotifEnumeration(Dna, k, d)
    Patterns <- เซ็ตว่าง
    for แต่ละลำดับเบส k-mer ของ Dna
        for แต่ละ PatternP ที่ต่างจาก k-mer อย่างมาก d เบส
            if PatternP ปรากฏอยู่ในชุดของสาย Dna
                เพิ่ม PatternP เข้าในเซ็ต Patterns
    ส่งกลับเซ็ต Patterns
```

จากรหัสเทียมข้างต้นจะเห็นว่าวิธีการนี้ใช้เวลาในการหาโมทิฟนานเป็นอย่างมากโดยเฉพาะเมื่อ k และ d มีค่ามาก เพื่อเป็นการลดเวลาในการประมวลผล ได้มีการเสนอวิธีการค้นหาโมทิฟโดยการหา k-mer สองสายที่มีความแตกต่างกันไม่เกิน d เบสจากคู่ของสายดีเอ็นเอ อย่างไรก็ตามวิธีการนี้ไม่สามารถแก้ปัญหาได้ ถ้าพิจารณาตัวอย่าง 15-mer สองสายคือ **AgAAgAAAGGttGGG** และ **cAAtAAAAcGGGGcG** ซึ่งแต่ละสายต่างจาก k-mer ที่ถูกต้อง **AAAAAAAAGGGGGGG** 4 เบส แต่ k-mer สองสายนี้เองต่างกันถึง 8 ตำแหน่งตามภาพข้างล่าง ซึ่งถ้า k-mer สองสายนี้เป็นตัวอย่างคำตอบสุดท้าย วิธีการหาโมทิฟ (k,d) โดยการพยายามหา k-mer ในดีเอ็นเอสองสายที่ต่างกันไม่เกิน d เบสจะไม่ได้ k-mer ทั้งสองสายข้างต้นเป็นคำตอบสุดท้าย

ฝึกหัด	จงเขียนโค้ดเพื่อสร้างสายดีเอ็นยาว 600 เบส จำนวน 10 เส้น โดยการสุ่มแต่ละเบสมาจาก “A”, “T”, “C” และ “G” โดยมีโอกาสที่จะเกิดแต่ละเบสเท่าๆกัน จากนั้นทำการสร้างโมทิฟที่เป็นคำตอบสุดท้ายขนาด 15-mer โดยแต่ละเบสสุ่มมาแบบเดียวกับการสร้างสายดีเอ็นเอข้างต้น และโมทิฟที่สร้างขึ้นนี้สามารถต่างจากโมทิฟ **AAAAAAAA GGGGGGG** ได้มากสุด 4 เบส จากนั้นสุ่มตำแหน่งในดีเอ็นเอเส้นที่หนึ่งและทำการแทรกโมทิฟที่สร้างขึ้นในตำแหน่งนั้น ทำการสร้างโมทิฟตามเงื่อนไขเดียวกันและทำการแทรกในดีเอ็นเอสายถัดไปจนครบ 10 เส้น จากนั้นใช้สายดีเอ็นเอที่มีการแทรกโมทิฟแล้วเป็นข้อมูลเข้า ลองทำการหาโมทิฟโดยการพยายามหา 15-mer ที่เหมือนกันที่สุดระหว่างดีเอ็นสองสายใดๆที่เป็นข้อมูลเข้า และรายงานผล

การให้คะแนนโมทิฟ

การใช้ชุดของโมทิฟเพื่อสร้างโพรไฟล์เมทริกซ์และสายอักขระเสียงข้างมาก

การอธิบายปัญหาการหาโมทิฟโดยใช้ตัวอย่างการแทรกโมทิฟเข้าไปในสายดีเอ็นเอข้างต้นมีข้อจำกัด เนื่องจากวิธีการหาโมทิฟข้างต้นมีสมมติฐานว่าสายดีเอ็นเอที่เป็นข้อมูลเข้าทั้งหมดจะต้องมีโมทิฟที่ถูกต้องแทรกอยู่ ซึ่งสมมติฐานนี้ *ไม่เป็นจริง* ในมุมมองของนักชีววิทยา ตัวอย่างเช่น การทดลองของสตีฟ เคย์ ที่ใช้ดีเอ็นเออาร์เรย์ในการตรวจสอบว่ามียีนใดบ้างใน *Arabidopsis thaliana* ที่มีการแสดงออกในช่วงเวลาต่างๆ ของวัน เคย์ไม่ได้คาดหวังว่าลำดับเบสส่วนหน้าของทุกยีนจะต้องมี evening element ทั้งนี้ด้วยเหตุผลจากการทดลองดีเอ็นเออาร์เรย์ที่มักมีสัญญาณรบกวนและยีนหลายยีนที่แสดงออกในรูปแบบเดียวกับกลุ่มยีนเป้าหมายอาจไม่เกี่ยวข้องกับการทำงานของนาฬิกาเซอร์คาร์เดียนก็ได้ ในกรณีของการหาชุดของยีนเป้าหมายของแฟกเตอร์ถอดรหัส HOXA5 ก็เช่นกัน

ด้วยเหตุผลข้างต้นวิธีการในการค้นหาโมทิฟได้ถูกนำเสนอในแนวทางที่สอดคล้องกับลักษณะข้อมูลการทดลองในห้องปฏิบัติการมากขึ้นโดยใช้การให้คะแนนโมทิฟหนึ่งๆ ที่สามารถถูกจับได้โดยแฟกเตอร์ถอดรหัส เทียบกับโมทิฟที่เป็นอุดมคติคือสามารถถูกจับได้โดยแฟกเตอร์ถอดรหัสได้ดีที่สุด อย่างไรก็ตามในทางปฏิบัติเรามักไม่ทราบโมทิฟที่เป็นอุดมคตินั้น ดังนั้นแนวทางที่เป็นไปได้คือพยายามเลือก k-mer จากแต่ละสายดีเอ็นเอและพยายามให้คะแนนโมทิฟเหล่านี้จากความเหมือนกับโมทิฟอื่นๆ ที่ถูกเลือกมาเช่นกัน

ในการกำหนดวิธีการให้คะแนน ข้อมูลที่เกี่ยวข้องประกอบด้วย t คือจำนวนสายดีเอ็นเอ โดยแต่ละสายยาว n เบส และทำการเลือก k-mer จากดีเอ็นเอแต่ละสาย เพื่อให้ได้ชุดของโมทิฟ *Motifs* ซึ่งถูกแสดงด้วย โมทิฟเมทริกซ์ (motif matrix) ขนาด $t \cdot k$ ดังแสดงในรูปที่ 4.5(ก) ซึ่งเป็นตัวอย่างโมทิฟเมทริกซ์ของตำแหน่งจับ (binding site) ของ HOXA5 จากรูปที่ 4.4 โดยในรูปที่ 4.5 นี้ตัวอักษรใหญ่ในแต่ละคอลัมน์แสดงเบสที่พบบ่อยสุดในตำแหน่งนั้นๆ ในตัวอย่างนี้คอลัมน์ที่ 1, 5, 6, และ 7 มีความอนุรักษ์ของเบสมากโดยในคอลัมน์ที่ 7 เป็นเบส “T” ทั้งหมด ในขณะที่คอลัมน์ที่ 2 และ 3 มีความอนุรักษ์ของเบสจำเพาะน้อยสุด เราสามารถสร้างโมทิฟเมทริกซ์ได้มากมายโดยการเลือกชุดของ k-mers จาก t ดีเอ็นเอ อย่างไรก็ตาม เป้าหมายคือการเลือก k-mers ที่ทำให้ได้โมทิฟเมทริกซ์ที่มีความอนุรักษ์มากสุด หรืออีกนัยหนึ่งคือมีตัวอักษรใหญ่มากที่สุดในทุกคอลัมน์ จากรูปที่ 4.5(ข) ฟังก์ชัน `SCORE(Motifs)` จะแสดงค่าผลรวมของตัวอักษรเล็กทั้งหมดที่ปรากฏอยู่ในโมทิฟเมทริกซ์ โดยค่าผลรวมนี้สามารถนำมาใช้วัดความอนุรักษ์ของโมทิฟเมทริกซ์โดยค่ายิ่งน้อยโมทิฟเมทริกซ์จะมีความอนุรักษ์มาก

ฝึกหัด	ค่าน้อยสุดที่เป็นไปได้ของ `SCORE(Motifs)` คือ 0 ซึ่งหมายความว่าเบสในแต่ละ คอลัมน์เหมือนกันทุกบรรทัด คำถามคือค่าที่มากที่สุดที่เป็นไปได้ของ `SCORE(Motifs)` เป็นเท่าใด ถ้าแสดงในรูปแบบของตัวแปร t และ k ข้างต้น

(ก) Motifs

	คอลัมน์ 1	2	3	4	5	6	7	8
1	C	A	C	a	A	A	T	g
2	C	A	C	T	A	A	T	g
3	C	g	t	a	A	A	T	c
4	C	t	t	T	t	g	T	T
5	C	g	C	T	A	A	T	g
6	C	A	C	T	A	A	T	T
7	C	A	t	a	A	A	T	T
8	C	t	g	T	A	A	T	T
9	C	A	t	a	A	A	T	T
10	C	t	C	T	A	A	T	T
11	C	A	C	T	A	A	T	g
12	a	A	g	a	A	A	T	g
13	C	t	g	a	A	A	T	g
14	a	g	C	T	t	A	T	T
15	C	g	g	T	A	A	T	T

(ข) SCORE(Motifs) 2 + 8 + 8 + 6 + 2 + 1 + 0 + 7 = 34

(ค) COUNT(Motifs)

A:	2	7	0	6	13	14	0	0
C:	13	0	7	0	0	0	0	1
G:	0	4	4	0	0	1	0	6
T:	0	4	4	9	2	0	15	8

(ง) PROFILE(Motifs)

A:	.13	.46	0	.4	.87	.97	0	0
C:	.87	0	.46	0	0	0	0	.07
G:	0	.27	.27	0	0	.03	0	.4
T:	0	.27	.27	.6	.13	0	1	.53

(จ) CONSENSUS(Motifs) C A C T A A T T

รูปที่ 4.5 (ก) โมทิฟเมทริกซ์ (ข) ผลของ `SCORE(Motifs)` *(ค) ผลของ* `COUNT(Motifs)` *(ง) โพรไฟล์เมทริกซ์ และ (จ) สายอักขระเสียงข้างมากของแฟกเตอร์ถอดรหัส HOXA5*

(*ที่มา*: Sequence Logo จาก http://jaspar.genereg.net/matrix/MA*0158.1/*)

เราสามารถสร้างเมทริกซ์ 4 x *k* ที่แสดงจำนวนครั้งที่แต่ละเบสปรากฏในแต่ละคอลัมน์ โดยแสดงผ่านฟังก์ชัน `COUNT(Motifs)` (รูปที่ 4.5 (ค)) นอกจากนี้ถ้าเรานำจำนวน t มาหารแต่ละค่าที่ปรากฏใน เมทริกซ์ 4 x *k* จะได้เมทริกซ์ใหม่เรียกว่า โพรไฟล์เมทริกซ์ `P = PROFILE(Motifs)` ซึ่งแต่ละค่าในโพรไฟล์เมทริกซ์แสดงค่าความถี่ของการเกิดเบสจำเพาะในแต่ละคอลัมน์ (รูปที่ 4.5(ง)) และท้ายสุดเราสามารถสร้างสายอักขระเสียงข้างมาก (consensus string) ผ่านฟังก์ชัน `CONSENSUS(Motifs)` (รูปที่ 4.5(จ)) จากเบสที่พบบ่อยสุดในแต่ละคอลัมน์ ถ้าเราสามารถเลือกชุด k-mers ที่ถูกต้องจากลำดับเบสส่วนหน้า ผลของสร้างสายอักขระเสียงข้าง

มาก CONSENSUS(*Motifs*)จะให้โมทิฟที่เป็นอุดมคติของลำดับเบสส่วนหน้าชุดนี้ ตัวอย่างจากรูปที่ 4.5(จ) สายอักขระเสียงข้างมากซึ่งเป็นตำแหน่งที่ HOXA5 มาจับคือ CACTAATT

การปรับปรุงการให้คะแนน

จากรูปที่ 4.5 คอลัมน์ที่ 1 และคอลัมน์ที่ 5 ของโมทิฟเมทริกซ์มี 2 คะแนนเท่ากันโดยคอลัมน์ที่ 1 มีเบส a เป็นเสียงข้างน้อยทั้ง 2 คะแนน ในขณะที่คอลัมน์ที่ 5 มีเบส t เป็นเสียงข้างน้อยทั้ง 2 คะแนน

หยุดคิด	สองคะแนนของสองคอลัมน์ที่เท่ากันนี้มีความหมายทางชีววิทยาเท่ากันหรือไม่

ในทางชีววิทยาบางตำแหน่งของโมทิฟที่แฟกเตอร์ถอดรหัสมาจับอาจเป็นนิวคลีโอไทด์ได้มากกว่า 1 แบบ ตัวอย่างเช่น โมทิฟ C6 zinc cluster factors ในยีสต์ (*Saccharomyces cerevisiae*) ที่ยาว 15-mer ประกอบด้วย 8 ตำแหน่งที่มีความอนุรักษ์มาก ในขณะที่อีก 7 ตำแหน่งมีความอนุรักษ์น้อยโดยแต่ละตำแหน่งมีเบสที่แตกต่างกันดังแสดงในรูปที่ 4.6

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
G	G	A/C	A	A	C/T	C/T	C/G	T/A	T/A	A	G	A	A/G	C

รูปที่ 4.6 โมทิฟ C6 zinc cluster factors ในยีสต์ (Saccharomyces cerevisiae) มีความอนุรักษ์มากในตำแหน่งที่ 1, 2, 4, 5, 11-13 และ 15 ในขณะที่เจ็ดตำแหน่งที่เหลือสามารถเป็นนิวคลีโอไทด์ได้สองประเภท

รูปที่ 4.7 แสดงสายอักขระเสียงข้างมากของ HOXA5 ข้างต้น โดยปรับให้ละเอียดมากขึ้นตามตัวอย่างการแสดงโมทิฟ CSRE ในยีสต์ โดยตำแหน่งที่ 1, 2, 3, 5, 6, และ 7 มีความอนุรักษ์มากในขณะที่ตำแหน่งที่ 4 และ 8 อาจเป็นนิวคลีโอไทด์ได้สองประเภทเท่าๆ กัน

1	2	3	4	5	6	7	8
C	A	C	T/A	A	A	T	T/G

รูปที่ 4.7 สายอักขระเสียงข้างมากแสดงโมทิฟของแฟกเตอร์ถอดรหัส HOXA5 ที่มีการเพิ่มข้อมูลนิวคลีโอไทด์ที่เป็นไปได้ในแต่ละตำแหน่ง

เอนโทรปีและโลโก้โมทิฟ

พิจารณาโพรไฟล์เมทริกซ์ในรูปที่ 4.5(ง) แต่ละคอลัมน์แสดงการแจกแจงความน่าจะเป็นของแต่ละนิวคลีโอไทด์ที่ตำแหน่งนั้น และผลรวมของทุกแถวจะต้องเป็น 1 ตัวอย่างเช่น ในคอลัมน์ที่สองหรือตำแหน่งที่สองของโมทิฟมีความน่าจะเป็นของนิวคลีโอไทด์ “A”, “C”, “G” และ “T” เท่ากับ 0.46, 0.0, 0.27 และ 0.27 ตามลำดับ

เอนโทรปี (entropy) ใช้วัดความไม่แน่นอนของการแจกแจงความน่าจะเป็น (p_1,..., p_N) โดยแสดงด้วยสูตรต่อไปนี้

$$H(p_1, \dots, p_n) = -\sum_{i=1}^{N} p_i \log_2(p_i)$$

ตัวอย่างเช่น ค่าเอนโทรปีของการแจกแจงความน่าจะเป็น (0.46, 0.0, 0.27, 0.27) ในคอลัมน์ที่สองมีค่าเท่ากับ

$$-(0.46 \log_2 0.46 + 0.0 \log_2 0.0 + 0.27 \log_2 0.27 + 0.27 \log_2 0.27) \approx 1.54$$

ในขณะที่คอลัมน์ที่มีความอนุรักษ์มากกว่าอย่างคอลัมน์ที่หนึ่ง ค่าเอนโทรปีของการแจกแจงความน่าจะเป็น (0.13, 0.87, 0.0, 0.0) มีค่าเท่ากับ

$$-(0.13 \log_2 0.13 + 0.87 \log_2 0.87 + 0.0 \log_2 0.0 + 0.0 \log_2 0.0) \approx 0.56$$

และคอลัมน์ที่มีความอนุรักษ์มากอย่างคอลัมน์ที่ 6 ค่าเอนโทรปีของการแจกแจงความน่าจะเป็น (0.97, 0.0, 0.03, 0.0) มีค่าเท่ากับ

$$-(0.97 \log_2 0.97 + 0.0 \log_2 0.0 + 0.03 \log_2 0.03 + 0.0 \log_2 0.0) \approx 0.19$$

ในเชิงเทคนิคค่า $\log_2 0$ ไม่มีการนิยามค่า อย่างไรก็ตามในการคำนวณค่าเอนโทรปี $0 \log_2 0$ ถูกกำหนดให้มีค่าเท่ากับ 0

หยุดคิด	ค่าเอนโทรปีที่มากและน้อยที่สุดเป็นเท่าใดถ้าการแจกแจงความน่าจะเป็นในแต่ละคอลัมน์มีค่าที่เป็นไปได้ทั้งหมดสี่แบบ

สำหรับคอลัมน์ที่มีความอนุรักษ์มากสุด เช่นคอลัมน์ที่ 7 มีค่าเอนโทรปีเท่ากับ 0 ซึ่งเป็นค่าเอนโทรปีน้อยสุดที่เป็นไปได้ ในทางกลับกันคอลัมน์ที่มีการปรากฏของนิวคลีโอไทด์ทั้งสี่ตัวเท่ากัน (นิวคลีโอไทด์ละ ¼) มีค่าเอนโทรปีเท่ากับ $-4 \cdot \frac{1}{4} \cdot \log_2 \frac{1}{4} = 2$ โดยทั่วไปคอลัมน์ที่มีความอนุรักษ์มากกว่าจะมีค่าเอนโทรปีน้อยกว่า ดังนั้นเราสามารถใช้ค่าเอนโทรปีในการปรับปรุงการให้คะแนนโมทิฟเมทริกซ์ โดยค่าเอนโทรปีของโมทิฟเมทริกซ์จะเท่ากับผลบวกของค่าเอนโทรปีในแต่ละคอลัมน์ การใช้เอนโทรปีเพื่อกำหนดคะแนนให้โมทิฟเมทริกซ์ถูกใช้มากกว่าการนับ `SCORE(Motifs)` ที่อธิบายไปก่อนหน้า อย่างไรก็ตามเพื่อให้การอธิบายหัวข้อถัดไปไม่ซับซ้อน เราจะยังคงใช้ `SCORE(Motifs)` แทนการใช้เอนโทรปี

ฝึกหัด	จงคำนวณค่าเอนโทรปีของโมทิฟเมทริกซ์ HOXA5 ที่แสดงในรูปที่ 4.5

sequence logo หรือ motif logo ที่แสดงในรูปที่ 4.5(จ) ข้างต้นเป็นตัวอย่างของการประยุกต์ใช้เอนโทรปี โดย sequence logo แสดงความอนุรักษ์ของนิวคลีโอไทด์จำเพาะในแต่ละตำแหน่งในรูปแบบของตัวอักษรที่กองซ้อนกันอยู่ ทั้งนี้ขนาดของตัวอักษรแต่ละตัวในกองซ้อนแต่ละคอลัมน์บ่งชี้ความถี่ของการปรากฏของนิวคลีโอไทด์นั้นๆ ในตำแหน่งนั้น และความสูงรวมของกองซ้อนในแต่ละคอลัมน์ขึ้นอยู่กับเนื้อหาข้อมูล (information content) ในคอลัมน์นั้นๆ ซึ่งคำนวณจาก $2 - H(p_1, \ldots, p_N)$ โดยค่าเอนโทรปียิ่งน้อยก็จะมีค่าเนื้อหาข้อมูลมาก คอลัมน์ที่มีกองซ้อนสูงมากจะมีความอนุรักษ์ของนิวคลีโอไทด์จำเพาะมาก

การหาโมทิฟโดยวิธีการหามีเดียนสตริง

หลังจากมีการกำหนดการให้คะแนนชุดของ k-mers แล้ว เราสามารถนิยามปัญหาการหาโมทิฟได้ดังต่อไปนี้

นิยามปัญหาที่ 4.2 ปัญหาการหาโมทิฟ

ปัญหาการหาโมทิฟ (Motif Finding Problem) รับข้อมูลเข้าเป็นชุดของสายอักขระ ให้หาหนึ่ง k-mer จากแต่ละสายอักขระ เพื่อมาสร้างโมทิฟเมทริกซ์ที่มีค่าคะแนนน้อยที่สุด	
ข้อมูลเข้า	ชุดของสายอักขระ *Dna* และค่าจำนวนเต็ม *k*
ผลลัพธ์	โมทิฟเมทริกซ์ (*Motifs*) ที่ประกอบด้วย k-mer จากแต่ละสายอักขระใน *Dna* โดยให้ค่า SCORE(*Motifs*)น้อยสุด

จากปัญหาข้างต้น ถ้าใช้วิธีการค้นหาทุกรูปแบบ (brute force) จะต้องพิจารณาชุดของ k-mers ทั้งหมดที่เป็นไปได้ และทดสอบว่า k-mers ชุดใดให้คะแนนโมทิฟน้อยสุด เนื่องจากสายอักขระแต่ละเส้นมี k-mer ที่เป็นไปได้ทั้งหมด *n*-*k*+1 สายและเนื่องจาก *Dna* มีทั้งหมด *t* เส้น จะมีชุดของ k-mers สำหรับสร้างโมทิฟเมทริกซ์ทั้งสิ้น $(n\text{-}k\text{+}1)^t$ ชุด และเมื่อได้แต่ละโมทิฟที่เป็นไปได้แล้วต้องนำมาคำนวณคะแนนโดยใช้ฟังก์ชัน SCORE(*Motifs*)ซึ่งต้องใช้ $k \cdot t$ ขั้นในการคำนวณ ถ้ามีสมมติฐานว่า *k* สั้นกว่า *n* มาก เวลาที่ใช้ในการหาโมทิฟโดยวิธีการค้นหาทุกรูปแบบจะเท่ากับ $O(n^t \cdot k \cdot t)$ ซึ่งช้ามาก

กำหนดแนวทางแก้ปัญหาใหม่อีกครั้ง

ในการค้นหาโมทิฟ เราสามารถเริ่มจากการหาชุดของ k-mers เพื่อสร้างเป็นโมทิฟเมทริกซ์ที่มีคะแนนน้อยสุดและหาโมทิฟที่เป็นคำตอบจากสายอักขระเสียงข้างมากของโมทิฟเมทริกซ์นั้นตามลักษณะการทำงานต่อไปนี้

Motifs -> CONSENSUS(*Motifs*)

เพื่อเป็นการลดเวลาในการหาโมทิฟโดยวิธีการทำทุกรูปแบบก่อนหน้า จึงมีคำถามว่าเราจะสามารถสร้างโมทิฟเมทริกซ์ย้อนกลับจากสายอักขระเสียงข้างมากได้หรือไม่ จากรูปที่ 4.8 ต่อไปนี้พบว่า SCORE(*Motifs*) มีค่าเท่ากับ 34 ตามที่คำนวณมาก่อนหน้าในรูปที่ 4.5 โดยนับผลรวมของตัวอักษรตัวเล็กในแต่ละคอลัมน์บวกกันและจะมีค่าเท่ากับผลรวมของตัวอักษรเล็กที่นับได้ในแต่ละแถว ตัวอักษรเล็กที่นับได้ในแต่ละแถวนี้แสดงความแตกต่างหรือระยะทางแฮมมิง (Hamming distance) d ของโมทิฟสายนั้นเทียบกับสายอักขระเสียงข้างมาก ตัวอย่างเช่น ในรูปที่ 4.8 d(**CACTAATT**, **CACaAATg**) = 2

	1	C	A	C	a	A	A	T	g	2
	2	C	A	C	T	A	A	T	g	1
	3	C	g	t	a	A	A	T	c	4
	4	C	t	t	T	t	g	T	T	4
	5	C	g	C	T	A	A	T	g	2
	6	C	A	C	T	A	A	T	T	0
	7	C	A	t	a	A	A	T	T	2
Motifs	8	C	t	g	T	A	A	T	T	2
	9	C	A	t	a	A	A	T	T	2
	10	C	t	C	T	A	A	T	T	1
	11	C	A	C	T	A	A	T	g	1
	12	a	A	g	a	A	A	T	g	4
	13	C	t	g	a	A	A	T	g	4
	14	a	g	C	T	t	A	T	T	3
	15	C	g	g	T	A	A	T	T	2
SCORE(Motifs)		2	+ 8	+ 8	+ 6	+ 2	+ 1	+ 0	+ 7	= 34
CONSENSUS(Motifs)		C	A	C	T	A	A	T	T	

รูปที่ 4.8 การคำนวณคะแนนโมทิฟเมทริกซ์ผ่านฟังก์ชัน SCORE(Motifs) ซึ่งผลบวกตัวอักษรเล็กตามคอลัมน์เท่ากับผลบวกของอักษรเล็กตามแถว

ถ้ากำหนดข้อมูลเข้าเป็น *Pattern* ซึ่งแสดงโมทิฟจากสายอักขระเสียงข้างมากและ $Motifs = \{Motif_1, \ldots, Motif_t\}$ *ซึ่งแสดงชุดของ k-mers ของโมทิฟเมทริกซ์* เรานิยามระยะทางแฮมมิงได้ดังสมการต่อไปนี้

$$d(Pattern, Motifs) = \sum_{i=1}^{t} HAMMINGDISTANCE(Pattern, Motif_i)$$

เมื่อย้อนกลับไปพิจารณา SCORE(*Motifs*)ข้างต้น จะพบว่า

$$SCORE(Motifs) = d(CONSENSUS(Motifs), Motifs)$$

สมการนี้ให้แนวคิดว่าแทนการหาชุดของ k-mers เพื่อสร้าง *Motifs* ให้ได้ SCORE(*Motifs*)น้อยสุด เราสามารถหาสายอักขระเสียงข้างมาก *Pattern* ที่ทำให้ค่า $d(Pattern, Motifs)$ มีค่าน้อยสุดจาก

Patterns และ *Motifs* ทั้งหมดที่เป็นไปได้จากชุดของสายดีเอ็นเอ ปัญหานี้เทียบเท่ากับปัญหาการหาโมทิฟในนิยามปัญหาที่ 4.2

ปัญหามีเดียนสตริง

การแก้ปัญหาการหาโมทิฟโดยใช้สายอักขระเสียงข้างมากข้างต้นมีความซับซ้อนเพิ่มขึ้นหรือไม่ เพราะนอกจากต้องพิจารณาชุดของ k-mers เพื่อสร้าง *Motifs* ทั้งหมดที่เป็นไปได้แล้วยังต้องพิจารณาสายอักขระเสียงข้างมากที่เป็นไปได้ทั้งหมดอีกด้วย อย่างไรก็ตามถ้าพิจารณาการแก้ปัญหานี้ในรายละเอียดจะพบว่าไม่จำเป็นต้องพิจารณาทุกชุดของ k-mers ที่เป็นไปได้เพื่อหา $d(Pattern, Motifs)$ ที่มีค่าน้อยสุด

หยุดคิด	ถ้ามีข้อมูลเข้าเป็นชุดของสายดีเอ็นเอ *Dna* และ *Pattern* ที่มีขนาด k-mer จงออกแบบอัลกอริทึมที่สามารถสร้าง `MOTIFS(Pattern, Dna)`ได้อย่างรวดเร็ว

เหตุผลที่ไม่ต้องพิจารณาทุกชุดของ k-mers เพื่อสร้าง *Motifs* ทั้งหมดที่เป็นไปได้ เนื่องจากเราสามารถเลือก k-mers ใน `MOTIFS(Pattern,Dna)` โดย k-mer ที่ถูกเลือกจากแต่ละ Dna_i เป็นอิสระต่อกัน ถ้า `d(Pattern, Text)`แสดงระยะทางแฮมมิงที่น้อยที่สุดระหว่าง *Pattern* กับ k-mer ใดๆ ใน Text

$$d(Pattern, Text) = \min_{all\ k-mer\ Pattern'\ in\ Text} HAMMINGDISTANCE(Pattern, Pattern')$$

ตัวอย่างเช่น

$$d(GATTCTCA, gcaaaGACGCTGAccaa) = 3$$

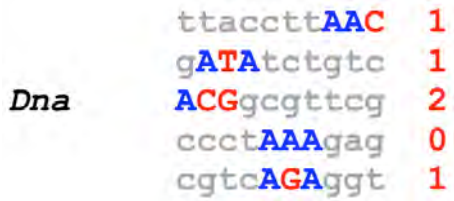
และ k-mer ใน Text ที่มีระยะทางแฮมมิงน้อยสุดเพื่อเทียบกับ *Pattern* ถูกแสดงด้วย

$$MOTIF(GATTCTCA, gcaaaGACGCTGAccaa) = GACGCTGA$$

นิยาม `MOTIF()`นี้อาจมีมากกว่าหนึ่ง k-mers ที่มีระยะทางแฮมมิงน้อยสุดเมื่อเทียบกับ *Pattern* ตัวอย่างเช่น `MOTIF(AAG, gcAATcctCAGc)` มีทั้ง **AAT** และ **CAG** เป็นต้น อย่างไรก็ตามการมี k-mers มากกว่า 1 สายที่เข้าเงื่อนไขนี้ไม่มีผลต่อการหาโมทิฟ ดังนั้นถ้ามีข้อมูลสายอักขระเสียงข้างมาก *Pattern* และชุดของสายดีเอ็นเอ *Dna* = {Dna_1, ..., Dna_t} เรากำหนด $d(Pattern, Dna)$ เป็นผลรวมของระยะทางแฮมมิงระหว่าง *Pattern* กับสายดีเอ็นเอแต่ละเส้น ดังต่อไปนี้

$$d(Pattern, Dna) = \sum_{i=1}^{t} d(Pattern, Dna_i)$$

ตัวอย่างเช่น จากชุดของสายดีเอ็นเอ *Dna* ต่อไปนี้ $d(AAA, Dna)$ = 1 + 1 + 2 + 0 + 1 = 5



เป้าหมายคือหา *Pattern* ที่ทำให้ $d(Pattern, Dna)$ มีค่าน้อยสุดจากชุดของ *Patterns* ทั้งหมดที่เป็นไปได้ ซึ่งเป็นปัญหาเทียบเท่ากับปัญหาการหาโมทิฟก่อนหน้า โดย *Pattern* นี้เรียกว่ามีเดียนสตริง (median string)

นิยามปัญหาที่ 4.3 ปัญหามีเดียนสตริง

ปัญหามีเดียนสตริง (Median String Problem) หามีเดียนสตริง	
ข้อมูลเข้า	ชุดของสายอักขระ *Dna* และค่าจำนวนเต็ม *k*
ผลลัพธ์	k-mer *Pattern* ที่ทำให้ $d(Pattern, Dna)$ ที่มีค่าน้อยที่สุดจากชุดของ k-mer *Patterns* ทั้งหมดที่เป็นไปได้

สังเกตว่าการหามีเดียนสตริงนี้เป็นปัญหาการหาค่าเหมาะที่สุด (optimization) 2 ชั้น ชั้นแรกคือการหา *Pattern* ที่ดีที่สุดจาก *Patterns* ทั้งหมดที่เป็นไปได้ และชั้นที่สองฟังก์ชัน $d(Pattern, Dna)$ เองต้องหา *Pattern* จากชุดของ k-mers ทั้งหมดที่เป็นไปได้ในชุดของสายดีเอ็นเอที่ใกล้เคียงกับ *Pattern* ที่นำมาเปรียบเทียบ การหามีเดียนสตริงแสดงไว้ในรหัสเทียมที่ 4.2 `MedianString()`

เปรียบเทียบวิธีการหาโมทิฟข้างต้น

ถึงแม้ตัวอย่างวิธีการหาโมทิฟข้างต้นทั้งหมดใช้แนวทางทำทุกรูปแบบ แต่ถ้าเปรียบเทียบเวลาที่ใช้ในการหาโมทิฟโดยวิธีการมีเดียนสตริงกับการทำทุกรูปแบบโดยใช้โมทิฟเมทริกซ์จะพบว่าวิธีการมีเดียนสตริงใช้เวลา $O(4^k \cdot n \cdot k \cdot t)$ ในขณะที่วิธีการก่อนหน้าใช้เวลา $O(n^t \cdot k \cdot t)$ ทั้งนี้ในกรณีของวิธีการมีเดียนสตริง $d(Pattern, Dna)$ จะถูกเรียกใช้งานสำหรับ *Pattern* ทั้งหมด 4^k รูปแบบ ในขณะที่การทดสอบ *Pattern* แต่ละรูปแบบจะกราดตรวจ (scan) สายดีเอ็นเอแต่ละเส้น 1 ครั้ง ซึ่งเวลารวมในการกราดตรวจดีเอ็นเอทุกเส้นเท่ากับ $k \cdot n \cdot t$ โดย *k* คือความยาวของโมทิฟที่ต้องการหา *n* คือความยาวของสายดีเอ็นเอแต่ละเส้น และ *t* คือจำนวนเส้นของสายดีเอ็นเอ จะเห็นว่าวิธีการหาโมทิฟโดยใช้มีเดียนสตริงนั้นมีประสิทธิภาพดีกว่า เนื่องจาก *k* มักมีขนาดสั้นไม่เกิน 20 นิวคลีโอไทด์ ในขณะยีนที่แสดงออกในเงื่อนไขหนึ่งๆ อาจมีจำนวนหลายร้อยเส้น

บทเรียนที่ได้จากตัวอย่างนี้คือการเปลี่ยนมุมมองหรือแนวทางการวิเคราะห์ปัญหา (เปลี่ยนจากการหาผลรวมในแนวคอลัมน์มาเป็นการหาผลรวมในแนวแถว ซึ่งได้ผลรวมเท่ากัน) อาจนำมาซึ่งวิธีการในการแก้ปัญหาที่ดีกว่าถึงแม้ยังใช้แนวทางในการแก้ปัญหาแบบเดิม (หาคะแนนรวมน้อยสุด) อย่างไรก็ตามวิธีการหาโมทิฟโดยการหา

มีเดียนสตริงยังมีประสิทธิภาพไม่เพียงพอในการนำมาใช้งานจริง เนื่องจากต้องมีการพิจารณาจำนวนรูปแบบที่เป็นไปได้ทั้งหมด 4^k รูปแบบ ถ้า k = 15 วิธีการนี้จะใช้เวลานานในการหาคำตอบ นอกจากนี้สมมติฐานหลักข้อนึงในการหาโมทิฟคือเราทราบความยาวของโมทิฟ (ค่า k) ที่ต้องการหาล่วงหน้า อย่างไรก็ตามสมมติฐานนี้ไม่เป็นจริงในทางปฏิบัติ ผลที่ตามมาคือเราจำเป็นต้องออกแบบให้อัลกอริทึมสามารถทดสอบค่า k ต่างๆ ได้ รวมทั้งต้องสามารถอนุมานความยาว k ที่ถูกต้องได้

รหัสเทียมที่ 4.2 MedianString

```
MedianString(Dna, k)
    distance <- อิฟินิตี้
    for แต่ละ k-mer Pattern ในรูปแบบ AA..TT ถึง TT..TT
        if distance > d(Pattern, Dna)
            distance <- d(Pattern, Dna)
            Median <- Pattern
    ส่งกลับ Median
```

วิธีการหาโมทิฟแบบละโมบ

โพรไฟล์เมทริกซ์กับการโยนลูกเต๋า

หลายอัลกอริทึมมีลักษณะการทำงานแบบทำซ้ำและในแต่ละรอบที่ทำงานมักมีทางเลือกมากกว่า 1 ทาง อัลกอริทึมแบบละโมบ (greedy algorithm) จะเลือกเส้นทางที่ให้ผลคำตอบรวมดีที่สุดในรอบทำซ้ำนั้นๆ โดยยึดหลักทำให้เร็วแลกกับความถูกต้องที่ลดลงหรือได้คำตอบเพียงโดยประมาณ อย่างไรก็ตามการประยุกต์ใช้อัลกอริทึมแบบละโมบกับปัญหาทางชีววิทยาหลายๆ ปัญหาพบว่าสามารถแก้ปัญหาได้ในระดับหนึ่ง หัวข้อนี้พิจารณาวิธีการหาทิฟแบบละโมบ (greedy motif search) จากรูปที่ 4.5(ง) `PROFILE(Motifs)`มีลักษณะข้อมูลเสมือนเป็นลูกเต๋าที่มี 4 หน้าคือ “A”, “C”, “G”, และ “T” เราสามารถสร้าง k-mer หนึ่งๆ โดยการโยนลูกเต๋าโพรไฟล์เมทริกซ์นี้ k ครั้ง ตัวอย่างเช่น ในคอลัมน์แรกโอกาสที่จะเป็นนิวคลีโอไทด์ “A”, “C”, “G” และ “T” เป็น 0.13, 0.87, 0.0, และ 0.0 ตามลำดับ

PROFILE(Motifs)

A:	.13	.46	0	.4	.87	.97	0	0
C:	.87	0	.46	0	0	0	0	.07
G:	0	.27	.27	0	0	.03	0	.4
T:	0	.27	.27	.6	.13	0	1	.53

$$\Pr(CGTATGTC|\ Profile) = .87 \cdot .27 \cdot .27 \cdot .4 \cdot .13 \cdot .03 \cdot 1 \cdot .07 = 0.00000693$$

*รูปที่ 4.9 การใช้โพรไฟล์เมทริกซ์ในการหาค่าความน่าจะเป็นในการเกิดลำดับเบส k-mer **CGTATGTC***

รูปที่ 4.9 แสดงตัวอย่างโพรไฟล์เมทริกซ์ของ HOXA5 จากรูปที่ 4.5 ที่ถูกนำมาใช้ในการหาค่าความน่าจะเป็นในการเกิดลำดับเบส k-mer ในตัวอย่างนี้ค่าน่าจะเป็นของ k-mer **CGTATGTC** เท่ากับผลคูณของค่าความน่าจะเป็นในการเกิดนิวคลีโอไทด์ “C”, “G, “T”, ..., “G”, “T”, “C” ในคอลัมน์ที่ 1, 2, 3, ..., 6, 7, 8 ในโพรไฟล์เมทริกซ์

k-mer ใดๆ จะมีค่าความน่าจะเป็นสูงถ้าการเรียงตัวของลำดับเบสในสายมีความใกล้เคียงกับลำดับเบสของสายอักขระเสียงข้างมากของโพรไฟล์เมทริกซ์ จากรูปที่ 4.9 ข้างต้นถ้า k-mer ที่ทดสอบเป็นเส้นเดียวกับสายอักขระเสียงข้างมากค่าความน่าจะเป็นจะเท่ากับ 0.049 ตามการคำนวณต่อไปนี้ ซึ่งมีค่ามากกว่าค่าความน่าจะเป็นของ k-mer **CACTAATT** ข้างต้น

$$\mathbf{Pr}(CACTAATT|\boldsymbol{Profile}) = .87 \cdot .46 \cdot .46 \cdot .6 \cdot .87 \cdot .97 \cdot 1 \cdot .53 = 0.049$$

ฝึกหัด	จงคำนวณค่าความน่าจะเป็น $\mathbf{Pr(CCCCTAGC|\boldsymbol{Profile})}$ โดยใช้โพรไฟล์เมทริกซ์ในรูปที่ 4.9

ถ้ามีโพรไฟล์เมทริกซ์เราจะสามารถหาได้ว่า k-mer ใดในสายดีเอ็นเอที่มีความใกล้เคียงกับสายอักขระเสียงข้างมากที่สุดหรือสอดคล้องกับโพรไฟล์ที่สุด ซึ่งเรียกว่าเป็น profile-most probable k-mer ตัวอย่างเช่นถ้าเราใช้โพรไฟล์เมทริกซ์ข้างต้นในการกราดตรวจหา 8-mer ในสายดีเอ็นเอ **CTCCTCATAAATTATCCGCC** จะได้ **CTCCTCATAAATTATCCGCC** เป็น 8-mer (CATAAATT) ที่ใกล้เคียงกับสายอักขระเสียงข้างมากที่สุด ซึ่งในตัวอย่างนี้ค่าความน่าจะเป็นของ 8-mer อื่นๆ จะมีค่าเป็น 0 โดยทั่วไปอาจมีมากกว่าหนึ่ง k-mer ที่ให้ค่าความน่าจะเป็นสูงสุด ซึ่งในกรณีนี้เราจะเลือก k-mer แรกที่พบ วิธีการหาโมทิฟโดยอัลกอริทึมแบบละโมบอาจทำได้โดยในแต่ละรอบใช้ k-mer ใน Dna_1 เป็นโมทิฟแรก จากนั้นสร้างโพรไฟล์เมทริกซ์จาก k-mer นี้ และหา $Motif_2$ ใน Dna_2 ที่สอดคล้องกับโพรไฟล์เมทริกซ์ที่สร้างขึ้นที่สุด เมื่อได้ $Motif_2$ แล้วทำการอัพเดตโพรไฟล์เมทริกซ์โดยใช้ข้อมูลทั้ง $Motif_1$ และ $Motif_2$ และหา $Motif_3$ ใน Dna_3 ที่สอดคล้องกับโพรไฟล์เมทริกซ์ที่สร้างขึ้นและวนซ้ำการทำงานจนครบ t-1 รอบ จะได้โพรไฟล์เมทริกซ์ ของ k-mers ที่ได้จากชุดของสายดีเอ็นเอ *Dna* ซึ่ง k-mers หรือโมทิฟชุดนี้จะถูกนำไปหาคะแนน ถ้าดีกว่าโมทิฟชุดเดิมทำการอัพเดตชุดของโมทิฟให้เป็นชุดใหม่ และทำการวนซ้ำเพื่อใช้ k-mer ถัดๆ ไปของ Dna_1 จนครบดังรหัสเทียมที่ 4.3 `GreedyMotifSearch()`

นิยามปัญหาที่ 4.4 ปัญหาการหา k-mer ที่สอดคล้องกับโพรไฟล์ที่สุด

ปัญหาการหา k-mer ที่สอดคล้องกับโพรไฟล์ที่สุด (Profile-most Probable k-mer Problem) หา k-mer ที่สอดคล้องกับโพรไฟล์ที่สุด	
ข้อมูลเข้า	สายอักขระ Text จำนวนเต็ม k และโพรไฟล์เมทริกซ์ขนาด 4 x k
ผลลัพธ์	k-mer ในสายอักขระ Text ที่สอดคล้องกับโพรไฟล์ที่สุด

รหัสเทียมที่ 4.3 GreedyMotifSearch

```
GreedyMotifSearch(Dna, k, t)
    BestMotifs <- โมทิฟเมทริกซ์ที่สร้างจาก k-mer แรกของทุกสาย Dnai ใน Dna
    for แต่ละ k-mer ในดีเอ็นเอสายที่ 1
        Motif_1 <- k-mer
        for i = 2 to i #ดีเอ็นเอแต่ละสายเริ่มจากสายที่ 2 ถึงสายที่ t
            สร้างโปรไฟล์เมทริกซ์จาก Motif_1 ถึง Motif_i-1
            Motif_i <- k-mer ในดีเอ็นเอสายที่ i ที่สอดคล้องกับโปรไฟล์เมทริกซ์ที่สร้างขึ้น
        Motifs <- {Motif_1, ..., Motif_t}
        if SCORE(Motifs) < SCORE(BestMotifs)
            BestMotifs <- Motifs
    ส่งกลับ BestMotifs
```

วิเคราะห์การทำงานของการหาโมทิฟแบบละโมบ

ถ้าพิจารณาโดยผิวเผินอาจเห็นว่าวิธีการแก้ปัญหาโดยอัลกอริทึมแบบละโมบข้างต้นมีตรรกะการทำงานที่ยอมรับได้ อย่างไรก็ดีถ้าพิจารณาในรายละเอียดจะพบว่าวิธีการนี้ใช้ไม่ได้ ลองพิจารณาการหาโมทิฟ **ACGT** ที่ถูกต้องจากชุดของดีเอ็นเอต่อไปนี้ โดยมีสมมติฐานว่าอัลกอริทึมได้เลือก 4-mer **ACCT** ที่ถูกต้องแล้วจากดีเอ็นเอสายแรก

ttACCTtaac
gATGTctgtc
acgGCGTtag
ccctaACGAg
cgtcagAGGT

โพรไฟล์เมทริกซ์ที่ถูกสร้างจาก 4-mer (**ACCT**) รอบแรกนี้จะเท่ากับ

A:	1	0	0	0
C:	0	1	1	0
G:	0	0	0	0
T:	0	0	0	1

และพร้อมนำไปใช้ในการค้นหาโมทิฟในดีเอ็นเอสายที่สองที่สอดคล้องกับโพรไฟล์เมทริกซ์ข้างต้น อย่างไรก็ตามเนื่องจากโพรไฟล์เมทริกซ์มีค่า 0 ในหลายตำแหน่ง ทำให้ค่าความน่าจะเป็นของ 4-mer ใดๆ ที่ไม่ใช่ **ACCT** มีค่าเป็น 0 ทั้งหมด ยกเว้นมีลำดับเบส **ACCT** ปรากฏอยู่ในดีเอ็นเอทุกเส้น จากตัวอย่างนี้จะพบว่าโอกาสที่ `GreedyMotifSearch()` จะพบคำตอบที่ถูกต้องมีน้อยมาก การปรากฏของ 0 ในหลายตำแหน่งของโพรไฟล์เมทริกซ์นี้เป็นปัญหาที่จำเป็นต้องพิจารณาเพิ่มเติม

การหาโมทิฟจากมุมมองของโอลิเวอร์ ครอมเวลล์

มีความน่าจะเป็นเท่าใดที่จะไม่มีพระอาทิตย์ขึ้นในวันพรุ่งนี้

กฎของโอลิเวอร์ ครอมเวลล์ (Oliver Cromwell) กล่าวว่าค่าความน่าจะเป็นไม่ควรเท่ากับ 0 หรือ 1 นอกจากกำลังพิจารณาข้อมูลเชิงตรรกะที่มีค่าเป็นจริงหรือเท็จเท่านั้น อีกนัยหนึ่งคือควรมีค่าความน่าจะเป็นให้กับการเกิดเหตุการณ์บางเหตุการณ์ถึงแม้มีโอกาสเกิดขึ้นจริงน้อยมาก เช่น ความน่าจะเป็นที่จะไม่มีพระอาทิตย์ขึ้นในวันพรุ่งนี้ ซึ่ง

ในศตวรรษที่ 18 นักคณิตศาสตร์ชาวฝรั่งเศสชื่อ ปิแอร์ ไซมอน ลาปลาซ (Pierre-Simon Laplace) ได้ประมาณค่าโอกาสที่จะไม่มีพระอาทิตย์ขึ้นในวันพรุ่งนี้ไว้เท่ากับ 1/1826251 โดยอาศัยข้อมูลก่อนหน้าว่าใน 5000 ปีที่ผ่านมาพระอาทิตย์ขึ้นในตอนเช้าทุกวัน การประมาณโอกาสที่จะไม่มีพระอาทิตย์ขึ้นในเช้าวันพรุ่งนี้อาจดูเป็นเรื่องขำขันอย่างไรก็ตามแนวคิดและวิธีการของลาปลาซมีบทบาทสำคัญต่อวิธีการทางสถิติในปัจจุบัน

จากการพิจารณาชุดข้อมูลหนึ่งๆ อาจมีบางเหตุการณ์ไม่เคยเกิดขึ้นถ้าข้อมูลมีจำนวนไม่มากพอหรือเหตุการณ์นั้นเป็นเหตุการณ์ที่โดยพื้นฐานมีโอกาสเกิดขึ้นน้อยมาก ซึ่งหมายความว่าค่าความน่าจะเป็นจากการสังเกตการณ์จะเป็น 0 อย่างไรก็ตาม การกำหนดค่า 0 ให้กับเหตุการณ์ต่างๆ โดยดูจากข้อมูลที่มีอยู่เท่านั้นอาจไม่ตรงกับความเป็นจริงรวมทั้งอาจทำให้เกิดความผิดพลาดได้ ลาปลาซแก้ปัญหานี้โดยการปรับค่าความน่าจะเป็นเทียมให้กับเหตุการณ์เหล่านี้

กฎการสืบทอดของลาปลาซ

กฎของครอมเวลล์มีความเกี่ยวข้องกับการคำนวณโอกาสการเกิด k-mer หนึ่งๆ โดยใช้โพรไฟล์เมทริกซ์

PROFILE(Motifs)

A:	.13	.46	0	.4	.87	.97	0	0
C:	.87	0	.46	0	0	0	0	.07
G:	0	.27	.27	0	0	.03	0	.4
T:	0	.27	.27	.6	.13	0	1	.53

$$\Pr(CACTACTT|Profile) = .87 \cdot .46 \cdot .46 \cdot .6 \cdot .87 \cdot 0 \cdot 1 \cdot .53 = 0$$

จากโพรไฟล์ข้างต้นเบสที่ 6 ทำให้ค่าความน่าจะเป็นของสาย k-mer นี้เป็น 0 ถึงแม้ **CACTACTT** ต่างจากสายอักขระหลักเพียงตำแหน่งนี้ตำแหน่งเดียว เพื่อเป็นการเพิ่มความยุติธรรมในการให้คะแนน นักชีวสารสนเทศมักจะแทนค่า 0 ในแต่ละตำแหน่งด้วยตัวเลขจำนวนเต็มที่มีค่าน้อยและเรียกค่านี้ว่าสูโดเคาท์ (pseudo-count) และแนวทางที่ง่ายที่สุดในการเพิ่มสูโดเคาท์เรียกว่ากฎการสืบทอดของลาปลาซ ซึ่งเป็นไปในแนวทางเดียวกับที่ลาปลาซใช้ในการคำนวณค่าความน่าจะเป็นที่พระอาทิตย์จะไม่ขึ้นในวันพรุ่งนี้ ในกรณีของโมทิฟ สูโดเคาท์มักถูกกำหนดให้มีค่าเป็น 1 โดยสามารถเพิ่มเข้าไปในผลของ `COUNT(Motifs)` ตั้งต้นต่อไปนี้

Motifs

T	A	A	C
G	T	C	T
A	C	T	A
A	G	G	T

COUNT(*Motifs*)

A:	2	1	1	1
C:	0	1	1	1
G:	1	1	1	0
T:	1	1	1	2

Profile(*Motifs*)

A:	2/4	1/4	1/4	1/4
C:	0	1/4	1/4	1/4
G:	1/4	1/4	1/4	0
T:	1/4	1/4	1/4	2/4

และเมื่อมีการเพิ่ม 1 ให้กับทุกค่าของ `COUNT(Motifs)` ตามกฎการสืบทอดของลาปลาซทั้งสองเมทริกซ์ ข้างต้น จะถูกปรับค่าเป็นดังต่อไปนี้

```
              A: 2+1 1+1 1+1 1+1                A: 3/8  2/8  2/8  2/8
              C: 0+1 1+1 1+1 1+1                C: 1/8  2/8  2/8  2/8
COUNT(Motifs) G: 1+1 1+1 1+1 0+1 Profile(Motifs)G: 2/8  2/8  2/8  1/8
              T: 1+1 1+1 1+1 2+1                T: 2/8  2/8  2/8  3/8
```

ปรับปรุงการหาโมทิฟแบบละโมบ

เราสามารถปรับปรุงขั้นตอนการหาโมทิฟแบบละโมบโดยเพิ่มการใช้สูโดเคาท์ จากรหัสเทียมที่ 4.3 ข้างต้นจะเปลี่ยนจาก

`สร้างโปรไฟล์เมทริกซ์จาก Motif_1 ถึง Motif_i-1` เป็น

`ประยุกต์ใช้กฎการสืบทอดของลาปลาซในการสร้างโปรไฟล์เมทริกซ์จาก Motif_1 ถึง Motif_i-1`

พิจารณาตัวอย่างการหาโมทิฟขนาด 4-mer ในตัวอย่างก่อนหน้า หลังการประยุกต์ใช้กฎการสืบทอดของลาปลาซได้ผลดังต่อไปนี้

```
ttACCTtaac
gATGTctgtc
acgGCGTtag
ccctaACGAg
cgtcagAGGT

Motifs: ACCT
```

```
              A: 1+1 0+1 0+1 0+1                A: 2/5 1/5 1/5 1/5
              C: 0+1 1+1 1+1 0+1                C: 1/5 2/5 2/5 1/5
COUNT(Motifs) G: 0+1 0+1 0+1 0+1 PROFILE(Motifs)G: 1/5 1/5 1/5 1/5
              T: 0+1 0+1 0+1 1+1                T: 1/5 1/5 1/5 2/5
```

และใช้โพรไฟล์เมทริกซ์ในการคำนวณค่าความน่าจะเป็นของ 4-mer ทั้งหมดที่อยู่ในดีเอ็นเอเส้นที่สองได้ผลดังต่อไปนี้

gATG	ATGT	TGTc	GTct	Tctg	ctgt	tgtc
$1/5^4$	$4/5^4$	$1/5^4$	$4/5^4$	$2/5^4$	$2/5^4$	$1/5^4$

ซึ่งได้โมทิฟที่สอดคล้องกับโพรไฟล์เมทริกซ์มากที่สุดสองโมทิฟคือ **ATGT** และ **GTct** สมมติว่าเราโชคดี ในการเลือก k-mer ที่ถูกต้องระหว่าง k-mers ที่ดีที่สุดสองสายนี้ จะได้โมทิฟเมทริกซ์และโพรไฟล์เมทริกซ์ใหม่ ดังต่อไปนี้

```
Motifs: ACCT
        ATGT
```

```
              A: 2+1 0+1 0+1 0+1                A: 3/6 1/6 1/6 1/6
              C: 0+1 1+1 1+1 0+1                C: 1/6 2/6 2/6 1/6
COUNT(Motifs) G: 0+1 0+1 1+1 0+1 PROFILE(Motifs)G: 1/6 1/6 2/6 1/6
              T: 0+1 1+1 0+1 2+1                T: 1/6 2/6 1/6 3/6
```

และใช้โพรไฟล์เมทริกซ์นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mer ทั้งหมดที่อยู่ในดีเอ็นเอเส้นที่สามได้ผลดังต่อไปนี้

acgG	cgGC	gGCG	GCGT	CGTt	GTta	Ttag
$12/6^4$	$2/6^4$	$2/6^4$	$12/6^4$	$3/6^4$	$2/6^4$	$2/6^4$

ซึ่งได้โมทิฟที่สอดคล้องกับโพรไฟล์เมทริกซ์มากที่สุดสองโมทิฟคือ acgG และ GCGT ถ้าในรอบนี้ acgG ถูกเลือกใช้ จะได้โมทิฟเมทริกซ์และโพรไฟล์เมทริกซ์ใหม่ดังต่อไปนี้

Motifs:
ACCT
ATGT
acgG

COUNT(*Motifs*)

A:	3+1	0+1	0+1	0+1
C:	0+1	2+1	1+1	0+1
G:	0+1	0+1	2+1	1+1
T:	0+1	1+1	0+1	2+1

PROFILE(*Motifs*)

A:	4/7	1/7	1/7	1/7
C:	1/7	3/7	2/7	1/7
G:	1/7	1/7	3/7	2/7
T:	1/7	2/7	1/7	3/7

และใช้โพรไฟล์เมทริกซ์นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mer ทั้งหมดที่อยู่ในดีเอ็นเอเส้นที่สี่ได้ผลดังต่อไปนี้

ccct	ccta	ctaA	taAC	aACG	ACGA	CGAg
$18/7^4$	$3/7^4$	$2/7^4$	$1/7^4$	$16/7^4$	$36/7^4$	$2/7^4$

ถึงแม้เราจะเลือก k-mer ที่ไม่ถูกต้องมาสร้างโมทิฟเมทริกซ์และโพรไฟล์เมทริกซ์จากดีเอ็นเอเส้นที่สาม โพรไฟล์เมทริกซ์ที่สร้างขึ้นจะยังสามารถนำมาใช้ในการหาโมทิฟที่ถูกต้อง ACGA ในดีเอ็นเอสายที่สี่ และนำมาสร้างเป็นโมทิฟเมทริกซ์และโพรไฟล์เมทริกซ์ใหม่ในรอบถัดไปดังต่อไปนี้

Motifs:
ACCT
ATGT
acgG
ACGA

COUNT(*Motifs*)

A:	4+1	0+1	0+1	1+1
C:	0+1	3+1	1+1	0+1
G:	0+1	0+1	3+1	1+1
T:	0+1	1+1	0+1	2+1

PROFILE(*Motifs*)

A:	5/8	1/8	1/8	2/8
C:	1/8	4/8	2/8	1/8
G:	1/8	1/8	4/8	2/8
T:	1/8	2/8	1/8	3/8

และใช้โพรไฟล์เมทริกซ์นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mer ทั้งหมดที่อยู่ในดีเอ็นเอเส้นที่ห้าได้ผลดังต่อไปนี้

cgtc	gtca	tcag	cagA	agAG	gAGG	AGGT
$1/8^4$	$8/8^4$	$8/8^4$	$8/8^4$	$10/8^4$	$8/8^4$	$60/8^4$

และได้ 4-mer ที่สอดคล้องกับโพรไฟล์เมทริกซ์มากที่สุดในดีเอ็นเอเส้นที่ห้าคือ **AGGT** ซึ่งหมายความว่า `GreedyMotifSearch()`สามารถสร้างโมทิฟเมทริกซ์ต่อไปนี้ ที่นำไปสู่การสร้างสายอักขระเสียงข้างมากที่มีความถูกต้อง

```
                ACCT
                ATGT
Motifs:         acgG
                ACGA
                AGGT

CONSENSUS(Motifs): ACGT
```

จะเห็นว่าการใช้สูโดเคาท์สามารถทำให้ประสิทธิภาพการหาโมทิฟโดยวิธีการแบบละโมบได้ผลที่ดีขึ้น อย่างไรก็ตามยังมีวิธีการหาโมทิฟอื่นที่มีประสิทธิภาพมากกว่า

การหาโมทิฟแบบสุ่ม

อัลกอริทึมแบบสุ่ม (randomized algorithms) เกือบทั้งหมด รวมทั้งอัลกอริทึมที่ใช้ในการหาโมทิฟแบบสุ่มที่จะกล่าวถึงต่อไปเป็นอัลกอริทึมกลุ่มมอนติคาร์โล (Monte Carlo algorithms) มีคุณสมบัติไม่รับประกันว่าคำตอบที่ได้ถูกต้อง 100% หรือเป็นคำตอบที่ดีที่สุด แต่สามารถหาคำตอบแบบประมาณได้รวดเร็ว ทำให้สามารถรันอัลกอริทึมได้หลายครั้งและเลือกคำตอบที่มีการประมาณค่าที่ถูกต้องที่สุดได้

ถ้ามีข้อมูลเข้าเป็นชุดของสายดีเอ็นเอ *Dna* และโพรไฟล์เมทริกซ์ขนาด 4 x k เรานิยาม `MOTIFS(Profile,Dna)` เป็นชุดของ k-mers ที่มีความสอดคล้องกับ *Profile* มากที่สุดโดยแต่ละ k-mer มาจากสายดีเอ็นเอแต่ละเส้น พิจารณาโพรไฟล์และชุดของสายดีเอ็นเอต่อไปนี้

```
                A: 4/5  0    0    1/5          ttaccttaac
                C:  0   3/5  1/5   0           gatgtctgtc
PROFILE(Motifs) G: 1/5  1/5  4/5   0     Dna   acggcgttag
                T:  0   1/5  0    4/5          ccctazcgag
                                                cgtcagaggt
```

ถ้าใช้ `PROFILE()` ข้างต้นในการหา 4-mer ที่สอดคล้องกับโพรไฟล์มากที่สุดจากดีเอ็นเอแต่ละเส้น จะได้ชุดของ k-mers (อักษรสีแดงในแต่ละบรรทัด) ต่อไปนี้

```
                                ttaccttaac
                                gatgtctgtc
MOTIFS(Profile,Dna)             acggcgttag
                                ccctaacgag
                                cgtcagaggt
```

โดยทั่วไปในรอบแรกเราสามารถสุ่มเลือก k-mer ใดๆ จากดีเอ็นเอแต่ละเส้น และนำมาสร้างเป็นโมทิฟเมทริกซ์และโพรไฟล์เมทริกซ์ จากนั้นโพรไฟล์เมทริกซ์นี้จะถูกนำไปหาชุดของโมทิฟเพื่อสร้างโมทิฟเมทริกซ์และโพรไฟล์เมทริกซ์ในรอบถัดๆ ไปดังสมการต่อไปนี้

$$\texttt{MOTIFS(PROFILE(Motifs),}\mathit{Dna}\texttt{)}$$

สมการนี้อยู่บนสมมติฐานว่าชุดของโมทิฟที่ได้จาก `MOTIFS()` จะได้คะแนนดีกว่าโมทิฟ (k-mers) ชุดแรกที่เลือกมาแบบสุ่มจากดีเอ็นเอแต่ละเส้น โมทิฟชุดใหม่นี้จะถูกนำไปสร้างโพรไฟล์เมทริกซ์ในรอบถัดไปตามสมการ

$$\texttt{PROFILE(MOTIFS(PROFILE(}\mathit{Motifs}\texttt{,}\mathit{Dna}\texttt{))}$$

และโพรไฟล์ใหม่ที่ได้จะถูกนำไปใช้หาโมทิฟที่สอดคล้องกับโพรไฟล์นี้ที่สุดจากดีเอ็นเอแต่ละเส้น

$$\texttt{MOTIFS(PROFILE(MOTIFS(PROFILE(}\mathit{Motifs}\texttt{,}\mathit{Dna}\texttt{)),}\mathit{Dna}\texttt{)}$$

ทำวนซ้ำถ้าคะแนนรวมของโมทิฟเมทริกซ์ยังลดลง

วิธีการข้างต้นเป็นวิธีการหาโมทิฟแบบสุ่ม (randomized motif search) (รหัสเทียมที่ 4.4 `RandomizedMotifSearch()`) เนื่องจากการรัน `RandomizedMotifSearch()` เพียงครั้งเดียวอาจได้คำตอบที่ไม่ดี โดยทั่วไปนักชีวสารสนเทศจะทำการรันฟังก์ชันซ้ำเป็นหลักหลายพันครั้ง โดยในแต่ละรอบของการรันชุดของ k-mers ในรอบแรกจะถูกเลือกมาแบบสุ่ม และชุดของ k-mers (โมทิฟ) ที่ดีที่สุดจะถูกเลือกจากผลการรันทั้งหมด

รหัสเทียมที่ 4.4 RandomizedMotifSearch

```
RandomizedMotifSearch(Dna, k, t)
    Motifs <- สุ่มเลือก k-mer หนึ่งเส้นมาจากดีเอ็นเอแต่ละเส้นมาใส่ใน Motifs
    BestMotifs <- Motifs #นำชุดของโมทิฟนี้มาเก็บไว้เป็นชุดของโมทิฟที่ดีที่สุด
    while True วนลูปไม่รู้จบ
        #สร้างโปรไฟล์เมทริกซ์จากชุดของโมทิฟที่สุ่มเลือกมา
        Profile <- PROFILE(Motifs)
        #หาโมทิฟชุดใหม่ที่สอดคล้องกับโปรไฟล์มากสุดจากดีเอ็นเอแต่เส้น เก็บเข้าตัวแปร Motifs
        Motifs <- MOTIFS(Profile, Dna)
        if คะแนนของ Motifs < คะแนนของ BestMotifs
            #นำโมทิฟชุดใหม่ที่คะแนนรวมน้อยกว่ามาเก็บใน BestMotifs แทนชุดเดิม
            BestMotifs <- Motifs
        else
            ส่งกลับ BestMotifs

```

ทำไมการหาโมทิฟแบบสุ่มถึงให้ผลลัพธ์ที่ถูกต้องได้

อาจมีคำถามว่าการหาโมทิฟแบบสุ่มให้ผลลัพธ์ที่ถูกต้องได้อย่างไร ในเมื่อในแต่ละรอบของการทำงานเริ่มจากการสุ่มเลือกชุดของ k-mers ลองพิจารณาชุดของสายดีเอ็นเอที่มีโมทิฟ (4,1) ACGT แทรกอยู่ในตัวอย่างต่อไปนี้ โดย

ตัวอักษรใหญ่ในดีเอ็นเอแต่ละเส้นแสดงโมทิฟที่เป็นคำตอบ และสมมติว่าในรอบแรกชุดของ k-mers (ตัวอักษรสีแดง) ที่สุ่มเลือกมาจากดีเอ็นเอแต่ละเส้นแทบไม่ถูกต้องเลย

```
          ttACCTtaac
          gATGTctgtc
Dna       ccgGCGTtag
          cactaACGAg
          cgtcagAGGT
```

จากชุด k-mers ที่สุ่มเลือกมาสามารถนำมาสร้างเป็นโมทิฟเมทริกซ์และโพรไฟล์เมทริกซ์ดังต่อไปนี้

```
Motifs
t a a c
G T c t
c c g G
a c t a
A G G T
```

```
PROFILE(Motifs)
A: 0.4  0.2  0.2  0.2
C: 0.2  0.4  0.2  0.2
G: 0.2  0.2  0.4  0.2
T: 0.2  0.2  0.2  0.4
```

เมื่อนำโพรไฟล์เมทริกซ์ที่สร้างขึ้นไปหา k-mer ที่สอดคล้องกับโพรไฟล์นี้มากที่สุด (ถูกแสดงด้วยคะแนนสีแดง) จากดีเอ็นเอแต่ละเส้น ได้ผลคะแนนความน่าจะเป็นดังต่อไปนี้

```
.0016/ttAC .0016/tACC .0128/ACCT .0064/CCTt .0016/Ctta .0016/Ttaa .0016/taac
.0016/gATG .0128/ATGT .0016/TGTc .0032/GTct .0032/Tctg .0032/ctgt .0016/tgtc
.0064/ccgG .0036/cgGC .0016/gGCG .0128/GCGT .0032/CGTt .0016/Gtta .0016/Ttag
.0032/cact .0064/acta .0016/ctaA .0016/taAC .0032/aACG .0128/ACGA .0016/CGAg
.0016/cgtc .0016/gtca .0016/tcag .0032/cagA .0032/agAG .0032/gAGG .0128/AGGT
```

เมื่อนำ 4-mer ชุดนี้ไปแสดงในชุดของดีเอ็นเอตั้งต้นจะพบว่าเป็นโมทิฟ (4,1) ที่เป็นคำตอบ

```
          ttACCTtaac
          gATGTctgtc
Dna       ccgGCGTtag
          cactaACGAg
          cgtcagAGGT
```

หยุดคิด	จงอธิบายว่าทำไมการเลือก k-mer แบบสุ่ม ถึงให้ผลลัพธ์ที่ถูกต้องได้

สำหรับปัญหาก่อนหน้าที่มีการแทรก 15-mer โมทิฟ-(15,4) ของ **AAAAAAAAGGGGGGG** เข้าไปในดีเอ็นเอ 10 เส้นโดยที่แต่ละเส้นมีความยาว 600 นิวคลีโอไทด์ ถ้าเราใช้วิธีการหาโมทิฟแบบสุ่มเพื่อแก้ปัญหานี้โดยการรันทั้งสิ้น 100,000 ครั้งโดยที่การรันทุกครั้ง k-mers ชุดแรกจะถูกเลือกมาแบบสุ่ม รูปที่ 4.10 แสดงชุดของโมทิฟ

ที่มีคะแนนรวมน้อยที่สุดเท่ากับ 43 จากการรัน 100,000 ครั้ง โดยได้สายอักขระเสียงข้างมากเป็น **AAAAAAAAacaGGGG** โมทิฟเหล่านี้มีความอนุรักษ์น้อยกว่าชุดของโมทิฟที่ถูกแทรกเข้าไปเพียงเล็กน้อยโดยโมทิฟชุดนั้นมีคะแนนรวมเท่ากับ 40 หรือ 41 โดยการรัน `GreedyMotifSearch()` อย่างไรก็ตาม `RandomizedMotifSearch()` สามารถรันได้จำนวนรอบมากกว่าในเวลาที่เท่ากัน ทำให้เพิ่มโอกาสในการพบโมทิฟที่ถูกต้องมากกว่า

หยุดคิด	จงเขียนโค้ดของฟังก์ชัน `RandomizedMotifSearch()` และตรวจสอบดูว่าได้ผลลัพธ์ของสายอักขระเสียงข้างมากที่คล้ายกับตัวอย่างข้างต้นหรือไม่ และต้องรันโค้ดทั้งหมดกี่รอบเพื่อให้ได้โมทิฟ-(15,4) เมทริกซ์ ที่มีคะแนนรวมเท่ากับ 40

รูปที่ 4.10 (ก) ชุดของโมทิฟที่เป็นผลลัพธ์จากการหาโมทิฟแบบสุ่มโดยมีคะแนนรวมน้อยที่สุดจากการรัน 100,000 ครั้ง (ข) สายอักขระเสียงข้างมากที่ได้จากโมทิฟเมทริกซ์

(*ที่มา*: *รูปที่* 2.7 *ของ* [52])

ถึงแม้ชุดของโมทิฟที่เป็นผลลัพธ์จาก `RandomizedMotifSearch()` จะมีความอนุรักษ์น้อยกว่า `MedianString()` เล็กน้อย ข้อดีของ `RandomizedMotifSearch()` คือสามารถหาโมทิฟที่มีขนาดยาวกว่า เพราะเวลาในการรัน `MedianString()` ขึ้นอยู่กับความยาวของโมทิฟ

ทำไมการหาโมทิฟแบบสุ่มถึงให้ผลลัพธ์ที่ดี

ในหัวข้อที่ผ่านมาเราสร้างโพรไฟล์เมทริกซ์จากชุดของโมทิฟ-(4,1) ที่นำมาแทรกในชุดของสายดีเอ็นเอ โดยมีสายอักขระเสียงข้างมากเป็น ACGT ดังต่อไปนี้

A:	**0.8**	0.0	0.0	0.2
C:	0.0	**0.6**	0.2	0.0
G:	0.2	0.2	**0.8**	0.0
T:	0.0	0.2	0.0	**0.8**

ถ้าแต่ละตำแหน่งในสายดีเอ็นเอแต่ละเส้นมีโอกาสเกิดนิวคลีโอไทด์ “A”, “C”, “G” และ “T” เท่าๆ กัน เราสามารถคาดหวังว่าโพรไฟล์เมทริกซ์ที่สร้างจากชุดของ k-mers ที่ถูกสุ่มเลือกมาจากชุดของสายดีเอ็นเอจะมีลักษณะต่อไปนี้

A:	0.25	0.25	0.25	0.25
C:	0.25	0.25	0.25	0.25
G:	0.25	0.25	0.25	0.25
T:	0.25	0.25	0.25	0.25

ซึ่งเป็นโพรไฟล์ที่ยูนิฟอร์ม (uniform) คือแสดงโอกาสในการเกิดนิวคลีโอไทด์แต่ละแบบเท่ากันในทุกตำแหน่ง ซึ่งเป็นโพรไฟล์ที่ไม่มีประโยชน์ ในทางกลับกันถ้าชุดของโมทิฟที่นำมาสร้างโพรไฟล์มีโอกาสเกิดนิวคลีโอไทด์แต่ละแบบในแต่ละตำแหน่งไม่เท่ากัน ดังตัวอย่างต่อไปนี้ โพรไฟล์ลักษณะนี้สามารถนำไปสู่การพบโมทิฟที่เป็นคำตอบ

A:	**0.4**	0.2	0.2	0.2
C:	0.2	**0.4**	0.2	0.2
G:	0.2	0.2	**0.4**	0.2
T:	0.2	0.2	0.2	**0.4**

หรือเข้าใกล้คำตอบได้ ซึ่งการทำงานของ `RandomizedMotifSearch()`ถูกออกแบบให้การทำงานในรอบวนซ้ำได้ผลที่เข้าใกล้คำตอบมากขึ้น จากตัวอย่างโพรไฟล์เมทริกซ์ข้างต้น ในรอบถัดไปจะมีการปรับค่าเข้าใกล้โมทิฟที่เป็นคำตอบมากขึ้นตามค่าความถี่ที่แตกต่างกันของแต่ละนิวคลีโอไทด์ในแต่ละตำแหน่ง ทั้งนี้อยู่บนสมมติฐานว่ามีโมทิฟที่ต้องการหาแทรกอยู่ในชุดของสายดีเอ็นเอ

ฝึกหัด	ถ้ามีการสุ่มเลือก 15-mer มาจากแต่ละสายดีเอ็นเอยาว 600 นิวคลีโอไทด์ จำนวน 10 เส้น จงคำนวณค่าความน่าจะเป็นที่มี 15-mer อย่างน้อยหนึ่งเส้นเป็นโมทิฟที่แทรกเข้าไป

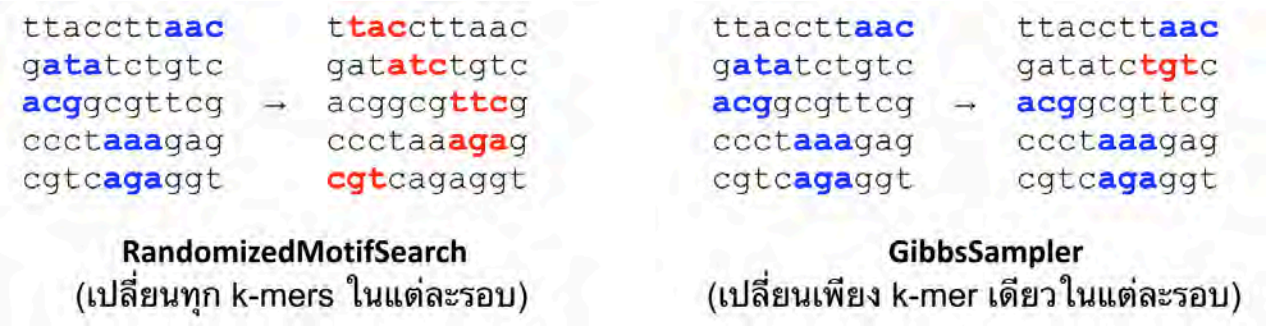
ถึงแม้มีค่าความน่าจะเป็นน้อยมากที่ชุดของ k-mers ที่สุ่มเลือกมาจะเป็นชุดเดียวกับโมทิฟที่แทรกเข้าไปทั้งหมด และถึงแม้ค่าความน่าจะเป็นที่มีอย่างน้อยหนึ่ง k-mer ที่ถูกเลือกมาแบบสุ่มตรงกับโมทิฟที่แทรกเข้าไปจะมีค่าน้อย ความน่าจะเป็นค่านี้ยังมีนัยสำคัญ เนื่องจากเราสามารถรัน `RandomizedMotifSearch()` หลายครั้งเพื่อเพิ่มโอกาสที่จะพบบาง k-mer ที่เป็นโมทิฟที่แทรกเข้าไป และนำไปสู่การสร้างโพรไฟล์เมทริกซ์ที่มีทิศทางจำเพาะต่อโมทิฟที่ต้องการหา

อย่างไรก็ตามการพบเพียงหนึ่ง k-mer ที่เป็นโมทิฟจริงมักไม่เพียงพอในการทำให้ `RandomizedMotifSearch()` หาคำตอบที่ดีที่สุดได้ เนื่องจากมีตำแหน่งเริ่มต้นของ k-mer ที่สามารถสุ่มเลือกจากดีเอ็นเอเส้นหนึ่งๆ จำนวนมากมาย การเลือกชุดของโมทิฟแบบสุ่มในทุกๆ รอบที่รันอาจไม่ได้คำตอบที่ดีเท่าตัวอย่างข้างต้นเพราะมีโอกาสน้อยที่ชุดของ k-mers ที่เกิดจากการสุ่มเลือกใหม่ในทุกรอบจะนำไปสู่โพรไฟล์เมทริกซ์ที่มีทิศทางความจำเพาะต่อโมทิฟที่ต้องการหา

ฝึกหัด	ถ้ามีการสุ่มเลือก 15-mer จากแต่ละสายดีเอ็นเอยาว 600 นิวคลีโอไทด์ จำนวน 10 เส้น จงคำนวณค่าความน่าจะเป็นที่มี 15-mer อย่างน้อยสองเส้น เป็นโมทิฟที่แทรกเข้าไป

กิบส์แซมพลิง

ในขณะที่ `RandomizedMotifSearch()`ทำการเลือก k-mers แบบสุ่มใหม่ทั้งหมดในแต่ละรอบของการทำงาน ทำให้มีโอกาสที่ k-mers ที่เป็นโมทิฟที่ถูกต้องแล้วถูกละทิ้งไปทั้งหมดในรอบใหม่ กิบส์แซมเพลอร์ (Gibbs sampler) ได้ปรับแนวทาง `RandomizedMotifSearch()`โดยในแต่ละรอบจะเลือกทิ้งเพียง k-mer เดียวจากรอบที่ผ่านมา ดังแสดงในตัวอย่างต่อไปนี้



โดย `GibbsSampler()` เริ่มการทำงานรอบแรกในลักษณะเดียวกับ `RandomizedMotifSearch()` คือชุดของ k-mers จะถูกเลือกแบบสุ่มจากดีเอ็นเอแต่ละเส้น รอบหลังจากนั้น `GibbsSampler()` จะทำการสุ่มเลือกค่าระหว่าง 1 ถึง t และทำการกำหนด k-mer ที่เลือกใหม่เฉพาะบรรทัดที่ถูกสุ่มเลือกมานั้น รหัสเทียมที่ 4.5 แสดงการทำงานของ `GibbsSampler()`โดยมีการทำซ้ำ N ครั้ง ในทางปฏิบัติการหยุดการทำงานของ `GibbsSampler()`มีได้หลายเงื่อนไขซึ่งจะไม่กล่าวถึงในบทเรียนนี้

หยุดคิด	ในขณะที่ค่าคะแนนที่เป็นผลลัพธ์จากการรัน `RandomizedMotifSearch()` จะลดลงในการรันแต่ละรอบ ในกรณีของ `GibbsSampler()` มีโอกาสเป็นไปได้ ที่คะแนนในการรันอาจแกว่งเพิ่มมากขึ้น ปรากฏการณ์นี้สมเหตุสมผลหรือไม่

รหัสเทียมที่ 4.5 GibbsSampler

```
GibbsSampler(Dna, k, t, N)
    Motifs <- สุ่มเลือก k-mer หนึ่งเส้นมาจากดีเอ็นเอแต่ละเส้นมาใส่ใน Motifs
    BestMotifs <- Motifs #นำชุดของโมทิฟนี้มาเก็บไว้เป็นชุดของโมทิฟที่ดีที่สุด
    for i <- 1 ถึง N
        i <- สุ่มเส้นของดีเอ็นเอที่ k-mer จะต้องถูกแทนที่โดยใช้ RANDOM(t)
        #สร้างโปรไฟล์เมทริกซ์จากชุดของโมทิฟยกเว้น Motif_i
        Profile <- PROFILE(Motifs ยกเว้น Motif_i)
        Motif_i <- เลือก k-mer ใหม่แบบสุ่มจากโปรไฟล์
        if คะแนนของ Motifs < คะแนนของ BestMotif
            #นำโมทิฟชุดใหม่ที่คะแนนรวมน้อยกว่ามาเก็บใน BestMotifs แทนชุดเดิม
            BestMotifs <- Motifs
    ส่งกลับ BestMotifs
```

ขั้นตอนการทำงานของกิบส์แซมพลิง

หัวข้อนี้แสดงขั้นตอนการทำงานของกิบส์แซมพลิงในรายละเอียดโดยใช้ชุดของสายดีเอ็นเอและ k-mers แบบสุ่มรอบแรก (ตัวอักษรสีแดง) เป็นชุดเดียวกับที่ใช้ในการอธิบายการหาโมทิฟแบบสุ่ม `RandomizedMotifSearch()` ในหัวข้อก่อนหน้า และในรอบที่สองของการทำงาน k-mer ของดีเอ็นเอสายที่สามถูกเลือกออกจากกลุ่ม

```
                ttACCTtaac          ttACCTtaac
                gATGTctgtc          gATGTctgtc
Dna             ccgGCGTtag    ⟶    ----------
                cactaACGAg          cactaACGAg
                cgtcagAGGT          cgtcagAGGT
```

ซึ่งได้ผลเป็นโมทิฟเมทริกซ์ เคาท์เมทริกซ์ และโพรไฟล์เมทริกซ์ต่อไปนี้

```
Motifs
t a a c
G T c t
a c t a
A G G T
```

```
COUNT(Motifs)          PROFILE(Motifs)
A: 2  1  1  1          A: 2/4  1/4  1/4  1/4
C: 0  1  1  1          C: 0    1/4  1/4  1/4
G: 1  1  1  0          G: 1/4  1/4  1/4  0
T: 1  1  1  2          T: 1/4  1/4  1/4  2/4
```

จะเห็นว่าโพรไฟล์เมทริกซ์มีความอนุรักษ์มากกว่าโพรไฟล์แบบยูนิฟอร์ม (uniform) เพียงเล็กน้อย ซึ่งอาจทำให้ไม่แน่ใจว่าโพรไฟล์เมทริกซ์นี้จะนำไปสู่การหาโมทิฟที่เป็นคำตอบได้อย่างไร ถ้าใช้โพรไฟล์เมทริกซ์นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mers ทั้งหมดในสายดีเอ็นเอที่ถูกคัดออกได้ผลดังนี้

ccgG	cgGC	gGCG	GCGT	CGTt	GTta	Ttag
0	0	0	1/128	0	1/256	0

จะเห็นว่าค่าความน่าจะเป็นของ 4-mers แทบทุกแบบมีค่าเป็น 0 ซึ่งเป็นปัญหาเดียวกับวิธีการหาโมทิฟแบบละโมบ และสามารถแก้ปัญหาโดยใช้สูโดเคาท์ ทำให้ได้เคาท์เมทริกซ์และโพรไฟล์เมทริกซ์ใหม่ต่อไปนี้

COUNT (*Motifs*)

A:	3	2	2	2
C:	1	2	2	2
G:	2	2	2	1
T:	2	2	2	3

PROFILE (*Motifs*)

A:	3/8	2/8	2/8	2/8
C:	1/8	2/8	2/8	2/8
G:	2/8	2/8	2/8	1/8
T:	2/8	2/8	2/8	3/8

หลังการเพิ่มสูโดเคาท์และใช้โพรไฟล์เมทริกซ์ใหม่นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mers ทั้งหมด ในสายดีเอ็นเอที่ถูกคัดออก (สายที่สาม) ได้ผลดังนี้

ccgG	cgGC	gGCG	GCGT	CGTt	GTta	Ttag
$4/8^4$	$8/8^4$	$8/8^4$	$24/8^4$	$12/8^4$	$16/8^4$	$8/8^4$

ดีเอ็นเอสายที่สามถูกนำกลับเข้ามาและ 4-mer **GCGT** จะถูกเลือกแทน 4-mer เดิม ccgG ที่ถูกเลือกแบบสุ่มไว้จากรอบที่ผ่านมาและเริ่มรอบการรันถัดไปโดยการสุ่มเลือกสายดีเอ็นเอที่จะถูกคัดออกใหม่โดยเป็นสายที่หนึ่งในตัวอย่างต่อไปนี้

Dna		
	ttACCT**taac**	----------
	gAT**GTct**gtc	gAT**GTct**gtc
	ccgGCGTtag →	ccg**GCGT**tag
	c**acta**ACGAg	c**acta**ACGAg
	cgtcag**AGGT**	cgtcag**AGGT**

ซึ่งได้ผลเป็นโมทิฟเมทริกซ์ เคาท์เมทริกซ์ และโพรไฟล์เมทริกซ์ต่อไปนี้

Motifs

G	T	c	t
G	C	G	T
a	c	t	a
A	G	G	T

COUNT (*Motifs*)

A:	2	0	0	1
C:	0	2	1	0
G:	2	1	2	0
T:	0	1	1	3

PROFILE (*Motifs*)

A:	2/4	0	0	1/4
C:	0	2/4	1/4	0
G:	2/4	1/4	2/4	0
T:	0	1/4	1/4	3/4

จะเห็นว่าโพรไฟล์เมทริกซ์เข้าใกล้โมทิฟจริงมากกว่ารอบแรก และเมื่อทำการเพิ่มสูโดเคาท์จะได้ผลดังต่อไปนี้

COUNT (*Motifs*)

A:	3	1	1	2
C:	1	3	2	1
G:	3	2	3	1
T:	1	2	2	4

PROFILE (*Motifs*)

A:	3/8	1/8	1/8	2/8
C:	1/8	3/8	2/8	1/8
G:	3/8	2/8	3/8	1/8
T:	1/8	2/8	2/8	4/8

หลังการเพิ่มสูโดเคาท์และใช้โพรไฟล์เมทริกซ์ใหม่นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mers ทั้งหมด ในสายดีเอ็นเอที่ถูกคัดออก (สายที่หนึ่ง) ได้ผลดังนี้

ttAC	tACC	ACCT	CCTt	CTta	Ttaa	taac
$2/8^4$	$2/8^4$	$72/8^4$	$24/8^4$	$8/8^4$	$4/8^4$	$1/8^4$

โดยดีเอ็นเอสายที่หนึ่งจะถูกนำกลับเข้ามาและ 4-mer **ACCT** จะถูกเลือกแทน 4-mer เดิม `taac` ที่ถูกเลือกแบบสุ่มไว้จากรอบที่ผ่านมา และเริ่มรอบการรันถัดไปโดยการสุ่มเลือกสายดีเอ็นเอที่จะถูกคัดออก (สายที่สี่) ในรอบถัดไป ดังตัวอย่างต่อไปนี้

Dna

ttACCTtaac
gATGTctgtc
ccgGCGTtag
cactaACGAg
cgtcagAGGT

→

ttACCTtaac
gATGTctgtc
ccgGCGTtag
----
cgtcagAGGT

และได้โมทิฟเมทริกซ์ เคาท์เมทริกซ์ และโพรไฟล์เมทริกซ์ที่มีสูโดเคาท์แล้วดังต่อไปนี้

Motifs

A	C	C	T
G	T	c	t
G	C	G	T
A	G	G	T

COUNT(*Motifs*)

A:	3	1	1	1
C:	1	3	3	1
G:	3	2	3	1
T:	1	2	1	5

PROFILE(*Motifs*)

A:	3/8	1/8	1/8	1/8
C:	1/8	3/8	3/8	1/8
G:	3/8	2/8	3/8	1/8
T:	1/8	2/8	1/8	5/8

ใช้โพรไฟล์เมทริกซ์ใหม่นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mers ทั้งหมดในสายดีเอ็นเอที่ถูกคัดออก (สายที่สี่) ได้ผลดังนี้

`cact`	`acta`	`ctaA`	`taAC`	`aACG`	`ACGA`	`CGAg`
$15/8^4$	$9/8^4$	$2/8^4$	$1/8^4$	$9/8^4$	$27/8^4$	$2/8^4$

นำดีเอ็นเอสายที่สี่กลับเข้ามาและ 4-mer **ACGA** จะถูกเลือกแทน 4-mer `acta` เดิมที่ถูกเลือกแบบสุ่มจากรอบที่ผ่านมาดังต่อไปนี้

Dna

ttACCTtaac
gATGTctgtc
ccgGCGTtag
cactaACGAg
cgtcagAGGT

จะเห็นว่าชุดของโมทิฟ (ตัวอักษรใหญ่) ในรอบนี้มีทิศทางที่เข้าใกล้ชุดของโมทิฟจริงมากแล้ว ถ้าสมมติว่ารอบถัดไปดีเอ็นเอสายที่สองถูกคัดออก เมื่อสร้างโพรไฟล์เมทริกซ์ และคำนวณค่าความน่าจะเป็นของ 4-mers ทั้งหมดที่เป็นไปได้ในดีเอ็นเอสายที่สอง 4-mer ที่มีค่าความน่าจะเป็นสูงสุดเป็นโมทิฟจริงหรือไม่

หยุดคิด	ลองรัน `GibbsSampler()` กับโจทย์ตอนต้นของบทเรียนที่มีชุดของสายดีเอ็นยาว 600 นิวคลีโอไทด์ จำนวน 10 เส้น โดยโมทิฟมีความยาว 15-mer และรายงานผล

ถึงแม้ `GibbsSampler()` จะให้ผลลัพธ์ที่ดีในหลายกรณี แนวคิดและวิธีการแก้ปัญหาผ่าน `Gibbs Sampler()` จะพิจารณาคำตอบที่เป็นไปได้แค่บางส่วน จึงมีข้อจำกัดในเรื่องของคำตอบโดยอาจไม่ใช่คำตอบที่ ดีที่สุดเนื่องจากติดค่าต่ำสุดเฉพาะที่ (local minimum) ด้วยเหตุผลนี้ การใช้งาน `GibbsSampler()`ควรมีการรันจำนวนครั้งมากๆ ด้วยหวังว่าผลการรันในครั้งใดครั้งหนึ่งอาจได้คำตอบที่ดีที่สุด

บทส่งท้าย

เชื้อวัณโรคที่อาศัยอยู่ในเซลล์ตัวให้อาศัย (host) หลบเลี่ยงจากยาปฏิชีวนะได้อย่างไร

โรควัณโรค หรือ Tuberculosis (TB) เป็นโรคติดเชื้อ (infectious disease) โดยเชื้อก่อโรคคือ *Mycobacterium tuberculosis* (MTB) และเป็นโรคที่ทำให้มีผู้เสียชีวิตเป็นหลักล้านคนในแต่ละปี ถึงแม้ในระยะหลังความรุนแรงและการระบาดของโรคลดลงเนื่องจากมียาปฏิชีวนะที่มีประสิทธิภาพ อย่างไรก็ตามก็มีสายพันธุ์ MTB ใหม่ๆ ที่ดื้อยา ในความเป็นจริงแล้ว MTB เป็นเชื้อก่อโรคที่ประสบความสำเร็จในการอยู่ร่วมกับมนุษย์มาหลายทศวรรษ โดยมีการประมาณการว่าประชากรจำนวน 1 ใน 3 ของโลกมีเชื้อ MTB อยู่ในร่างกายแต่ไม่แสดงออก (latent MTB infection) ซึ่งสำหรับหลายๆ คนเชื้อ MTB อาจไม่ก่อโรคเลย ในขณะที่ผู้ป่วยวัณโรคเป็นกลุ่มที่เชื้อมีการทำงาน ด้วยเหตุนี้นักชีววิทยาและนักวิทยาศาสตร์มีความสนใจเป็นอย่างมากว่าอะไรเป็นตัวกระตุ้นให้เชื้อ MTB ที่แอบซ่อนอยู่ในร่างกายเริ่มการทำงาน

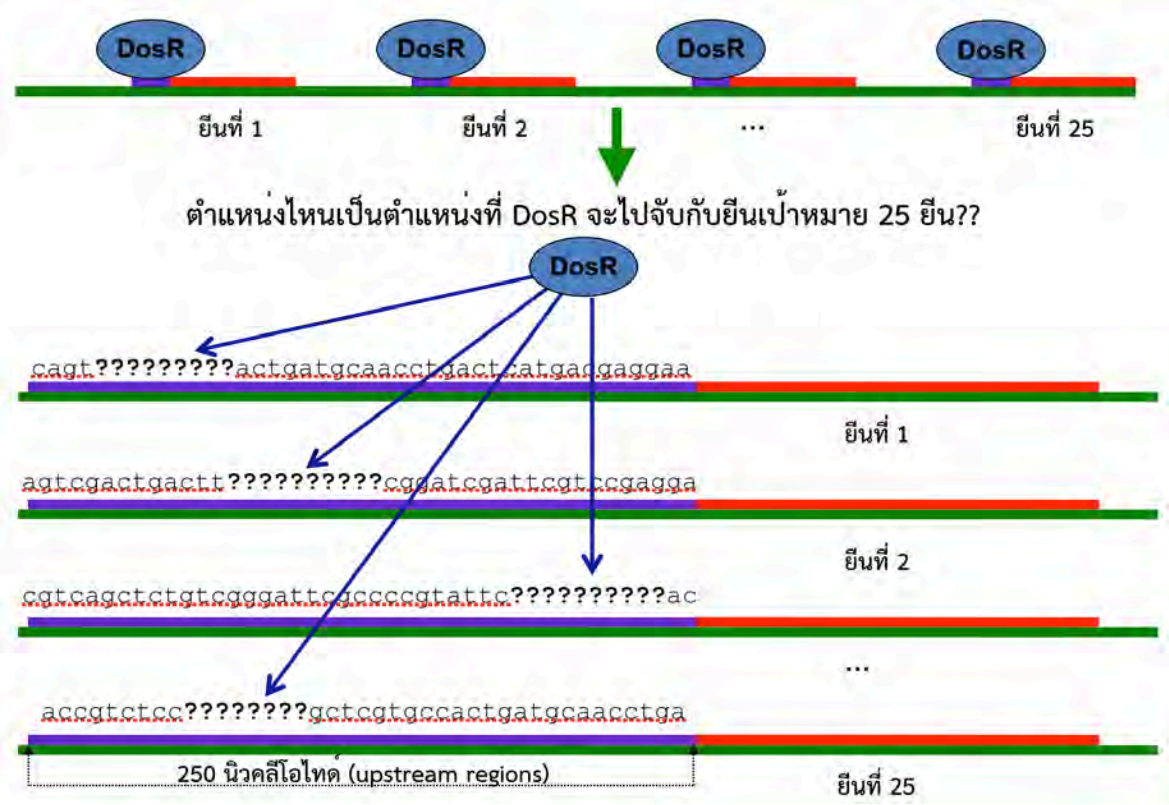
มีการรายงานว่าภาวะพร่องออกซิเจน (Hypoxia) มีความเกี่ยวข้องกับ MTB ที่ไม่แสดงออก โดยนักชีววิทยาพบว่า MTB จะไม่ทำงานในภาวะพร่องออกซิเจน โดยจะทำงานอีกครั้งและแพร่เชื้อเมื่อปอดของผู้ป่วยฟื้นฟูขึ้น เนื่องจากเชื้อ MTB มีความสามารถในการอยู่รอดเป็นเวลาหลายปีในภาวะพร่องออกซิเจน จึงมีความสนใจในการค้นหาว่ามียีนใดบ้างใน MTB ที่เกี่ยวข้องหรือควบคุมให้เกิดสภาวะที่ไม่แสดงออกของเชื้อ โดยนักชีววิทยาต้องการค้นหาแฟกเตอร์ถอดรหัสที่สามารถรับสัญญาณความพร่องของออกซิเจน และควบคุมการแสดงออกของยีนอื่นๆ เพื่อให้ปรับตัวและพร้อมเข้าสู่สภาวะไม่แสดงออก

ในปี ค.ศ. 2003 นักชีววิทยาพบว่ายีน Dormancy survival regulator (*DosR*) เป็นแฟกเตอร์ถอดรหัสที่ควบคุมการแสดงออกของยีนอื่นๆ หลายยีนในภาวะพร่องออกซิเจน อย่างไรก็ตามในการศึกษานั้นยังไม่ทราบว่า DosR ควบคุมการแสดงออกของยีนเหล่านั้นอย่างไร รวมทั้งไม่ทราบบริเวณที่เป็นตำแหน่งจับ (binding site) ของ DosR ในกลุ่มยีนเป้าหมาย เพื่อตอบคำถามนี้นักชีววิทยาได้ทำการทดลองโดยใช้ดีเอ็นเออาร์เรย์ และพบ 25 ยีนมีการเปลี่ยนแปลงการแสดงออกอย่างชัดเจนในสภาวะพร่องออกซิเจน ในส่วนของงานทางชีวสารสนเทศ ลำดับเบสส่วนหน้ายาว 250 นิวคลีโอไทด์ของ 25 ยีน (รูปที่ 4.11) ถูกนำมาเป็นข้อมูลเข้าเพื่อหาตำแหน่งจับของ DosR

เพื่อให้สามารถรันอัลกอริทึมได้เร็วขึ้น ลำดับเบสส่วนหน้าถูกเลือกมาวิเคราะห์เพียง 10 เส้น ทั้งนี้ในการหาโมทิฟของโจทย์นี้ไม่มีความรู้มาก่อนว่าความยาวของโมทิฟควรเป็นเท่าใด รูปที่ 4.12 แสดงผลการทำงานของ

`MedianString()` และ `RandomizedMotifSearch()` โดยมีการลองเปลี่ยนค่าความยาวของโมทิฟในช่วง 8-12 นิวคลีโอไทด์

หยุดคิด	จากตัวอย่างผลการทำงานของ `MedianString()` เราสามารถระบุตำแหน่งจับของ DosR ได้หรือไม่ และคิดว่าความยาวของโมทิฟที่ถูกต้องเป็นเท่าใด



รูปที่ 4.11 โจทย์ทางชีววิทยาที่ต้องการหาตำแหน่งจับของแฟกเตอร์ถอดรหัส DosR ในลำดับเบสส่วนหน้าของยีนเป้าหมาย 25 ยีนในเชื้อ MTB

	MedianString	Score	RandomizedMotifSearch	Score
k=8:	CATCGGCC	11	CCGACGGG	13
k=9:	GGCGGGGAC	16	CCATCGGCC	16
k=10:	GGTGGCCACC	19	CCATCGGCCC	21
k=11:	GGACTTCCGGC	20	ACCTTCGGCCC	25
k=12:	GGACTTCCGGCC	23	GGACCAACGGCC	28

รูปที่ 4.12 ผลการทำงานของ `MedianString()` และ `RandomizedMotifSearch()` จากลำดับเบสส่วนหน้าของยีนเป้าหมาย 10 ยีนของแฟกเตอร์ถอดรหัส DosR

ถึงแม้ผลของสายอักขระเสียงข้างมากที่ได้จาก `RandomizedMotifSearch()` จะมีความแปรผันและคะแนนมากกว่า `MedianString()` อย่างไรก็ตาม `MedianString()` ใช้เวลาในการประมวลผลนานกว่ามาก ดังนั้น `RandomizedMotifSearch()` มีประโยชน์ในการหาโมทิฟที่มีขนาดยาวเพราะใช้เวลาในการประมวลผลน้อยกว่าโดย `RandomizedMotifSearch()`สามารถหาโมทิฟ **CGGGACCTACGTCCCTAGCC** ที่เป็นอักขระเสียงข้างมากยาว 20 นิวคลีโอไทด์ (ได้คะแนน 57) ซึ่งโมทิฟ 20-mer นี้ครอบคลุมโมทิฟ 12-mer ที่เป็นอักขระเสียงข้างมากของทั้ง `MedianString()` และ `Randomized MotifSearch()`

GGACTTCCGGCC
CGGGACCTACGTCCCTAGCC
GGACCAACGGCC

และถ้ามีการรัน `GibbsSampler()` โดยใช้ *N*=200 จะได้สายอักขระเสียงข้างมากเส้นเดียวกับที่รันโดยใช้ `RandomizedMotifSearch()` ข้างต้น โดยมีชุดของโมทิฟที่นำมาสร้างโมทิฟเมทริกซ์แตกต่างกันและมีคะแนนที่น้อยกว่าคือ 55 จะเห็นว่าอัลกอริทึมในการหาโมทิฟที่แตกต่างกันให้ผลลัพธ์ที่ไม่เหมือนกัน สำหรับการหาโมทิฟของ DosR ข้างต้น ยังไม่มีคำตอบที่ชัดเจนว่าตำแหน่งจับจริงของ DosR มีรูปแบบอย่างไร นอกจากนี้ยังมีคำถามที่น่าสนใจว่า ถ้ามีโพรไฟล์เมทริกซ์ที่เป็นตัวแทนของตำแหน่งจับที่ถูกต้องแล้วจะสามารถหายีนเป้าหมายอื่นๆ ที่อาจถูกควบคุมโดยแฟกเตอร์ถอดรหัสเดียวกันได้หรือไม่ อย่างไร

ความท้าทายของการหาโมทิฟ

การหาโมทิฟที่ถูกต้องจะยากและซับซ้อนถ้าสายดีเอ็นเอแต่ละเส้นมีความถี่ของนิวคลีโอไทด์แบบใดแบบหนึ่งมากผิดปกติ วิธีการค้นหาชุด k-mers โดยให้ได้คะแนนรวมน้อยสุดจะใช้ไม่ได้ผล ตัวอย่างเช่น ถ้าดีเอ็นเอแต่ละเส้นมีนิวคลีโอไทด์ A มากเป็นพิเศษ จากอัลกอริทึมข้างต้น k-mer **AA...AA** มีโอกาสที่จะเป็นโมทิฟที่มีคะแนนรวมน้อยสุดและไม่สามารถหาโมทิฟที่เป็นคำตอบที่ถูกต้องอย่าง **GCCG** ได้เนื่องจาก aaaa มีคะแนนรวมน้อยกว่า ดังตัวอย่างต่อไปนี้

Dna
t**aaaa**GTCGa
acGCTG**aaaa**
aaaaGCCTat
aCCCGa**ata**a
ag**aaaa**GGCG

แนวทางแก้ปัญหานี้ สามารถทำได้โดยการใช้เอนโทรปีสัมพัทธ์ (relative entropy) ซึ่งจะกล่าวถึงในหัวข้อถัดไป นอกจากปัญหานี้แล้วอีกปัญหาหนึ่งคือมีโมทิฟจำนวนมากที่มีการใช้อักษรอื่นเพิ่มเติมในการแสดงผลเพื่อให้ได้ข้อมูลครบถ้วนกว่า เช่น ใช้ **W** ในการบอกว่าตำแหน่งนี้เป็น **A** หรือ **T** ก็ได้ ใช้ **S** ในการบอกว่าตำแหน่งนี้เป็น **G**

หรือ **C** ก็ได้ ใช้ **K** ในการบอกว่าตำแหน่งนี้เป็น **G** หรือ **T** ก็ได้ และใช้ **Y** ในการบอกว่าตำแหน่งนี้เป็น **C** หรือ **T** ก็ได้ เป็นต้น ลำดับอักขระอย่าง **CSKWYWWATKWATYYK** ซึ่งแสดงโมทิฟของ CSRE ในยีสต์ที่มีการกล่าวถึงในช่วงต้นของบทเรียนนี้ จะค้นหาไม่พบโดยวิธีการที่อธิบายในบทเรียนนี้

เอนโทรปีสัมพัทธ์

ถ้ามีข้อมูลเข้าเป็นชุดของสายดีเอ็นเอ เรากำหนดเอนโทรปีสัมพัทธ์ของโพรไฟล์เมทริกซ์ขนาด 4 x k ด้วยสมการต่อไปนี้

$$\sum_{j=1}^{k} \sum_{r\in\{A,C,T,G\}} p_{r,j} \cdot \log_2(p_{r,j}\,/b_r) =$$

$$\sum_{j=1}^{k} \sum_{r\in\{A,C,T,G\}} p_{r,j} \cdot \log_2(p_{r,j}) - \sum_{j=1}^{k} \sum_{r\in\{A,C,T,G\}} p_{r,j} \cdot \log_2(b_r)$$

โดยที่ b_r เป็นความถี่ของนิวคลีโอไทด์ r ในชุดของสายดีเอ็นเอ ทั้งนี้ในหัวข้อก่อนหน้าเราพยายามหาผลรวมเอนโทรปีที่น้อยที่สุด แต่ในกรณีของเอนโทรปีสัมพัทธ์นี้เราต้องการหาผลรวมที่มีค่ามากสุด โดยพจน์ต่อไปนี้ เรียกว่าครอสเอนโทรปี (cross-entropy) ของโพรไฟล์เมทริกซ์ และเอนโทรปีสัมพัทธ์คือผลต่างระหว่างค่าครอสเอนโทรปีและค่าเอนโทรปี

$$-\sum_{j=1}^{k} \sum_{r\in\{A,C,T,G\}} p_{r,j} \cdot \log_2(b_r)$$

ค่าเอนโทรปีสัมพัทธ์ของโมทิฟ **GCCG** ข้างต้นคือ 9.85-3.53 = 6.32 ดังแสดงต่อไปนี้ โดยในตัวอย่างนี้กำหนด b_A = 0.5, b_C = 0.18, b_G = 0.2 และ b_T = 0.12 ตามลำดับ

Motifs	G	T	C	G
	G	C	T	G
	G	C	C	T
	c	C	C	G
	G	G	C	G

PROFILE *(Motifs)*				
A:	0.0	0.0	0.0	0.0
C:	0.2	0.6	0.8	0.0
G:	0.8	0.2	0.0	0.8
T:	0.0	0.2	0.2	0.2

Entropy	0.72+ 1.37+ 0.72+ 0.72 = 3.53
Cross-entropy	2.35+ 2.56+ 2.47+ 2.47 = 9.85

สำหรับ k-mer **aaaa** ที่มีความอนุรักษ์มากกว่า ถึงแม้มีคะแนนรวมของโมทิฟน้อยกว่า แต่ไม่มีความหมายในทางชีววิทยา และมีค่าเอนโทรปีสัมพัทธ์เท่ากับ 4.18-0.72 = 3.46 ดังแสดงต่อไปนี้ การใช้เอนโทรปีสัมพัทธ์ทำให้เราสามารถหาโมทิฟที่ถูกต้องมากขึ้นได้

```
                a  a  a  a
                a  a  a  a
Motifs          a  a  a  a
                a  t  a  a
                a  a  a  a

              A: 1.0  0.8  1.0  1.0
              C: 0.0  0.0  0.0  0.0
PROFILE(Motifs)
              G: 0.0  0.0  0.0  0.0
              T: 0.0  0.2  0.0  0.0

      Entropy  0.0+ 0.72+ 0.0+ 0.0 = 0.72
Cross-entropy  0.94+ 1.36+ 0.94+ 0.94 =4.18
```

Position Weight Matrix

Position Weight Matrix (PWM) หรือ Position-Specific Scoring Matrix (PSSM) เป็นแบบจำลองที่ใช้ในการแสดงตำแหน่งจับของแฟกเตอร์ถอดรหัสหนึ่งๆ แทนการใช้สายอักขระเสียงข้างมาก (consensus string) แกรี่ สตอร์โม (Gary Stormo) และคณะนำเสนอ PWM เป็นครั้งแรก ในปี ค.ศ. 1982 [88] เพื่อใช้เป็นแบบจำลองแสดงบริเวณในสายอาร์เอ็นเอที่เป็นตำแหน่งเริ่มต้นของการแปลรหัสไปเป็นโปรตีน (translation start site) ในเชื้ออีโคไล (*E. coli*) รวมทั้งเป็นเครื่องมือในการกราดตรวจว่ายีนอื่นๆ ที่ไม่ได้ถูกพิจารณามีจุดเริ่มของการแปลรหัสจากอาร์เอ็นเอไปเป็นโปรตีนที่บริเวณไหน

การสร้าง PWM (รูปที่ 4.13) เริ่มจากนำชุดของโมทิฟมาสร้างเมทริกซ์ความถี่ของแต่ละตำแหน่ง (Position Frequency Matrix: PFM) ซึ่งเท่ากับ **COUNT(*Motifs*)** ในบทเรียนนี้ จากนั้นแปลงเป็นเมทริกซ์ความน่าจะเป็น (Position Probability Matrix: PPM) ซึ่งเท่ากับ **PROFILE(*Motifs*)** และแปลงจาก PPM ไปเป็น PWM โดยค่าในแต่ละช่องของ PWM คำนวณจากสมการต่อไปนี้ $PWM_{r,j} = \log_2(PPM_{r,j}/b_r)$ โดยที่ค่าโดยปริยาย (default value) ของ b_r เท่ากับ $1/|r|$ โดย $|r|$ คือจำนวนอักขระทั้งหมดที่เป็นไปได้ ในกรณีของดีเอ็นเอ $b_r = 1/|r| = ¼ = 0.25$ ในกรณีของกรดแอมิโน $b_r = 1/|r| = 1/20 = 0.05$ ทั้งนี้ค่า b_r โดยปริยายนี้อยู่บนสมมติฐานว่าโอกาสในการเกิดนิวคลีโอไทด์แต่ละแบบในสายดีเอ็นเอหนึ่งๆ หรือโอกาสในการเกิดกรดแอมิโนแต่ละแบบในสายโปรตีนหนึ่งๆ มีค่าใกล้เคียงกัน ถ้ามีนิวคลีโอไทด์หรือกรดแอมิโนบางแบบปรากฏมากเป็นพิเศษค่า b_r สามารถคำนวณได้โดยใช้ตัวอย่างในหัวข้อเอนโทรปีสัมพัทธ์ข้างต้น

Motifs

GAGGTAAAC
TCCGTAAGT
CAGGTTGGA
ACAGTCAGT
TAGGTCATT
TAGGTACTG
ATGGTAACT
CAGGTATAC
TGTGTGAGT
AAGGTAAGT

PFM
COUNT(*Motifs*)

A:	3	6	1	0	0	6	7	2	1
C:	2	2	1	0	0	2	1	1	2
G:	1	1	7	10	0	1	1	5	1
T:	4	1	1	0	10	1	1	2	6

PPM
PROFILE(*Motifs*)

A:	0.3	0.6	0.1	0.0	0.0	0.6	0.7	0.2	0.1
C:	0.2	0.2	0.1	0.0	0.0	0.2	0.1	0.1	0.2
G:	0.1	0.1	0.7	1.0	0.0	0.1	0.1	0.5	0.1
T:	0.4	0.1	0.1	0.0	1.0	0.1	0.2	0.2	0.6

PWM

A:	0.26	1.26	-1.32	-inf	-inf	1.26	1.49	-0.32	-1.32
C:	-0.32	-0.32	-1.32	-inf	-inf	-0.32	-1.32	-1.32	-0.32
G:	-1.32	-1.32	1.49	2.0	-inf	-1.32	-1.32	1.0	-1.32
T:	0.68	-1.32	-1.32	-inf	2.0	-1.32	-1.32	-.032	1.26

รูปที่ 4.13 ขั้นตอนการสร้าง Position Weight Matrix (PWM) จากชุดของโมทิฟ

(*ที่มา*: ดัดแปลงจาก *Position weight matrix*. [ONLINE] Available at: https://en.wikipedia.org [เข้าถึงออนไลน์เมื่อวันที่ 11 ก.ค. พ.ศ. 2564])

เราสามารถนำ PWM ไปกราดตรวจหาโมทิฟในลำดับเบสส่วนหน้าของยีนอื่นๆ หรือยีนทั้งจีโนมเพื่อดูว่ามีโมทิฟที่สนใจแทรกอยู่ในลำดับเบสส่วนหน้าของยีนเหล่านั้นหรือไม่ การกราดตรวจทำได้โดยนำ PWM ไปทาบกับลำดับเบสส่วนหน้า ทีละ *k* เบส (ตัวอักษรสีแดง) ตามตัวอย่างต่อไปนี้ และหาผลบวกของแต่ละเบสตามตำแหน่งที่ปรากฏใน PWM เช่น

```
ACCGTAAGTTTCAGAGATTACAG… = 0.26-0.32-1.32+2+2+1.26+1.49+1+1.26 = 7.63
ACCGTAAGTTTCAGAGATTACAG… = -0.32-.032+1.49-inf-inf… = -inf
```

ลำดับเบส **ACCGTAAGT** มีคะแนนมากกว่าลำดับเบส **CCGTAAGTT** ซึ่งหมายถึงลำดับเบส **ACCGTAAGT** มีโอกาสเป็นโมทิฟที่กำลังค้นหามากกว่า

การหาดีเอ็นเอโมทิฟในบทเรียนนี้สามารถนำไปประยุกต์ใช้กับการแก้ปัญหาอื่นๆ ที่เกี่ยวข้อง เช่น การหาตำแหน่งจับของไมโครอาร์เอ็นเอจำเพาะหนึ่งๆ บนชุดของดีเอ็นเอหรืออาร์เอ็นเอเป้าหมาย การหาตำแหน่งจับของโปรตีนบนชุดอาร์เอ็นเอเป้าหมาย เป็นต้น โดยผู้เขียนและคณะนำเสนอผลงานวิจัยชื่อ microlive [89] เพื่อใช้ออก

แบบไวรัสวัคซีนเชื้อเป็นที่ปลอดภัยมากขึ้น งานวิจัยนี้อ้างอิงกระบวนการพื้นฐานของสิ่งมีชีวิตทั้งพืชและสัตว์ที่มีการแสดงออกของไมโครอาร์เอ็นเอเพื่อยับยั้งการแปลรหัสหรือการทำให้เอ็มอาร์เอ็นเอสลายตัว (mRNA degradation) โดยเข้าจับบริเวณจำเพาะของสายเอ็มอาร์เอ็นเอ [90] และได้แนวคิดจากงานวิจัยก่อนหน้าที่มีการสร้างวัคซีนเชื้อเป็นของไวรัสไข้หวัดใหญ่ A (influenza A virus) ที่ลดความรุนแรงของเชื้อโดยการเพิ่มบริเวณที่ไมโครอาร์เอ็นเอของเซลล์ผู้ให้อาศัย (host) สามารถเข้าจับ ที่เรียกว่า microRNA response element (MRE) เข้าไปเป็นส่วนหนึ่งของสายโออาร์เอฟของนิวคลีโอโปรตีน (nucleoprotein) ของไวรัส ซึ่งผลการทดสอบวัคซีนเชื้อเป็น H1N1 และ H5N1 ที่มีการเติม MRE พบว่ามีความปลอดภัยมากขึ้น [91] โดย MRE นี้เป็นตัวอย่างของอาร์เอ็นเอโมทิฟ เพื่อให้การออกแบบวัคซีนเชื้อเป็นที่ใช้ไมโครอาร์เอ็นเอของเซลล์ตัวให้อาศัยเป็นตัวควบคุมสามารถทำได้รวดเร็วและเป็นอัตโนมัติ ผู้เขียนและคณะจึงออกแบบและพัฒนา microlive โดยเน้นการออกแบบวัคซีนเชื้อเป็นสำหรับมนุษย์ ซึ่งมีขั้นตอนวิจัยหลักประกอบด้วย รวบรวมสายอาร์เอ็นเอไวรัสจากเอ็นซีบีไอและชุดไมโครอาร์เอ็นเอของมนุษย์จาก miRBase [92] หาบริเวณจับที่เป็นไปได้ของแต่ละไมโครอาร์เอ็นเอบนสายของอาร์เอ็นเอไวรัสโดยใช้โปรแกรม miRanda [93, 94] โดยบริเวณเหล่านี้ถูกกำหนดเป็นชุด MRE ตั้งต้น ทำการเปลี่ยนลำดับเบสในตำแหน่งที่ไม่จับของแต่ละ MRE และทดสอบความสามารถในการจับโดยใช้โปรแกรม miRanda อีกครั้งพร้อมทั้งจัดลำดับคะแนนความสามารถในการจับระหว่างคู่ของไมโครอาร์เอ็นเอและ MRE ที่มีการเปลี่ยนลำดับเบส จากนั้นวัดผลกระทบของการเปลี่ยนลำดับเบสต่อการแปลรหัสไปเป็นลำดับกรดแอมิโนและจัดลำดับ MRE อีกครั้ง ทั้งนี้ผู้ใช้สามารถนำเข้าสายอาร์เอ็นเอไวรัสที่สนใจเป็นข้อมูลเข้าเพื่อให้ microlive หาชุดของ MRE หรือสามารถสืบค้น MRE ที่ระบบได้ทำการคำนวณและจัดลำดับไว้ล่วงหน้าของอาร์เอ็นเอไวรัส 7 ชนิดประกอบด้วยไวรัสไข้หวัดใหญ่ A (influenza A), เดงกี (dengue) ที่ก่อให้เกิดโรคไข้เลือดออก, ตับอักเสบซี (hepatitis C), หัด (measles), คางทูม (mumps), โปลิโอ (polio) และพิษสุนัขบ้า (rabies)

ตัวอย่างโปรแกรมค้นหาโมทิฟที่มีการใช้งานอย่างแพร่หลาย

MEME [95, 96] เป็นชุดของเครื่องมือที่ช่วยในการวิเคราะห์โมทิฟของทั้งดีเอ็นเอ อาร์เอ็นเอ และโปรตีน โดยชุดของเครื่องมือสามารถแบ่งออกเป็น 4 กลุ่มหลักประกอบด้วย (1) ชุดเครื่องมือที่ใช้ในการหา *de novo* โมทิฟหรือโมทิฟใหม่ซึ่งมีเป้าหมายเดียวกับเนื้อหาหลักของบทเรียนนี้ (2) ชุดเครื่องมือทางสถิติที่ใช้โมทิฟที่อยู่ในฐานข้อมูลที่มีการรายงานมาก่อนมาทดสอบการปรากฏของโมทิฟเหล่านี้อย่างมีนัยสำคัญในชุดของข้อมูลเข้าจากผู้ใช้ (3) เครื่องมือที่ใช้ในการค้นหาโมทิฟโดยนำ PWM ไปกราดตรวจลำดับเบสส่วนหน้าของยีนต่างๆ โดยแต่ละเครื่องมือในชุดนี้มีเป้าหมายแตกต่างกันไป เช่น อัลกอริทึม MAST รับข้อมูลเข้าเป็นชุดของโมทิฟจากผู้ใช้งานและฐานข้อมูลที่ผู้ใช้เลือก โดยผลของการทำงานคือการให้คะแนนแต่ละสายข้อมูลในฐานข้อมูลที่ถูกเลือกตามการปรากฏของชุดของโมทิฟที่เป็นข้อมูลเข้า ในขณะที่อัลกอริทึม MCAST เหมาะกับการใช้กราดตรวจจีโนมโดยเน้นการหาชุดของตำแหน่งจับหรือมอดูลควบคุมแบบซิส (*cis*-regulatory module: CRM) ซึ่งอาจถูกจับโดยชุดของแฟกเตอร์ถอดรหัสที่มีการรายงานมาก่อน เป็นต้น (4) เครื่องมือเพื่อการเปรียบเทียบโมทิฟใหม่ (*de novo* motif) ที่หาได้โดย

MEME กับที่หาได้จากเครื่องมืออื่นๆ นอกจากชุดเครื่องมือ MEME ที่มีการใช้งานอย่างแพร่หลายและมีการพัฒนาเพิ่มเติมอย่างต่อเนื่องแล้ว ยังมีผลงานวิจัยอื่นๆ ที่มีแนวทางในการหาโมทิฟแตกต่างกันไปตามตัวอย่างที่กล่าวถึงในบทปริทัศน์ [97] ผลการเปรียบเทียบประสิทธิภาพของบางเครื่องมือเหล่านี้สามารถศึกษาเพิ่มเติมได้จาก [98] เป็นต้น

ตัวอย่างฐานข้อมูลโมทิฟ

ตัวอย่างฐานข้อมูลโมทิฟ เช่น JASPAR (http://jaspar.genereg.net/) [99-107] เป็นฐานข้อมูลตำแหน่งจับของแฟกเตอร์ถอดรหัสในรูปแบบ Position Frequency Matrices หรือ PFM ที่สามารถนำไปสร้างเป็น Position Weight Matrices (PWM) ต่อได้ โดยรวบรวมข้อมูลจากผลการทดลองแบบต่างๆ จากห้องปฏิบัติการ ข้อมูลในฐานข้อมูลมาจากสิ่งมีชีวิตที่หลากหลายทั้งสัตว์เลี้ยงลูกด้วยนม สัตว์มีกระดูกสันหลัง พืช และแมลง เป็นต้น โดยมีการปรับเพิ่มข้อมูลอย่างต่อเนื่องจนปัจจุบัน CIS-BP (http://cisbp.ccbr.utoronto.ca/) [108] เป็นฐานข้อมูลตำแหน่งจับของโปรตีนบนสายดีเอ็นเอในรูปแบบ Position Weight Matrices (PWM) โดยข้อมูลจำนวนมากเป็นสิ่งมีชีวิตกลุ่มยูแคริโอต ฐานข้อมูล UniPROBE (http://thebrain.bwh.harvard.edu/uniprobe/) [109] เน้นการเก็บข้อมูลตำแหน่งจับจากการทดลอง protein binding microarray (PBM) โดยเน้นการวัดการจับของชุดโปรตีนที่สกัดจากสิ่งมีชีวิตต่างๆ ทั้งกลุ่มที่เป็นโพรแคริโอตเช่น เชื้อแบคทีเรีย *Vibrio harveyi* และกลุ่มยูแคริโอตเช่น ยีสต์ (*Saccharomyces cerevisiae*) หนอนตัวกลม (*Caenorhabditis elegans*) หนูและมนุษย์ เป็นต้น โดยข้อมูลอยู่ในรูปแบบ PWM ฐานข้อมูล TFBSshape [110] (https://tfbsshape.usc.edu/) เป็นผลงานวิจัยที่ได้รับการจัดกลุ่มโดยวารสาร Nucleic Acid Research ให้อยู่ในกลุ่มที่ช่วยทำให้เกิดความก้าวหน้าอย่างมาก (breakthrough) ฐานข้อมูล TFBSshape เก็บข้อมูลคุณลักษณะเชิงโครงสร้างของสายดีเอ็นเอที่เป็นผลการทำนายจากอัลกอริทึมที่ผู้วิจัยพัฒนามาก่อนหน้า โดยใช้ข้อมูลตำแหน่งจับจากฐานข้อมูล JASPAR และ UniPROBE เป็นข้อมูลเข้า ข้อมูลคุณลักษณะเชิงโครงสร้างนี้สามารถนำไปวิเคราะห์ความจำเพาะในการจับของแฟกเตอร์ถอดรหัสได้ละเอียดขึ้น

ฐานข้อมูล TRANSFAC [111-113] เป็นฐานข้อมูลหลักในการเก็บข้อมูลของแฟกเตอร์ถอดรหัสและยีนเป้าหมายในงานวิจัยสมัยแรกๆ อย่างไรก็ตามข้อมูลหลังปี ค.ศ. 2005 อยู่ในฐานข้อมูล TRANSFAC® Professional ซึ่งเป็นฐานข้อมูลปิดเชิงพาณิชย์ โดยข้อมูลก่อนปี ค.ศ. 2006 ยังเปิดเป็นสาธารณะ นอกจากฐานข้อมูลข้างต้นที่เก็บข้อมูลโมทิฟของสิ่งมีชีวิตที่หลากหลาย ยังมีฐานข้อมูลอีกกลุ่มที่เน้นการเก็บข้อมูลสิ่งมีชีวิตหรือกลุ่มของสิ่งมีชีวิตจำเพาะ เช่น ฐานข้อมูล HOCOMOCO (https://hocomoco11.autosome.ru/) [114-116] เก็บข้อมูลตำแหน่งจับของแฟกเตอร์ถอดรหัสในมนุษย์ในรูปแบบ PWM ที่เน้นความถูกต้องและคุณภาพของข้อมูล โดยในเวอร์ชันถัดๆ มามีการนำข้อมูลจากห้องปฏิบัติการมารวมด้วย ฐานข้อมูล PRODORIC (http://www.prodoric.de/) [117] เก็บข้อมูลตำแหน่งจับของสิ่งมีชีวิตกลุ่มโพรแคริโอต ข้อมูลสรุปเกี่ยวกับแต่ละฐานข้อมูลสามารถศึกษาเพิ่มเติมได้จาก [118] สำหรับเทคโนโลยีจากห้องปฏิบัติการที่ใช้ในการศึกษาการจับกันระหว่างโปรตีนกับดีเอ็นรวมทั้งผลกระทบ

ต่อการออกแบบอัลกอริทึมสามารถศึกษาเพิ่มเติมได้จาก [119, 120] ตัวอย่างเครื่องมือที่ใช้ในการแสดงตำแหน่งจับที่มีการใช้งานอย่างแพร่หลายคือ Sequence logos [121] และ WebLogo [122] รวมทั้งเครื่องมือที่ช่วยในการแปลงจาก sequence logo กลับเป็น PWM [123]

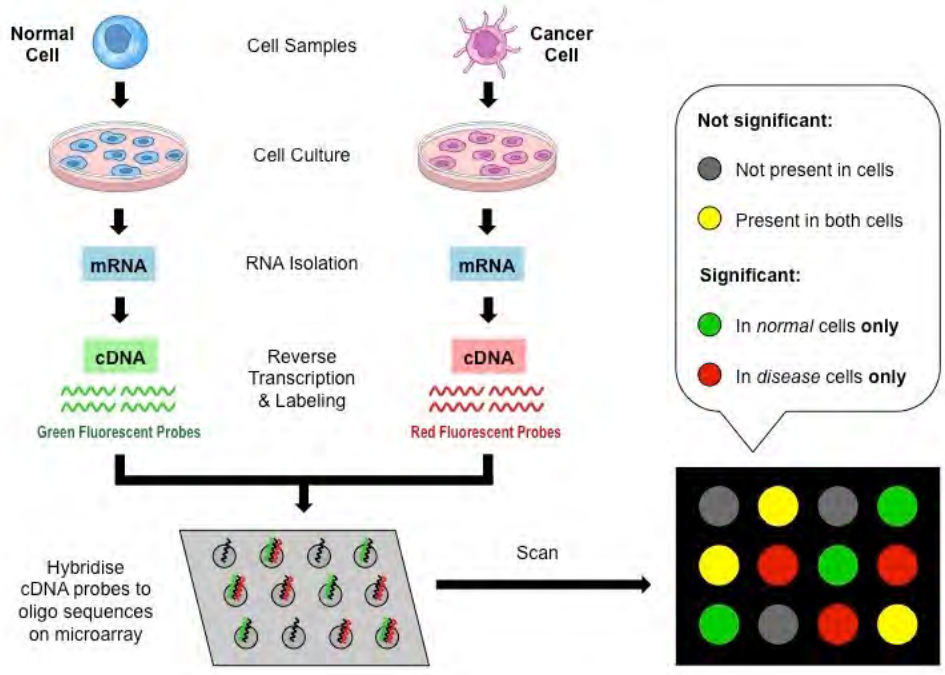
แบบฝึกหัดบทที่ 4

1. เขียนโปรแกรมเพื่อแก้ปัญหาที่เกี่ยวข้องกับการหาโมทิฟโดยใช้โจทย์ที่โรซาลินด์ (http://rosalind.info) ดังต่อไปนี้
    1) Transcribe DNA into RNA (http://rosalind.info/problems/rna/)
    2) Complementing a strand of DNA (http://rosalind.info/problems/revc/)
    3) Counting Point Mutations (http://rosalind.info/problems/hamm/)
    4) Finding a Motif in DNA (http://rosalind.info/problems/subs/)
    5) Rabbits and Recurrence Relations (http://rosalind.info/problems/fib/)
    6) Consensus and Profile (http://rosalind.info/problems/cons/)
    7) Translating RNA into Protein (http://rosalind.info/problems/prot/)
    8) Finding a Protein Motif (http://rosalind.info/problems/mprt/)
    9) Finding a Shared Motif (http://rosalind.info/problems/lcsm/)
    10) Finding a Spliced Motif (http://rosalind.info/problems/sseq/)
    11) Finding a Shared Spliced Motif (http://rosalind.info/problems/lcsq/)
    12) Implement Randomized Search (http://rosalind.info/problems/ba2f/)
    13) New Motif Discovery (http://rosalind.info/problems/meme/)
    14) Implement GibbsSampler (http://rosalind.info/problems/ba2g/)
2. อัลกอริทึมที่ใช้ในการหาโมทิฟในบทนี้สามารถนำไปประยุกต์ใช้ในการค้นหาโมทิฟในชุดของสายโปรตีนได้หรือไม่ อย่างไร
3. ถ้าเรามีโพรไฟล์เมทริกซ์ของแฟกเตอร์ถอดรหัส ABC และต้องการหาว่ายังมียีนใดอีกบ้างในจีโนมที่อาจเป็นยีนเป้าหมายของ ABC จะมีขั้นตอนในการหาอย่างไรและต้องใช้ข้อมูลเข้าอะไรบ้าง
4. จากข้อมูล Position Frequency Matrix ของ SPI1 ใน JASPAR (MA0080.4) จงสร้าง Position Weight Matrix (PWM) ตามตัวอย่าง Tang, D. 2013 ต่อไปนี้
   https://davetang.org/muse/2013/10/01/position-weight-matrix/

ภาคผนวกบทที่ 4

ดีเอ็นเออาร์เรย์

ดีเอ็นเออาร์เรย์ (DNA array) (รูปที่ 4.14) เป็นชุดของดีเอ็นเอโมเลกุลที่ถูกนำมาติดไว้กับชิปซิลิคอนหรือแผ่นแก้วที่มีลักษณะเป็นช่องๆ เหมือนตารางสองมิติ โดยในแต่ละช่องจะมีลำดับเบสดีเอ็นเอที่มีความจำเพาะเรียกว่าโพรบ (probe) ติดอยู่เพื่อใช้ในการวัดปริมาณการแสดงออกของยีนเป้าหมายหรือทาร์เกต (target) ที่จำเพาะกับโพรบนั้นๆ โดยโพรบถูกสังเคราะห์ขึ้นให้มีความจำเพาะกับแต่ละยีนในสิ่งมีชีวิตที่ต้องการนำมาทดสอบ การออกแบบการทดลองมักสกัดเอ็มอาร์เอ็นเอจากเซลล์มาตรฐานเพื่อใช้เป็นข้อมูลอ้างอิงกับเอ็มอาร์เอ็นเอของเซลล์ที่อยู่ในเงื่อนไขจำเพาะ เช่นเซลล์ของ MTB ในภาวะปกติกับเซลล์ของ MTB ในภาวะพร่องออกซิเจน โดยเอ็มอาร์เอ็นเอเหล่านี้ถูกแปลงย้อนกลับเป็นซีดีเอ็นเอ (cDNA) และติดแท็กด้วยฟลูออเรสเซนต์ที่มีสีที่แตกต่างกัน เช่น ซีดีเอ็นเอที่มาจากเซลล์ปกติติดแท็กสีเขียว ส่วนซีดีเอ็นเอที่มาจากเซลล์ที่อยู่ในเงื่อนไขที่สนใจติดแท็กสีแดง หลังจากนั้นนำซีดีเอ็นเอติดแท็กเหล่านี้ไปทดสอบการจับกับโพรบบนชิป ถ้าซีดีเอ็นเอเป็นคู่สมกับโพรบจะมีการปล่อยแสงฟลูออเรสเซนต์ออกมา โดยความเข้มของแสงขึ้นอยู่กับจำนวนซีดีเอ็นเอหรืออาร์เอ็นเอของยีนที่แสดงออก การทดลองหา evening elements ที่กล่าวถึงในตอนต้นบทเรียนนี้ ใช้ดีเอ็นเออาร์เรย์ชิปที่สามารถวัดการแสดงออกของยีนใน *Arabidopsis thaliana* ได้ 8,000 ยีนพร้อมกัน



รูปที่ 4.14 ดีเอ็นเออาร์เรย์

(*ที่มา*: Cornell, B. 2016. *DNA Microarrays*. [ONLINE] Available at: http://ib.bioninja.com.au [เข้าถึงออนไลน์เมื่อวันที่ 11 ก.ค. พ.ศ. 2564])

บทที่ 5 การเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีน (Sequence alignment)

วัตถุประสงค์

- เพื่อให้นิสิตเห็นแนวทางในการวิเคราะห์ข้อมูลแนวทางหนึ่งที่มีความสำคัญและเกี่ยวข้องกับการแก้ปัญหาทางชีววิทยาหลายปัญหาทั้งการเปรียบเทียบยีน/โปรตีนเพื่ออนุมานฟังก์ชัน การเปรียบเทียบยีน/โปรตีนเพื่อหาโดเมนหรือส่วนของสายข้อมูลที่มีความอนุรักษ์เพื่ออนุมานความสำคัญและฟังก์ชัน การเปรียบเทียบยีน/โปรตีนเพื่อการวิเคราะห์ความสัมพันธ์ในเชิงวิวัฒนาการเบื้องต้น
- เพื่อให้นิสิตคุ้นเคยกับข้อมูลและองค์ความรู้ที่เกี่ยวข้องและเข้าใจการพัฒนาการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีน
- เพื่อให้นิสิตเห็นตัวอย่างงานวิจัยรวมทั้งตัวอย่างโปรแกรมที่ใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนทั้งแบบเปรียบเทียบระหว่างคู่ของสายข้อมูล ชุดของสายข้อมูล และการสืบค้นสายข้อมูลกับฐานข้อมูลขนาดใหญ่
- เพื่อให้นิสิตเห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทาย รวมทั้งงานวิจัยอื่นๆ ที่เกี่ยวข้อง

ผลลัพธ์ที่คาดหวัง

- นิสิตเห็นที่มาของโจทย์ทางชีววิทยาที่ต้องการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นและหรือโปรตีน
- นิสิตเข้าใจคุณลักษณะของข้อมูลเข้า
- นิสิตสามารถอธิบายการทำงานของอัลกอริทึมหลักๆ ที่ใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนได้ เข้าใจความสำคัญและการใช้งานเมทริกซ์คะแนนแบบต่างๆ เช่นเมทริกซ์คะแนนแพมและบลอสซัม เป็นต้น
- นิสิตสามารถเขียนโปรแกรมเพื่อเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนได้
- นิสิตสามารถยกตัวอย่างโปรแกรมที่ใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีน

- นิสิตสามารถยกตัวอย่างความท้าทายที่ยังมีอยู่และสามารถนำเสนอแนวทางในการพัฒนาวิธีการแก้ปัญหาเหล่านี้ รวมทั้งสามารถประยุกต์องค์ความรู้จากบทเรียนเพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้

เนื้อหาโดยสรุป

การเปรียบเทียบความคล้ายคลึงระหว่างสายดีเอ็นเอและหรือโปรตีนเป็นขั้นตอนพื้นฐานขั้นตอนหนึ่งในวิธีการทางชีวสารสนเทศที่มักใช้ในการอนุมานฟังก์ชันของสายข้อมูลเข้าโดยเปรียบเทียบกับฐานข้อมูลของโปรตีน การศึกษาความเกี่ยวเนื่องกันของสายดีเอ็นเอและหรือโปรตีนนำไปสู่การอนุมานความสัมพันธ์ในเชิงวิวัฒนาการ ทั้งนี้ปัญหาการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนนี้เป็นปัญหาเชิงอัลกอริทึม เรียกรวมๆ ว่าการทำ sequence alignment ซึ่งประกอบด้วย (1) การเปรียบเทียบความคล้ายคลึงกันระหว่างสายข้อมูลสองเส้น (pair-wise alignment) โดยอัลกอริทึมพื้นฐานใช้กำหนดการพลวัต (dynamic programming) ในการขยับลำดับเบสหรือกรดแอมิโนให้ตรงกันมากที่สุดในองค์รวม (global alignment) หรือเฉพาะที่ (local alignment) (2) การเปรียบเทียบความคล้ายคลึงกันของสายข้อมูลมากกว่าสองสาย (multiple sequence alignment) โดยการทำ pair-wise alignment ระหว่างสายข้อมูลแต่ละเส้นกับสายข้อมูลเส้นอื่นทั้งหมดภายในชุด ซึ่งคู่ของสายข้อมูลที่คล้ายคลึงกันมากที่สุดจะถูกนำมารวมกันและแสดงด้วยสายข้อมูลตัวแทน และทำการเปรียบเทียบสายข้อมูลตัวแทนนี้กับสายข้อมูลเส้นที่เหลือทั้งหมดภายในชุด และวนทำซ้ำจนกว่าจะครบจำนวนสายข้อมูลภายในชุด (3) การเปรียบเทียบสายข้อมูลเข้ากับสายข้อมูลภายในฐานข้อมูลขนาดใหญ่ เช่น ฐานข้อมูลโปรตีนของเอ็นซีบีไอ และยูนิพรอต เป็นต้น โดยอัลกอริทึมหลักถูกพัฒนาในโปรแกรม BLAST [1] ที่มีการใช้งานกันอย่างแพร่หลาย

บทที่ 5 การเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีน (Sequence alignment)

ในปี ค.ศ. 1983 ดูลิตเติล (Doolittle) และคณะ [124] และวอเตอร์ฟิลด์ (Waterfield) และคณะ [125] ได้ทำการเปรียบเทียบลำดับกรดแอมิโนของยีน Platelet-derived growth factor (*PDGF*) ที่ถูกแปลรหัสออกมากับลำดับกรดแอมิโนของยีนอื่นๆ ที่มีข้อมูลอยู่ในช่วงเวลานั้น โดยผลการเปรียบเทียบทำให้นักชีววิทยาโรคมะเร็งประหลาดใจเป็นอย่างมาก เนื่องจากลำดับเบสของยีน *PDGF* มีความคล้ายคลึงกับลำดับเบสของยีน *v-sis* มาก ความคล้ายกันมากนี้เป็นเรื่องพิศวงเพราะยีนทั้งสองมีฟังก์ชันการทำงานต่างกันมาก โดยยีน *PDGF* จะแปลรหัสไปเป็นโปรตีนที่ทำงานเกี่ยวกับการกระตุ้นการเติบโตของเซลล์ ในขณะที่ยีน *v-sis* เป็นยีนก่อมะเร็ง (oncogene) หลังการค้นพบของดูลิตเติลนักวิทยาศาสตร์ได้ตั้งสมมติฐานว่ามะเร็งบางรูปแบบอาจเกิดจากยีนที่ดีแต่ทำงานในเวลาที่ไม่ถูกต้อง ความเชื่อมโยงระหว่างยีน *PDGF* และ *v-sis* สร้างกระบวนทัศน์ใหม่ในการศึกษาสายนิวคลีโอไทด์ที่ได้จากการหาลำดับเบส โดยสายนิวคลีโอไทด์จะถูกนำมาเปรียบเทียบความคล้ายคลึงกับสายดีเอ็นเอหรือโปรตีนในฐานข้อมูลเพื่อหาฟังก์ชัน และกลายเป็นขั้นตอนหลักของงานทางด้านจีโนมิกส์ในปัจจุบันโดยเฉพาะลำดับเบสเหล่านั้นอยู่ในจีโนมใหม่ เช่น ผลงานวิจัยของผู้เขียนในโครงการหาลำดับเบสจีโนมเชื้อรา [54] เมื่อทำนายบริเวณที่เป็นยีนทั้งหมดในจีโนมแล้วจะนำลำดับนิวคลีโอไทด์ของยีนที่ทำนายได้เหล่านี้ไปเทียบเคียงความคล้ายคลึงกับสายโปรตีนที่มีอยู่ในฐานข้อมูลเปิดสาธารณะผ่านโปรแกรม BLAST เช่น ฐานข้อมูล nr (non-redundant) ที่เอ็นซีบีไอซึ่งเก็บสายข้อมูลโปรตีนจำนวนมากมายของสิ่งมีชีวิตชนิดต่างๆ โดยผลของการเทียบเคียงจะถูกนำมาใช้เพื่ออนุมานฟังก์ชันของยีนที่ทำนายได้ จำนวนสายโปรตีนที่อยู่ในฐานข้อมูลโปรตีน (nr) เข้าถึงเมื่อวันที่ 2 กุมภาพันธ์ พ.ศ. 2561 มีจำนวนทั้งสิ้น 478,964,146 รายการ

ทำความรู้จักกับการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีน

การเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนในมุมมองของการเล่นเกมส์

ตัวอย่างปัญหาเริ่มต้นเป็นการเปรียบเทียบความคล้ายคลึงกันระหว่างสายข้อมูลสองเส้น โดยใช้ระยะทางแฮมมิง (Hamming distance) ซึ่งแสดงจำนวนเบสที่แตกต่างกันระหว่างสายข้อมูล โดยอาจตั้งเงื่อนไขเข้มงวดว่าต้องเทียบอักขระต่ออักขระที่อยู่ในลำดับเดียวกันระหว่างสายข้อมูลสองเส้น อย่างไรก็ตามสายดีเอ็นเอมักเกิดการเพิ่มหรือลดเบส ดังนั้นเงื่อนไขข้างต้นจึงไม่เหมาะสม และมีการปรับวัตถุประสงค์เป็นการหาอักขระในอีกสายที่ตรงกับอักขระตัวปัจจุบันในดีเอ็นเอสายแรก พิจารณาสายดีเอ็นเอ **ATGCATGG** และ **TGCATGCA** ที่ไม่มีตำแหน่งใดเลยที่มี

อักขระตรงกันและมีระยะทางแฮมมิงเท่ากับ 8 เมื่อเลื่อนสายดีเอ็นเอให้เหมาะสมเราสามารถลดระยะทางแฮมมิงเหลือ 3 ดังตัวอย่าง ต่อไปนี้

ATGCATGG-
-TGCATGCA

หรือตัวอย่างที่ดีเอ็นเอสองสายอาจมีความเหมือนในลำดับเบสย่อย เช่น

ATGC-TTA-
-TGCATTAA

ตัวอย่างข้างต้นนำไปสู่การวัดผลการเทียบดีเอ็นเอสองสายที่เหมาะสมโดยพยายามเลื่อนลำดับเบสในดีเอ็นเอสายแรกให้มีจำนวนเบสที่ตรงกับลำดับเบสในดีเอ็นเอสายที่สองให้มากที่สุด โดยวิธีการพื้นฐานเพื่อให้ได้จำนวนเบสที่ตรงกันมากที่สุดนี้สามารถทำได้โดยนำเบสแรกของดีเอ็นเอทั้งสองเส้นออกไปจากสายถ้าเป็นเบสเดียวกันและเพิ่มให้ 1 คะแนน แต่ถ้าไม่ตรงกันให้นำเบส 1 ตัวออกจากสายดีเอ็นเอเส้นใดเส้นหนึ่งเพื่อให้ได้เบสถัดไปตรงกับเบสปัจจุบันของดีเอ็นเออีกสาย และทำไปเรื่อยๆ จนกว่าดีเอ็นเอเส้นใดเส้นหนึ่งจะไม่มีเบสเหลือ โดยมีเป้าหมายให้ได้จำนวนเบสที่ตรงกันมากที่สุด

ปัญหาการหาสายอักขระย่อยร่วมที่ยาวที่สุด

เราสามารถเทียบความคล้ายคลึงกันของสายดีเอ็นเอ *v* และ *w* โดยใช้เมทริกซ์สองแถว แถวแรกเก็บลำดับเบสของ *v* และแถวที่สองเก็บลำดับเบสของ *w* โดยเรียงจากซ้ายไปขวา และอาจมี การแทรกอักขระ '-' ในตำแหน่งที่เบสไม่ตรงกัน ตัวอย่างต่อไปนี้เป็นการเลื่อนเบสให้สอดคล้องกันโดยใช้ข้อมูลสายดีเอ็น 2 เส้นคือ ATGTTATA และ ATCGTCC

v: A T – G T T A T A

w: A T C G T – C – C

จากตัวอย่างนี้แต่ละคอลัมน์ที่มีเบสตรงกันเรียกว่าแมช (match) แสดงถึงเบสที่อนุรักษ์ร่วมกันระหว่างดีเอ็นเอสองสาย สำหรับคอลัมน์ที่มีเบสต่างกันเรียกว่ามิสแมช (mismatch) และคอลัมน์ที่มี '-' แสดงการเกิดอินเดล (indel) โดยคอลัมน์ที่เป็น '-' ใน *v* แสดงถึงเกิดการสอดแทรก (insertion) ใน w ที่คอลัมน์นั้นเพื่อให้ *w* มีความคล้ายกับ *v* มากขึ้น ในขณะที่คอลัมน์ที่เป็น '-' ใน *w* แสดงถึงเกิดการขาดหาย (deletion) ของเบสนั้นใน *w* ในตัวอย่างนี้มี 4 แมช 2 มิสแมช 1 insertion และ 2 deletion แมชที่เกิดขึ้นระหว่างดีเอ็นเอสองสายเป็นตัวกำหนดส่วนของลำดับเบสที่เกิดขึ้นร่วมกัน (common subsequence) โดยไม่จำเป็นต้องอยู่ติดกันทั้งหมด ในตัวอย่างนี้ **ATGT** เป็นส่วนของลำดับเบสที่เกิดร่วมกันระหว่าง **ATGT**TATA และ **AT**C**GT**CC และเนื่องจากการขยับเบสระหว่างสายดีเอ็นเอมีเป้าหมายเพื่อให้ได้จำนวนเบสที่ตรงกันมากที่สุด ดังนั้น **ATGT** ซึ่งเป็นผลจากการขยับเบสข้างต้นจึงเป็นตัวแทนของส่วนของลำดับเบสที่เกิดขึ้นร่วมกันยาวที่สุด (longest common substring) ด้วย ทั้งนี้คู่ของสายดีเอ็นเอใดๆ อาจมีส่วนของลำดับเบสที่เกิดขึ้นร่วมกันที่ยาวที่สุดมากกว่า 1 เส้น

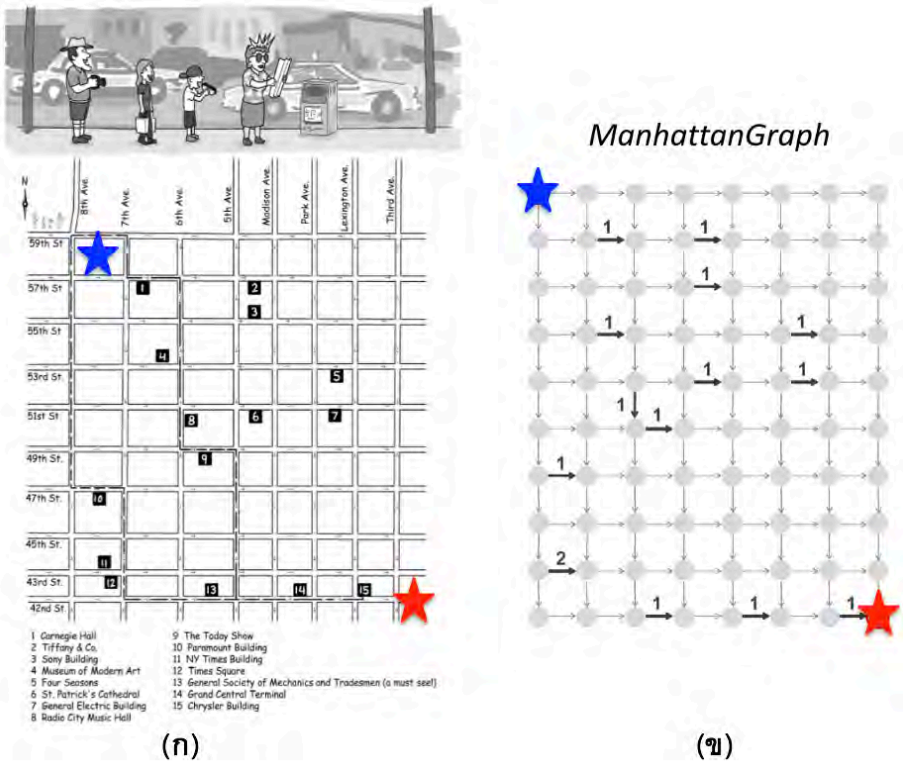
นิยามปัญหาที่ 5.1 ปัญหาการหาสายอักขระย่อยร่วมที่ยาวที่สุด

ปัญหาการหาสายอักขระย่อยร่วมที่ยาวที่สุด หาส่วนของสายอักขระที่เกิดขึ้นร่วมกันที่ยาวที่สุดระหว่างสายอักขระสองเส้น	
ข้อมูลเข้า	สายอักขระ 2 เส้น
ผลลัพธ์	ส่วนของสายอักขระที่เกิดขึ้นร่วมกันที่ยาวที่สุด

ปัญหาการหาเส้นทางเดินชมเมืองแมนฮัตตัน

วางแผนการเดินชมเมืองอย่างไรให้ผ่านจุดท่องเที่ยวมากที่สุด

การหาวิธีการเดินชมเมืองแมนฮัตตันให้ผ่านจุดท่องเที่ยวมากที่สุดนี้เรียกว่า Manhattan Tourist Problem โดยแผนที่ของเมืองสามารถแสดงได้โดยกราฟแบบมีทิศทาง (directed graph) เรียกว่า *Manhattan Graph* โดยทางแยกต่างๆ ถูกแสดงด้วยโหนด และเส้นเชื่อมระหว่างโหนดมีค่าน้ำหนักซึ่งแสดงจำนวนจุดท่องเที่ยวที่อยู่ในเส้นทางเดินนั้น สำหรับเส้นเชื่อมที่ไม่มีค่าหมายถึงไม่มีจุดท่องเที่ยวในบล็อกทางเดิน



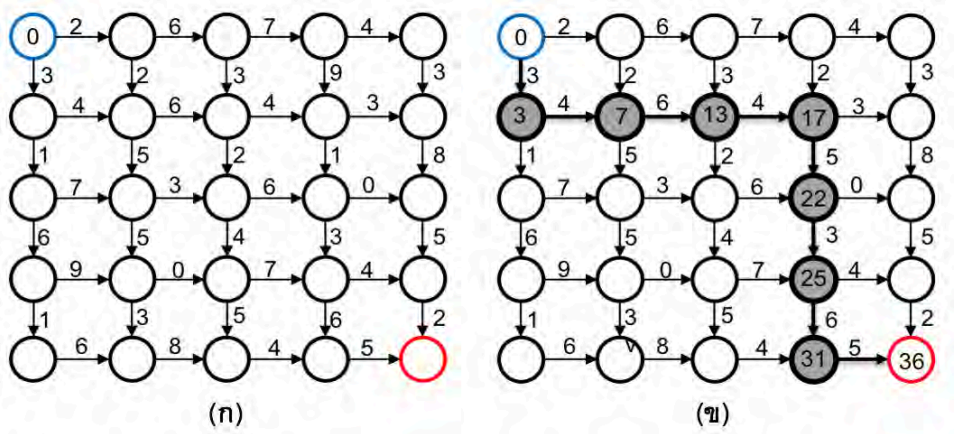
รูปที่ 5.1 (ก) แผนที่ใจกลางเมืองแมนฮัตตันที่มีจุดท่องเที่ยว (กล่องสีดำเล็กๆ) บนถนนสายต่างๆ และ (ข) กราฟแบบมีทิศทาง ManhattanGraph ที่แต่ละเส้นเชื่อมแสดงจำนวนจุดท่องเที่ยวในเส้นทางเดินนั้น

(ที่มา: รูปที่ 5.2 ของ [52])

ดาวสีน้ำเงิน (ตำแหน่งซ้ายบน) ในรูปที่ 5.1(ก) แสดงจุดตั้งต้นของการเดินเที่ยวเรียกว่าโหนดต้นทาง (source node) และจุดสิ้นสุดการเดินคือดาวสีแดง (ตำแหน่งขวาล่าง) เรียกว่าโหนดปลายทาง (sink node) โดยสามารถเดินได้เพียงสองทิศทางคือเดินลงล่างหรือเดินไปทางขวาเท่านั้น จาก *Manhattan Graph* (รูปที่ 5.1(ข)) ในโจทย์นี้เราต้องการหาเส้นทางเดินที่ผ่านจุดท่องเที่ยวมากที่สุดหรือต้องการเส้นทางเดินที่มีผลรวมของค่าน้ำหนักมากสุด รูปที่ 5.2 แสดงเมทริกซ์ที่ทำให้เป็นทั่วไปในการแก้ปัญหาโจทย์เดียวกัน

นิยามปัญหาที่ 5.2 ปัญหาการหาเส้นทางเดินชมเมืองแมนฮัตตัน

ปัญหาการหาเส้นทางเดินชมเมืองแมนฮัตตัน หาเส้นทางที่มีความยาวมากสุดจากแผนที่เมืองที่เป็นบล็อกสี่เหลี่ยม	
ข้อมูลเข้า	เมทริกซ์ขนาด n x m โดยมี n+1 แถว และ m+1 คอลัมน์
ผลลัพธ์	เส้นทางเดินที่ยาวที่สุดจากโหนดต้นทาง (0,0) ไปยังโหนดปลายทาง (n,m) ในเมทริกซ์



รูปที่ 5.2 (ก) เมทริกซ์ขนาด n x m ซึ่งแสดงแผนที่จุดตัดของเมืองๆ หนึ่งโดยโหนดสีฟ้าอยู่ตำแหน่ง (0,0) และโหนดสีแดงอยู่ที่ตำแหน่ง (4,4) (ข) เส้นทางเดินจากโหนดตั้งต้นไปยังโหนดปลายทางโดยวิธีการเลือกเส้นทางแบบละโมบ

ฝึกหัด	มีเส้นทางเดินที่เป็นไปได้ทั้งหมดกี่เส้นทางในเมทริกซ์ทางซ้ายของรูปที่ 5.2

จากนิยามปัญหาที่ 5.2 และตัวอย่างเมทริกซ์ดังในรูปที่ 5.2(ก) ถ้าใช้วิธีการทำทุกรูปแบบ (brute force) ต้องทดลองเส้นทางเดินทั้งหมดที่เป็นไปได้จำนวนมากซึ่งไม่มีประสิทธิภาพ ในขณะที่วิธีการหาเส้นทางแบบละโมบใช้เวลาน้อยแต่มักไม่ได้คำตอบที่ดีที่สุดดังตัวอย่างในรูปที่ 5.2(ข) ที่หาเส้นทางเดินได้ค่าความยาวเส้นทางรวมเท่ากับ 36 ซึ่งไม่ใช่ค่าที่ดีที่สุด

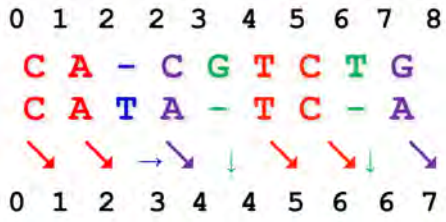
หยุดคิด	ความยาวเส้นทางรวมที่ยาวที่สุดหรือค่าน้ำหนักรวมมากสุดในรูปที่ 5.2(ก) เป็นเท่าใด

นิยามปัญหาที่ 5.2 ข้างต้นสามารถประยุกต์ใช้ได้กับกราฟที่มีทิศทางใดๆ โดยมีเงื่อนไขว่าต้องไม่มีลูป (loop) หรือไซเคิล (cycle) ในกราฟ (directed acyclic graph: DAG)

การเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนกับปัญหาการหาเส้นทางเดินชมเมืองแมนฮัตตัน

ในรูปที่ 5.3 มีการเพิ่มอาร์เรย์ของค่าจำนวนเต็มแสดงตำแหน่งของการเทียบเบสระหว่างดีเอ็นเอสองสาย โดยอาร์เรย์ [0 1 2 2 3 4 5 6 7 8] และอาร์เรย์ [0 1 2 3 4 4 5 6 6 7] แสดงจำนวนของเบสของ **CACGTCTG** และ **CATATCA** ที่ถูกใช้ไปแล้ว ณ คอลัมน์นั้นๆ ตามลำดับ นอกจากนี้อาร์เรย์ที่สาม [↘ ↘ → ↘ ↓↘ ↘↓ ↘] แสดงผลการเทียบเบสว่าเป็นแมชหรือมิสแมช (mathch/mismatch: ↘/↘) เกิด insertion (→) หรือเกิด deletion (↓) โดยอาร์เรย์นี้แสดงเส้นทางจากโหนดตั้งต้นไปยังโหนดปลายทางในเมทริกซ์ 8 x 7 ในรูปที่ 5.4 (ก) โดยโหนดที่ i ของเส้นทางนี้ประกอบด้วยค่าในตำแหน่งที่ i ของอาร์เรย์ [0 1 2 2 3 4 5 6 7 8] และ [0 1 2 3 4 5 5 6 6 7] ดังต่อไปนี้

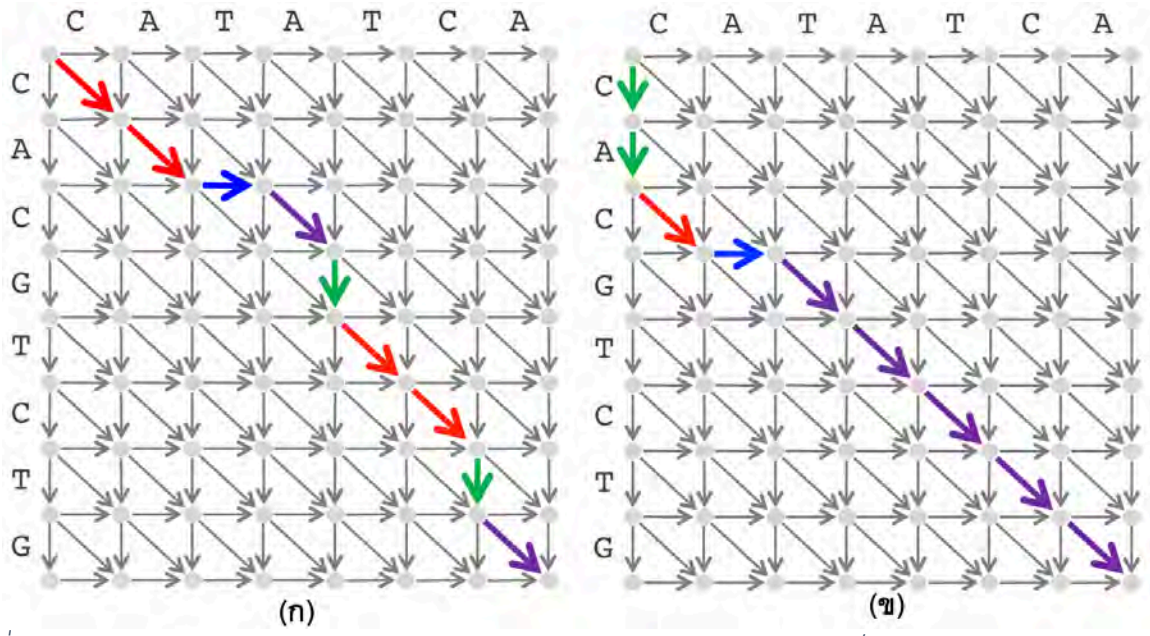
(0,0) ↘(1,1) ↘(2,2) →(2,3)↘(3,4) ↓(4,4)↘(5,5) ↘(6,6) ↓(7,6)↘(8,7)



รูปที่ 5.3 การเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอ CACGTCTG และ CATATCA โดยอาร์เรย์ของตัวเลขแถวบนสุดและล่างสุดแสดงจำนวนเบสของสายดีเอ็นเอ CACGTCTG และ CATATCA ที่ถูกใช้ไปแล้วในคอลัมน์หนึ่งๆ อาร์เรย์ของลูกศรแสดงผลการเปรียบเทียบในแต่ละคอลัมน์ว่าเป็นแมช มิสแมช หรืออินเดล

การเปรียบเทียบความคล้ายคลึงกันระหว่างดีเอ็นเอสองสายสามารถแสดงได้ด้วยเส้นทางหนึ่งๆ ในกราฟแสดงการเปรียบเทียบลำดับเบส (alignment graph) ในรูปที่ 5.4(ก) เส้นทางที่แสดงคือ (0,0) ↘(1,1) ↘(2,2) →(2,3))↘(3,4) ↓(4,4)↘(5,5) ↘(6,6) ↓(7,6)↘(8,7)ซึ่งเป็นเส้นทางที่สอดคล้องกับตัวอย่างการเปรียบเทียบดีเอ็นเอสองสายในรูปที่ 5.3 และรูปที่ 5.4(ข) แสดงตัวอย่างเส้นทางอื่นที่การเลื่อนเบสได้จำนวนเบสที่แมชเพียง 1 เบส

ฝึกหัด	แสดงผลการเปรียบเทียบความคล้ายคลึงกันระหว่างคู่ของสายดีเอ็นเอโดยอ้างอิงจากเส้นทางในกราฟรูป 5.4 ทางขวามือ



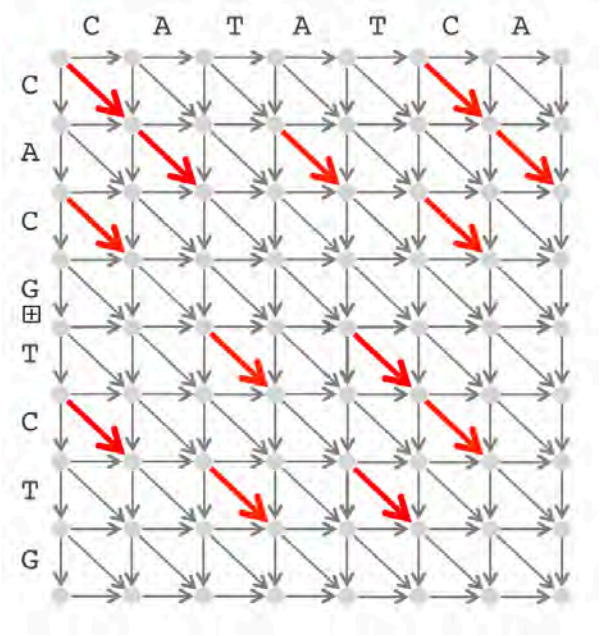
*รูปที่ 5.4 (ก) เส้นทางในกราฟแสดงการเปรียบเทียบความคล้ายคลึงกันระหว่างดีเอ็นเอสองสายคือ **CACGTCTG** และ **CATATCA** ที่สอดคล้องกับรูปที่ 5.3 (ข) ตัวอย่างเส้นทางอื่นซึ่งผลการเลื่อนเบสระหว่างสายดีเอ็นเอมีเพียง 1 เบสที่แมช*

หยุดคิด	เราสามารถใช้กราฟแสดงการเปรียบเทียบลำดับเบส (alignment graph) ในการหาส่วนของสายอักขระที่ยาวที่สุดที่ปรากฏอยู่ในสายอักขระทั้งสองสาย (longest common substring: LCS) ได้หรือไม่

รูปที่ 5.5 เป็นตัวอย่างกราฟแสดงการเปรียบเทียบลำดับเบสระหว่าง **ATGTTATA** และ **ATCGTCC** หรือ `AlignmentGraph(ATGTTATA, ATCGTCC)` ที่เน้นการแมช (↘) ระหว่างเบสทั้งหมดที่เป็นไปได้ โดยแต่ละเส้นเชื่อมที่แสดงการแมชนี้จะมีคะแนนเท่ากับ 1 ในขณะที่เส้นเชื่อมอื่นๆ ทั้งหมดมีคะแนนเป็น 0 กราฟแสดงการเปรียบเทียบลำดับเบสในรูปที่ 5.5 นี้ สามารถนำไปใช้ในการออกแบบอัลกอริทึมที่ใช้ในการหาเส้นทางที่ยาวที่สุดในกราฟแบบมีทิศทางและไม่มีลูป (DAG) โดยอัลกอริทึมหลักที่ใช้ในการแก้ปัญหานี้คือกำหนดการพลวัต (dynamic programming)

กำหนดการพลวัตกับกราฟแบบมีทิศทางและไม่มีลูป

ถ้ามีโหนด *b* อยู่ในกราฟแบบมีทิศทางและไม่มีลูป (DAG) และ s_b เป็นเส้นทางที่ยาวที่สุดจากโหนดตั้งต้นมายังโหนด *b* เราเรียกโหนด *a* ว่าเป็นตัวนำหน้า (predecessor) ของ *b* ถ้ามีเส้นเชื่อมจาก *a* มายัง *b* ใน DAG และเส้นชี้เข้าโหนด (in-degree) ของโหนดหนึ่งๆ จะเท่ากับจำนวนตัวนำหน้าของโหนดนั้นๆ คะแนน s_b ของโหนด *b* โดยมีระดับขั้นเข้าเท่ากับ *k* คำนวณได้จากสมการต่อไปนี้



*รูปที่ 5.5 AlignmentGraph(**CACGTCTG** และ **CATATCA**) ที่แสดงการแมช (↘) ทั้งหมดที่เป็นไปได้*

$$s_b = \max_{all\ predecessors\ a\ of\ node\ b} \{s_a + weight\ of\ edge\ from\ a\ to\ b\}$$

จากกราฟแสดงการเปรียบเทียบลำดับเบสในรูปที่ 5.5 สามารถคำนวณเส้นทางที่ยาวที่สุดได้โดยใช้สมการต่อไปนี้

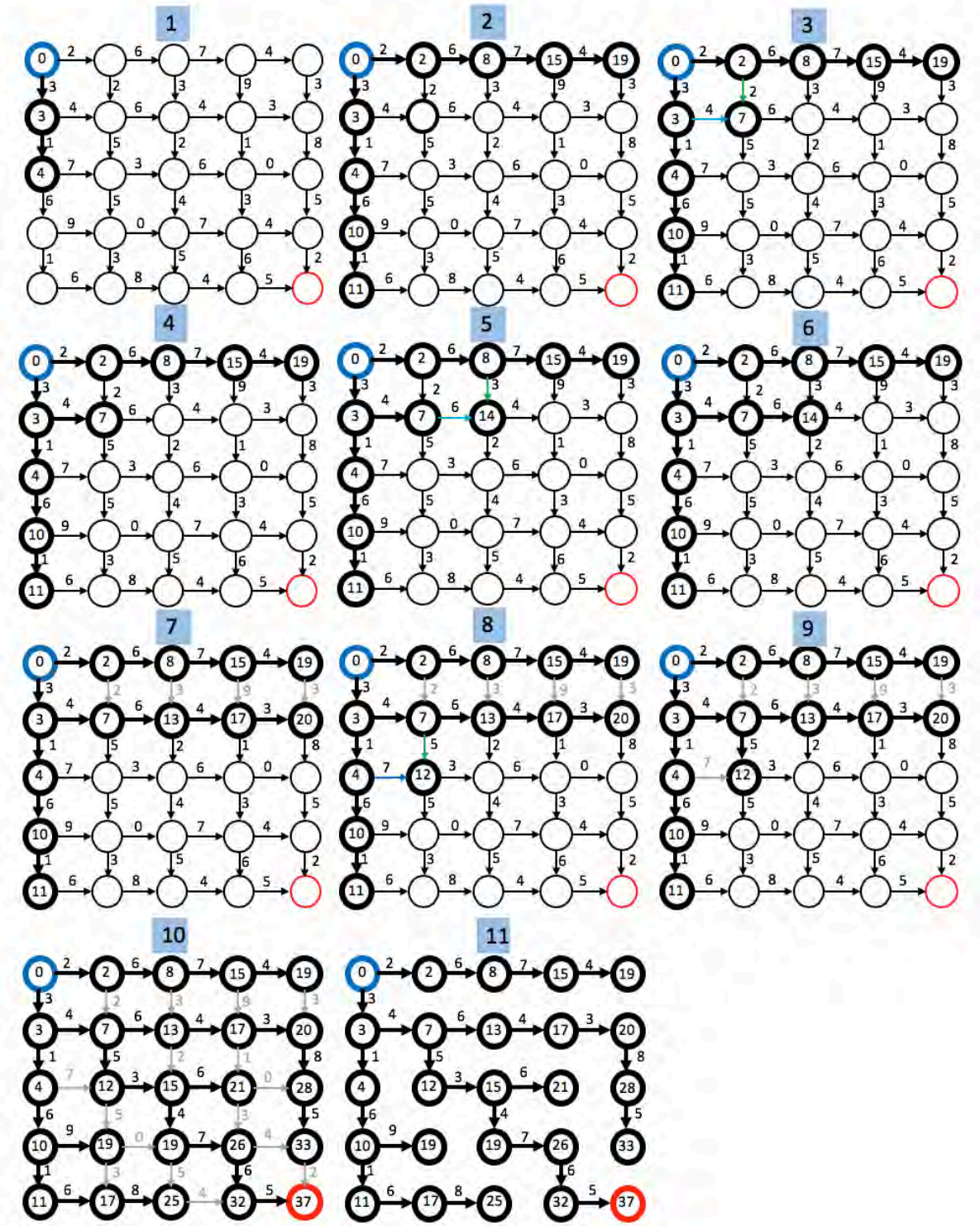
$$s_{i,j} = \max \begin{cases} s_{i-1,j} + 0 \\ s_{i,j-1} + 0 \\ s_{i-1,j-1} + 1\ if\ v_i = w_j \end{cases}$$

และสามารถคำนวณโดยอนุญาตให้ค่าน้ำหนักมีความเป็นทั่วไปมากขึ้น โดยใช้สมการการต่อไปนี้

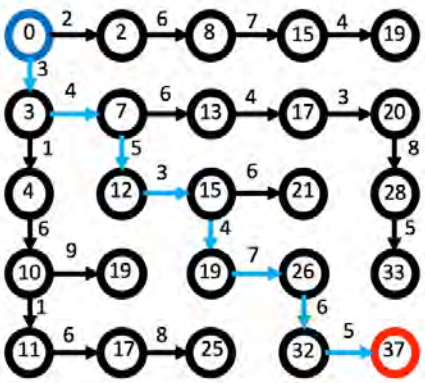
$$s_{i,j} = \max \begin{cases} s_{i-1,j} + weight\ of\ edge\ \downarrow between\ (i-1,j)\ and\ (i,j) \\ s_{i,j-1} + weight\ of\ edge\ \rightarrow between\ (i,j-1)\ and\ (i,j) \\ s_{i-1,j-1} + weight\ of\ edge\ \searrow between\ (i-1,j-1)\ and\ (i,j) \end{cases}$$

หยุดคิด	จากสมการการปรากฏซ้ำ (recurrence) ข้างต้น แม้ไม่การพิจารณาค่ามิสแมช (mismatch) เป็นส่วนหนึ่งของสมการ แต่สามารถใช้ในการหาเส้นทางที่ยาวที่สุด ซึ่งอนุมานถึงจำนวนเบสที่ตรงกันมากที่สุดได้ เพราะอะไร

รูปที่ 5.6 แสดงขั้นตอนการหาเส้นทางที่ยาวที่สุดโดยใช้สมการการปรากฏซ้ำข้างต้น โดยใช้เมทริกซ์เดียวกันกับการหาเส้นทางแบบละโมบในรูปที่ 5.2 และรูปที่ 5.7 เป็นเส้นทางที่ยาวที่สุดที่เป็นผลลัพธ์สุดท้ายของรูปที่ 5.6



รูปที่ 5.6 ขั้นตอนการหาเส้นทางที่ยาวที่สุดสำหรับเมทริกซ์ในรูปที่ 5.2 โดยใช้กำหนดการพลวัต



รูปที่ 5.7 เส้นทางที่มีผลรวมค่าน้ำหนักเส้นเชื่อมมากสุดจากผลลัพธ์ในรูปที่ 5.6

การเดินย้อนกลับในกราฟแสดงการเปรียบเทียบลำดับเบส

เราสามารถใช้แนวคิดของการเดินย้อนกลับจากโหนดปลายทางไปยังโหนดต้นทางเพื่อแสดงส่วนของดีเอ็นเอที่ยาวที่สุดร่วมกัน (LCS) ระหว่างสายดีเอ็นเอ *v* และ *w* จากรูปที่ 5.5 ข้างต้นถ้าเรากำหนดค่าน้ำหนักของเส้นเชื่อมที่แสดงสถานะแมช (match) เท่ากับ 1 และเส้นเชื่อมที่เหลือทั้งหมดเป็น 0 ค่าของ $s_{|v|,|w|}$ จะเท่ากับ LCS ของสายดีเอ็นเอ *v* และ *w* อัลกอริทึมต่อไปนี้ (รหัสเทียมที่ 5.1 `LCSBackTrack()`) มีการเก็บตัวชี้การย้อนรอย (backtracking pointer) ที่ถูกใช้ในระหว่างการหาเส้นทาง $s_{i,j}$ โดยมีค่าที่เป็นไปได้ 3 ค่าคือ ↓, → และ↘

รหัสเทียมที่ 5.1 LCSBackTrack

```
LCSBackTrack(v,w)
    # v เป็นอะเรย์ของลำดับเบสในดีเอ็นเอเส้นแรก
    # w เป็นอะเรย์ของลำดับเบสในดีเอ็นเอเส้นที่สอง
    Backtrack <- เมทริกซ์ขนาด v x w และให้ค่าตั้งต้นเป็น "" ทั้งหมด
    A <- เมทริกซ์ขนาด v x w และให้ค่าตั้งต้นเป็น 0 ทั้งหมด
    for i ที่มีค่าตั้งแต่ 1 ถึง จำนวนเบสใน v
        for j ที่มีค่าตั้งแต่ 1 ถึง จำนวนเบสใน w
            if v[i] == w[j]
                A[i][j] <- A[i-1][j-1] + 1
            else:
                A[i][j] <- max(A[i-1][j], A[i][j-1])
            if A[i][j] == A[i-1][j]
                Backtrack[i][j] <- "S" # S คือเส้นที่ชี้ลง
            else if A[i][j] == A[i][j-1]
                Backtrack[i][j] <- "E" # E คือเส้นชี้ไปทางขวา
            else if A[i][j] == A[i-1][j-1]+1 และ v[i] == w[j]
                Backtrack[i][j] <- "D" # D คือเส้นทแยงมุม
    ส่งกลับ Backtrack
```

และจากเมทริกซ์ Backtrack ที่สร้างขึ้น สามารถแสดงผลเป็นสายอักขระของผลการเทียบเบสในแต่ละคอลัมน์ตามรหัสเทียม `OutputLCS()` ต่อไปนี้

รหัสเทียมที่ 5.2 OutputLCS

```
OutputLCS(Backtrack,v,i,j)
    if i==0 หรือ j==0
        ส่งกลับค่าอักขระว่าง
    if Backtrack[i][j] == "S" # S คือเส้นที่ชี้ลง
        OutputLCS(Backtrack,v,i-1,j)
    else if Backtrack[i][j] == "E" # E คือเส้นชี้ไปทางขวา
        OutputLCS(Backtrack,v,i,j-1)
    else if Backtrack[i][j] == "D" # D คือเส้นทแยงมุม
        OutputLCS(Backtrack,v,i-1,j-1)
        แสดงผล v[i]
```

ฝึกหัด	ตัวอย่างรหัสเทียม `OutputLCS()` ข้างต้นจะแสดงผลของ LCS เพียงสายเดียว จงปรับโค้ด `OutputLCS()` และ `LCSBrackTrack()` ข้างต้น ให้สามารถหา LCS ทั้งหมดที่มีอยู่ในสายอักขระทั้งสองเส้น

การให้คะแนนความคล้ายคลึงกัน

ข้อจำกัดในการให้คะแนนความคล้ายคลึงกันระหว่างดีเอ็นเอสองเส้นโดยให้ค่าแมช (match) เป็น 1 ในขณะที่ค่ามิสแมช (mismatch) และอินเดล (indel) เป็น 0 อาจทำให้เราพยายามขยับสายดีเอ็นเอเพื่อให้ได้จำนวนแมชมากที่สุดโดยไม่สนใจว่าต้องเพิ่มอินเดลเข้าไปเท่าใด อย่างไรก็ตามอินเดลมีความหมายทางชีววิทยาซึ่งหลายกรณีเกี่ยวข้องกับกระบวนการวิวัฒนาการ ดังนั้นการเพิ่มลดเบสในระหว่างการเทียบลำดับเบสควรมีการพิจารณาการลงโทษในกรณีที่เกิดอินเดลด้วย

เมทริกซ์คะแนน

เพื่อให้การให้คะแนนการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอมีการนำเรื่องอินเดลเข้ามาพิจารณา นอกจากให้คะแนนแมชของแต่ละเบสเป็น 1 ตามตัวอย่างข้างต้น ได้มีการเพิ่มส่วนของการลงโทษ โดยเพิ่ม μ เป็นตัวคูณกับจำนวนมิสแมชที่เกิดขึ้นและ σ เป็นตัวคูณกับจำนวนอินเดลที่เกิดขึ้น ซึ่งค่าคะแนนใหม่นี้สามารถแสดงด้วยชุดตัวแปรต่อไปนี้

$$\#matches - \mu \cdot \#mismatches - \sigma \cdot \#indels$$

จากตัวอย่างการเปรียบเทียบความคล้ายคลึงกันของดีเอ็นเอสองเส้นต่อไปนี้ ถ้ามีการกำหนดค่า $\mu = 1$ และ $\sigma = 2$ จะได้คะแนนรวมของความคล้ายคลึงอยู่ที่ -4

```
A T - G T T A T A
A T C G T - C - C
+1+1-2+1+1-2-1-2-1
```

นักชีววิทยาได้กำหนดรายละเอียดของการลงโทษเพิ่มเติมโดยใช้องค์ความรู้ที่มีมาก่อนว่า โอกาสหรือความถี่ของการเกิดความไม่ตรงกันเหล่านี้มีไม่เท่ากันสำหรับนิวคลีโอไทด์และกรดแอมิโนแต่ละตัว ดังนั้นคะแนนลงโทษของการเกิดมิสแมชและอินเดลสำหรับแต่ละนิวคลีโอไทด์หรือกรดแอมิโนที่จำเพาะจะมีค่าแตกต่างกันไป โดยคะแนนลงโทษที่แตกต่างกันนี้สามารถกำหนดอยู่ในรูปแบบเมทริกซ์คะแนน ตัวอย่างเช่น ถ้าสายอักขระมีอักขระที่เป็นไปได้ทั้งหมด k แบบ จะมีการสร้างเมทริกซ์คะแนนขนาด (k+1) x (k+1) โดยเก็บคะแนนของการเทียบระหว่างทุกคู่ของอักขระ สำหรับเมทริกซ์คะแนนของดีเอ็นเอที่มีนิวคลีโอไทด์ 4 ประเภท (k=4) ถ้าคะแนนลงโทษของมิสแมช (mismatch) ทั้งหมดมีค่าเท่ากันคือ μ และคะแนนลงโทษของอินเดล (indel) ทั้งหมดมีค่าเท่ากันคือ σ จะได้เมทริกซ์คะแนนของนิวคลีโอไทด์ที่สามารถนำไปใช้ในการเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอดังต่อไปนี้

	A	C	G	T	–
A	+1	$-\mu$	$-\mu$	$-\mu$	$-\sigma$
C	$-\mu$	+1	$-\mu$	$-\mu$	$-\sigma$
G	$-\mu$	$-\mu$	+1	$-\mu$	$-\sigma$
T	$-\mu$	$-\mu$	$-\mu$	+1	$-\sigma$
–	$-\sigma$	$-\sigma$	$-\sigma$	$-\sigma$	

โดยทั่วไปเมทริกซ์คะแนนที่ใช้เปรียบเทียบสายดีเอ็นเอมักมีการกำหนดค่าตัวแปร μ และ σ เท่านั้น ในขณะที่เมทริกซ์คะแนนที่ใช้เปรียบเทียบสายโปรตีนจะมีรายละเอียดมากกว่ามากตามจำนวนกรดแอมิโนและตามวิธีการที่ได้ มาซึ่งคะแนนในเมทริกซ์ โดยเมทริกซ์คะแนนหลักที่ใช้ในการเปรียบเทียบสายโปรตีนประกอบด้วยเมทริกซ์คะแนนแพม (PAM) และเมทริกซ์คะแนนบลอสซัม (BLOSUM) (ภาคผนวกบทที่ 5)

การเปรียบเทียบความคล้ายคลึงกันแบบครอบคลุมและแบบเฉพาะที่

การเปรียบเทียบความคล้ายคลึงกันแบบครอบคลุม

วิธีการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนที่ผ่านมาเป็นการเปรียบเทียบแบบครอบคลุม (global alignment) โดยมีนิยามปัญหาดังต่อไปนี้

นิยามปัญหาที่ 5.3 ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนแบบครอบคลุม

ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนแบบครอบคลุม (Global Alignment Problem) หาคะแนนที่มากที่สุดในการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนโดยใช้เมทริกซ์คะแนน	
ข้อมูลเข้า	สายอักขระสองเส้นซึ่งเป็นตัวแทนของสายดีเอ็นเอหรือโปรตีนและเมทริกซ์คะแนน
ผลลัพธ์	ผลการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนโดยมีคะแนนรวมมากสุด

เพื่อเป็นการแก้ปัญหาข้างต้นให้หาเส้นทางที่ยาวที่สุดในกราฟแสดงการเปรียบเทียบลำดับเบสหลังมีการปรับค่าเส้นเชื่อมในกราฟโดยใช้เมทริกซ์คะแนน จากนั้นคำนวณคะแนนแต่ละโหนดในกราฟด้วยสมการต่อไปนี้

$$s_{i,j} = \max \begin{cases} s_{i-1,j} + Score(v_i, -) \\ s_{i,j-1} + Score(-, w_j) \\ s_{i-1,j-1} + Score(v_i, w_j) \end{cases}$$

ข้อจำกัดของการเปรียบเทียบความคล้ายคลึงกันแบบครอบคลุม

การวิเคราะห์ชุดของยีนในกลุ่มฮอมีโอบอกซ์ (homeobox genes) ถูกนำมาใช้แสดงข้อจำกัดของการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนแบบครอบคลุม (global alignment) ซึ่งไม่สามารถค้นพบความหมายทางชีววิทยาที่ซ่อนอยู่ ยีนในกลุ่มฮอมีโอบอกซ์มีหน้าที่ในการควบคุมการพัฒนาเอ็มบริโอ (embryo) และพบในสิ่งมีชีวิตหลายชนิดรวมทั้งแมลงวันและมนุษย์ (รูปที่ 5.8) ยีนกลุ่มฮอมีโอบอกซ์มีความยาวและแตกต่างกันค่อนข้างมากระหว่างสิ่งมีชีวิต อย่างไรก็ตามมีบริเวณย่อยในยีนเหล่านี้ที่มีความอนุรักษ์ระหว่างสิ่งมีชีวิตซึ่งเรียกว่าฮอมีโอโดเมน (homeodomain) คำถามคือจะสามารถหาบริเวณเหล่านี้ซึ่งเป็นเพียงส่วนสั้นๆ ในสายของยีนที่มีความยาวมากได้อย่างไร เพราะคะแนนความคล้ายคลึงแบบครอบคลุมมักมีค่าน้อย เนื่องจากการเปรียบเทียบแบบครอบคลุมจะพยายามหาความเหมือนกันตลอดความยาวของสายข้อมูลทั้งสองเส้น ถ้าต้องการหาความคล้ายคลึงกันเฉพาะที่ เช่น บริเวณที่เป็นฮอมีโอโดเมนข้างต้น จำเป็นต้องปรับอัลกอริทึมเพิ่มเติม

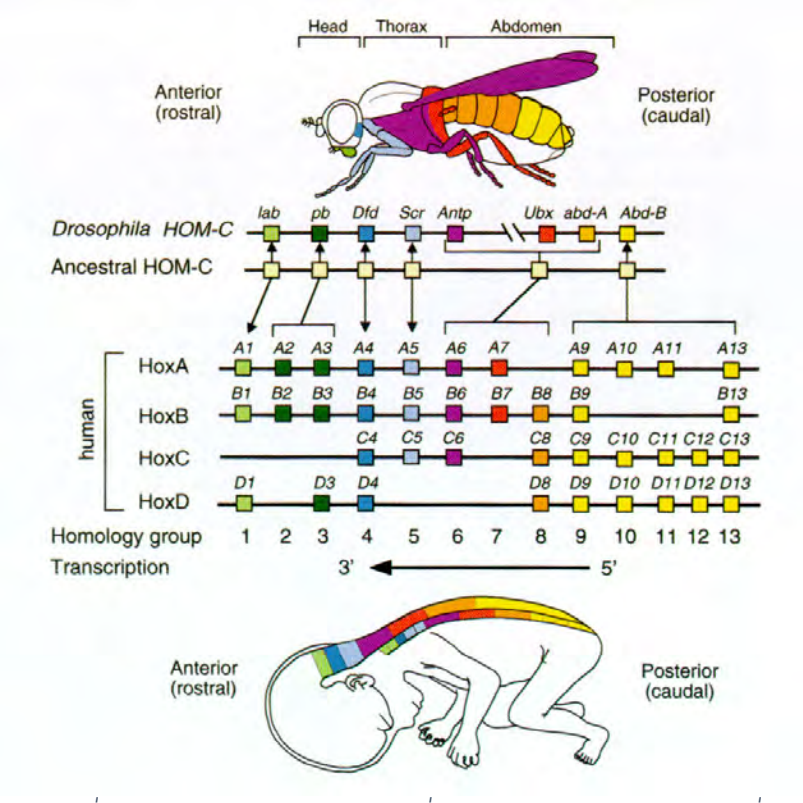
พิจารณาตัวอย่างการเปรียบเทียบดีเอ็นเอสองสายต่อไปนี้ ซึ่งคะแนนแบบครอบคลุมประกอบด้วย 22 แมช 18 อินเดล และ 2 มิสแมช และได้คะแนนรวมเป็น 22-18-2 = 2 คะแนน (โดยสมมติว่า μ และ σ มีค่าเป็น 1 ทั้งคู่)

```
GCC-C-AGTC-TATGT-CAGGGGGCACG--A-GCATGCACA-
GCCGCC-GTCGT-T-TTCAG----CA-GTTATGT-T-CAGAT
```

อย่างไรก็ตามคู่ของสายดีเอ็นเอนี้ยังมีเส้นทางอื่นในกราฟแสดงการเปรียบเทียบลำดับเบส เช่น การขยับลำดับเบสโดยเน้นให้เกิดชุดของเบสที่แมชอยู่ติดกันดังตัวอย่างต่อไปนี้

```
---G----C-----C--CAGTCTATG-TCAGGGGGCACGAGCATGCAGA
GCCGCCGTCGTTTTCAGCAGT-TATGTTCAG-----A------T-----
```

ซึ่งประกอบด้วย 17 แมชและ 32 อินเดล และได้คะแนนรวมเท่ากับ -15 ถึงแม้บริเวณที่มีความอนุรักษ์จะให้คะแนนถึง 12-2 = 10 คะแนน ซึ่งโอกาสที่จะเกิดโดยบังเอิญมีน้อย เส้นทางที่สองนี้เน้นให้ชุดของเบสที่แมชอยู่ติดกัน ซึ่งถ้าดูคะแนนรวมแบบครอบคลุมข้างต้นจะได้น้อยกว่าคะแนนรวมของเส้นทางแรกมากเพราะเกิดอินเดลจำนวนมาก ในขณะที่ผลของการคำนวณคะแนนแบบครอบคลุมที่มากกว่ากลับไม่สามารถสื่อความหมายในเชิงชีววิทยาได้ในกรณีนี้



รูปที่ 5.8 ชุดของยีนฮอมีโอบอกซ์ที่พบในมนุษย์เทียบกับแมลงหวี่

(ที่มา: รูปที่ 1 ของ [126])

ดังนั้นในกรณีที่ความคล้ายคลึงกันเกิดเฉพาะบางบริเวณของสายดีเอ็นเอหรือโปรตีนนักชีววิทยาจะไม่สนใจการเปรียบเทียบแบบครอบคลุม (global alignment) แต่จะเน้นการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอหรือโปรตีนเฉพาะที่ (local alignment) ที่มีคะแนนของการเปรียบเทียบแบบครอบคลุมเฉพาะที่มากสุดตามนิยามปัญหาที่ 5.4

นิยามปัญหาที่ 5.4 การเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนเฉพาะที่

ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนเฉพาะที่ (Local Alignment Problem) หาคะแนนที่มากสุดในการเปรียบเทียบความคล้ายคลึงกันระหว่างดีเอ็นเอและหรือโปรตีนสองสายแบบเฉพาะที่	
ข้อมูลเข้า	สายอักขระสองเส้น *v* และ *w* ซึ่งเป็นตัวแทนของสายดีเอ็นเอหรือโปรตีนและเมทริกซ์คะแนน
ผลลัพธ์	ผลการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนแบบเฉพาะที่ที่มีคะแนนรวมมากสุด

วิธีการพื้นฐานที่ใช้ในการแก้ปัญหาคือการหาเส้นทางที่มีคะแนนรวมมากสุดจากกราฟแสดงการเปรียบเทียบลำดับเบสที่มีการเชื่อมต่อคู่ของโหนดทุกคู่

หยุดคิด	เวลาที่ใช้ในการแก้ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนเฉพาะที่โดยใช้วิธีการพื้นฐานข้างต้นเป็นเท่าใด

การเปรียบเทียบลำดับเบสกับการนั่งแท็กซี่ฟรี

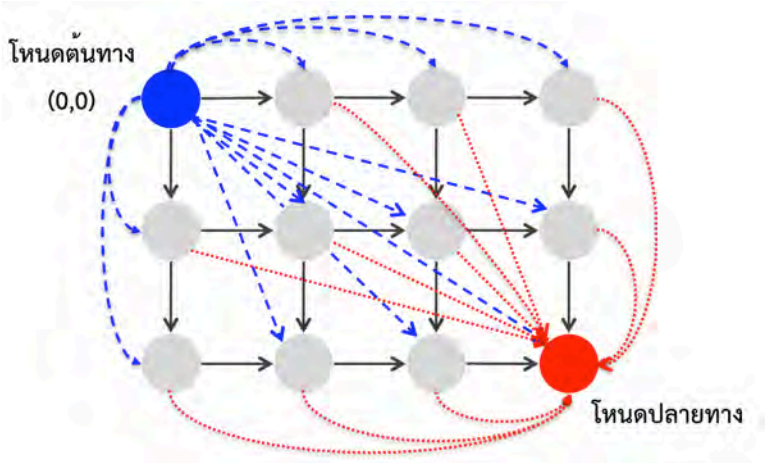
เพื่อให้สามารถหาคำตอบได้เร็วขึ้น ลองนึกถึงการนั่งแท็กซี่ฟรีจากโหนดเริ่มต้นที่ตำแหน่ง (0,0) ตรงไปยังโหนดที่เป็นจุดเริ่มต้นของส่วนของสายข้อมูลที่มีความอนุรักษ์ถ้ามีโหนดนั้นอยู่ และเริ่มนับจำนวนแมชที่ปรากฏไปเรื่อยๆ จนพบโหนดสุดท้ายที่เป็นส่วนของสายข้อมูลที่มีความอนุรักษ์ จากนั้นนั่งแท็กซี่ฟรีอีกครั้งโดยตรงไปยังโหนดปลายทาง คะแนนการเปรียบเทียบความคล้ายคลึงกันของสายอักขระสองสายนี้จะเท่ากับคะแนนของการเปรียบเทียบเฉพาะที่ที่เกิดความอนุรักษ์ระหว่างสายอักขระสองสาย จากตัวอย่างของการนั่งแท็กซี่ฟรีข้างต้นเทียบได้กับการเพิ่มเส้นเชื่อมจากโหนดต้นทาง (0,0) ไปยังโหนดอื่นๆ ทั้งหมดโดยมีน้ำหนักเป็น 0 และเพิ่มเส้นเชื่อมจากโหนดใดๆ ที่ไม่ใช้โหนดต้นทางไปยังโหนดปลายทางโดยมีน้ำหนักเป็น 0 เช่นกัน ซึ่งจะได้กราฟแบบมีทิศทางและไม่เกิดลูปดังแสดงในรูปที่ 5.9 และสามารถนำไปใช้ในการแก้ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอหรือโปรตีนเฉพาะที่ข้างต้น ด้วยแนวคิดของแท็กซี่ฟรีเราไม่จำเป็นต้องหาเส้นทางที่ยาวที่สุด (มีน้ำหนักรวมมากที่สุด) ระหว่างทุกคู่ของโหนดในกราฟ เนื่องจากเส้นทางที่ยาวที่สุดจากโหนดต้นทางไปยังโหนดปลายทางเป็นเส้นทางที่ดีที่สุดแล้ว

จำนวนเส้นเชื่อมทั้งหมดในกราฟรูปที่ 5.9 มีค่าเท่ากับ $O(|v||w|)$ซึ่งมีค่าไม่มากและเนื่องจากเวลาที่ใช้ในการหาเส้นทางที่ยาวที่สุดถูกกำหนดโดยจำนวนของเส้นเชื่อมที่อยู่ในกราฟ การเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอหรือโปรตีนเฉพาะที่จะทำงานได้รวดเร็ว

ในการคำนวณค่า $s_{i,j}$ โดยการเพิ่มเส้นเชื่อมน้ำหนัก 0 จากโหนดต้นทางไปยังทุกโหนดในกราฟ ทำให้โหนดต้นทางเป็นตัวนำหน้า (predecessor) ของทุกโหนด ดังนั้นการคำนวณคะแนนในสมการก่อนหน้าต้องปรับเพิ่มเติมดังต่อไปนี้

$$s_{i,j} = max \begin{cases} 0 \\ s_{i-1,j} + Score(v_i, -) \\ s_{i,j-1} + Score(-, w_j) \\ s_{i-1,j-1} + Score(v_i, w_j) \end{cases}$$

และเนื่องจากโหนดปลายทางก็มีเส้นเชื่อมตรงจากโหนดก่อนหน้าทุกโหนด ดังนั้นการหาเส้นทางข้างต้นจะครอบคลุมความยาวรวมของสายดีเอ็นหรือโปรตีนทั้งเส้น



รูปที่ 5.9 กราฟเปรียบเทียบลำดับเบสที่มีการเพิ่มเส้นเชื่อมที่มีค่าน้ำหนักเป็น 0 (เส้นประสีน้ำเงิน) ที่เชื่อมโหนดตั้งต้นสีน้ำเงิน (0,0) ไปยังทุกโหนดในกราฟและเพิ่มเส้นเชื่อมที่มีค่าน้ำหนักเป็น 0 (เส้นไข่ปลาสีแดง) ที่เชื่อมทุกโหนดที่ไม่ใช่โหนดตั้งต้นไปยังโหนดปลายทางสีแดง

(*ที่มา*: *ปรับจากรูปที่* 5.21 *ของ* [52])

การประยุกต์ใช้การเปรียบเทียบความคล้ายคลึงกันของสายอักขระกับปัญหาอื่น

Edit distance

ในปี ค.ศ. 1966 วลาดิเมียร์ เลเวนชเตย์น (Vladimir Levenshtein) ได้นิยามปัญหา edit distance หรือระยะทางระหว่างสายอักขระสองสายว่าเป็นการหาจำนวนการดำเนินการตรวจแก้ (edit operation) ที่ต้องใช้ในการแปลงสายอักขระเส้นหนึ่งให้เป็นสายอักขระอีกเส้นหนึ่ง โดยการดำเนินการตรวจแก้ประกอบด้วย การสอดแทรก (insertion) การขาดหาย (deletion) และการแทนที่ (substitution) ในตำแหน่งหนึ่งๆ ตัวอย่างเช่น สายอักขระ **TGCATAT** สามารถแปลงได้เป็น **ATCCGAT** โดยใช้ 5 การดำเนินการ ซึ่งอนุมานได้ว่าระยะทางที่มากที่สุดระหว่างสายอักขระสองสายเท่ากับ 5 ดังแสดงในขั้นตอนต่อไปนี้

TGCATAT
↓ ลบ (delete) นิวคลีโอไทด์สุดท้าย
TGCATA
↓ ลบ (delete) นิวคลีโอไทด์สุดท้าย
TGCAT
↓ เพิ่ม (insert) **A** ที่ต้นสายอักขระ
ATGCAT
↓ เปลี่ยน (substitute) **G** เป็น **C**
ATCCAT
↓ เพิ่ม (insert) **G** หลังตำแหน่งที่ 4
ATCCGAT

หยุดคิด	เราสามารถเปลี่ยนสายอักขระ TGCATAT ให้เป็น ATCCGAT โดยใช้จำนวนการดำเนินการน้อยกว่า 5 ได้ไหม

ในความเป็นจริงแล้วระยะทางระหว่างสายอักขระสองสายนี้มีค่าเท่ากับ 4 ดังแสดงต่อไปนี้

TGCATAT
↓ เพิ่ม (insert) A ที่ต้นสายอักขระ
ATGCATAT
↓ ลบ (delete) T ในตำแหน่งที่ 6
ATGCAAT
↓ เปลี่ยน (substitute) A เป็น G ในตำแหน่งที่ 5
ATGCGAT
↓ เปลี่ยน (substitute) G เป็น C ในตำแหน่งที่ 3
ATCCGAT

เลเวนชเตย์นได้นิยามปัญหาการหาระยะทางระหว่างสายอักขระสองสายไว้แต่ไม่ได้อธิบายอัลกอริทึมในการแก้ปัญหา

ฝึกหัด	จงเขียนอัลกอริทึมที่ใช้ในการหาระยะทางระหว่างสายอักขระสองสาย

Fitting alignment

สมมติเรามีสายโปรตีนยาว 20,000 กรดแอมิโน (v) ในเชื้อแบคทีเรีย *Bacillus brevis* และต้องการหาส่วนของโปรตีนที่มีความคล้ายคลึงกับโปรตีนโดเมน A (A-domain) ที่มีความยาว 600 กรดแอมิโน (w) ในเชื้ออื่นๆ การเปรียบเทียบความคล้ายคลึงกันของสายอักขระแบบครอบคลุม (global alignment) จะไม่ได้ผลลัพธ์ตามที่คาดหวัง เพราะวิธีการนี้จะพยายามเทียบ 600 กรดแอมิโนกับทั้ง 20,000 กรดแอมิโนและหาคะแนนรวมที่มากที่สุดโดยไม่สนใจความหมายทางชีววิทยา ในขณะที่การเปรียบเทียบความคล้าย คลึงกันเฉพาะที่จะพยายามหาส่วนของสายอักขระของทั้ง v และ w ที่มีความอนุรักษ์ร่วมกันที่ให้คะแนนมากสุด ดังนั้นจึงจำเป็นต้องมีอัลกอริทึมจำเพาะที่จะทำการเทียบส่วนของสายอักขระ v' ใดๆ ของ v ซึ่งทำให้ได้คะแนนรวมของการเปรียบเทียบความเหมือนแบบครอบคลุมระหว่าง v' กับ w มากที่สุด (Fitting Alignment Problem) ตัวอย่างต่อไปนี้แสดงผลคะแนนการเปรียบเทียบความเหมือนระหว่าง v = GTAGGCTTAAGGTTA และ w = TAGATA โดยมีสมมติฐานว่าค่าลงโทษทั้งมิสแมช μ และอินเดล σ เป็น 1 ทั้งคู่

Global	Local	Fitting
GTAGGCTTAAGGTTA	GTAGGCTTAAGGTTA	GTAGGCTTAAGGTTA
-TAG----A---T-A	-TAGATA	-TAGA--TA

ในตัวอย่างข้างต้นนี้คะแนนของการเปรียบเทียบความเหมือนเฉพาะส่วน (local alignment) มีค่าเท่ากับ 3 ในขณะที่คะแนนของการเปรียบเทียบความเหมือนแบบครอบคลุม (global alignment) เท่ากับ 6-9 = -3 และคะแนนของ Fitting alignment มีค่าเท่ากับ 5-1-2 = 2

Overlap alignment

ในบทที่ 2 มีตัวอย่างการประกอบร่างจีโนมโดยใช้กราฟแสดงความคาบเกี่ยว (overlap graph) ซึ่งความซับซ้อนของความคาบเกี่ยวนี้มีเพิ่มมากขึ้นเมื่อรีดที่อ่านได้มีความผิดพลาด การเปรียบเทียบความคล้ายคลึงกันระหว่างซัฟฟิกซ์ของรีดที่ 1 กับพรีฟิกซ์ของรีดที่ 2 แสดงโดยตัวอย่างต่อไปนี้

ATGCATGCCGG
T-CC-GAAAC

การเปรียบเทียบความคล้ายคลึงกันในส่วนของสายข้อมูลที่คาบเกี่ยว (overlap alignment) ของสายอักขระ $v = v_1 \ldots v_n$ และ $w = w_1 \ldots w_m$ เป็นการเปรียบเทียบความคล้ายคลึงของสายอักขระแบบครอบคลุมเฉพาะบริเวณที่คาบเกี่ยวกัน

การกำหนดคะแนนลงโทษในกรณีที่เกิด insertion หรือ deletion

Affine gap penalties

การลงโทษในกรณีที่เกิด indels โดยใช้ σ ข้างต้น ถึงแม้จะทำให้การให้คะแนนความคล้ายคลึงกันมีความหมายทางชีววิทยามากขึ้น แต่ก็ยังมีรายละเอียดของการเพิ่มหรือลดเบสที่ต้องพิจารณาเพิ่มเติม ตัวอย่างเช่น ในการคำนวณคะแนนลงโทษของอินเดล ณ จุดนี้แต่ละตำแหน่งที่เกิดอินเดลถือว่าเป็นอิสระต่อกันซึ่งหมายถึงถ้าเกิดอินเดล k ตำแหน่ง คะแนนลงโทษส่วนอินเดลนี้จะเท่ากับ $\sigma \cdot k$ อย่างไรก็ตาม หลายๆ ครั้งที่เกิดอินเดลขึ้นในสายดีเอ็นเอเป็นการเพิ่มหรือลดชุดของเบส ดังนั้นคะแนนลงโทษ $\sigma \cdot k$ ข้างต้นจะเป็นการลงโทษมากเกินไป ในตัวอย่างการเปรียบเทียบสายดีเอ็นเอต่อไปนี้ พบว่าทั้งด้านซ้ายและขวาได้คะแนนเท่ากัน อย่างไรก็ตามในทางชีววิทยาผลการเทียบในด้านขวานั้นมีความเหมาะสมกว่า

GATCCAG GATCCAG
GA-C-AG GA--CAG

จากตัวอย่างข้างต้น จึงได้มีการเพิ่มตัวแปรอีกหนึ่งตัวเพื่อใช้ระบุช่องว่าง (gap) ซึ่งเป็นจำนวนอินเดลที่เกิดขึ้นติดต่อกันในผลการเปรียบเทียบคู่ของสายดีเอ็นเอหรือโปรตีน และมีการนำเสนอการกำหนดคะแนนลงโทษในกรณีที่เกิดช่องว่างความยาว k เบสโดยใช้สมการ เช่น $\sigma + \epsilon \cdot (k-1)$ โดยที่ σ แสดงค่า gap opening penalty หรือค่าลงโทษการเริ่มต้นช่องว่าง โดยคำนวณจากเบสแรกของช่องว่าง ในขณะที่ ϵ หรือ gap extension penalty แสดงค่าสัมประสิทธิ์ของจำนวนอินเดลที่เหลือในช่องว่าง โดยทั่วไป ϵ จะมีค่าน้อยกว่า σ เพื่อแสดง affine penalty ที่การลงโทษในกรณีที่เกิดอินเดลต่อเนื่องจะมีโทษน้อยกว่าการเกิดอินเดลเดี่ยวๆ ตัวอย่างเช่น ถ้า $\sigma = 5$

และ $\epsilon = 1$ ค่าคะแนนลงโทษของการเปรียบเทียบคู่ของสายดีเอ็นเอข้างต้นทางซ้ายได้เท่ากับ $2\sigma = 10$ ในขณะที่ทางขวาจะเท่ากับ $\sigma + \epsilon = 6$

นิยามปัญหาที่ 5.5 ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนโดยใช้ affine gap penalty

ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนโดยใช้ affine gap penalty เปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนแบบครอบคลุมให้ได้คะแนนมากสุดโดยใช้ affine gap penalty	
ข้อมูลเข้า	สายอักขระสองเส้นซึ่งเป็นตัวแทนของสายดีเอ็นเอหรือโปรตีน เมทริกซ์คะแนน σ และ ϵ
ผลลัพธ์	ผลการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนแบบครอบคลุมที่คำนวณจากเมทริกซ์คะแนน ค่าลงโทษการเริ่มต้นช่องว่าง σ และค่าสัมประสิทธิ์ของจำนวนอินเดลที่อยู่ในช่องว่างโดยไม่รวมอินเดลแรกสุด ϵ

หยุดคิด	ต้องปรับแต่งกราฟแสดงการเปรียบเทียบลำดับเบส (alignment graph) อย่างไรให้สามารถแสดงช่องว่างในกราฟ

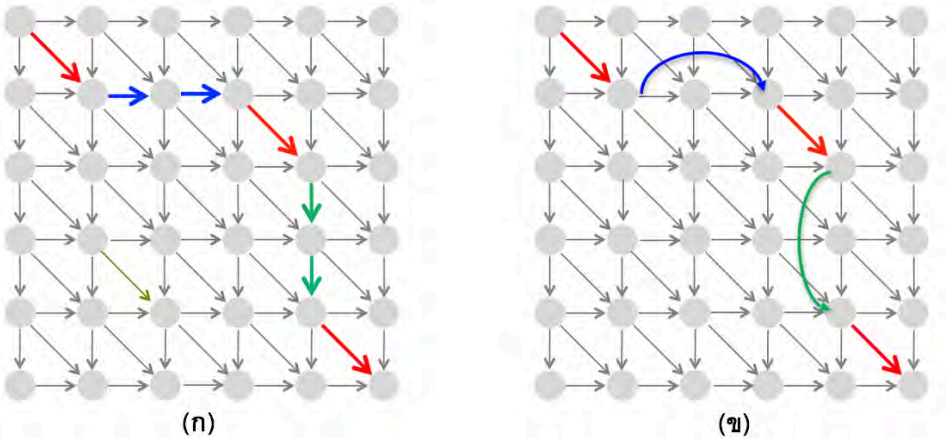
รูปที่ 5.10(ก) แสดงกราฟเปรียบเทียบลำดับเบสแบบปกติ ในขณะที่รูปที่ 5.10(ข) แสดงวิธีการปรับแต่งกราฟแสดงการเปรียบเทียบลำดับเบสที่มีการนำช่องว่างเข้ามาร่วมแสดงผลโดยมีการเพิ่มเส้นเชื่อมที่มีความยาวตามขนาดของแต่ละช่องว่างและเนื่องจากไม่สามารถทราบล่วงหน้าว่ามีช่องว่างเกิดขึ้นในบริเวณไหนบ้างในผลของการเปรียบเทียบจึงจำเป็นต้องเพิ่มเส้นเชื่อมที่แสดงทุกช่องว่างที่เป็นไปได้ ซึ่งหมายถึงต้องเพิ่มเส้นเชื่อมระหว่างโหนด (i,j) ไปยังโหนด $(i+k,j)$ และ $(i,j+k)$ โดยเส้นเชื่อมเหล่านี้มีค่าน้ำหนักเท่ากับ $\sigma + \epsilon \cdot (k-1)$ สำหรับทุกค่า k ที่เป็นไปได้ ดังแสดงในรูปที่ 5.11 สำหรับสายดีเอ็นเอสองเส้นที่แต่ละเส้นยาว n นิวคลีโอไทด์ จำนวนเส้นเชื่อมที่มีการพิจารณา affine gap penalty จะเพิ่มจาก $O(n^2)$ เป็น $O(n^3)$

หยุดคิด	DAG ที่มีจำนวนเส้นเชื่อมเท่ากับ $O(n^2)$ สามารถใช้แก้ปัญหาที่ 5.5 ได้หรือไม่

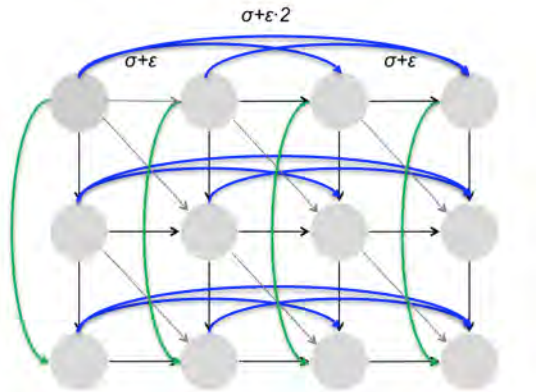
แผนที่สามระดับของเมืองแมนฮัตตัน

เราสามารถใช้ DAG ที่มีเส้นเชื่อม $O(n^2)$ ในการแก้ปัญหาที่ 5.5 โดยเพิ่มจำนวนโหนดผ่านการสร้างกราฟแสดงการเปรียบเทียบลำดับเบส 3 ระดับ (รูปที่ 5.12) โดยแต่ละโหนด (i,j) ในกราฟเดิมจะถูกจำลองออกมาเป็น 3 ระดับ คือ $(i,j)_{lower}$, $(i,j)_{middle}$, และ $(i,j)_{upper}$ โดยระดับกลาง (middle) เก็บเฉพาะเส้นเชื่อมในแนวเส้นทแยงมุมซึ่งแสดง

สถานะแมชหรือมิสแมช โดยมีน้ำหนักของเส้นเชื่อมเป็น $\mathtt{Score}(v_i, w_j)$ กราฟระดับล่าง (lower) และกราฟระดับบน (upper) เก็บเฉพาะเส้นเชื่อมที่ชี้ลงในแนวดิ่งซึ่งแสดงสถานะ gap extension ใน *v* และเส้นเชื่อมในแนวนอนซึ่งแสดงสถานะ gap extension ใน *w* โดยมีน้ำหนักของเส้นเชื่อมเป็น -ϵ ตามลำดับ



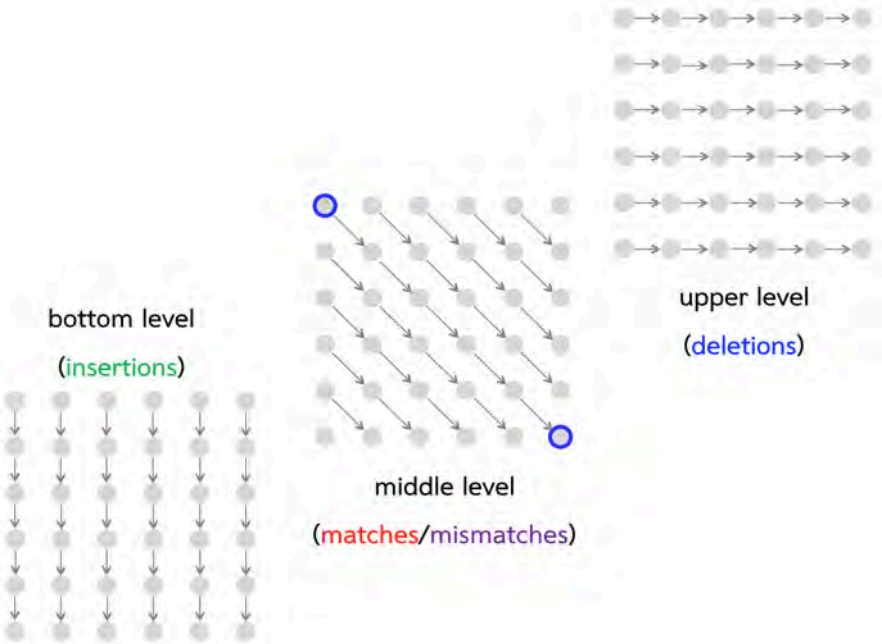
รูปที่ 5.10 (ก) กราฟเปรียบเทียบลำดับเบส (alignment graph) แบบปกติ (ข) กราฟเปรียบเทียบลำดับเบสโดยนำช่องว่าง (gap) เข้ามาแสดงเป็นส่วนหนึ่งของกราฟ



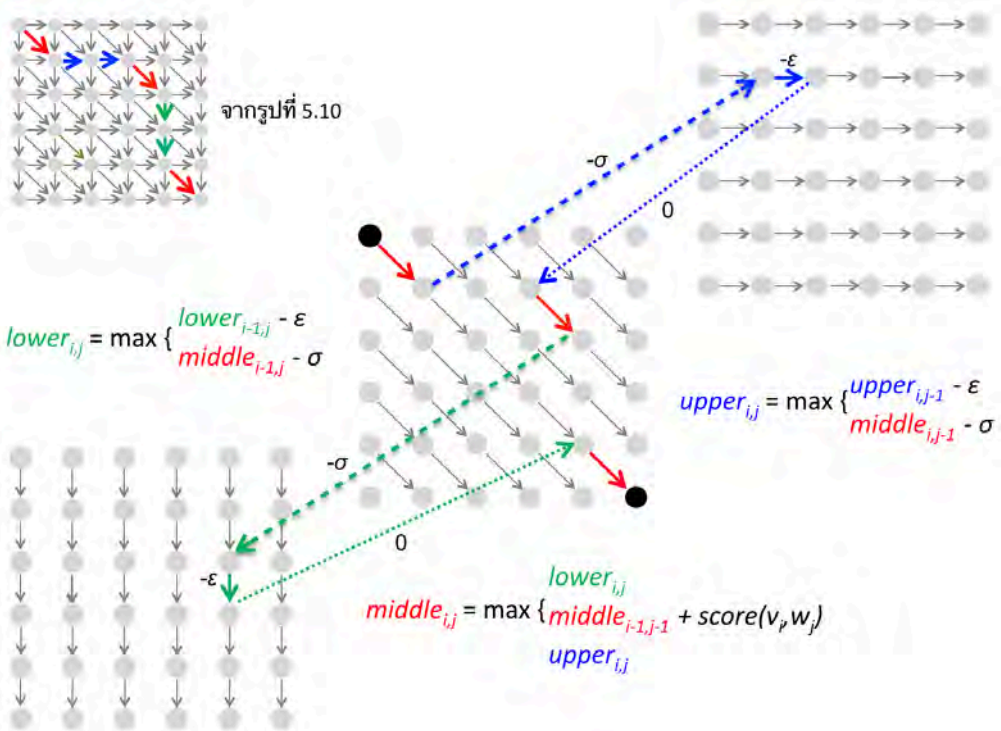
รูปที่ 5.11 จำนวนของเส้นเชื่อมที่เพิ่มขึ้นเมื่อมีการพิจารณาเรื่อง affine gap penalty

(ที่มา: ปรับจากรูปที่ 5.23 ของ [52])

เพื่อให้สามารถนำเรื่องของช่องว่าง (gap) มาร่วมพิจารณา แต่ละโหนด $(i,j)_{\text{middle}}$ จะมีเส้นเชื่อมไปยังโหนด $(i+1,j)_{\text{lower}}$ และ $(i,j+1)_{\text{upper}}$ ซึ่งเส้นเชื่อมทั้งสองนี้มีค่าน้ำหนักเท่ากับ $-\sigma$ สำหรับอินเดลแรกและ $-\epsilon$ สำหรับอินเดลถัดๆ ไปของช่องว่าง และ 0 สำหรับตำแหน่งที่ปิดช่องว่างซึ่งทำให้ได้คะแนนการลงโทษเป็น $\sigma + \epsilon \cdot (k-1)$ ตามที่ต้องการ รูปที่ 5.13 แสดงเส้นทางเดินของรูปที่ 5.10(ข) โดยใช้กราฟแสดงการเปรียบเทียบลำดับเบส 3 ระดับ DAG ที่แสดงในรูปที่ 5.13 นี้ถึงแม้ดูซับซ้อนแต่จำนวนเส้นเชื่อมที่ใช้จะเท่ากับ $\mathrm{O(nm)}$ สำหรับคู่ของสายดีเอ็นเอหรือโปรตีนที่มีความยาว n และ m ตามลำดับ และการหาเส้นทางที่ยาวที่สุดยังสอดคล้องกับวิธีการเปรียบเทียบความคล้ายคลึงกันโดยใช้ affine gap penalty ทั้งนี้กราฟแสดงการเปรียบเทียบลำดับเบส 3 ระดับสามารถแปลงให้เป็นชุดของความสัมพันธ์เวียนเกิด (recurrence relation) ดังแสดงในรูปที่ 5.13



รูปที่ 5.12 กราฟเปรียบเทียบลำดับเบส 3 ระดับเพื่อลดจำนวนเส้นเชื่อมที่ต้องใช้ในการแก้ปัญหาที่ 5.5 (ที่มา: รูปที่ 5.24 ของ [52])



รูปที่ 5.13 กราฟเปรียบเทียบลำดับเบส 3 ระดับเพื่อลดจำนวนเส้นเชื่อมที่ต้องใช้ในการแก้ปัญหาที่ 5.5 โดย $lower_{i,j}$, $middle_{i,j}$,และ $upper_{i,j}$ เป็นความยาวของเส้นทางที่ยาวที่สุดจากโหนดต้นทางไปยังโหนด $(i,j)_{lower}$, $(i,j)_{middle}$ และ $(i,j)_{upper}$ ตามลำดับ

บทส่งท้าย

วิธีการในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนข้างต้นเป็นการเปรียบเทียบระหว่างสายข้อมูลสองเส้นหรือที่เรียกว่า pairwise alignment ซึ่งสามารถนำไปประยุกต์ใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นหรือโปรตีนในชุดของข้อมูลหรือที่เรียกว่า multiple sequence alignment

การเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอหรือโปรตีนหลายเส้น

การเปรียบเทียบความคล้ายคลึงกันของสายอักขระจำนวน t เส้นประกอบด้วย v^1, ..., v^t เรียกว่าการทำ multiple sequence alignment หรือ t-way alignment ซึ่งสามารถแสดงโดยเมทริกซ์จำนวน t แถวโดยแถวที่ i จะมีลำดับอักขระของสายอักขระเส้นที่ i และอาจมีการแทรกช่องว่างในบางตำแหน่ง โดยมีสมมติฐานว่าไม่มีคอลัมน์ใดเลยที่มีแต่ช่องว่าง ในตัวอย่างของ 3-way alignment ต่อไปนี้ อักขระที่พบมากสุดในแต่ละตำแหน่งแสดงโดยตัวอักษรใหญ่

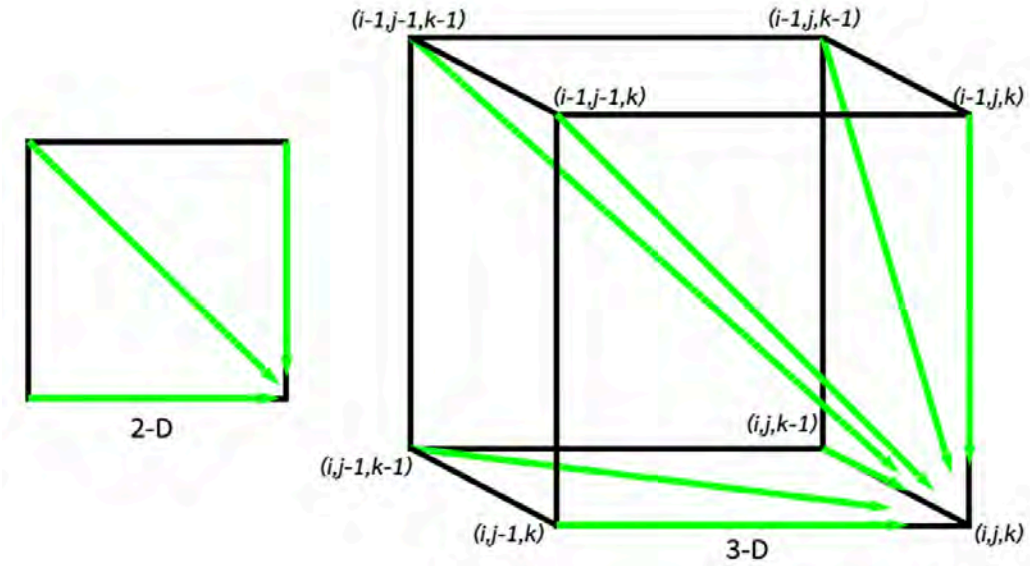
```
A  T  -  G  T  T  a  T  A
A  g  C  G  a  T  C  -  A
A  T  C  G  T  -  C  T  c
0  1  2  2  3  4  5  6  7  8
0  1  2  3  4  5  6  7  7  8
0  1  2  3  4  5  5  6  7  8
```

โดยเมทริกซ์นี้เป็นเมทริกซ์แบบทั่วไปของเมทริกซ์ที่แสดงผล pairwise alignment ส่วนอาร์เรย์ของค่าที่ตามมาอีก 3 บรรทัดแสดงจำนวนของอักขระที่ถูกใช้ไปแล้ว ณ คอลัมน์นั้นๆ ของสายอักขระสายที่ 1, 2 และ 3 ตามลำดับ โดยอาร์เรย์ของค่าเหล่านี้สอดคล้องกับเส้นทางในกริดสามมิติต่อไปนี้

(0,0,0) → (1,1,1) → (2,2,2) → (2,3,3) → (3,4,4) → (4,5,5) → (5,6,5) → (6,7,6) → (7,7,7) → (8,8,8)

ในขณะที่กราฟแสดงการเปรียบเทียบลำดับเบสระหว่างดีเอ็นหรือโปรตีนสองสายเป็นกริดในสองมิติ กราฟที่แสดงการเปรียบเทียบของดีเอ็นเอหรือโปรตีนสามสายสามารถแสดงโดยกริดในกล่องสามมิติ หรือที่เรียกว่าลูกบาศก์หรือคิวบ์ (cube) ดังแสดงในรูปที่ 5.14 การคำนวณคะแนนของเส้นทางหนึ่งๆ ในลูกบาศก์สามารถขยายจากการคำนวณคะแนนในการเปรียบเทียบระหว่างดีเอ็นเอหรือโปรตีนสองสายโดยการใช้กำหนดการพลวัตดังชุดของสมการที่แสดงในรูปที่ 5.14 โดยในกรณีที่มีสายข้อมูลจำนวน t เส้นและแต่ละเส้นยาว n อักขระ กราฟแสดงการเปรียบเทียบลำดับเบสจะประกอบด้วย n^t โหนด และแต่ละโหนดจะมีเส้นเชื่อมเข้ามากสุดจำนวน 2^t -1 เส้น ซึ่งหมายถึงต้องใช้เวลาในการรันเท่ากับ $O(n^t 2^t)$ ถ้า t มีจำนวนมากอัลกอริทึมที่ใช้กำหนดการพลวัตจะมีประสิทธิภาพไม่ดีพอ ในทางปฏิบัติจึงได้มีการนำเสนออัลกอริทึมต่างๆ ที่ใช้ฮิวริสติกในการหาคำตอบที่ใกล้เคียงแต่อาจไม่ใช่คำตอบที่ดีที่สุดโดยเน้นการลดเวลาในการรันอัลกอริทึมเพื่อหาคำตอบ

$$s_{i,j,k} = \max \begin{cases} s_{i-1,j,k} + Score(v_i, -, -) \\ s_{i,j-1,k} + Score(-, w_j, -) \\ s_{i,j,k-1} + Score(-, -, u_k) \\ s_{i-1,j-1,k} + Score(v_i, w_j, -) \\ s_{i-1,j,k-1} + Score(v_i, -, u_k) \\ s_{i,j-1,k-1} + Score(-, w_j, u_k) \\ s_{i-1,j-1,k-1} + Score(v_i, w_j, u_k) \end{cases}$$



รูปที่ 5.14 ลูกบาศก์แสดงกราฟเปรียบเทียบลำดับอักขระของสายอักขระสามสาย

(ที่มา: ปรับจากรูปที่ 5.31 ของ [52])

การเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอหรือโปรตีนหลายเส้นแบบละโมบ

จากตัวอย่างการเปรียบเทียบความคล้ายคลึงกันระหว่างดีเอ็นเอสามสายต่อไปนี้

```
AT-GTTaTA
AgCGaTC-A
ATCGT-CTc
```

สามารถนำไปสู่การเปรียบเทียบความคล้ายคลึงกันระหว่างคู่ของดีเอ็นเอสองสาย (pairwise alignment) จำนวนสามคู่

```
AT-GTTaTA    AT-GTTaTA    AgCGaTC-A
AgCGaTC-A    ATCGT-CTc    ATCGT-CTc
```

คำถามคือเราสามารถรวมผลการเปรียบเทียบความคล้ายคลึงกันระหว่างคู่ของสายดีเอ็นเอไปเป็นผลของการเปรียบเทียบชุดของสายดีเอ็นข้างต้นได้หรือไม่

หยุดคิด	1.ผลการเปรียบเทียบชุดของสายดีเอ็นหรือโปรตีนที่ดีที่สุดสามารถแยกออกเป็นชุดผลการเปรียบเทียบคู่ของสายดีเอ็นเอหรือโปรตีนที่ดีที่สุดได้หรือไม่ 2.จงรวมผลการเปรียบเทียบคู่ของสายดีเอ็นเอต่อไปนี้ให้เป็นผลการเปรียบเทียบชุดของสายดีเอ็น **CCCCTTTT**, **TTTTGGGG** และ **GGGGCCCC** **CCCCTTTT----** **----CCCCTTTT** **TTTTGGGG----** **----TTTTGGGG** **GGGGCCCC----** **----GGGGCCCC**

จากตัวอย่างการเปรียบเทียบชุดของสายดีเอ็นเอ **CCCCTTTT**, **TTTTGGGG** และ **GGGGCCCC** แสดงให้เห็นว่าเรา *ไม่* สามารถรวมชุดผลการเปรียบเทียบคู่ของสายดีเอ็นเอให้เป็นผลการเปรียบเทียบชุดของสายดีเอ็นเอได้เสมอไป ดังตัวอย่าง “หยุดคิด” ข้างต้น ผลการเปรียบเทียบคู่ของสายดีเอ็นเอทางซ้ายมืออนุมานได้ว่า ลำดับเบส **CCCC** มาก่อน **TTTT** ในขณะที่ผลการเปรียบเทียบคู่ของสายดีเอ็นเอทางขวามืออนุมานได้ว่า ลำดับเบส **TTTT** มาก่อน **GGGG** ในขณะที่ผลการเปรียบเทียบคู่ของสายดีเอ็นเอตรงกลางอนุมานได้ว่าลำดับเบส **GGGG** มาก่อน **CCCC** ซึ่งสรุปได้ว่า **CCCC** ต้องมาก่อน **TTTT** และ **TTTT** ต้องมาก่อน **GGGG** และ **GGGG** ต้องมาก่อน **CCCC** ซึ่งเกิดความขัดแย้งกับผลการถ่ายทอด (transitive) ที่ **CCCC** ต้องมาก่อน **GGGG**

เพื่อเป็นการหลีกเลี่ยงการเกิดความขัดแย้งกันเองนี้ อัลกอริทึมแบบละโมบบางอัลกอริทึมจะพยายามสร้างผลการเปรียบเทียบชุดของสายดีเอ็นเอจากชุดผลการเปรียบเทียบระหว่างคู่ของสายดีเอ็นเอโดยใช้ฮิวริสติก (heuristic) ในการเลือกคู่ของสายดีเอ็นเอที่มีความคล้ายคลึงกันที่สุดก่อน และใช้ผลการเปรียบเทียบของสายดีเอ็นเอคู่นี้เป็นจุดตั้งต้น โดยในแต่ละรอบจะทำการเลือกดีเอ็นเอมา 1 สายจากชุดของสายดีเอ็นเอที่ยังเหลืออยู่ที่มีความคล้ายคลึงกับผลการเปรียบเทียบคู่ของสายดีเอ็นเอตั้งต้นที่สุด มาสร้างเป็นผลการเปรียบเทียบสายดีเอ็นเอ 3 สาย จากนั้นทำการเลือกดีเอ็นเอสายถัดไป และวนซ้ำกระบวนการข้างต้นจนกว่าจะไม่มีสายดีเอ็นเอเหลือในชุด คำถามคือการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอกับผลการเปรียบเทียบชุดของสายดีเอ็นเอที่กำลังสร้างอยู่นั้นต้องทำอย่างไร

ทั้งนี้ผลการเปรียบเทียบลำดับนิวคลีโอไทด์จำนวน k คอลัมน์สามารถแสดงโดยใช้โพรไฟล์เมทริกซ์ต่อไปนี้ (4 x k ในกรณีของดีเอ็นเอ และ 20 x k ในกรณีของโปรตีน) โดยอัลกอริทึมแบบละโมบจะทำการเพิ่มดีเอ็นเอเข้าไป 1 เส้นที่ใกล้เคียงกับโพรไฟล์ปัจจุบันมากที่สุด จากนั้นสร้างโพรไฟล์ใหม่โดยรวมเส้นที่เพิ่มเข้ามาและทำซ้ำจนหมดชุดของสายดีเอ็นเอ โดยปัญหาการเปรียบเทียบชุดของสายดีเอ็นเอหรือโปรตีนจะกลายเป็นปัญหาการเปรียบเทียบคู่ของสายดีเอ็นเอหรือโปรตีนจำนวน t-1 ครั้ง

วิธีการเปรียบเทียบความคล้ายคลึงกันของชุดของสายดีเอ็นเอหรือโปรตีนแบบละโมบข้างต้นจะทำงานได้ดีในกรณีที่สายดีเอ็นเอหรือโปรตีนมีความคล้ายคลึงกันมาก อย่างไรก็ตามถ้าสายดีเอ็นเอหรือโปรตีนที่นำมาเปรียบเทียบมีความแตกต่างกันมากประสิทธิภาพของอัลกอริทึมจะลดลงอย่างมากโดยเฉพาะในกรณีสายดีเอ็นเอหรือโปร

ตีนที่ถูกเลือกเป็นคู่ตั้งต้นไม่ใช่ตัวแทนของคำตอบที่ดีที่สุด รวมทั้งมีผลต่อการเพิ่มความผิดพลาดในการเลือกสายดีเอ็นเอมาสร้างเป็นโพรไฟล์เมทริกซ์ในรอบถัดๆ ไป

		T	C	G	G	G	-	g	T	T	T	t	t
		c	C	-	-	t	G	A	c	T	T	a	C
		a	C	G	-	G	G	A	T	T	T	t	C
		T	t	G	G	G	-	A	c	T	T	t	t
Alignment		a	-	-	-	G	-	-	-	T	-	C	-
		T	t	G	G	G	G	A	c	T	T	C	C
		T	C	G	-	-	G	A	T	T	c	a	t
		-	-	-	G	G	G	A	T	T	c	C	-
		T	a	G	G	G	G	A	a	c	-	-	C
		T	C	G	G	G	t	A	T	a	a	C	C
Profile	A:	.2	.1	0	0	0	0	.8	.1	.1	.1	.2	0
	C:	.1	.5	0	0	0	0	0	.3	.1	.2	.4	.5
	G:	0	0	.7	.6	.8	.6	.1	0	0	0	0	0
	T:	.6	.2	0	0	.1	.1	0	.5	.8	.6	.2	.3

วิธีการอื่นๆ ที่มีการนำเสนอ เช่น การทำ progressive alignment ซึ่งเป็นวิธีการที่ใช้ในโปรแกรม CLUSTAL [127] มีขั้นตอนหลักประกอบด้วยการเปรียบเทียบความคล้ายคลึงกันระหว่างทุกคู่ของสายข้อมูลโดยใช้อัลกอริทึมนีเดอมาน-วานซ์ (Needleman-Wunsch) [128] ซึ่งทำการเปรียบเทียบคู่ของสายข้อมูลแบบครอบคลุม (global alignment) และสร้างผลการเปรียบเทียบเป็นเมทริกซ์แสดงระยะทางระหว่างทุกคู่ของสายข้อมูล ข้อมูลในเมทริกซ์นี้จะถูกนำไปใช้ในการสร้างต้นไม้แสดงความสัมพันธ์ระหว่างสายข้อมูล (guide tree) จากต้นไม้นี้สายข้อมูลสองเส้นที่มีความคล้ายคลึงกันมากที่สุดจะถูกเลือกมาเปรียบเทียบกันอีกครั้งโดยใช้อัลกอริทึมนีเดอมาน-วานซ์แบบข้างต้น และผลการเปรียบเทียบจะถูกแปลงเป็นสายอักขระเสียงข้างมาก (consensus string) ซึ่งจะนำไปใช้เปรียบเทียบกับสายข้อมูลอื่นๆ ต่อ เสมือนกับเป็นสายข้อมูลหนึ่งเส้น โดยผลของการเปรียบเทียบระหว่างสายข้อมูลเส้นที่มีความคล้ายคลึงมากสุดกับสายอักขระเสียงข้างมากข้างต้นจะถูกนำมาสร้างเป็นสายอักขระเสียงข้างมากสายใหม่และทำซ้ำจนหมดชุดของสายข้อมูล

การเปรียบเทียบสายดีเอ็นเอหรือโปรตีนกับฐานข้อมูลขนาดใหญ่

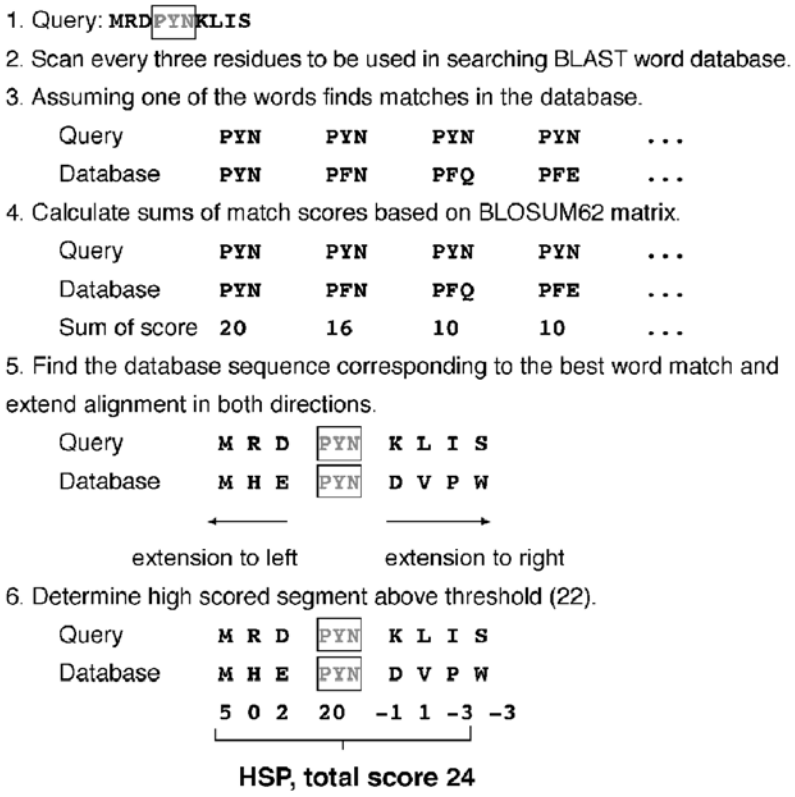
โปรแกรม BLAST (Basic Local Alignment Search Tool) [1] ถูกพัฒนาโดยสตีเฟ่น อัลท์ชูล(Stephen Altschul) และคณะ ที่เอ็นซีบีไอในปี ค.ศ. 1990 และกลายเป็นโปรแกรมที่มีการใช้งานกันอย่างแพร่หลายและต่อเนื่องมาจนปัจจุบัน (โปรแกรม BLAST ถูกอ้างอิง 70,106 ครั้ง จากผลการสืบค้นกูเกิลเมื่อวันที่ 10 ก.พ. พ.ศ. 2561/ค.ศ.

2018) โดยโปรแกรม BLAST ทำหน้าที่ในการรับสายดีเอ็นเอหรือโปรตีนเป็นข้อมูลเข้าและใช้ฮิวริสติกในการเทียบสายข้อมูลเข้านั้นกับสายข้อมูลทั้งหมดที่อยู่ในฐานข้อมูล มีวัตถุประสงค์หลักเพื่อหาส่วนของสายข้อมูลในฐานข้อมูลที่มีความอนุรักษ์ร่วมกับสายข้อมูลเข้า ส่วนของสายข้อมูลที่พบความอนุรักษ์ในฐานข้อมูลบ่งชี้ความคล้ายคลึงกันมากกว่าความคล้ายคลึงกันโดยบังเอิญ รูปที่ 5.15 แสดงการทำงานของโปรแกรม BLAST โดยขั้นตอนแรกสร้างรายการคำจากสายข้อมูลเข้า แต่ละคำประกอบด้วย 3 กรดแอมิโนของสายโปรตีน (ในกรณีที่เป็นสายดีเอ็นเอ แต่ละคำจะประกอบด้วย 11 นิวคลีโอไทด์) ชุดของคำนี้เรียกว่า seeding ขั้นตอนที่สองตรวจสอบในฐานข้อมูลว่ามีสายโปรตีนใดบ้างที่ประกอบด้วยคำที่อยู่ในสายข้อมูลเข้า การตรวจสอบความตรงกันของคำนี้ให้คะแนนโดยใช้เมทริกซ์คะแนนบลอสซัม 62 (BLOSUM62) (สามารถเปลี่ยนเป็นเมทริกซ์คะแนนอื่นได้) ในการคำนวณคะแนนการแทนที่ของกรดแอมิโนหนึ่งด้วยกรดแอมิโนอื่น การตัดสินว่าแต่ละคำที่นำมาเทียบนั้นตรงกับคำในฐานข้อมูลหรือไม่ขึ้นอยู่กับเกณฑ์คะแนนที่กำหนดไว้ ขั้นถัดไปทำการเปรียบเทียบสายข้อมูลเข้ากับสายโปรตีนหนึ่งๆ ที่มีคำที่ตรงกันโดยการทำ pairwise alignment การเปรียบเทียบนี้เริ่มจากส่วนของคำที่เหมือนกันและทำการขยายการเปรียบเทียบโดยใช้ลำดับกรดแอมิโนทั้งซ้ายและขวาของคำตั้งต้น โดยการเปรียบเทียบนี้จะทำการขยายจำนวนกรดแอมิโนทั้งซ้ายและขวาไปเรื่อยๆ จนกว่าคะแนนที่ได้จะลดลงต่ำกว่าค่าเกณฑ์ที่กำหนดเนื่องจากเกิดมิสแมช (ค่าเกณฑ์คะแนนในการหยุดการเปรียบเทียบนี้เท่ากับ 22 สำหรับโปรตีนและ 20 สำหรับดีเอ็นเอ) ส่วนของสายโปรตีนที่ตรงกันและไม่มีช่องว่าง (gap) เรียกว่า high-scoring segment pair (HSP) โดยโปรแกรม BLAST เวอร์ชันแรกรายงาน HSP ที่มีคะแนนสูงสุด ในเวอร์ชันหลังจากนั้นมีการอนุญาตให้มีช่องว่างในการเปรียบเทียบได้เรียกว่า gapped BLAST ซึ่งในกรณีนี้ HSP ที่มีคะแนนสูงสุดจะถูกเลือกมาเปรียบเทียบกับสายข้อมูลเข้าโดยใช้กำหนดการพลวัตและทำการขยายจำนวนกรดแอมิโนที่นำมาเปรียบเทียบทั้งซ้ายและขวาไปเรื่อยๆ จนกว่าค่าคะแนนจะลดลงเหมือนในเวอร์ชันก่อนหน้า อย่างไรก็ตามโปรแกรม BLAST อนุญาตให้คะแนนลดลงได้ชั่วคราวถ้าหลังจากนั้นยังสามารถเพิ่มคะแนนให้กลับมาสูงกว่าเกณฑ์ได้

ชุดโปรแกรม BLAST

โปรแกรม BLAST ได้ถูกออกแบบและพัฒนามาเป็นชุด โดยโปรแกรมภายชุดประกอบด้วย BLASTN, BLASTP, BLASTX, TBLASTN, และ TBLASTX ความแตกต่างหลักของแต่ละโปรแกรมภายในชุดคือประเภทของสายข้อมูลเข้า โปรแกรม BLASTN ทำการเทียบสายข้อมูลเข้าที่เป็นลำดับนิวคลีโอไทด์กับฐานข้อมูลลำดับเบสนิวคลีโอไทด์ (เช่น ฐานข้อมูลนิวคลีโอไทด์ nt ที่เอ็นซีบีไอ มีลำดับเบสนิวคลีโอไทด์ 254,826,802 เส้น เข้าถึงเมื่อวันที่ 10 ก.พ. พ.ศ. 2561) โปรแกรม BLASTP ทำการเทียบสายข้อมูลเข้าที่เป็นลำดับกรดแอมิโนกับฐานข้อมูลโปรตีน (เช่น non-redundant protein database (nr) ที่เอ็นซีบีไอ มีลำดับกรดแอมิโน 482,053,249 เส้น เข้าถึงเมื่อวันที่ 10 ก.พ. พ.ศ. 2561) โปรแกรม BLASTX ทำการเทียบสายข้อมูลเข้าที่เป็นลำดับนิวคลีโอไทด์กับฐานข้อมูลโปรตีนโดยก่อนนำไปเทียบจะทำการแปลงลำดับนิวคลีโอไทด์ให้เป็นลำดับกรดแอมิโนทั้ง 6 เฟรม ในขณะที่โปรแกรม TBLASTN จะทำการเทียบสายข้อมูลเข้าที่เป็นลำดับกรดแอมิโนกับฐานข้อมูลนิวคลีโอไทด์โดยก่อนนำไปเทียบจะทำการแปลง

ลำดับกรดแอมิโนเป็น 6 เฟรมในระดับนิวคลีโอไทด์ก่อน โปรแกรม TBLASTX ทำการเทียบสายข้อมูลเข้าที่เป็นลำดับนิวคลีโอไทด์กับฐานข้อมูลนิวคลีโอไทด์โดยก่อนเทียบจะทำการแปลงลำดับนิวคลีโอไทด์ทั้ง 6 เฟรมเป็นลำดับกรดแอมิโนเพื่อเทียบกับลำดับนิวคลีโอไทด์ในฐานข้อมูลที่ถูกแปลงทั้ง 6 เฟรมเป็นกรดแอมิโนก่อนเทียบเช่นกัน



รูปที่ 5.15 ขั้นตอนหลักในการทำงานของโปรแกรม BLAST โดยคะแนนที่ใช้ในตัวอย่างเป็นเมทริกซ์คะแนน BLOSUM62 ตัวอย่างคำที่ตรงกับส่วนของสายข้อมูลเข้า (PYN) แสดงในกล่อง

(*ที่มา*: *รูปที่* 4.1 *ของ [129]*)

จากแนวทางการเปรียบเทียบสายดีเอ็นเอหรือโปรตีนกับฐานข้อมูลขนาดใหญ่ข้างต้น และชุดโปรแกรม BLAST ที่มีโปรแกรมย่อยที่ทำงานแตกต่างกันไป งานวิจัยของผู้เขียนในโครงการชีวสารสนเทศเพื่อการศึกษาและวิเคราะห์ระบบจังหวะหนึ่งวันนาฬิกาและความสัมพันธ์ของระบบต่อการออกดอกในพืช ได้ใช้แนวทางข้างต้นเป็นส่วนประกอบในการหาไมโครอาร์เอ็นเอ (microRNA) หรือเอ็มไออาร์เอ็นเอ (miRNA) ในพืช โดยจำลองการทำงานของนักชีววิทยาให้เป็นขั้นตอนทางคอมพิวเตอร์ที่สามารถทำงานได้กับข้อมูลขนาดใหญ่และมีความเป็นอัตโนมัติมากขึ้น งานวิจัยนี้สร้างฐานข้อมูลขนาดใหญ่ในรูปแบบฐานข้อมูล BLAST จากข้อมูลอีเอสที (EST: Expressed Sequence Tags) หรือลำดับเบสสายสั้นยาวประมาณ 200-500 คู่เบส ซึ่งเป็นลำดับเบสของดีเอ็นเอคู่สมหรือซีดีเอ็เอ (cDNA) รวบรวมจาก PlantGDB [130, 131] และใช้ข้อมูลไมโครอาร์เอ็นเอ (mature microRNA) ในพืชที่มี

การรายงานมาก่อนหน้าในฐานข้อมูล miRBase [92] เป็นข้อมูลเข้าเพื่อนำมาเทียบกับฐานข้อมูลที่สร้างขึ้นผ่านโปรแกรม BLASTN เพื่อหาสายอีเอสทีจากฐานข้อมูลที่มีบริเวณที่น่าจะเป็นไมโครอาร์เอ็นเอ จากนั้นทำการคัดกรองสายอีเอสเหล่านี้โดยใช้เป็นข้อมูลเข้าของโปรแกรม BLASTX กับฐานข้อมูลยูนิพรอตที่นำมาติดตั้งภายในหน่วยงาน และทำการคัดออกสายอีเอสทีที่มีความคล้ายคลึงกับสายโปรตีนในฐานข้อมูลยูนิพรอต จากนั้นทำการคัดกรองสายอีเอสทีที่เหลือเฉพาะกลุ่มที่เป็นอาร์เอ็นเอไม่กำหนดรหัส (noncoding RNA) โดยใช้เป็นข้อมูลเข้าของโปรแกรม BLASTN กับฐานข้อมูล Rfam [132] ทำการคัดออกสายอีเอสทีที่มีความคล้ายคลึงกับสายอาร์เอ็นเอในฐานข้อมูล Rfam และใช้สายอีเอสทีที่ผ่านการคัดกรองเป็นข้อมูลเข้าเพื่อทำนายโครงสร้างทุติยภูมิของอาร์เอ็น (RNA secondary structure) โดยใช้โปรแกรม UNAFold [133] ตัดกิ่งผลลัพธ์โครงสร้างทุติยภูมิของอาร์เอ็นเอออกเป็นชุดของ precursor miRNAs และทำการคัดออกสาย precursor miRNAs ที่อาจเป็นส่วนของลำดับกำหนดรหัส (coding sequence) โดยนำไปทำการ BLASTX กับฐานข้อมูลยูนิพรอตอีกครั้ง สาย precursor miRNAs ที่ผ่านการคัดกรองจะถูกนำไปตรวจสอบโดยชุดของกฎของวิกเตอร์ แอมบรอสและคณะ (Victor Ambros) [134] โดย precursor miRNAs ที่ผ่านการคัดกรองจะเป็นผลการทำนายไมโครอาร์เอ็นเอสุดท้าย ในการทำนายโปรตีนเป้าหมายของชุดไมโครอาร์เอ็นเอที่เป็นผลลัพธ์นี้ ไมโครอาร์เอ็นเอชุดนี้จะถูกนำไปใช้เป็นข้อมูลเข้าเพื่อทำการกราดตรวจ (scan) อีเอสทีทั้งหมดในฐานข้อมูล PlantGDB เพื่อหาอีเอสทีที่มีบริเวณที่เป็นคู่สมกับไมโครอาร์เอ็นเอ อีเอสทีที่มีบริเวณที่เป็นคู่สมจะถูกนำไปคัดกรองอีกครั้งโดยชุดของกฎของโรดส์ (Rhoades) และคณะ [135, 136] และนำอีเอสทีที่ผ่านการคัดกรองไปหาฟังก์ชันโดยการ BLASTX กับฐานข้อมูลยูนิพรอตต่อไป จากการประยุกต์ใช้ขั้นตอนข้างต้นกับสายอีเอสทีจำนวน 5,306,503 เส้นจากพืชรวม 173 ชนิดใน PlantGDB พบไมโครอาร์เอ็นเอ 128 กลุ่ม (family) รวม 4,006 รายการ จากพืชทั้งสิ้น 125 ชนิด โดยมีไมโครอาร์เอ็นเอ 78 กลุ่มที่พบโปรตีนเป้าหมายรวม 2,995 โปรตีนจากสายอีเอสทีที่ผ่านการกราดตรวจจำนวน 4,953 เส้น ผลลัพธ์ที่ได้จากการวิเคราะห์ทั้งหมดถูกนำมาสร้างเป็นฐานข้อมูลออนไลน์ชื่อ microPC [137] โดยรวมข้อมูลไมโครอาร์เอ็นเอที่พบในพืชที่มีการรายงานใน miRBase เข้ามาด้วย เพื่อให้ผู้ใช้สามารถสืบค้นข้อมูลได้โดยง่ายทั้งจากมุมมองกลุ่มไมโครอาร์เอ็นเอและชนิดของพืช ผู้ใช้สามารถเปรียบข้อมูลลำดับเบสอาร์เอ็นเอของตนเองกับข้อมูลที่อยู่ในระบบรวมทั้งใช้ระบบเพื่อทำนายไมโครอารเอ็นเอและโปรตีนเป้าหมายจากสายข้อมูลนิวคลีโอไทด์

นอกจากฐานข้อมูลออนไลน์ microPC ข้างต้น เพื่อให้ผู้ใช้สามารถวิเคราะห์หาไมโครอาร์เอ็นเอในพืช กับชุดข้อมูลจำนวนมากของตนเอง ผู้เขียนและคณะได้ออกแบบและพัฒนาซอฟต์แวร์ชื่อ C-mii [138] ซึ่งผู้ใช้สามารถดาวน์โหลดไปติดตั้งที่เครื่องของตนเองได้ โดยขั้นตอนการทำนายไมโครอาร์เอ็นเอและโปรตีนเป้าหมายเป็นชุดขั้นตอนเดียวกับที่ใช้ใน microPC โดย C-mii มีส่วนต่อประสานกราฟิกกับผู้ใช้ (graphical user interface) หรือกุย (GUI) เพื่อให้ผู้ใช้สามารถทำงานในแต่ละขั้นตอนได้โดยง่าย และผลลัพธ์ของการรันแต่ละขั้นตอนอยู่ในรูปแบบกราฟิก เช่น โครงสร้างทุติยภูมิของ primary miRNA และ precursor miRNA ผลของการทำ multiple sequence aligment ของสาย precursor miRNAs ผลการกำหนดฟังก์ชันรวมทั้งยีนออนโทโลยี (Gene

Ongology) หรือโก (GO) ให้กับโปรตีนเป้าหมาย ซึ่งแสดงผลได้ทั้งในรูปแบบตารางและต้นไม้ของยีนออนโทโลยี (GO tree) โดยผู้ใช้สามารถสำรวจผลการทำนาย ทำการคัดกรองเพิ่มเติม และส่งออกผลเฉพาะส่วนที่สนใจได้

ตัวอย่างโปรแกรมที่มีการใช้งานอย่างแพร่หลาย

นอกจากโปรแกรม BLAST ที่เป็นโปรแกรมหลักในการเทียบสายดีเอ็นเอหรือโปรตีนกับฐานข้อมูลขนาดใหญ่ เช่น ฐานข้อมูลโปรตีนที่เอ็นซีบีไอ (NCBI) (https://www.ncbi.nlm.nih.gov) และฐานข้อมูลโปรตีนยูนิพรอต UniProt (http://www.uniprot.org) แล้ว ยังมีชุดของโปรแกรมที่ใช้เปรียบเทียบความคล้ายคลึงกันระหว่างคู่ของสายข้อมูล (pair-wise alignment) เช่น โปรแกรม NEEDLE (EMBOSS) ที่ใช้อัลกอริทึมนีเดอมาน-วานซ์ (Needleman-Wunsch algorithm) ซึ่งทำงานแบบ global alignment และโปรแกรม Water ที่ใช้อัลกอริทึมสมิธ-วอเตอร์แมน (Smith-Waterman algorithm) [139] และโปรแกรม LALIGN ที่ EMBL-EBI (https://www.ebi.ac.uk/Tools/psa/) ซึ่งทำงานแบบ local alignment และตัวอย่างโปรแกรมที่ใช้ในการทำ multiple sequence alignment เช่น CLUSTAL W [140], T-Coffee [141], MAFFT [142], MUSCLE [143], Clustal Omega [144, 145] ซึ่งหลายเครื่องมือในกลุ่มนี้มีทั้งเวอร์ชันที่เป็นเว็บที่ EMBL-EBI (https://www.ebi.ac.uk/Tools/msa/) เวอร์ชันที่สามารถเรียกผ่าน REST API/SOAP API และเวอร์ชันที่สามารถดาวน์โหลดมาติดตั้งที่เครื่องผู้ใช้งาน

แบบฝึกหัดบทที่ 5

เขียนโปรแกรมเพื่อแก้ปัญหาที่เกี่ยวข้องกับการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีน โดยใช้โจทย์ที่โรซาลินด์ต่อไปนี้

1) Global Alignment with Scoring Matrix (http://rosalind.info/problems/glob/)
2) Local Alignment with Scoring Matrix (http://rosalind.info/problems/loca/)
3) Global Alignment with Scoring Matrix and Affine Gap Penalty (http://rosalind.info/problems/gaff/)

ภาคผนวกบทที่ 5

เมทริกซ์คะแนนแพม

ในกระบวนการคัดเลือกทางธรรมชาติ กรดแอมิโนแต่ละตัวสามารถถูกแทนที่ด้วยกรดแอมิโนอื่นด้วยความถี่ที่แตกต่างกัน เมทริกซ์คะแนนแพม (PAM: point accepted mutation scoring matrices) เป็นเมทริกซ์ที่แต่ละแถวและคอลัมน์เป็นตัวแทนของแต่ละกรดแอมิโน ในกรณีของชีวสารสนเทศเมทริกซ์คะแนนแพมถูกใช้เป็นเมทริกซ์คะแนนของการเกิดมิสแมชหรือการแทนที่กรดแอมิโนหนึ่งด้วยกรดแอมิโนอื่นซึ่งเรียกโดยทั่วไปว่า substitution matrix ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายของโปรตีน ค่าในแต่ละช่องของเมทริกซ์คะแนนแพม

$M(i,j)$ บ่งชี้โอกาสที่กรดแอมิโนแถว i จะถูกแทนที่ด้วยกรดแอมิโนในแต่ละคอลัมน์ j ผ่านกระบวนการแปรผันในธรรมชาติในวิวัฒนาการของสิ่งมีชีวิต โดยไม่ใช่การแทนที่แบบสุ่ม เมทริกซ์คะแนนแพมแต่ละเมทริกซ์ เช่น PAM_1, PAM_{50} ตัวเลข 1 และ 50 เกี่ยวเนื่องกับระยะเวลาในขบวนการวิวัฒนาการของลำดับกรดแอมิโนในสายของโปรตีนที่นำมาใช้ในการสร้างเมทริกซ์ PAM_n หมายถึงในลำดับกรดแอมิโนทุก 100 ตำแหน่งจะมี n การแปรผัน เมทริกซ์คะแนนแพมนำเสนอครั้งแรกโดย มากาเร็ต เดย์ฮอฟ (Margaret Dayhoff) ในปี ค.ศ. 1978 โดยเมทริกซ์คะแนนเหล่านี้สร้างจากการแปรผันของกรดแอมิโนจำนวน 1,572 ตำแหน่งที่พบในแผนภูมิวิวัฒนาการชาติพันธุ์ (phylogenetic tree) จากชุดของสายโปรตีนที่มีความคล้ายคลึงและเกี่ยวเนื่องกันมากจำนวน 71 กลุ่มโปรตีน (protein family) และมีความอนุรักษ์โดยรวมอย่างน้อย 85%

เมทริกซ์คะแนน PAM_1 สร้างขึ้นจากผลของการเปรียบเทียบชุดของโปรตีนที่มีความเหมือนกัน 99% โดย $M(i, j)$ หรือ mutation matrix จะเก็บจำนวนครั้งที่กรดแอมิโน i และ j ปรากฏในคอลัมน์เดียวกัน (i กลายเป็น j) หารด้วยจำนวนครั้งทั้งหมดที่กรดแอมิโน i ปรากฏในตำแหน่งต่างๆ ของทุกสายโปรตีนที่นำมาเปรียบเทียบ และถ้าให้ $f(j)$ เป็นความถี่ของการปรากฏกรดแอมิโน j ในสายของโปรตีนทั้งหมดที่นำมาเปรียบเทียบ ค่าในแต่ละช่องของเมทริกซ์คะแนนแพม 1 หรือ $PAM_1(i,j)$ มีค่าเท่ากับ

$$log\left(\frac{M(i,j)}{f(j)}\right)$$

สำหรับค่าในแต่ละช่องของเมทริกซ์คะแนนแพม n โดยที่ n มีค่ามากกว่า 1 สามารถคำนวณได้จากสมการต่อไปนี้

$$log\left(\frac{M^n(i,j)}{f(j)}\right)$$

รูปที่ 5.14 แสดงเมทริกซ์คะแนนแพม 250 (PAM_{250})

PAM250 matrix

Ala	A	2																	
Arg	R	-2	6																
Asn	N	0	0	2															
Asp	D	0	-1	2	4														
Cys	C	-2	-4	-4	-5	12													
Gln	Q	0	1	1	2	-5	4												
Glu	E	0	-1	1	3	-5	2	4											
Gly	G	1	-3	0	1	-3	-1	0	5										
His	H	-1	2	2	1	-3	3	1	-2	6									
Ile	I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5								
Leu	L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6							
Lys	K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5						
Met	M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6					
Phe	F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9				
Pro	P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6			
Ser	S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2		
Thr	T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3	
Trp	W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17
Tyr	Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10
Val	V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4
		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V

รูปที่ 5.16 เมทริกซ์คะแนนแพม 250 (PAM250)

(*ที่มา*: *รูปที่* 6.9 *ของ [146]*)

เมทริกซ์คะแนนบลอสซัม

เมทริกซ์คะแนนบลอสซัม (BLOSUM scoring matrices หรือ BLOcks SUbstitution Matrix) เป็นเมทริกซ์ที่มีลักษณะเดียวกับเมทริกซ์คะแนนแพม แต่ต่างจากเมทริกซ์คะแนนแพมในการได้มาซึ่งคะแนนในแต่ละช่อง สตีเวน เฮนนิคอฟ (Steven Henikoff) และโจรา เฮนนิคอฟ (Jora Henikoff) นำเสนอเมทริกซ์คะแนนบลอสซัมครั้งแรกในปี ค.ศ. 1992 [147] โดยพิจารณาลำดับกรดแอมิโนในบริเวณที่มีความอนุรักษ์สูงและไม่มีช่องว่าง ระหว่างชุดของโปรตีนที่อยู่ในกลุ่มเดียวกันในฐานข้อมูล BLOCKS [148] และทำการนับความถี่ของการเกิดกรดแอมิโนแต่ละตัวและความน่าจะเป็นที่กรดแอมิโนหนึ่งจะถูกแทนที่ด้วยอีกกรดแอมิโนหนึ่ง จากนั้นคำนวณคะแนนโดยใช้ log-odds ทุกเมทริกซ์คะแนนบลอสซัมคำนวณจากผลการเปรียบเทียบความคล้ายคลึงกันระหว่างสายโปรตีนภายในชุดที่เปอร์เซ็นต์หนึ่งๆ ตัวอย่างเช่น เมทริกซ์คะแนน BLOSUM80 สร้างจากชุดของโปรตีนที่มีความคล้ายคลึงกันโดยเฉลี่ย 80% โดยสายโปรตีนที่อยู่ในชุดไม่ได้อ้างอิงจากความใกล้เคียงกันในเชิงวิวัฒนาการอย่างในกรณีของเมทริกซ์คะแนนแพม ตัวอย่างของเมทริกซ์ BLOSUM62 แสดงในรูปที่ 5.15 ที่มาของหัวข้อนี้เรียบเรียงจาก [146]

BLOSUM62 matrix

Ala	A	4																	
Arg	R	-1	5																
Asn	N	-2	0	6															
Asp	D	-2	-2	1	6														
Cys	C	0	-3	-3	-3	9													
Gln	Q	-1	1	0	0	-3	5												
Glu	E	-1	0	0	2	-4	2	5											
Gly	G	0	-2	0	-1	-3	-2	-2	6										
His	H	-2	0	1	-1	-3	0	0	-2	8									
Ile	I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4								
Leu	L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4							
Lys	K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5						
Met	M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5					
Phe	F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6				
Pro	P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7			
Ser	S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4		
Thr	T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	
Trp	W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11
Tyr	Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7
Val	V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4
		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V

รูปที่ 5.17 เมทริกซ์คะแนนบลอสซัม 62 (BLOSUM62)

(*ที่มา*: *รูปที่* 6.10 *ของ [146]*)

ความแตกต่างระหว่างเมทริกซ์คะแนนแพมและบลอสซัม

สำหรับเมทริกซ์คะแนนแพมค่าตัวเลขที่มากเช่น PAM250 ใช้เป็นเมทริกซ์คะแนนเพื่อเปรียบเทียบความคล้ายคลึงกันของสายโปรตีนได้ดีในกรณีที่สายโปรตีนที่นำมาเปรียบเทียบกันนั้นค่อนข้างห่างกันในเชิงวิวัฒนาการ ในขณะที่ค่าที่น้อยลงจะใช้ได้ดีกับการเปรียบเทียบความคล้ายคลึงกันของสายโปรตีนที่มีความใกล้เคียงกันในเชิงวัฒนาการ สำหรับเมทริกซ์บลอสซัมค่าตัวเลขต่อท้ายแต่ละเมทริกซ์มีความหมายในเชิงตรงกันข้ามกับแพม โดยค่าตัวเลขมากเหมาะกับการนำไปใช้ในการเปรียบเทียบความคล้ายคลึงกันของสายโปรตีนที่มีความใกล้เคียงกัน ในขณะที่ตัวเลข

ต่อท้ายเมทริกซ์ที่มีค่าน้อย เช่น BLOSUM45 เหมาะกับการนำไปใช้ในการเปรียบเทียบชุดของโปรตีนที่แตกต่างกันมากกว่า สตีเวนและโจรา เฮนนิคอฟเปรียบเทียบและสรุปความเกี่ยวเนื่องกันระหว่างเมทริกซ์คะแนนสองแบบดังต่อไปนี้

PAM250 ≈ BLOSUM45

PAM120 ≈ BLOSUM60

PAM160 ≈ BLOSUM62

โปรแกรม BLAST ใช้เมทริกซ์คะแนน BLOSUM62 เป็นเมทริกซ์หลักในการให้คะแนนการแทนค่ากรดแอมิโนเมื่อเกิดมิสแมช (mismatch) ซึ่งใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอหรือโปรตีนที่เป็นข้อมูลเข้ากับฐานข้อมูลของสายโปรตีน โดยสรุปเมทริกซ์คะแนนแพมและบลอสซัมมีความแตกต่างกันดังต่อไปนี้

1. เมทริกซ์คะแนนแพมถูกสร้างโดยอ้างอิงจากความเกี่ยวข้องของโปรตีนในเชิงวิวัฒนาการ อัตราในการถูกแทนที่ของกรดแอมิโนในแต่ละตำแหน่งด้วยกรดแอมิโนอื่นคำนวณจากแผนภูมิวิวัฒนาการชาติพันธุ์ (phylogenetic tree) ที่สร้างขึ้นและจากสายของโปรตีนที่เป็นบรรพบุรุษ ในขณะที่เมทริกซ์คะแนนบลอสซัมได้จากการสังเกตข้อมูลที่เป็นผลจากการเปรียบเทียบชุดของสายโปรตีนในบริเวณที่ไม่มีช่องว่างและมีความอนุรักษ์ร่วมกันสูง ดังนั้นในทางปฏิบัติเมทริกซ์คะแนนแพมมักถูกใช้ในการเปรียบเทียบชุดของสายโปรตีนเพื่อสร้างต้นไม้วิวัฒนาการ ในขณะที่เมทริกซ์คะแนนบลอสซัมเหมาะกับการเปรียบเทียบสายของโปรตีนแบบเฉพาะที่ (local alignment)
2. เมทริกซ์คะแนนแพมถูกสร้างจากความยาวโดยรวมของสายโปรตีนซึ่งรวมทั้งบริเวณที่อนุรักษ์และไม่อนุรักษ์ดังนั้นจึงมีความเกี่ยวเนื่องกับการเปรียบเทียบสายของโปรตีนแบบครอบคลุม (global alignment) ในขณะที่เมทริกซ์คะแนนบลอสซัมถูกสร้างจากผลการเปรียบเทียบความคล้ายคลึงกันของสายโปรตีนเฉพาะที่ (local alignment) จำเพาะบริเวณที่มีความอนุรักษ์ระหว่างสายของโปรตีน

ถึงแม้เครื่องมือทางชีวสารสนเทศมักมีการเตรียมเมทริกซ์คะแนนหลักไว้ให้ใช้ เช่นโปรแกรม BLAST ใช้ BLOSUM62 เป็นเมทริกซ์คะแนนหลัก อย่างไรก็ตามในมุมมองของนักชีวสารสนเทศการเข้าใจคุณลักษณะ ข้อดี และข้อจำกัดของแต่ละเมทริกซ์ และสามารถเลือกใช้ได้ถูกต้องเหมาะสมตามวัตถุประสงค์ เป็นสิ่งที่สำคัญ โดยมีคำแนะนำในการเลือกใช้เมทริกซ์คะแนนดังต่อไปนี้

PAM100 ≈ BLOSUM90 (สำหรับเปรียบเทียบสายโปรตีนที่มีความใกล้เคียงกันมาก)

PAM120 ≈ BLOSUM80 (สำหรับเปรียบเทียบสายโปรตีนทั่วไปแทบทั้งหมด)

PAM160 ≈ BLOSUM62 (สำหรับเปรียบเทียบสายโปรตีนทั่วไปแทบทั้งหมด)

PAM200 ≈ BLOSUM52 (สำหรับเปรียบเทียบสายโปรตีนทั่วไปแทบทั้งหมด)

PAM250 ≈ BLOSUM45 (สำหรับเปรียบเทียบสายโปรตีนที่แตกต่างกันมาก)

รูปแบบไฟล์ที่เกี่ยวข้อง

CLUSTAL

ไฟล์ที่ใช้ในการเก็บผลการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนมีหลายรูปแบบ โดยรูปแบบที่เป็นที่รู้จักและมีการใช้งานกันทั่วไปคือรูปแบบ clustal (รูปที่ 5.18) ซึ่งเป็นรูปแบบไฟล์ผลลัพธ์จากโปรแกรม CLUSTAL W โดยเป็นแฟ้มข้อความ (text file) ที่มีรูปแบบจำเพาะ บรรทัดแรกเริ่มด้วยคำว่า CLUSTAL W แล้วตามด้วยเลขเวอร์ชันในวงเล็บ ส่วนชุดของบรรทัดที่เป็นผลการเปรียบเทียบจะกำหนดความยาวของสายข้อมูลในการแสดงผลไม่เกิน 60 อักขระต่อบรรทัด บรรทัดแรกในส่วนนี้แสดงสายข้อมูลตั้งต้น บรรทัดที่สองแสดงสายข้อมูลที่นำมาเทียบ และบรรทัดลำดับที่สามแสดงผลการเปรียบเทียบของแต่ละตำแหน่ง โดยสัญลักษณ์ ‘*’ แสดงสถานะแมช สัญลักษณ์ ‘:’ แสดงถึงการใช้แทนกันได้ระหว่างกรดแอมิโน สัญลักษณ์ ‘.’ แสดงถึงการใช้แทนกันได้แต่มีโอกาสเกิดน้อยกว่า ส่วนสัญลักษณ์ ‘-’ แสดงช่องว่าง (gap) และ ช่องว่าง ‘ ’ แสดงสถานะไม่แมช สำหรับเลขต่อท้ายแต่ละบรรทัดเช่น 60 ในสายข้อมูลส่วนแรก และ 120 ในสายข้อมูลส่วนที่สอง อาจมีหรือไม่มีก็ได้ โดยใช้แสดงจำนวนอักขระในบรรทัดนั้นๆ และเป็นค่าสะสม ทั้งนี้แต่ละเวอร์ชันอาจมีความแตกต่างกันไปบ้าง

```
CLUSTAL W (1.82) multiple sequence alignment


FOSB_MOUSE      MFQAFPGDYDSGSRCSSSPSAESQYLSSVDSFGSPPTAAASQECAGLGEMPGSFVPTVTA 60
FOSB_HUMAN      MFQAFPGDYDSGSRCSSSPSAESQYLSSVDSFGSPPTAAASQECAGLGEMPGSFVPTVTA 60
                ************************************************************

FOSB_MOUSE      ITTSQDLQWLVQPTLISSMAQSQGQPLASQPPAVDPYDMPGTSYSTPGLSAYSTGGASGS 120
FOSB_HUMAN      ITTSQDLQWLVQPTLISSMAQSQGQPLASQPPVVDPYDMPGTSYSTPGMSGYSSGGASGS 120
                *******************************.***************:*.**:****** 

FOSB_MOUSE      GGPSTSTTTSGPVSARPARARPRRPREETLTPEEEEKRRVRRERNKLAAAKCRNRRRELT 180
FOSB_HUMAN      GGPSTSGTTSGPGPARPARARPRRPREETLTPEEEEKRRVRRERNKLAAAKCRNRRRELT 180
                ****** ***** .********************************************** 

FOSB_MOUSE      DRLQAETDQLEEEKAELESEIAELQKEKERLEFVLVAHKPGCKIPYEEGPGPGPLAEVRD 240
FOSB_HUMAN      DRLQAETDQLEEEKAELESEIAELQKEKERLEFVLVAHKPGCKIPYEEGPGPGPLAEVRD 240
                ************************************************************

FOSB_MOUSE      LPGSTSAKEDGFGWLLPPPPPPPLPFQSSRDAPPNLTASLFTHSEVQVLGDPFPVVSPSY 300
FOSB_HUMAN      LPGSAPAKEDGFSWLLPPPPPPPLPFQTSQDAPPNLTASLFTHSEVQVLGDPFPVVNPSY 300
                ****:.******.**************:*:**************************.*** 

FOSB_MOUSE      TSSFVLTCPEVSAFAGAQRTSGSEQPSDPLNSPSLLAL 338
FOSB_HUMAN      TSSFVLTCPEVSAFAGAQRTSGSDQPSDPLNSPSLLAL 338
                ***********************:************** 
```

รูปที่ 5.18 รูปแบบไฟล์ CLUSTAL

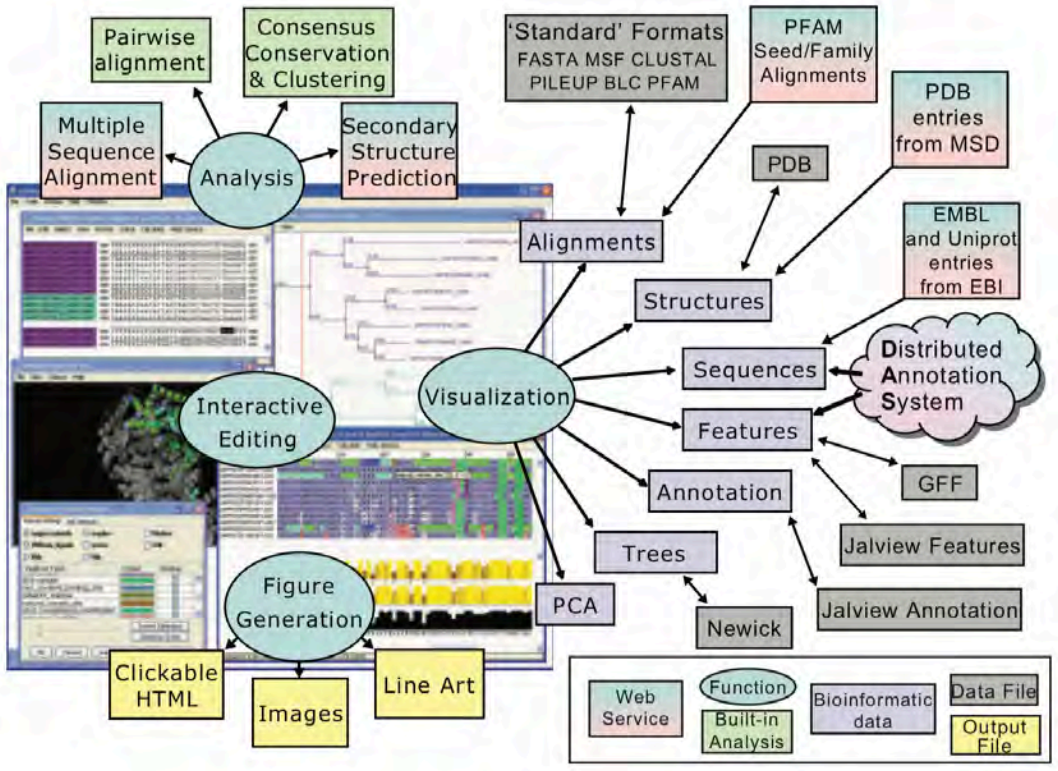
(*ที่มา*: http://web.mit.edu/meme_v4.11.4/share/doc/clustalw-format.html)

สำหรับรายละเอียดของรูปแบบไฟล์อื่นๆ ที่เป็นไปได้สามารถอ่านเพิ่มเติมได้จาก
http://emboss.open-bio.org/html/use/ch05s04.html

โปรแกรมที่ใช้ในการแก้ไข แสดงผล และเปรียบเทียบความคล้ายคลึงกันของชุดของสายดีเอ็นเอและหรือโปรตีน

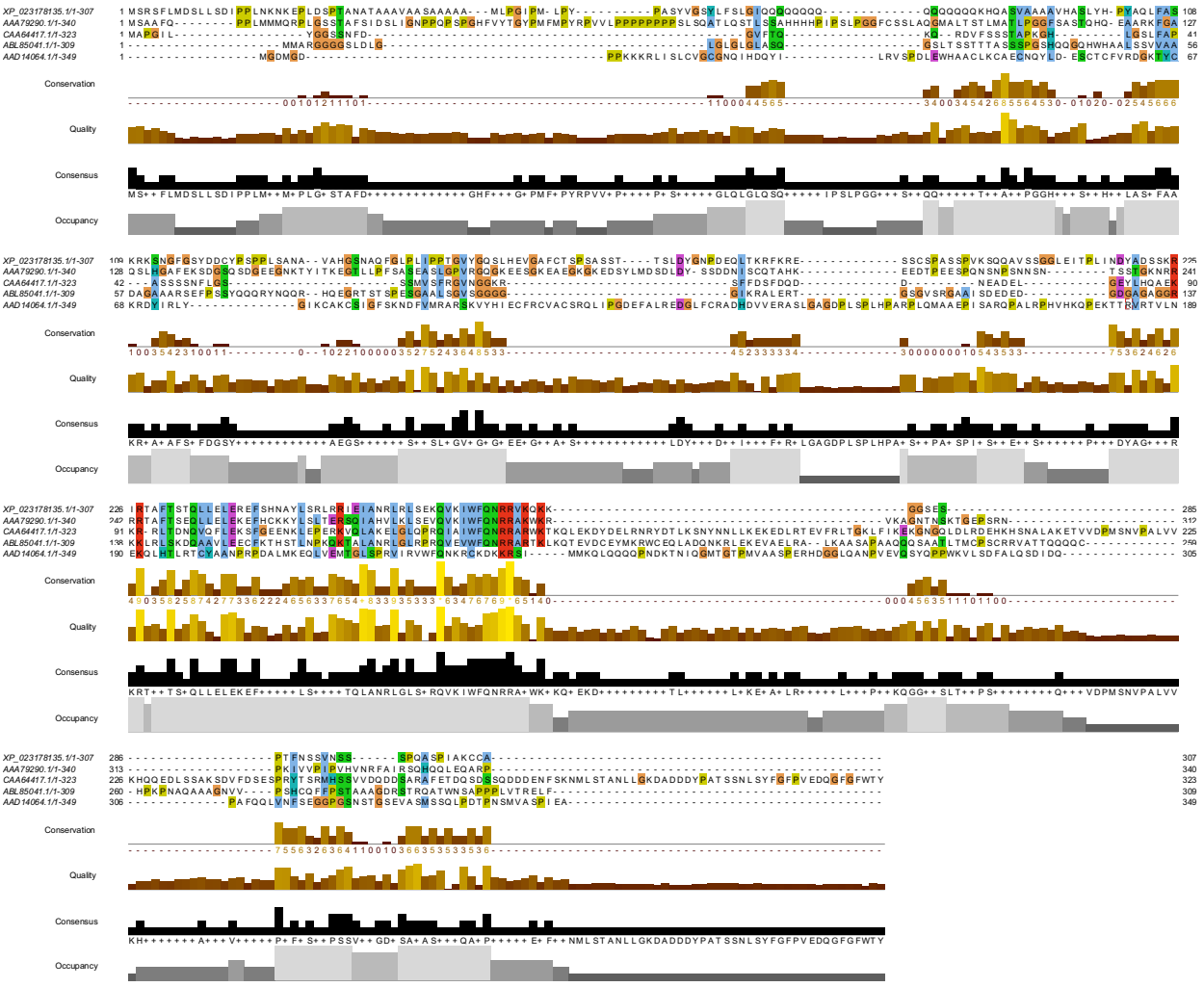
Jalview

โปรแกรม Jalview [149, 150] (รูปที่ 5.19) เป็นโปรแกรมโอเพนซอร์ส (open source) ที่ใช้ในการแก้ไข แสดงผล และเปรียบเทียบความคล้ายคลึงกันระหว่างสายข้อมูลประเภทเดียวกันภายในชุด ซึ่งอาจเป็นชุดของสายข้อมูลดีเอ็นเอ ชุดของสายข้อมูลอาร์เอ็นเอ หรือชุดของสายข้อมูลโปรตีน นอกจากนี้โปรแกรม Jalview ยังช่วยเชื่อมโยงข้อมูลอื่นๆ ที่เกี่ยวข้องเข้ามาในการวิเคราะห์และแสดงผลได้โดยอัตโนมัติ เช่นเชื่อมโยงโครงสร้างสามมิติของแต่ละโปรตีน (ถ้ามีโครงสร้างในฐานข้อมูลเปิด) ข้อมูลโปรตีนโดเมนจาก Pfam ข้อมูลกลุ่มอาร์เอ็นเอ (RNA family) จากฐานข้อมูล Rfam เป็นต้น รูปที่ 5.20 โปรแกรม Jalview แสดงผลของการรันโปรแกรม MUSCLE ซึ่งทำ multiple sequence alignment โดยมีข้อมูลเข้าเป็นโปรตีนฮอมีโอบอกซ์ 5 เส้นจากมนุษย์ (*Homo sapiens*), แมลงวัน (*Drosophila hydei*), กบเล็บแอฟริกา (*Xenopus laevis),* มะเขือเทศ (*Solanum lycopersicum)* และหญ้า (*Brachypodium sylvaticum*) โดยพบบริเวณจำเพาะมากกว่า 1 บริเวณที่มีความอนุรักษ์ร่วมกันระหว่างทั้ง 5 สิ่งมีชีวิต



รูปที่ 5.19 หน้าจอของโปรแกรม Jalview ฟังก์ชันการทำงาน และความสามารถในการเชื่อมโยงข้อมูลกับฐานข้อมูลสาธารณะ

(*ที่มา*: *รูปที่* 1 *ของ [150]*)



รูปที่ 5.20 โปรแกรม Jalview แสดงผลจากโปรแกรม MUSCLE ในการเปรียบเทียบความคล้ายคลึงกันของโปรตีนฮอมีโอบอกซ์ของสิ่งมีชีวิต 5 ชนิดคือ มนุษย์ (Homo sapiens)*, แมลงวัน* (Drosophila hydei)*, กบเล็บแอฟริกา* (Xenopus laevis)*, มะเขือเทศ* (Solanum lycopersicum) *และหญ้า* (Brachypodium sylvaticum)

บทที่ 6 การจำแนกฟีโนไทป์ของไวรัสเอชไอวี (HIV phenotypic classification)

วัตถุประสงค์

- เพื่อให้นิสิตเห็นตัวอย่างของปัญหาทางชีววิทยาที่อัลกอริทึมในบทเรียนก่อนหน้ายังตอบโจทย์ได้ไม่ดีพอ
- เพื่อให้นิสิตเห็นแนวทางที่แตกต่างในการเปรียบเทียบความคล้ายคลึงกันของสายข้อมูลดีเอ็นเอและหรือโปรตีน
- เพื่อให้นิสิตเข้าใจองค์ประกอบพื้นฐานของแบบจำลองมาร์คอฟซ่อนเร้น (Hidden Markov Model: HMM) แผนภาพ HMM ปัญหา Decoding อัลกอริทึมวิเทอบิ อัลกอริทึมฟอร์เวิร์ดแบคเวิร์ด อัลกอริทึมบอม-เวลช์
- เพื่อให้นิสิตเห็นตัวอย่างงานวิจัยและผลงานวิจัย รวมทั้งตัวอย่างโปรแกรมที่ใช้ HMM
- เพื่อให้นิสิตเห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทาย รวมทั้งงานวิจัยอื่นๆ ที่เกี่ยวข้อง

ผลลัพธ์ที่คาดหวัง

- นิสิตสามารถอธิบายความแตกต่างของการเปรียบเทียบความคล้ายคลึงกันระหว่างสายข้อมูลโดยวิธีการทำ sequence alignment ในบทที่ 5 และการใช้โพรไฟล์ HMM ในบทเรียนนี้
- นิสิตสามารถอธิบายองค์ประกอบหลักของ HMM สามารถเขียนแผนภาพ HMM กราฟวิเทอบิ และสามารถหาค่าพารามิเตอร์ที่เกี่ยวข้อง เช่น ค่าความน่าจะเป็นในการเปลี่ยนสถานะ ค่าความน่าจะเป็นในการส่งออกอักขระในสายลำดับข้อมูลที่ส่งออก การทำงานของอัลกอริทึมหลักที่เกี่ยวข้อง
- นิสิตสามารถเขียนโปรแกรมเพื่อใช้ HMM ในการแก้ปัญหาอย่างง่ายได้
- นิสิตสามารถยกตัวอย่างโปรแกรมและฐานข้อมูลที่ใช้ HMM เพื่อเปรียบเทียบความคล้ายคลึงกันระหว่างสายโปรตีนได้
- นิสิตสามารถประยุกต์องค์ความรู้จากบทเรียนนี้เพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้

เนื้อหาโดยสรุป

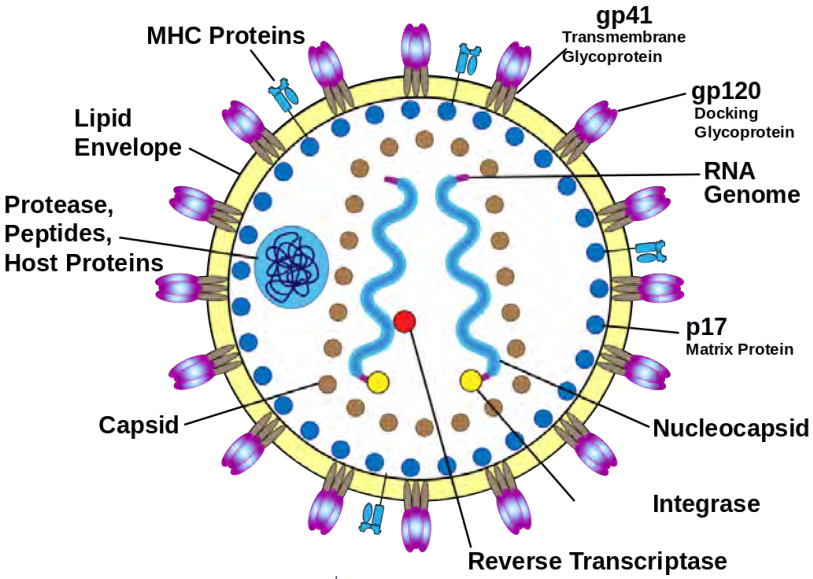
บทเรียนนี้ยกตัวอย่างโจทย์ทางชีววิทยาในเรื่องของการเปรียบเทียบลำดับกรดแอมิโนของโปรตีน gp120 บริเวณ V3 loop ของไวรัสเอชไอวีที่มีการแปรผันค่อนข้างมากและมีเพียงตำแหน่ง 11 และ 25 ที่ถูกใช้ระบุว่าเกี่ยวข้องกับลักษณะปรากฏหรือฟีโนไทป์ที่จำเพาะของเชื้อ ซึ่งการเปรียบเทียบความคล้ายคลึงกันของสายข้อมูลในบทเรียนก่อนหน้าไม่มีการพิจารณาค่าความน่าจะเป็นของการส่งออกอักขระในลำดับที่จำเพาะ บทเรียนนี้อธิบายแนวทางการเปรียบเทียบความคล้ายคลึงกันของสายข้อมูลโดยการประยุกต์ใช้แบบจำลองมาร์คอฟซ่อนเร้น (Hidden Markov Model: HMM) องค์ประกอบของ HMM แผนภาพ HMM กราฟวิเทอบิ และอัลกอริทึมที่เกี่ยวข้อง เช่น อัลกอริทึมวิเทอบิ (Viterbi algorithm) อัลกอริทึมฟอร์เวิร์ด-แบคเวิร์ด (forward-backward algorithm) อัลกอริทึมบอม-เวลช์ (Baum-Welch algorithm) ตัวอย่างโปรแกรมและฐานข้อมูลที่ประยุกต์ใช้โพรไฟล์ HMM ในการเปรียบเทียบความคล้ายคลึงกันของสายโปรตีน รวมทั้งตัวอย่างการประยุกต์ใช้ HMM ในการแก้ปัญหาอื่นๆ ทางชีววิทยา

บทที่ 6 การจำแนกฟีโนไทป์ของไวรัสเอชไอวี (HIV phenotypic classification)

ไวรัสเอชไอวีหลบเลี่ยงระบบภูมิคุ้มกันในร่างกายมนุษย์อย่างไร

ในปี ค.ศ. 1984 มาร์กาเร็ต เฮคเคลอร์ (Margaret Heckler) รัฐมนตรีประจำกระทรวงสาธารณสุข (US Health and Human Services) ของประเทศสหรัฐอเมริกาในขณะนั้นประกาศว่าจะมีวัคซีนเพื่อป้องกันโรคเอชไอวีภายในสองปีและในปี ค.ศ. 1997 ประธานาธิบดี บิล คลินตัน (Bill Clinton) อนุมัติศูนย์วิจัยใหม่ภายใต้สถาบันสุขภาพแห่งชาติ (National Institute of Health) หรือเอ็นไอเอช (NIH) โดยมีเป้าหมายเพื่อพัฒนาวัคซีนเอชไอวี ในปี ค.ศ. 2005 บริษัทเมอร์ค (Merck) ได้เริ่มการทดลองวัคซีนเอชไอวีทางคลินิกและหยุดการทดลองหลังผ่านไป 2 ปี หลังจากพบว่าวัคซีนที่ทดลองนั้นไม่ได้ลดความเสี่ยงของการติดเชื้อ

ปัจจุบันยังไม่มีวัคซีนเอชไอวีที่ได้รับการรับรองแม้ว่าจะมีการลงทุนอย่างมากมายรวมถึงมีการทดลองเชิงคลินิกอย่างต่อเนื่อง และมีประชากร 36.7 ล้านคนทั่วโลกที่เป็นผู้ติดเชื้อ (ที่มา: https://www.hiv.gov/hiv-basics/overview/data-and-trends/global-statistics เข้าถึงออนไลน์เมื่อวันที่ 11 ก.พ. พ.ศ. 2561) ทั้งนี้งานวิจัยและพัฒนาในเชิงการรักษามีความก้าวหน้าเป็นอย่างมาก และประสบความสำเร็จในการพัฒนายาต้านรีโทรไวรัสประกอบด้วยชุดของยาที่ทำให้อาการของผู้ป่วยติดเชื้อคงที่ อย่างไรก็ตามการรักษาด้วยวิธีนี้ไม่สามารถทำให้หายขาดจากโรค และไม่สามารถควบคุมการแพร่กระจายของเชื้อ



รูปที่ 6.1 ไวรัสเอชไอวี

(*ที่มา:* Zionlion77, Public domain, via Wikimedia Commons. 2009. *Schematic representation of an HIV-Virion.* [*เข้าถึงออนไลน์เมื่อวันที่* 11 *ก.ค. พ.ศ.* 2564])

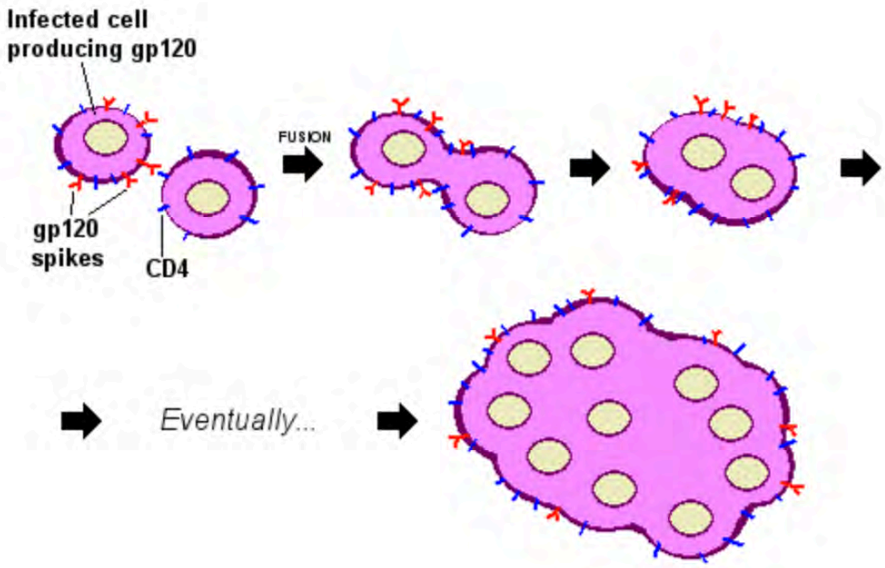
วัคซีนที่ใช้ในการป้องกันเชื้อไวรัสมักถูกสร้างจากโปรตีนที่ผนังเซลล์ของไวรัส (รูปที่ 6.1) โดยวัคซีนเหล่านี้จะกระตุ้นภูมิคุ้มกันของมนุษย์ให้รู้จักโปรตีนที่ผิวเซลล์ของไวรัสว่าเป็นโปรตีนแปลกปลอม ต้องกำจัด และจดจำไว้เพื่อคอยคุ้มกันเซลล์ภายในร่างกายจากไวรัส อย่างไรก็ตามโปรตีนที่ผิวเซลล์ของไวรัสเอชไอวีมีการแปรผันของลำดับกรดแอมิโนมากเนื่องจากไวรัสต้องพยายามเปลี่ยนแปลงตัวเองให้รวดเร็วเพื่อความอยู่รอด ประชากรของเชื้อเอชไอวีในผู้ติดเชื้อรายหนึ่งๆ มีการเปลี่ยนแปลงตัวเองไปอย่างรวดเร็วเพื่อให้สามารถหลีกเลี่ยงภูมิคุ้มกันของผู้ติดเชื้อได้ รูปที่ 6.2 แสดงผลของการทำ multiple sequence alignment ส่วนของโปรตีน gp120 ที่เก็บจากผู้ติดเชื้อ 1 รายใน 9 ช่วงเวลาที่แตกต่างกัน โดยคอลัมน์ที่เป็นสีเข้มแสดงส่วนที่แตกต่างกันระหว่างช่วงเวลา และคอลัมน์สีฟ้าแสดงกรดแอมิโนที่แตกต่างจากกรดแอมิโนหลักของคอลัมน์นั้นๆ ทั้งนี้เพื่อแสดงให้เห็นว่าเชื้อเอชไอวีมีการปรับเปลี่ยนตัวเองได้รวดเร็วมาก ซึ่งถ้านำเชื้อเอชไอวีของผู้ติดเชื้อหลายๆ คนมาเทียบกันก็จะมีความแปรผันมากขึ้นอีก ดังนั้นวัคซีนเอชไอวีที่มีประสิทธิภาพต้องมีความครอบคลุมในการรู้จักโปรตีนของไวรัสเอชไอวีที่มีรูปแบบที่หลากหลาย โดยอาจสร้างสายเพปไทด์เพื่อจำลองส่วนที่มีการแปรผันน้อยสุดของโปรตีนที่ผิวเซลล์ของไวรัสเอชไอวีและใช้เพปไทด์นี้เป็นวัคซีน อย่างไรก็ตามนอกจากการแปรผันที่รวดเร็วของโปรตีนที่ผิวเซลล์แล้ว โปรตีนเหล่านี้ยังสามารถหลบเลี่ยงโดยกระบวนการไกลโคซิเลชัน (glycosylation) ซึ่งเป็นการดัดแปรโมเลกุลของโปรตีนหลังการแปลรหัส (posttranslational modification) เหมือนการใส่หน้ากากซึ่งทำให้สามารถหลบซ่อนได้จากระบบภูมิคุ้มกันของผู้ติดเชื้อ ผลคือยังไม่มีวัคซีนเอชไอวีที่ใช้งานได้อย่างมีประสิทธิภาพ

รูปที่ 6.2 ผลของการทำ multiple sequence alignment ส่วนของโปรตีน gp120 ที่เก็บจากผู้ติดเชื้อ 1 รายใน 9 ช่วงเวลาที่แตกต่างกัน

(*ที่มา*: *รูปที่* 10.1 *ของ* [52])

ไวรัสเอชไอวีประกอบด้วย 9 ยีน ในบทเรียนนี้ให้ความสนใจยีน *env* ที่มีอัตราการเปลี่ยนแปลงลำดับเบสสูงมากคือประมาณ 1-2% ต่อนิวคลีโอไทด์ต่อปี โดยโปรตีนที่แปลรหัสจากยีน *env* นี้จะถูกตัดออกเป็นสองโปรตีนคือ ไกลโคโปรตีน จีพี 120 (glycoprotein gp 120) ที่มีความยาวประมาณ 480 กรดแอมิโน และไกลโคโปรตีน จีพี 41 (glycoprotein gp 41) ที่มีความยาวประมาณ 345 กรดแอมิโน โดยโปรตีนทั้งสองนี้จับกันเป็น envelope spike ทำให้สามารถนำไวรัสเอชไอวีเข้าสู่เซลล์เจ้าบ้านได้

เนื่องจากไวรัสเอชไอวีมีการกลายพันธุ์อย่างรวดเร็ว เชื้อเอชไอวีที่ตรวจพบอาจมีลักษณะที่ปรากฏหรือฟีโนไทป์ (phenotype) ที่แตกต่างกัน ซึ่งการรักษาต้องใช้ชุดของยาที่แตกต่างกัน ตัวอย่างเช่น เชื้อเอชไอวีอาจแบ่งออกเป็นกลุ่มที่สามารถเพิ่มจำนวนได้อย่างรวดเร็วและชักนำให้เกิดเซลล์หลายนิวเคลียส (syncytium inducing: SI) และกลุ่มที่เพิ่มจำนวนช้าและไม่ชักนำให้เกิดเซลล์หลายนิวเคลียส (non-syncytium inducing: NSI) โดยกลุ่มที่เป็น SI นั้น โปรตีน gp120 ของไวรัสจะถูกเคลื่อนย้ายไปที่ผิวเซลล์ของไวรัสโดยโปรตีนนี้สามารถทำให้ผิวเซลล์หลายๆ เซลล์ของผู้ติดเชื้อรวมเข้าด้วยกันกลายเป็นเซลล์หลายนิวเคลียส (syncytium) (รูปที่ 6.3) และทำงานไม่ได้ โดยกระบวนการนี้ทำให้ไวรัสเอชไอวีที่เป็น SI สามารถฆ่าเซลล์ของผู้ติดเชื้อได้หลายเซลล์พร้อมกันโดยการเข้าสู่เซลล์เพียงครั้งเดียว



รูปที่ 6.3 ขั้นตอนการเกิดเซลล์หลายนิวเคลียส (syncytium) ในผู้ป่วยเอชไอวี

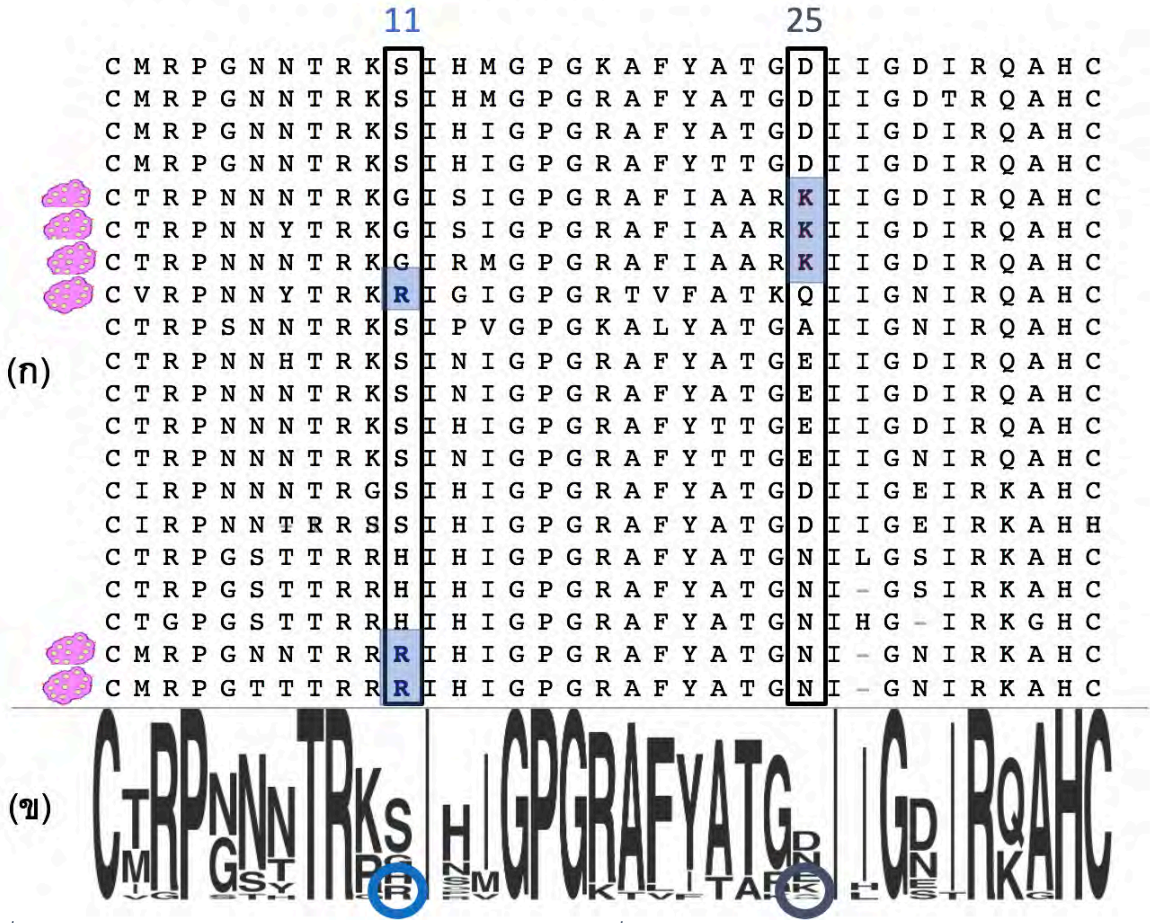
(ที่มา: Stowell, D. ©(2002-2006). *syncytia.* [ONLINE] Available at: http://www.mcld.co.uk [เข้าถึงออนไลน์เมื่อวันที่ 11 ก.ค. พ.ศ.2564]*)*

เนื่องจาก gp120 มีความสำคัญในการนำมาจำแนกการแสดงออกของไวรัสเอชไอวีกลุ่ม SI และ NSI นักชีววิทยาจึงมีความสนใจในการตรวจสอบและตัดสินว่ากรดแอมิโนในตำแหน่งใดของ gp120 ที่สามารถนำมาใช้ในการจำแนกฟีโนไทป์นี้

ในปี ค.ศ. 1992 ฌอง-ฌาคส์ เดอ ยอง (Jean-Jacques De jong) ได้ทำการวิเคราะห์ผลการเปรียบเทียบชุดของลำดับกรดแอมิโนในบริเวณที่เป็น V3 loop ของโปรตีน gp120 ดังแสดงในรูปที่ 6.4 และได้สร้างกฎ 11/25 โดยระบุว่าเชื้อเอชไอวีที่มีโอกาสแสดงฟีโนไทป์ SI มีกรดแอมิโนอาร์จินีน (arginine: Arg: R) หรือ ไลซีน (lysine: Lys: K) ที่ตำแหน่ง 11 หรือ 25 ในบริเวณที่เป็น V3 loop การศึกษาหลังจากนั้นพบว่ามีอีกหลายตำแหน่งที่มีผลต่อการจำแนก SI/NSI ฟีโนไทป์

ข้อจำกัดของการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีน

ก่อนที่นักชีววิทยาจะสามารถสร้างกฎเพื่อใช้ในการจำแนกฟีโนไทป์ของไวรัสเอชไอวีข้างต้น ปัญหาพื้นฐานที่พบ คือ การได้มาซึ่งผลของการเปรียบเทียบบริเวณ V3 loop ของโปรตีน gp120 จากคนไข้หลายๆคน ที่มีความถูกต้องสูง ซึ่งการเปรียบเทียบความคล้ายคลึงกันของชุดโปรตีน (multiple sequence alignment) (รูปที่ 6.4(ก)) ความผิดพลาดในการขยับลำดับกรดแอมิโนเพียงตำแหน่งเดียวมีผลต่อชนิดของกรดแอมิโนที่ปรากฏในตำแหน่ง 11 และ 25 ซึ่งส่งผลต่อการจำแนก SI/NSI ฟีโนไทป์ รูปที่ 6.4(ข) แสดงโมทิฟโลโก้ของ V3 loop ซึ่งบางคอลัมน์มีความอนุรักษ์สูงในขณะที่คอลัมน์อื่นๆ มีการแปรผันสูง นอกจากนี้โมทิฟโลโก้ของ V3 loop ไม่มีส่วนของ insertion/deletion ซึ่งมักปรากฏในส่วนอื่นๆ ที่มีความอนุรักษ์น้อยกว่าบริเวณ V3 loop โดย insertion/deletion จะทำให้การวิเคราะห์โปรตีน gp120 โดยรวมมีความซับซ้อนขึ้นไปอีก เนื่องจากแต่ละคอลัมน์มีความอนุรักษ์ของชุดของกรดแอมิโนมากน้อยแตกต่างกันไป



รูปที่ 6.4 (ก) ผลการเปรียบเทียบลำดับกรดแอมิโนบริเวณที่เป็น V3 loop ของโปรตีน gp120 จากผู้ป่วยเอชไอวี 20 ราย โดยคอลัมน์ที่ 11 และ 25 ของผู้ป่วยที่มี SI ฟีโนไทป์มีกรดแอมิโนเป็นอาร์จินีน (R) หรือ ไลซีน (K) (ข) โมทิฟโลโก้ของ V3 loop

(*ที่มา: ดัดแปลงจากรูปที่* 10.3 *ของ [52]*)

คำถามที่ตามมาคือการใช้เมทริกซ์คะแนนอย่างบลอสซัมหรือแพม รวมทั้งคะแนนลงโทษอินเดลเดียวกันสำหรับทุกคอลัมน์จะสามารถตอบโจทย์การเทียบบริเวณ V3 loop ข้างต้นที่คาดหวังให้ลำดับกรดแอมิโนหนึ่งๆ อยู่ในคอลัมน์ที่ถูกต้องได้หรือไม่ จากคำถามเหล่านี้จึงมีการเสนอแนวทางการให้คะแนนที่แตกต่างกันสำหรับแต่ละคอลัมน์ ตัวอย่างเช่น กรดแอมิโนที่ไม่ใช่อาร์จินีน (R) ในคอลัมน์ที่ 3 ในรูปที่ 6.4(ก) ข้างต้น ควรมีคะแนนลงโทษมากกว่ากรดแอมิโนที่ไม่ใช่เซอร์รีน (S) ในคอลัมน์ที่ 11 โดยสรุปคือ การเปรียบเทียบความคล้ายคลึงกันของชุดของโปรตีนในบทที่ 5 ไม่สามารถตอบโจทย์การเปรียบเทียบ V3 loop ในบทนี้ได้ จึงต้องมีการออกแบบและพัฒนาอัลกอริทึมเพิ่มเติมในการเปรียบเทียบชุดของสายโปรตีน โดยมีการนำวิธีการทางสถิติเข้ามาร่วมพิจารณา

เล่นพนันกับยากูซ่า

เกมส์พนันเกมส์หนึ่งที่เป็นที่นิยมคือการทายว่าลูกเต๋าสองลูกที่เจ้ามือเขย่าอยู่ เมื่อเปิดออกมาแล้วผลรวมของหน้าลูกเต๋าจะเป็นเลขคู่หรือคี่ ซึ่งเกมส์ที่มีลักษณะเดียวกันแต่ซับซ้อนน้อยกว่าคือเกมส์ทายหัวหรือก้อย ในเกมส์ทายหัวก้อยนี้ถ้าในรอบการทายหนึ่งๆ มีคนทายก้อยมากกว่าหัว เจ้ามือขี้โกงอาจเปลี่ยนเหรียญให้เป็นเหรียญที่เมื่อโยนแล้วมีโอกาสที่จะออกหัวมากกว่า เช่น ออกหัวเป็น ¾ เท่า แทนเหรียญปกติที่โอกาสในการออกหัวและก้อยเป็น ½ เท่ากัน

หยุดคิด	ถ้ามีการโยนเหรียญ 100 ครั้ง และออกหัว 63 ครั้ง คำถามคือเจ้ามือขี้โกงหรือไม่ และใช้เหรียญถ่วงน้ำหนักหรือไม่

เราไม่สามารถตอบปัญหาข้างต้นได้ชัดเจน เนื่องจากตัวคำถามไม่ได้กำหนดรายละเอียดชัดเจน เหรียญแต่ละเหรียญเมื่อโยนแต่ละครั้งมีโอกาสออกทั้งหัวและก้อย อย่างไรก็ตามสิ่งที่เราสามารถตอบได้คือเหรียญที่ใช้นั้นน่าจะเป็นเหรียญปกติ (fair coin: F) หรือเหรียญถ่วงน้ำหนัก (biased coin: B) โดยค่าความน่าจะเป็นในการออกหัวและก้อยสำหรับเหรียญปกติ แสดงดังต่อไปนี้

$$\Pr_{\mathrm{F}}(\text{"H"}) = 1/2 \quad \Pr_{\mathrm{F}}(\text{"T"}) = ½$$

และค่าความน่าจะเป็นในการออกหัวและก้อยสำหรับเหรียญถ่วงน้ำหนักคือ

$$\Pr_{\mathrm{B}}(\text{"H"}) = 3/4 \quad \Pr_{\mathrm{B}}(\text{"T"}) = ¼$$

เนื่องจากการโยนเหรียญแต่ละครั้งเป็นอิสระต่อกัน ดังนั้นค่าความน่าจะเป็นที่การโยนเหรียญปกติ n ครั้งโดยมีลำดับเป็น $x = x_1 x_{2\ldots} x_n$ แล้วออกหัว “H” จำนวน k ครั้ง มีค่าเท่ากับ

$$\Pr(x|F) = \prod_{i=1}^{n} Pr_F(x_i) = (1/2)^n$$

ในขณะที่เหรียญถ่วงน้ำหนักมีค่าความน่าจะเป็นในการเกิดลำดับ x เดียวกันเท่ากับ

$$\Pr(x|B) = \prod_{i=1}^{n} Pr_B(x_i) = (1/4)^{n-k} \cdot (3/4)^k = \frac{3^k}{4^n}$$

ถ้า $\Pr(x|F) > \Pr(x|B)$ เจ้ามือน่าจะใช้เหรียญปกติ แต่ถ้า $\Pr(x|F) < \Pr(x|B)$ เจ้ามือน่าจะใช้เหรียญถ่วงน้ำหนัก เนื่องจากค่าของ $(1/2)^n$ และค่า $\frac{3^k}{4^n}$ มีค่าน้อยมากสำหรับ n ที่มีค่ามาก ดังนั้นเราจึงเลือกใช้ log-odds ratio ในการเปรียบเทียบแทน ดังต่อไปนี้

$$\log_2\left(\frac{\Pr(x|F)}{\Pr(x|B)}\right) = \log_2(\frac{2^n}{3^k}) = n - k \cdot \log_2 3$$

ฝึกหัด	จงแสดงว่า $\Pr(x\|F)$ มีค่ามากกว่า $\Pr(x\|B)$ ถ้าค่า log-odds ratio เป็นบวก และน้อยกว่า $\Pr(x\|B)$ ถ้าค่า log-odds ratio เป็นลบ

กลับไปที่ตัวอย่างข้างต้นที่มีการโยนเหรียญ 100 ครั้ง และออกหัว 63 ครั้ง ค่า log-odds ratio จะมีค่าเป็นบวกเนื่องจาก

$$\frac{k}{n} = \log_2 3 \approx 0.6309 > 0.63$$

ถ้าพิจารณาผลการเปรียบเทียบค่าความน่าจะเป็นข้างต้นจะพบว่า เจ้ามือน่าจะใช้เหรียญปกติ ถึงแม้ 63 เข้าใกล้ 75 มากกว่า 50

เจ้ามือแอบใช้เหรียญถ่วงน้ำหนักสลับกับเหรียญปกติ

หากเรามีสมมติฐานว่าก่อนโยนเหรียญในแต่ละรอบเจ้ามือมีโอกาสเปลี่ยนเหรียญที่โยนจากเหรียญปกติไปเป็นเหรียญถ่วงน้ำหนักด้วยความน่าจะเป็น 0.1 หลังเห็นผลของการโยนเหรียญในแต่ละรอบ จะสามารถทราบได้อย่างไรว่ารอบไหนเจ้ามือใช้เหรียญปกติและรอบไหนเจ้ามือใช้เหรียญถ่วงน้ำหนัก (นิยามปัญหาที่ 6.1)

นิยามปัญหาที่ 6.1 ปัญหาคาสิโน

ปัญหาคาสิโน (Casino Problem) ถ้ามีผลของการโยนเหรียญในแต่ละรอบ สามารถบอกได้หรือไม่ว่ารอบไหนเจ้ามือใช้เหรียญปกติและรอบไหนเจ้ามือใช้เหรียญถ่วงน้ำหนัก	
ข้อมูลเข้า	ผลของการโยนเหรียญ $x = x_1 x_{2\ldots} x_n$ ในแต่ละรอบซึ่งมาจากการโยนเหรียญปกติ (F) หรือเหรียญถ่วงน้ำหนัก (B)
ผลลัพธ์	ลำดับของเหรียญที่ถูกใช้ในแต่ละรอบ $\pi = \pi_1 \pi_{2\ldots} \pi_n$ โดยที่ π_i มีค่าเท่ากับ *F* หรือ *B* ซึ่งเป็นการระบุว่า x_i เกิดจากการโยนเหรียญปกติหรือเหรียญถ่วงน้ำหนัก

การนิยามปัญหาคาสิโนข้างต้นต้องอยู่ในรูปแบบที่สามารถเปรียบเทียบและเลือก π ใดๆ ที่น่าจะเป็นคำตอบที่ดีที่สุดได้

นิยามปัญหาที่ 6.1 นี้ไม่ชัดเจนในเชิงคำนวณ เนื่องจากทั้งเหรียญปกติและเหรียญถ่วงน้ำหนักสามารถออกหัวหรือก้อยก็ได้ สิ่งที่เราต้องการหาคือความน่าจะเป็นของลำดับการใช้เหรียญของเจ้ามือ

หยุดคิด	เราสามารถนิยามปัญหาคาสิโนข้างต้นใหม่ได้อย่างไร

การหา CG-islands

ก่อนย้อนกลับไปเรื่องการโยนเหรียญ ลองพิจารณาปัญหาทางชีววิทยาต่อไปนี้ ซึ่งสามารถเทียบเคียงกับการโยนเหรียญข้างต้น โดยวิธีการแก้ปัญหาสามารถนำไปประยุกต์ใช้กับโจทย์ทางชีววิทยาได้อีกหลากหลายปัญหา รวมทั้งปัญหาการจำแนกฟีโนไทป์ของไวรัสเอชไอวี

ต้นคริสต์ศตวรรษที่ 20 ฟีบัส เลวีน (Phoebus Levene) ค้นพบนิวคลีโอไทด์ 4 ตัวที่ประกอบกันเป็นสายดีเอ็นเอ อย่างไรก็ตามในเวลานั้นความรู้เกี่ยวกับดีเอ็นเอยังมีไม่มากนัก (ผลงานวิจัยของวัตสันและคริกเกี่ยวกับดีเอ็นเอสายคู่ได้รับการตีพิมพ์หลังจากนั้นประมาณ 50 ปี) ผลคือเลวีนมีคำถามว่าดีเอ็นเอเก็บข้อมูลทางพันธุกรรมโดยใช้เพียง 4 ตัวอักษรได้อย่างไร และได้ตั้งสมมติฐานว่าดีเอ็นเอประกอบด้วย 4 นิวคลีโอไทด์นี้จำนวนเท่าๆ กัน ศตวรรษถัดมาเรามีองค์ความรู้เพิ่มเติมว่าคู่สมของแต่ละนิวคลีโอไทด์ที่อยู่สายตรงข้ามประกอบเป็นดีเอ็นเอสายคู่นั้นมีสัดส่วนของแต่ละนิวคลีโอไทด์ในปริมาณเท่าๆ กัน อย่างไรก็ตามสมมติฐานที่ว่าดีเอ็นเอสายเดี่ยวประกอบด้วย 4 นิวคลีโอไทด์จำนวนเท่าๆ กันนั้นไม่เป็นความจริง นอกจากนี้จีโนมของสิ่งมีชีวิตที่แตกต่างกันจะมีองค์ประกอบของนิวคลีโอไทด์กัวนีน **G** และไซโทซีน **C** (GC-content) ไม่เท่ากัน ตัวอย่างเช่น มนุษย์มี GC-content ประมาณ 42% และเมื่อพิจารณาเฉพาะส่วนของ GC-content เราอาจคาดว่าคู่ของนิวคลีโอไทด์ (dinucleotide) ที่อยู่ติดกัน เช่น **CC**, **CG**, **GC** และ **GG** จะถูกพบในจีโนมมนุษย์ด้วยความถี่ 0.21 x 0.21 = 4.41% อย่างไรก็ตามความถี่ของ **CG** ในจีโนมมนุษย์มีเพียงประมาณ 1% เนื่องจากคู่นิวคลีโอไทด์ **CG** มักเกิดการดัดแปลงดีเอ็นเอโดยการเติมกลุ่มเมทิล (CH_3) ให้กับนิวคลีโอไทด์ไซโทซีน (**C**) ที่อยู่ติดเป็นคู่กับกัวนีน (**G**) ผ่านการเติมหมู่เมทิล (methylation) ซึ่งเป็นกระบวนการที่พบเป็นปกติในธรรมชาติ ทำให้ไซโทซีนที่ถูกเติมหมู่เมทิลมีโอกาสเปลี่ยนไปเป็นไทมีน (**T**) และเป็นที่มาว่าทำไมความถี่ของการเกิดคู่นิวคลีโอไทด์ **CG** จึงต่ำกว่าการเกิดนิวคลีโอไทด์คู่อื่นๆ ในจีโนมของสิ่งมีชีวิตหลายชนิด อย่างไรก็ตามกระบวนการเติมหมู่เมทิลมักถูกยับยั้งในบริเวณรอบๆ ยีนที่เรียกว่า CG-island ซึ่งเป็นบริเวณที่มีความถี่ของ **CG** สูงกว่าบริเวณอื่น ดังนั้นการหาบริเวณที่น่าจะเป็นยีนในจีโนมวิธีการหนึ่งคือการหาบริเวณที่เป็น CG-island

วิธีการแบบง่ายในการหาบริเวณที่เป็น CG-island ในจีโนมคือการไล่ดูลำดับเบสในสายของจีโนมผ่านการใช้หน้าต่างเลื่อน (sliding window) โดยขนาดของหน้าต่างใช้กำหนดความยาวของสายจีโนมที่ต้องการพิจารณาในบริเวณหนึ่งๆ และนับความถี่ของ **CG** ที่พบในหน้าต่างนั้น ถ้ามีจำนวนมากจีโนมบริเวณนั้นก็น่าจะเป็น CG-island และน่าจะมียีนอยู่ใกล้ๆ ข้อจำกัดของวิธีการนี้คือเราไม่ทราบขนาดของหน้าต่างที่เหมาะสม นอกจากนี้ในขณะที่เลื่อนหน้าต่างไปในบริเวณที่คาบเกี่ยวกันอาจถูกระบุว่าเป็น CG-island หรือ ไม่เป็น CG-island ก็ได้

แบบจำลองมาร์คอฟซ่อนเร้น

พิจารณาการโยนเหรียญโดยใช้แบบจำลองมาร์คอฟซ่อนเร้น

เป้าหมายของหัวข้อนี้ คือพัฒนาแนวคิดและแบบจำลองที่สามารถนำไปประยุกต์ใช้ในการแก้ปัญหาคาสิโนและการหา CG-island ข้างต้น ถ้าลองจินตนาการโดยเปลี่ยนเจ้ามือในคาสิโนจากคนให้เป็นเครื่องจักรเครื่องหนึ่งซึ่งเราไม่ทราบว่าเครื่องจักรนี้ถูกสร้างขึ้นมาและมีกลไกการทำงานภายในอย่างไร อย่างไรก็ตามสิ่งที่เราทราบคือเครื่องจักรนี้มีการทำงานเป็นรอบ โดยในแต่ละรอบนั้นเครื่องจักรจะอยู่ในสถานะใดสถานะหนึ่งที่ซ่อนเร้นอยู่ระหว่าง F และ B และจะแสดงผลออกมาให้เห็นเป็น “H” หรือ “T” เท่านั้น โดยในแต่ละรอบที่เครื่องจักรทำงาน เครื่องจักรจะต้องตัดสินใจสองอย่าง

1) จะย้ายไปที่สถานะซ่อนเร้นใดระหว่าง F และ B
2) จะแสดงผลออกไปเป็น “H” หรือ “T”

เครื่องจักรตอบคำถามแรกโดยการเลือกแบบสุ่มระหว่างสองสถานะโดยมีค่าความน่าจะเป็นที่จะอยู่ในสถานะเดิม 0.9 และย้ายไปอีกสถานะหนึ่งเท่ากับ 0.1 เครื่องจักรตอบคำถามที่สองโดยเลือกว่าจะส่งออกตัวอักษร “H” หรือ “T” ด้วยความน่าจะเป็นที่ถูกกำหนดไว้จำเพาะสำหรับแต่ละสถานะซ่อนเร้นที่อยู่ในรอบนั้นๆ จากตัวอย่างของการโยนเหรียญในปัญหาคาสิโน ค่าความน่าจะเป็นในการส่งออกตัวอักษร “H” หรือ “T” เป็น 0.5 เท่ากันสำหรับสถานะซ่อนเร้น F และเป็น 0.75 และ 0.25 สำหรับสถานะซ่อนเร้น B เป้าหมายของเราคือการอนุมานลำดับของสถานะการทำงานภายในเครื่องจักรโดยการวิเคราะห์จากข้อมูลที่ส่งออกมา

จากแนวคิดข้างต้นเราได้ทำการแปลงเจ้ามือที่เป็นมนุษย์ไปเป็นเครื่องจักรที่เรียกว่า แบบจำลองมาร์คอฟซ่อนเร้น (Hidden Markov Model: HMM) ความแตกต่างอย่างเดียวระหว่างเครื่องจักรโยนเหรียญข้างต้นกับแบบจำลองมาร์คอฟซ่อนเร้นแบบทั่วไปคือแบบจำลองมาร์คอฟซ่อนเร้นทั่วไปไม่ได้จำกัดจำนวนสถานะซ่อนเร้น ไม่ได้จำกัดค่าความน่าจะเป็นในการย้ายจากสถานะซ่อนเร้นหนึ่งไปยังสถานะซ่อนเร้นอื่น และไม่ได้จำกัดค่าความน่าจะเป็นในการส่งออกผลแต่ละแบบของแต่ละสถานะซ่อนเร้น โดยทั่วไป HMM มีองค์ประกอบหลัก 4 ส่วนคือ Σ, States, Transition, และ Emission โดยถูกกำหนดไว้ดังต่อไปนี้

- Σ คือ ชุดของอักขระที่เป็นไปได้ในการแสดงออก
- States คือชุดของสถานะซ่อนเร้นทั้งหมด
- เมทริกซ์ขนาด |States| x |States| แสดงค่าความน่าจะเป็นในการเปลี่ยนจากสถานะซ่อนเร้นหนึ่งไปยังอีกสถานะซ่อนเร้นหนึ่ง (transition probability) โดยเมทริกซ์นี้เรียกว่าเมทริกซ์เปลี่ยนสถานะ (transition matrix) ค่าในเมทริกซ์เช่น $transition_{l,k}$ แสดงค่าความน่าจะเป็นในการเปลี่ยนจากสถานะซ่อนเร้น l ไปยังสถานะซ่อนเร้น k

- เมทริกซ์ขนาด |States| x |Σ| แสดงค่าความน่าจะเป็นในการแสดงผลเป็นอักขระจำเพาะหนึ่งๆ ของแต่ละสถานะซ่อนเร้น (emission probability) โดยเมทริกซ์นี้เรียกว่าเมทริกซ์อิมิชชัน (emission matrix) ค่าในเมทริกซ์ เช่น $emission_k(b)$ แสดงค่าความน่าจะเป็นในการแสดงผลเป็นอักขระ *b* ที่เป็นสมาชิกของ Σ เมื่อ HMM อยู่ในสถานะซ่อนเร้น *k*

ผลรวมค่าความน่าจะเป็นในการเปลี่ยนจากสถานะซ่อนเร้น *l* ไปยังสถานะซ่อนเร้น k ใดๆ ที่เป็นสมาชิกของ States มีค่าเท่ากับ 1 ดังสมการต่อไปนี้

$$\sum_{all\ states\ k} transition_{l,k} = 1$$

และในสถานะซ่อนเร้น *k* ใดๆ ผลรวมค่าความน่าจะเป็นในการแสดงผลเป็นอักขระ *b* ที่เป็นสมาชิกของ Σ มีค่าเท่ากับ 1 ดังสมการต่อไปนี้

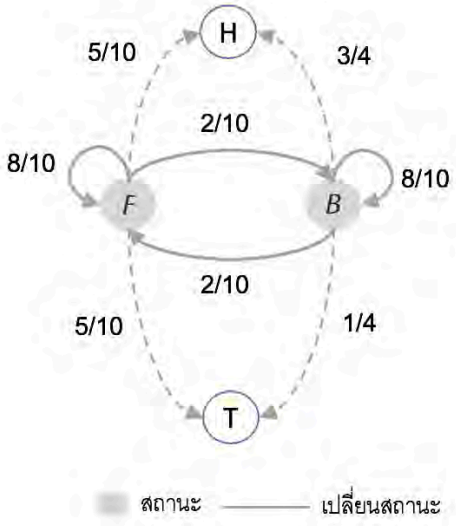
$$\sum_{all\ symbols\ b\ from\ \Sigma} emission_k(b) = 1$$

ฝึกหัด	จงใช้ HMM โดยแสดงองค์ประกอบทั้งสี่ส่วนเพื่อจำลองปัญหาคาสิโน

แผนภาพ HMM

แบบจำลอง HMM สามารถแสดงโดยใช้แผนภาพ HMM (HMM diagram) ซึ่งในแผนภาพโหนดที่มีเส้นทึบเชื่อมแสดงสถานะซ่อนเร้น โดยเส้นทึบที่ชี้ออกจากโหนด *l* ไปโหนด *k* แสดงการเปลี่ยนสถานะจากสถานะซ่อนเร้น *l* ไปยังสถานะซ่อนเร้น *k* โดยมีค่าความน่าจะเป็นในการเปลี่ยนสถานะแสดงบนเส้นเชื่อม โหนดที่มีเส้นประชี้เข้าหาแสดงอักขระที่แสดงผล โดยค่าบนเส้นประที่ชี้ออกจากโหนดสถานะซ่อนเร้นหนึ่งๆ มายังโหนดอักขระนี้แสดงค่าความน่าจะเป็นที่โหนดสถานะซ่อนเร้นนี้จะแสดงผลเป็นอักขระที่ถูกชี้ รูปที่ 6.5 แสดงแผนภาพ HMM เพื่อจำลองปัญหาคาสิโนข้างต้น ประกอบด้วยสองสถานะซ่อนเร้น คือ ใช้เหรียญปกติ (F) และใช้เหรียญถ่วงน้ำหนัก (B) โดยค่าความน่าจะเป็นในการเปลี่ยนสถานะเป็น 0.2 เท่ากัน และค่าความน่าจะเป็นในการอยู่ในสถานะเดิมเป็น 0.8 สถานะซ่อนเร้นเหรียญปกติ (F) มีค่าความน่าจะเป็นในการแสดงผลเป็นหัว “H” และก้อย “T” เท่ากันคือ 0.5 ในขณะที่สถานะซ่อนเร้นเหรียญถ่วงน้ำหนัก (B) มีค่าความน่าจะเป็นในการแสดงผลเป็นหัวและก้อยเท่ากับ 0.75 และ 0.25 ตามลำดับ

วิถีซ่อนเร้น (hidden path) $\pi = \pi_1\pi_{2\ldots}\pi_n$ ใน HMM คือลำดับของสถานะซ่อนเร้นที่ HMM ได้เดินผ่านในแต่ละรอบ โดยทางเดินที่ซ่อนอยู่นี้คือชุดของเส้นทึบในแผนภาพ HMM รูปที่ 6.6 แสดงตัวอย่างของเครื่องจักร HMM ที่ทำหน้าที่เป็นเจ้ามือขี้โกง โดย $\Pr(\pi_{(i-1)} \rightarrow \pi_i)$ แสดงค่าความน่าจะเป็นในการเปลี่ยนสถานะ



เมทริกซ์เปลี่ยนสถานะ (Transition matrix)

	F	B
F	8/10	2/10
B	2/10	8/10

เมทริกซ์อิมิชชัน (Emission matrix)

	T	H
F	5/10	5/10
B	1/4	3/4

รูปที่ 6.5 การจำลองปัญหาคาสิโนโดยใช้แผนภาพ HMM

$\Pr(x_i|\pi_i)$ แสดงค่าความน่าจะเป็นในการแสดงผลเป็น x_i โดยสถานะซ่อนเร้น π_i มีลำดับหน้าเหรียญที่แสดงออกเป็น x = “TTHHHTHTHT” และมีลำดับการใช้เหรียญหรือลำดับของสถานะซ่อนเร้นเป็น π = *FFBBBFFFBB* โดยสองครั้งแรกและครั้งที่ 6-8 ใช้เหรียญปกติในขณะที่การโยนครั้งอื่นๆ ใช้เหรียญถ่วงน้ำหนัก

x	T	T	H	H	H	T	H	T	H	T
π	F	F	B	B	B	F	F	F	B	B
$\Pr(\pi_{i-1}\rightarrow\pi_i)$	.8	.2	.8	.8	.2	.8	.8	.2	.8	
$\Pr(x_i\|\pi_i)$	½	½	¾	¾	¾	½	½	½	¾	¼

รูปที่ 6.6 ตัวอย่างลำดับการออกหน้าเหรียญและสถานะซ่อนเร้นที่ใช้ในแต่ละลำดับ

หยุดคิด	ค่า $\Pr(x,\pi)$ สำหรับ x และ π ในรูปที่ 6.6 เป็นเท่าใด

กำหนดวิธีการแก้ปัญหาคาสิโนใหม่

หัวข้อนี้แสดงการใช้ HMM เป็นเครื่องมือเพื่อแก้ปัญหาคาสิโนซึ่งมีเป้าหมายในการหา π ที่สอดคล้องกับลำดับการออกหัวก้อย x มากที่สุด โดยเริ่มพิจารณาจากปัญหาที่ง่ายกว่าคือ หาค่าความน่าจะเป็น $\Pr(x,\pi)$ที่ HMM จะใช้เส้นทาง $\pi = \pi_1\pi_{2\ldots}\pi_n$ และแสดงผลลำดับของอักขระเป็น $x = x_1x_{2\ldots}x_n$ โดยที่

$$\sum_{all\ strings\ of\ emitted\ symbol\ x} \sum_{all\ hidden\ paths\ \pi} \Pr(x,\pi) = 1$$

สำหรับแต่ละสายอักขระ x ที่แสดงออกมานั้นมีค่าความน่าจะเป็นเท่ากับ $\Pr(x)$ ซึ่งเป็นอิสระจากเส้นทางแสดงลำดับสถานะซ่อนเร้นที่ถูกเลือกโดย HMM

$$\Pr(x) = \sum_{all\ hidden\ paths\ \pi} \Pr(x, \pi)$$

และแต่ละเส้นทางแสดงลำดับของสถานะซ่อนเร้น π มีค่าความน่าจะเป็นเท่ากับ $\Pr(\pi)$ ซึ่งเป็นอิสระจากสายอักขระ x ที่ HMM แสดงออกมา ดังนั้น

$$\Pr(\pi) = \sum_{all\ strings\ of\ emitted\ symbols\ x} \Pr(x, \pi)$$

เหตุการณ์ที่ HMM ใช้เส้นทางแสดงลำดับสถานะซ่อนเร้น π และแสดงออกสายอักขระ x ออกมานั้นสามารถพิจารณาว่าเกิดจากการรวมสองเหตุการณ์เข้าด้วยกันคือ

- **เหตุการณ์ที่ 1:** HMM เลือกเส้นทางแสดงลำดับสถานะซ่อนเร้น π ซึ่งมีค่าความน่าจะเป็นเท่ากับ $\Pr(\pi)$
- **เหตุการณ์ที่ 2:** HMM แสดงออกสายอักขระ x โดยกำหนดให้ HMM ใช้เส้นทาง π โดยเราเรียกค่าความน่าจะเป็นรูปแบบนี้ว่า ความน่าจะเป็นแบบมีเงื่อนไข (conditional probability) ของ x เมื่อกำหนด π ให้ และแสดงสัญลักษณ์ได้เป็น $\Pr(x|\pi)$

โดยทั้งสองเหตุการณ์ข้างต้นจะต้องเกิดในกรณีที่ HMM ใช้เส้นทาง π และให้ผลเป็นสายอักขระ x ซึ่งสามารถเขียนสมการได้เป็น

$$\Pr(x, \pi) = \Pr(x|\pi) \cdot \Pr(\pi)$$

ในการคำนวณ $\Pr(x, \pi)$ เราจะทำการคำนวณ $\Pr(\pi)$ ก่อนดังแสดงในรูปที่ 6.6 ในส่วนที่เป็น $\Pr(\pi_i \rightarrow \pi_{i+1})$ ซึ่งแสดงค่าความน่าจะเป็นในการเปลี่ยนจากสถานะซ่อนเร้น π_i ไปเป็น π_{i+1} ในปัญหาคาสิโนเรามีสมมติฐานว่าในการโยนเหรียญครั้งแรกโอกาสที่เจ้ามือจะเลือกเหรียญปกติหรือเหรียญถ่วงน้ำหนักมาใช้มีเท่าๆกัน ดังนั้นในรูปที่ 6.6 $\Pr(\pi_0 \rightarrow \pi_1) = 1/2$ โดย π_0 เป็นสถานะซ่อนเร้นเริ่มต้น (initial state) และถือว่าเป็น silent node ที่ไม่มีการส่งออกอักขระใดๆ ค่าความน่าจะเป็นของ π คำนวณจากผลคูณของค่าความน่าจะเป็นในการเปลี่ยนจากสถานะซ่อนเร้นหนึ่งไปยังอีกสถานะซ่อนเร้นหนึ่งตลอดเส้นทางของลำดับสถานะใน π ดังสมการต่อไปนี้

$$\Pr(\pi) = \prod_{i=1}^{n} \Pr(\pi_{i-1} \rightarrow \pi_i) = \prod_{i=1}^{n} transition_{\pi_{i-1}, \pi_i}$$

สำหรับค่าความน่าจะเป็นที่จะแสดงผลเป็นสายอักขระ x โดยกำหนดให้ HMM ใช้เส้นทาง π จะเท่ากับ

$$\Pr(x|\pi) = \prod_{i=1}^{n} \Pr(x_i|\pi_i)$$

$$= \prod_{i=1}^{n} emission_{\pi_i}(x_i)$$

ดังนั้นเมื่อย้อนกลับไปที่ค่าความน่าจะเป็นตั้งต้น $\Pr(x, \pi)$ ซึ่งเป็นค่าความน่าจะเป็นที่ HMM จะใช้เส้นทาง π และแสดงผลสายอักขระเป็น x จะสามารถเขียนสมการใหม่ได้ดังต่อไปนี้

$$\Pr(x, \pi) = \Pr(x|\pi) \cdot \Pr(\pi)$$
$$= \prod_{i=1}^{n} \Pr(x_i|\pi_i) \cdot \Pr(\pi_{i-1} \to \pi_i)$$
$$= \prod_{i=1}^{n} emission_{\pi_i}(x_i) \cdot transition_{\pi_{i-1},\pi_i}$$

ฝึกหัด	จงคำนวณ $\Pr(x, \pi)$ สำหรับ x และ π ในรูปที่ 6.6 และหาว่ามีเส้นทาง π อื่นที่ดีกว่าที่ไม่ใช่ FFBBBFFFBB ที่ให้ผลการแสดงออกเป็น x = “TTHHHTHTHT” หรือไม่ ถ้ามีเป็นเส้นทางไหน

หยุดคิด	หลังจากทราบองค์ประกอบหลักของ HMM แล้ว เราสามารถนำ HMM นี้ไปแก้ปัญหาการหา CG-island ในจีโนมที่กล่าวถึงก่อนหน้าได้อย่างไร และมีข้อจำกัดหรือไม่ อย่างไร

The Decoding Problem

กราฟวิเทอบิ

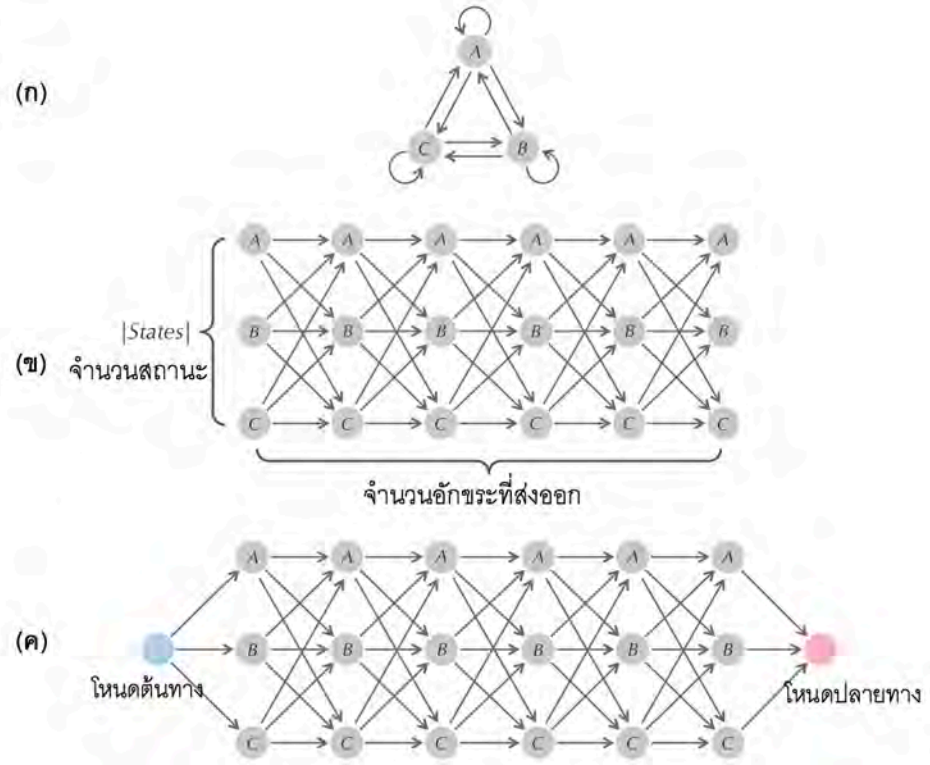
ดังได้กล่าวในข้างต้นทั้งปัญหาคาสิโนและปัญหาการหา CG-island ในจีโนม เป้าหมายคือหาเส้นทางแสดงลำดับสถานะซ่อนเร้น π ที่มีโอกาสแสดงผลออกเป็น x มากที่สุด หรืออีกนัยหนึ่งคือพยายามหา $\Pr(x, \pi)$ ที่มีค่ามากสุด

นิยามปัญหาที่ 6.2 ปัญหา Decoding

Decoding Problem หาเส้นทางแสดงลำดับสถานะซ่อนเร้น π ที่ดีที่สุดใน HMM โดยให้ผลออกมาเป็นสายอักขระ x	
ข้อมูลเข้า	สายอักขระ $x = x_1x_2\ldots x_n$ ที่ส่งออกจาก HMM โดยที่ HMM ประกอบด้วย Σ, States, Transition และ Emission
ผลลัพธ์	เส้นทางแสดงลำดับสถานะซ่อนเร้น π ที่ให้คะแนน $\Pr(x, \pi)$ มากสุด

ในปี ค.ศ. 1967 แอนดรู วิเทอบิ (Andrew Viterbi) ใช้ HMM โดยจัดรูปแบบให้เหมือนตาราง 2 มิติที่แสดงแผนที่ของเมืองแมนฮัตตัน (ดังตัวอย่างในบทที่ 5 เรื่องการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีน) ในการแก้ปัญหา Decoding สำหรับ HMM ที่มีการส่งออกสายข้อมูลเป็น $x = x_1x_2\ldots x_n$ เมื่อนำมา

จัดให้อยู่ในรูปแบบกราฟวิเทอบิ (รูปที่ 6.7) สถานะซ่อนเร้นแต่ละสถานะจะถูกนำมาเรียงกันแถวละ 1 สถานะ ซึ่งหมายถึงจำนวนแถวจะเท่ากับ |States| และในแต่ละแถวจะประกอบด้วย n คอลัมน์ตามจำนวนของอักขระที่ HMM ส่งออก โดยโหนด (k,i) ในกราฟวิเทอบิ แสดงสถานะซ่อนเร้น k ที่แสดงผลอักขระลำดับที่ i ใน x โดยแต่ละโหนดจะมีเส้นเชื่อมไปยังทุกโหนดที่อยู่ในคอลัมน์ถัดไปทางขวามือ เส้นเชื่อมจากโหนด (l, i-1) มายัง (k,i) แสดงการเปลี่ยนสถานะซ่อนเร้นจากสถานะ l มาเป็นสถานะ k ด้วยความน่าจะเป็นเท่ากับ $\text{transition}_{l,k}$ และมีการส่งออกอักขระ x_i ด้วยค่าความน่าจะเป็นเท่ากับ $emission_k(x_i)$ ดังนั้นทุกเส้นทางที่เชื่อมต่อโหนดจากคอลัมน์แรกของ-



รูปที่ 6.7 (ก) แผนภาพ HMM ที่ประกอบด้วยสถานะซ่อนเร้น 3 สถานะ โดยไม่ได้แสดงค่าในส่วนของ Σ, Transition และ Emission (ข) HMM ในรูปแบบกราฟวิเทอบิที่ส่งออกสายข้อมูลเป็น $x = x_1x_2...x_n$ (ค) กราฟวิเทอบิที่มีการเพิ่มโหนดต้นทางและโหนดปลายทาง

(ที่มา: ปรับจากรูปที่ 10.7 ของ [52])

กราฟวิเทอบิไปยังโหนดในคอลัมน์สุดท้ายแสดง π ทั้งหมดที่เป็นไปได้ โดยค่าน้ำหนักของเส้นเชื่อมระหว่างโหนดแต่ละเส้นกำหนดโดย

$$WEIGHT_i(l,k) = transition_{\pi_{i-1},\pi_i} \cdot emission_{\pi_i}(x_i)$$

และสามารถกำหนดผลคูณของค่าน้ำหนักเส้นเชื่อมของเส้นทาง π ดังสมการต่อไปนี้

$$\prod_{i=2}^{n} transition_{\pi_{i-1},\pi_i} \cdot emission_{\pi_i}(x_i) = \prod_{i=1}^{n-1} WEIGHT_i(l,k)$$

หยุดคิด	สมการแสดงผลคูณของค่าน้ำหนักเส้นเชื่อมในกราฟวิเทอบินี้แตกต่างจาก $\Pr(x,\pi)$ ที่แสดงข้างต้นอย่างไร

สมการแสดงผลคูณของค่าน้ำหนักเส้นเชื่อมในกราฟวิเทอบินี้แตกต่างจาก $\Pr(x,\pi)$ ที่แสดงข้างต้นเพียงอย่างเดียวคือ สมการแสดงผลคูณค่าน้ำหนักยังไม่มีการคูณ $transition_{\pi_0,\pi_1} \cdot emission_{\pi_1}(x_1)$ ซึ่งเป็นการเปลี่ยนจากสถานะเริ่มต้น π_0 ไปยังสถานะ π_1 และส่งออกอักขระแรก เพื่อให้กราฟวิเทอบิมีสถานะเริ่มต้นด้วย จึงมีการเพิ่มโหนดตั้งต้น (source) ทางซ้ายสุดและทำการเชื่อมโหนดตั้งต้นนี้ไปยังทุกโหนดในคอลัมน์แรกโดยมีน้ำหนักของเส้นเชื่อมเท่ากับ $WEIGHT_0(source,k) = transition_{\pi_0,k} \cdot emission_k(x_1)$ นอกจากนี้เรายังสามารถเพิ่มโหนดปลายทาง (sink) และเพิ่มเส้นเชื่อมจากทุกโหนดให้คอลัมน์สุดท้ายไปยังโหนดปลายทางนี้โดยมีค่าน้ำหนักของเส้นเชื่อมเหล่านี้เท่ากับ 1 ด้วยกราฟวิเทอบินี้เราสามารถแก้ปัญหา Decoding โดยการหาเส้นทางในกราฟที่เชื่อมระหว่างโหนดต้นทางไปยังโหนดปลายทางที่ให้ผลคูณของค่าน้ำหนักมากสุด

ฝึกหัด	จงหาเส้นทาง π ที่ให้ผลคูณของค่าน้ำหนักมากสุดโดยสายอักขระที่ส่งออกจาก HMM เป็น x = “HHTT”

อัลกอริทึมวิเทอบิ

เราสามารถใช้กำหนดการพลวัตในการแก้ปัญหา Decoding ในหัวข้อที่ผ่านมา โดยกำหนดให้ $s_{k,i}$ เป็นผลคูณของค่าน้ำหนักที่มากที่สุดจากโหนดต้นทางมายังโหนด (k,i) โดยอัลกอริทึมวิเทอบิ (Viterbi algorithm) มีสมมติฐานว่าเส้นเชื่อมจำนวน i-1 เส้นแรกของเส้นทางที่ดีที่สุดจากโหนดต้นทางมายังโหนด (k,i) นั้น มาจากเส้นทางที่ดีที่สุดจากโหนดต้นทางมายังโหนด (l, i-1) สำหรับสถานะซ่อนเร้น l ใดๆ ซึ่งสมมติฐานนี้ทำให้เราสามารถสร้างสมการความสัมพันธ์เวียนเกิดต่อไปนี้

$$\begin{aligned} s_{k,i} &= \max_{all\ states\ l} \{s_{l,i-1} \cdot (weight\ of\ edge\ between\ nodes\ (l,i-1) and\ (k,i))\} \\ &= \max_{all\ states\ l} \{s_{l,i-1} \cdot WEIGHT_i(l,k)\} \\ &= \max_{all\ states\ l} \{s_{l,i-1} \cdot transition_{\pi_{i-1},\pi_i} \cdot emission_{\pi_i}(x_i)\} \end{aligned}$$

และเนื่องจากโหนดตั้งต้นเชื่อมต่อกับทุกโหนดในคอลัมน์แรกของกราฟวิเทอบิ

$$\begin{aligned} s_{k,1} &= s_{source} \cdot \big(weight\ of\ edge\ between\ source\ and\ (k,1)\big) \\ &= s_{source} \cdot\ WEIGHT_0(source, \mathrm{k}) \\ &= s_{source} \cdot transition_{source,k} \cdot emission_k(x_1) \end{aligned}$$

โดย s_{source} ในความสัมพันธ์เวียนเกิดนี้มีค่าเท่ากับ 1 และสามารถหาผลคูณค่าน้ำหนักที่มากที่สุดโดยใช้สมการต่อไปนี้

$$s_{sink} = \max_{all\ states\ l} s_{l,n}$$

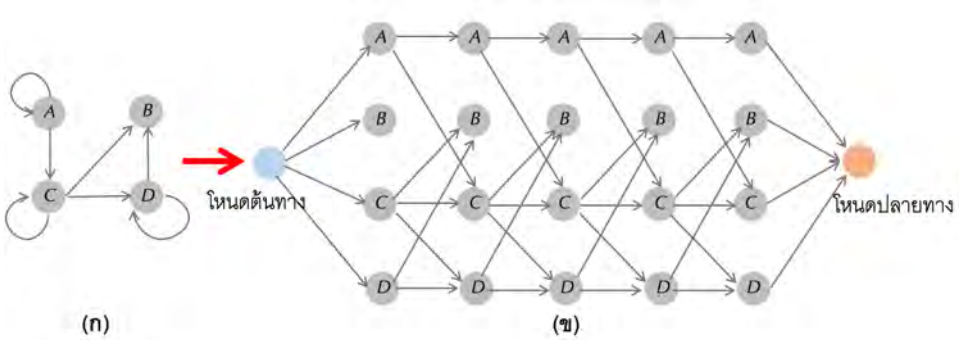
ประสิทธิภาพของอัลกอริทึมวิเทอบิ

เราสามารถแก้ปัญหา Decoding โดยจำลองปัญหาเป็นการหาเส้นทางที่ยาวที่สุดในกราฟแบบมีทิศทางและไม่มีลูป (directed acyclic graph: DAG) ลักษณะเดียวกับปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นและหรือโปรตีนในบทที่ 5 โดยการหาผลคูณของค่าน้ำหนักที่มากที่สุดในสมการ $\prod_{i=1}^{n} WEIGHT_i(\pi_{i-1}, \pi_i)$ และสามารถเปลี่ยนรูปของสมการเป็นผลบวกโดยการใส่ฟังก์ชันลอการิทึม $\sum_{i=1}^{n} \log(WEIGHT_i(\pi_{i-1}, \pi_i))$ และหาเส้นทางที่มีผลบวกมากสุดแทน ทั้งนี้เวลาที่ใช้ในการทำงานของอัลกอริทึมวิเทอบิเป็นสมการเส้นตรงแปรผันตามจำนวนเส้นเชื่อมในกราฟและมีค่าเท่ากับ $O(|States^2| \cdot n)$ โดย n คือจำนวนอักขระที่ส่งออก

ฝึกหัด	จงประยุกต์ใช้ HMM เพื่อแก้ปัญหาการหา CG-island ในโครโมโซม X ของมนุษย์ในช่วง 1 ล้านลำดับเบสแรก โดยสมมติว่าการเปลี่ยนสถานะจาก CG-island ไปเป็น non CG-island มีค่าความน่าจะเป็นเท่ากับ 0.001 ในขณะที่การเปลี่ยนสถานะจาก non CG-island มาเป็น CG-island มีค่าความน่าจะเป็น 0.0001 คำถามคือจาก 1 ล้านนิวคลีโอไทด์ที่เป็นข้อมูลเข้า พบ CG-island ทั้งหมดกี่บริเวณ

การเปลี่ยนสถานะใน HMM จากสถานะซ่อนเร้นหนึ่งไปยังอีกสถานะซ่อนเร้นหนึ่งในทางปฏิบัติอาจไม่มีทางเกิดขึ้น (forbidden transition) ดังนั้นเส้นเชื่อมระหว่างสถานะเหล่านี้สามารถลบออกจากแผนภาพ HMM และกราฟวิเทอบิได้ จากรูปที่ 6.8(ก) เนื่องจากไม่มีการเปลี่ยนสถานะจาก A ไป B และ D หรือจาก B ไปยังสถานะอื่นๆ เป็นต้น เส้นเชื่อมเหล่านี้จะถูกลบออกจากกราฟวิเทอบิที่จำลองแผนภาพ HMM โดยแสดงผล 5 อักขระ ดังรูปที่ 6.8 (ข) ซึ่งการลบเส้นเชื่อมที่ไม่มีทางเกิดขึ้นนี้ทำให้อัลกอริทึมวิเทอบิทำงานได้เร็วขึ้น

ฝึกหัด	ในปัญหาคาสิโนก่อนหน้า สายอักขระใดระหว่าง “HHTT” และ “HTHT” มีโอกาสในการส่งออกมากกว่ากัน



รูปที่ 6.8 (ก) แผนภาพ HMM ที่ประกอบด้วยสถานะซ่อนเร้น 4 สถานะและมีการเปลี่ยนสถานะเพียงบางแบบ (ข) การลดเส้นเชื่อมในกราฟวิเทอบิที่ไม่มีทางเกิดขึ้น

การหาสายข้อมูลส่งออกที่มีโอกาสเกิดขึ้นมากที่สุด

หัวข้อที่ผ่านมาเราศึกษากำหนดการพลวัตในการหา $\Pr(\pi)$ ที่มีค่ามากสุด ในหัวข้อนี้เราสนใจค่าความน่าจะเป็นที่ HMM จะส่งออกสายอักขระ x หนึ่งๆ หรือคำนวณหาค่า $\Pr(x)$ นั่นเอง

นิยามปัญหาที่ 6.3 ปัญหาการหาความน่าจะเป็นที่ HMM จะส่งออกสายอักขระหนึ่งๆ

ปัญหาการหาความน่าจะเป็นที่ HMM จะส่งออกสายอักขระหนึ่งๆ (Outcome Likelihood Problem) หาค่าความน่าจะเป็นที่ HMM ส่งออกสายอักขระ x ใดๆ	
ข้อมูลเข้า	สายอักขระ $x = x_1x_2\ldots x_n$ ที่ถูกส่งออกจาก HMM ซึ่งประกอบด้วย Σ, States, Transition และ Emission
ผลลัพธ์	ค่าความน่าจะเป็น $\Pr(x)$ ที่ HMM ส่งออกสายอักขระ x

หยุดคิด	เราสามารถดัดแปลงสมการความสัมพันธ์เวียนเกิดของวิเทอบิต่อไปนี้ เพื่อแก้ปัญหา Outcome Likelihood ได้อย่างไร $s_{k,i} = \max_{all\ states\ l}\{s_{l,i-1} \cdot WEIGHT_i(l,k)\}$

ในหัวข้อก่อนหน้าเราทราบว่า $\Pr(x)$ เท่ากับผลบวกของ $\Pr(x,\pi)$ สำหรับทุกเส้นทาง π อย่างไรก็ตาม จำนวนของเส้นทางจะเพิ่มมากขึ้นแบบเลขชี้กำลัง (exponential) ตามจำนวนอักขระที่ส่งออกในสายอักขระ x ดังนั้นการเลือกใช้กำหนดการพลวัต (dynamic programming) จึงเป็นแนวทางที่ดีกว่าในการคำนวณ $\Pr(x)$

กำหนดให้ผลรวมผลคูณทุกเส้นทางจากโหนดตั้งต้น (source) ถึงโหนด (k,i) ในกราฟวิเทอบิเป็น $forward_{k,i}$ และ $forward_{sink}$ เท่ากับ $\Pr(x)$ ในการคำนวณ $forward_{k,i}$ เราแบ่งเส้นทางทั้งหมดที่เชื่อมจากโหนดตั้งต้น source ถึง (k, i) ออกเป็นเซตย่อยของ |States| โดยแต่ละเซตย่อยมีเส้นทางที่ผ่านโหนด (l, i-1)

ซึ่งมีผลรวมผลคูณเป็น $forward_{l,i-1}$ ก่อนที่จะมาถึงโหนด (k, i) สำหรับบาง l ที่เป็นสถานะซ่อนเร้นใดๆ ดังนั้น $forward_{k,i}$ จึงเป็นผลรวมของ |States| ดังต่อไปนี้

$$forward_{k,i} = \sum_{all\ states\ l} forward_{l,i-1} \cdot \big(weight\ of\ edge\ connecting\ (l, i-1)\ and\ (k,i)\big)$$
$$= \sum_{all\ states\ l} forward_{l,i-1} \cdot WEIGHT_i(l,k)$$

สังเกตว่าความแตกต่างเดียวระหว่างสมการเวียนเกิดข้างต้นและสมการเวียนเกิดวิเทอบิก่อนหน้าที่แสดงอีกครั้งต่อไปนี้

$$s_{k,i} = \max_{all\ states\ l} \{s_{l,i-1} \cdot WEIGHT_i(l,k)\}$$

คือฟังก์ชันการหาค่าที่มากสุด $(\max)$ ในอัลกอริทึมวิเทอบิก่อนหน้ากลายมาเป็นฟังก์ชันการหาผลรวม (Σ) ในสมการนี้ ซึ่งเราสามารถแก้ปัญหา Outcome Likelihood โดยการคำนวณ $forward_{sink}$ ซึ่งแสดงโดยสมการต่อไปนี้

$$\sum_{\text{all states k}} forward_{k,n}$$

จากสมการนี้เราสามารถคำนวณค่าความน่าจะเป็น $\Pr(x)$ ในการส่งออกสายอักขระ x โดยถ้าต้องการหาสายอักขระส่งออกที่มีโอกาสเกิดขึ้นมากสุด เราสามารถนิยามปัญหาได้ดังต่อไปนี้

นิยามปัญหาที่ 6.4 ปัญหาการหาสายอักขระที่มีโอกาสส่งออกมากที่สุด

ปัญหาการหาสายอักขระที่มีโอกาสส่งออกมากที่สุด (Most Likely Outcome Problem) หาสายอักขระที่มีโอกาสถูกส่งออกมากที่สุด	
ข้อมูลเข้า	โมเดล HMM ที่ประกอบด้วย Σ, States, Transition และ Emission และจำนวนเต็ม n
ผลลัพธ์	สายอักขระ $x = x_1x_2\ldots x_n$ ที่ถูกส่งออกจาก HMM โดยเป็นสายอักขระที่ทำให้ค่าความน่าจะเป็น $\Pr(x)$ มากสุด

การสร้างโพรไฟล์ HMM เพื่อใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นและหรือโปรตีน

HMMs เกี่ยวข้องกับการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนอย่างไร

หัวข้อนี้แสดงการนำ HMM ไปประยุกต์ใช้ในการเปรียบเทียบความคล้ายคลึงกันของส่วนที่เป็น V3 loop ระหว่างชุดโปรตีน gp120 และแสดงให้เห็นว่า HMM สามารถแก้ปัญหาการเปรียบเทียบสายโปรตีนในโจทย์นี้ได้ดีกว่าวิธีการที่อธิบายในบทก่อนหน้าอย่างไร

ถ้ามีข้อมูลลำดับกรดแอมิโนของโปรตีนที่อยู่ในกลุ่มเดียวกัน เราสามารถตรวจสอบได้ว่าสายโปรตีนที่เข้ามาใหม่อยู่ในกลุ่มนี้ด้วยหรือไม่ โดยการเปรียบเทียบความคล้ายคลึงกันระหว่างลำดับกรดแอมิโนของโปรตีนที่เข้ามาใหม่กับลำดับกรดแอมิโนของสายโปรตีนที่อยู่ในกลุ่ม ถ้าผลการเปรียบเทียบผ่านเกณฑ์คะแนนความเหมือนกับอย่างน้อยหนึ่งโปรตีนในกลุ่ม จะสามารถอนุมานได้ว่าโปรตีนเส้นใหม่นี้อยู่ในกลุ่มนี้ด้วย อย่างไรก็ตามการเปรียบเทียบในแนวทางนี้จะให้คำตอบที่ไม่ถูกต้องถ้าโปรตีนที่นำมาเปรียบเทียบนั้นแตกต่างกันค่อนข้างมากอย่างในกรณี V3 loop ของชุดโปรตีน gp120 ของเชื้อเอชไอวี ที่ได้จากการแยกเชื้อแต่ละครั้งจากผู้ป่วยคนเดียวกัน

จากรูปที่ 6.9(ก) แสดงการเปรียบความคล้ายคลึงกันระหว่างส่วนของโปรตีน 5 เส้น โดยแต่ละเส้นมีความยาวของสายโปรตีนเท่ากับ 10 กรดแอมิโน โดยคอลัมน์ที่ <1> และ <2> ในรูป มีหลายบรรทัดที่เป็นสัญลักษณ์แสดงช่องว่าง (‘-’) ซึ่งไม่มีความหมายในเชิงการเกิดความอนุรักษ์ร่วมกันภายในกลุ่ม ทำให้นักชีววิทยามักตัดคอลัมน์เหล่านี้ออกจากผลการเปรียบเทียบความคล้ายคลึงกันระหว่างสายข้อมูลภายในชุดถ้าคอลัมน์มีจำนวนบรรทัดที่มีช่องว่างมากกว่าหรือเท่ากับค่า column removal threshold (θ) โดยรูปที่ 6.9(ข) มีการตัดคอลัมน์ <1> และ <2> ออก และผลในรูปนี้เรียกว่า seed alignment (*Alignment**) ซึ่งจะถูกนำไปสร้างโพรไฟล์เมทริกซ์ ดังแสดงในรูปที่ 6.9(ค) จากนั้นสร้าง HMM จากข้อมูล *Alignment** และโพรไฟล์เมทริกซ์ของ *Alignment** ในการทดสอบความคล้ายคลึงกันของสายโปรตีนใหม่ (Text) กับชุดโปรตีนในกลุ่ม HMM จะคำนวณค่าความน่าจะเป็นของการส่งออกสายโปรตีน Text โดย Text ที่มีความคล้ายคลึงกับ seed alignment มากกว่า จะมีค่าความน่าจะเป็นในการส่งออกสูงกว่า

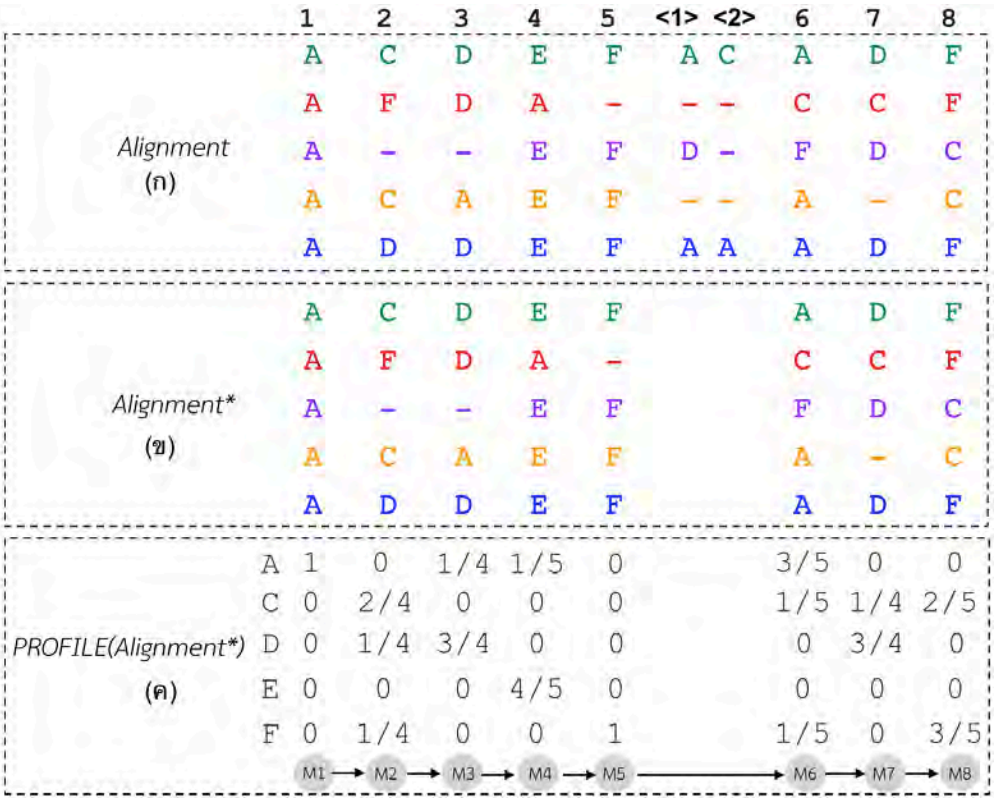
จากลำดับคอลัมน์ของ *Alignment** ในรูปที่ 6.9(ข) เราสามารถสร้าง HMM ที่ประกอบด้วยสถานะแมช (match) เรียงต่อกัน k สถานะ เมื่อ HMM เข้าสู่สถานะ MATCH(i) จะส่งออกอักขระ x_i โดยมีค่าความน่าจะเป็นเท่ากับความถี่ของการเกิดอักขระนั้นในคอลัมน์ที่ i ของ PROFILE(*Alignment**) และมีค่าความน่าจะเป็นในการเปลี่ยนจากสถานะที่ i ไปยังสถานะที่ i+1 เท่ากับ 1 โดยค่าคะแนนความคล้ายคลึงกันระหว่าง *Alignment** กับ Text ที่เป็นสายโปรตีนใหม่ มีค่าเท่ากับค่าความน่าจะเป็น $\Pr(Text)$ ที่ HMM จะส่งออก Text และคะแนนนี้มีค่าเท่ากับผลคูณของค่าความถี่ของอักขระที่แมชในแต่ละคอลัมน์ใน PROFILE(*Alignment**) เช่น ค่าความน่าจะเป็นที่ HMM ในรูปที่ 6.9 จะส่งออกสายอักขระ ADDAFFDF มีค่าเท่ากับ

$$1 \cdot \frac{1}{4} \cdot \frac{3}{4} \cdot \frac{1}{5} \cdot 1 \cdot \frac{1}{5} \cdot \frac{3}{4} \cdot \frac{3}{5} = 0.003375$$

หยุดคิด	อะไรคือข้อจำกัดของ HMM ในรูปที่ 6.9

ถึงแม้ HMM ข้างต้นมีการให้คะแนนในแต่ละคอลัมน์แตกต่างกัน โดย Text ที่มีความคล้ายคลึงกับ *Alignment** มาก จะมีค่าความน่าจะเป็นในการส่งออกสายอักขระมากกว่า Text ที่แตกต่างจาก *Alignment** อย่างไรก็ตาม HMM ข้างต้นยังขาดส่วนที่เป็นหัวใจของ HMM เนื่องจากมีเส้นทางของสถานะซ่อนเร้นเพียง 1

เส้นทาง และยังไม่สามารถจำลองการเกิดการสอดแทรก (insertion) และ การขาดหาย (deletion) ของสายอักขระ รวมทั้งยังจำกัดความยาวของสายข้อมูลเข้าที่ต้องยาวเท่ากับจำนวนคอลัมน์ใน *Alignment** เท่านั้น



รูปที่ 6.9 (ก) ผลการเปรียบเทียบความคล้ายคลึงกันระหว่างสายโปรตีน 5 เส้นในชุด (ข) ผลการเปรียบเทียบความคล้ายคลึงกันระหว่างสายโปรตีน 5 เส้นในชุดโดยตัดคอลัมน์ <1> และ <2> ออก เนื่องจากอัตราส่วนของ '-' เกินค่า θ (0.35) ที่กำหนด (ค) HMM ที่แสดงสถานะแมช (match)

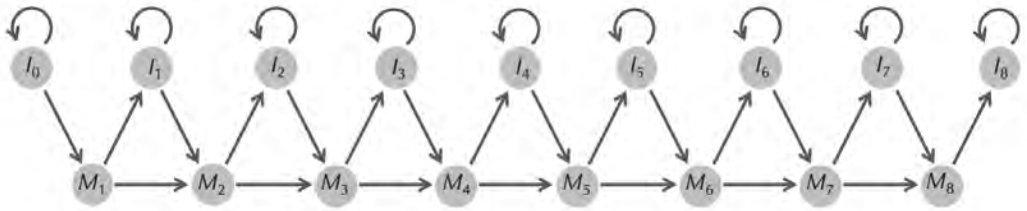
(*ที่มา: ปรับจากรูปที่* 10.9 *ของ* [52])

การสร้างโพรไฟล์ HMM

เพื่อกำจัดข้อจำกัดของ HMM ข้างต้น ได้มีการเสนอ HMM ในรูปแบบที่เรียกว่า โพรไฟล์ HMM (profile HMM) ซึ่งสร้างจาก *Alignment** และแสดงโดย **`HMM(Alignment*)`** เมื่อมีข้อมูลเข้าเป็นโปรตีนสายใหม่ Text เป้าหมายคือการหาเส้นทางแสดงลำดับสถานะซ่อนเร้นที่ให้ค่าความน่าจะเป็นมากสุด ซึ่งเท่ากับการแก้ปัญหา De-coding สำหรับโพรไฟล์ HMM ที่ส่งออกสายอักขระ Text

จากโพรไฟล์ HMM ที่แสดงเฉพาะลำดับสถานะซ่อนเร้นแมชในรูปที่ 6.9 เพื่อให้สามารถนำ Text ที่มีความยาวที่แตกต่างมาเปรียบเทียบได้ ต้องทำการเพิ่มสถานะซ่อนเร้นอื่นๆ นอกเหนือจากสถานะแมชจำนวน k สถานะข้างต้น โดยขั้นแรกทำการเพิ่มสถานะ insertion จำนวน k+1 สถานะแสดงโดย **`INSERTION(0)`**,..., **`INSERTION(k)`** (รูปที่ 6.10) การเพิ่มสถานะ **`INSERTION(i)`** นี้ ทำให้โพรไฟล์ HMM สามารถส่งออก

อักขระเพิ่มเติมหลังจากผ่านคอลัมน์ที่ i ของ PROFILE(*Alignment**) และก่อนเข้าสู่สถานะที่ i+1 ดังนั้นจึงต้องลากเส้นเชื่อมจากสถานะ MATCH(i) ไปยังสถานะ INSERTION(i) และจากสถานะ INSERTION(i) ไปยังสถานะ MATCH(i+1) และเพื่อให้สามารถแทรกได้มากกว่า 1 อักขระในระหว่างคอลัมน์ ใน PROFILE(*Alignment**) จึงต้องเพิ่มเส้นเชื่อมที่สถานะINSERTION(i) ที่ชี้เข้าตัวเองด้วย



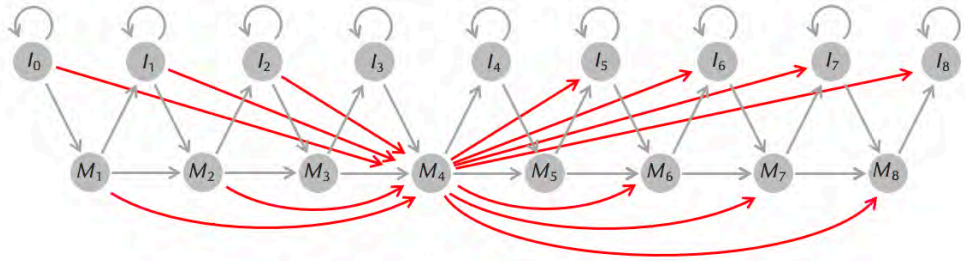
รูปที่ 6.10 แผนภาพ HMM ที่มีการเพิ่มสถานะซ่อนเร้น insertion จำนวน k+1 สถานะ จากรูปที่ 6.9

(*ที่มา*: *รูปที่* 10.11 *ของ* [52])

หยุดคิด	เราสามารถใช้ HMM ในรูปที่ 6.10 ในการเปรียบเทียบความคล้ายคลึงกับสายโปรตีนเข้าที่ยาวน้อยกว่า 8 กรดแอมิโนได้หรือไม่

หลังปรับปรุง HMM ให้สามารถรองรับการเปรียบเทียบสายโปรตีนเข้าที่ยาวกว่าจำนวนลำดับกรดแอมิโน k ใน HMM ตั้งต้น โดยการเพิ่มสถานะซ่อนเร้น insertion ลำดับถัดไปเป็นการปรับปรุง HMM เพิ่มเติมให้รองรับสาย ข้อมูลเข้าที่บางลำดับกรดแอมิโนหายไป (deletion) โดยอนุญาตให้โพรไฟล์ HMM สามารถข้ามบางคอลัมน์ใน PROFILE(*Alignment**) ด้วยการเพิ่มเส้นเชื่อมจากแต่ละสถานะที่มีอยู่ไปยังสถานะที่อยู่ทางขวามืออื่นทั้ง หมดดังตัวอย่างรูปที่ 6.11

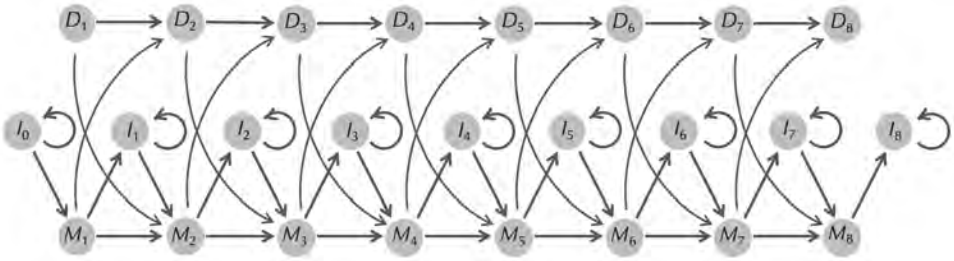
หยุดคิด	จากตัวอย่างของการเพิ่มเส้นเชื่อม (สีแดงทั้งหมด) ที่ชี้เข้าและออกจากโหนด MATCH(4) เพื่อรองรับการเกิด deletion ถ้าให้ลากเส้นเชื่อมจนครบจะมีเส้นเชื่อมเพิ่มทั้งหมดกี่เส้น



รูปที่ 6.11 การปรับ HMM เพื่อให้รองรับสถานะ deletion โดยการลากเส้นเชื่อมเพิ่มเติมจากสถานะหนึ่งๆ ไปยังสถานะอื่นๆ ทั้งหมดทางขวา เส้นเชื่อมสีแดงแสดงการรองรับสถานะ deletion ของโหนด MATCH(4)

(*ที่มา*: *รูปที่* 10.12 *ของ* [52])

จากตัวอย่างการปรับ HMM เพื่อรองรับการเกิด deletion ข้างต้น จะต้องลากเส้นเชื่อมเพิ่มจำนวนมากเพื่อลดจำนวนเส้นเชื่อมเหล่านี้ รูปที่ 6.12 แสดงการปรับ HMM โดยเพิ่มสถานะซ่อนเร้น deletion จำนวน k สถานะซึ่งแทนด้วย `DELETION(i),…,DELETION(k)` ตามรูปที่ 6.12 ซึ่งเส้นเชื่อมจากสถานะ `MATCH(i-1)` ไปยังสถานะ `MATCH(i+1)` เดิมจะถูกแทนที่ด้วยเส้นเชื่อมจากสถานะ `MATCH(i-1)` ไปยังสถานะ`DELETION(i)` และจากสถานะ `DELETION(i)` ไปยังสถานะ `MATCH(i+1)` ทั้งนี้การเข้าสู่สถานะ `DELETION(i)` จะอนุญาตให้ HMM ข้ามบางคอลัมน์ไปโดยไม่มีการส่งออกอักขระในคอลัมน์นั้น



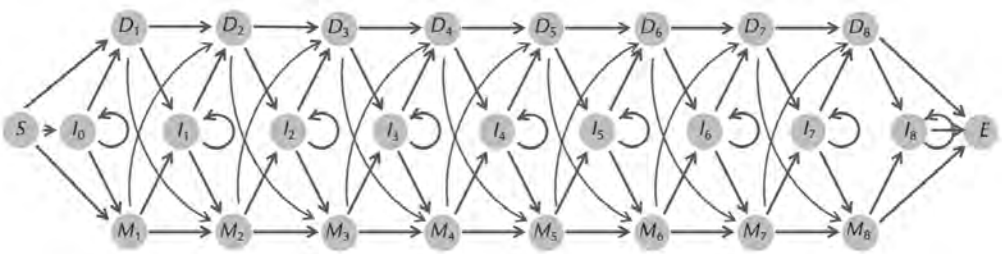
รูปที่ 6.12 การปรับ HMM โดยการเพิ่มสถานะ deletion (โหนด D_i)

(*ที่มา*: *รูปที่* 10.13 *ของ* [52])

หยุดคิด	HMM ในรูปที่ 6.12 นี้ครบถ้วนในการจำลองการเกิดทั้ง insertion และ deletion หรือไม่

จาก HMM ในรูปที่ 6.12 เราสามารถเปลี่ยนสถานะทั้งไปและกลับระหว่างสถานะแมชกับ insertion และระหว่างสถานะแมชกับ deletion แต่ยังไม่มีการเปลี่ยนสถานะระหว่าง insertion กับ deletion ดังนั้นต้องมีการปรับโพรไฟล์ HMM โดยการเพิ่มเส้นเชื่อมระหว่างสถานะ `INSERTION(i)` ไปยังสถานะ `DELETION(i+1)` และจากสถานะ `DELETION(i)` ไปยังสถานะ `INSERTION(i)` รูปที่ 6.13 แสดงโพรไฟล์ HMM ที่สมบูรณ์ หลังจากมีการลากเส้นเชื่อมจากโหนดสถานะตั้งต้น (initial state: S) ไปยังสถานะแมช insertion และ deletion ในคอลัมน์แรก และเส้นเชื่อมจากสถานะแมช insertion และ deletion ในคอลัมน์สุดท้ายไปยังสถานะปลายทาง (terminal state: E)

หยุดคิด	1. แผนภาพ HMM ในรูปที่ 6.13 มีจำนวนเส้นเชื่อมทั้งหมดเท่าใด และแตกต่างจากจำนวนเส้นเชื่อมทั้งหมดในรูปที่ 6.11 อย่างไร 2. กราฟวิเทอบิที่สร้างจากแผนภาพ HMM ในรูปที่ 6.13 มีลักษณะอย่างไร และมีจำนวนโหนดและเส้นเชื่อมเท่าใด



รูปที่ 6.13 โพรไฟล์ HMM ที่ถูกปรับปรุงโดยเพิ่มส่วนที่รองรับการเปลี่ยนสถานะระหว่าง insertion และ deletion รวมทั้งมีการเพิ่มโหนดต้นทาง S และโหนดปลายทาง E

(*ที่มา*: *รูปที่* 10.14 *ของ* [52])

ค่าความน่าจะเป็น Transition และ Emission ของโพรไฟล์ HMM

รูปที่ 6.14 แสดงเส้นทางในโพรไฟล์ HMM ของลำดับกรดแอมิโนในแต่ละบรรทัดของ *Alignment* ในรูปที่ 6.9 โดยแต่ละเส้นทางมีการแยกสีตามบรรทัดของสายข้อมูลโปรตีน กรดแอมิโนที่อยู่ใน seed alignment (*Alignment**) (ไม่รวมคอลัมน์ที่มีพื้นหลังสีเทา) อาจอยู่ในสถานะแมช (เป็นอักขระแสดงกรดแอมิโน) หรือสถานะ deletion (เป็นอักขระ '-') สำหรับอักขระที่ไม่ได้อยู่ใน seed alignment (คอลัมน์ที่มีพื้นหลังเป็นสีเทา) ถ้าเป็นอักขระ '-' จะไม่ถูกนำมาพิจารณา แต่ถ้าเป็นอักขระอื่นจะหมายถึงอักขระที่ถูกส่งออกโดยสถานะ insertion

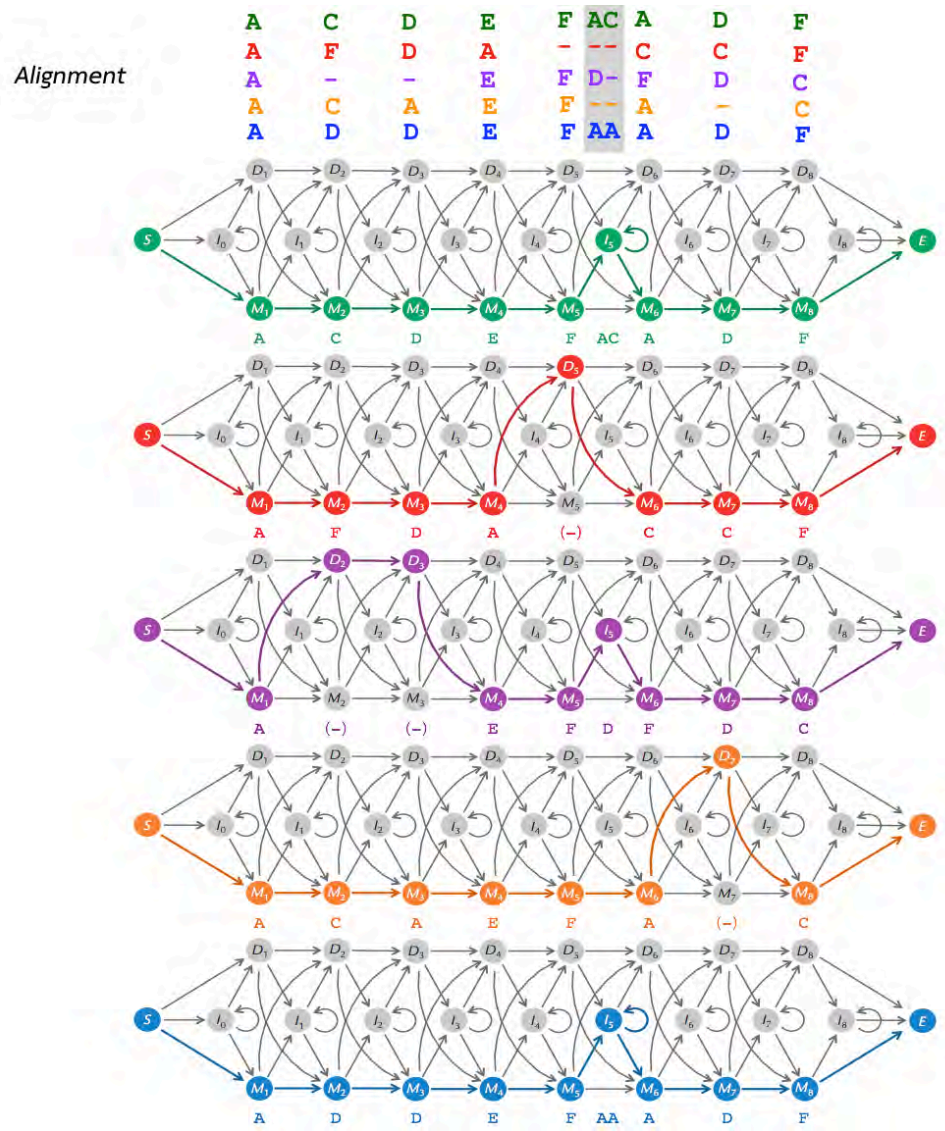
หยุดคิด	ความน่าจะเป็นในการเปลี่ยนจากสถานะหนึ่งไปอีกสถานะหนึ่ง (transition) และความน่าจะเป็นในการส่งออกอักขระหนึ่งๆ (emission) ในแต่ละลำดับจำเพาะของโพรไฟล์ HMM ในรูปที่ 6.14 มีค่าเท่าใด

เราสามารถกำหนดค่าความน่าจะเป็นในการเปลี่ยนสถานะ $\text{transition}_{l,k}$ ได้จากการนับความถี่ในการเปลี่ยนจากสถานะ l ไปยังสถานะ k ของห้าเส้นทางในรูปที่ 6.14 เทียบกับจำนวนเส้นทางทั้งหมดที่ผ่านสถานะ l จากรูปที่ 6.14 มีจำนวน 4 จาก 5 เส้นทางที่ผ่าน `MATCH(5)` และ 3 ใน 4 เส้นทางนี้เปลี่ยนสถานะไปเป็น `INSERTION(5)` ในสถานะถัดไป ในขณะที่อีก 1 เส้นทางจะเปลี่ยนไปเป็นสถานะ `MATCH(6)` ซึ่งสามารถคำนวณค่าความน่าจะเป็นในการเปลี่ยนสถานะได้ดังต่อไปนี้

$$transition_{MATCH(5),INSERTION(5)} = \frac{3}{4}$$

$$transition_{MATCH(5),MATCH(6)} = \frac{1}{4}$$

$$transition_{MATCH(5),DELETION(6)} = 0$$



รูปที่ 6.14 เส้นทางในโพรไฟล์ HMM ที่แสดงลำดับกรดแอมิโนในแต่ละบรรทัดของ Alignment ในรูปที่ 6.9 อักขระ '-' ใต้แต่ละแผนภาพแสดงสถานะ deletion ซึ่งไม่มีการส่งออกกรดแอมิโนในคอลัมน์นั้น

(*ที่มา*: *รูปที่* 10.15 *ของ* [52])

จากรูปที่ 6.14 ความน่าจะเป็นในการเปลี่ยนจากสถานะเริ่มต้น (initial state: S) ไปเป็นสถานะ `MATCH(1)` เท่ากับ 1 สำหรับกรณีโพรไฟล์ HMM ทั่วไป สถานะเริ่มต้นสามารถเปลี่ยนไปเป็นสถานะ `INSERTION(0)` และ `DELETION(1)` ได้อีกสองสถานะ

ค่าความน่าจะเป็นในการส่งออกอักขระ $\text{emission}_k(b)$ สามารถหาได้โดยการหารจำนวนอักขระ b ที่ส่งออกโดยสถานะ k ด้วยจำนวนอักขระที่ถูกส่งออกทั้งหมดโดยสถานะ k ในรูปที่ 6.14 สถานะ `INSERTION(5)` ส่งออกอักขระ **A**, **D** และ **C** เป็นจำนวน 3, 1, และ 1 ครั้งตามลำดับ หรือสถานะ `MATCH(2)` ส่งออกอักขระ **C**,

D และ **F** เป็นจำนวน 2, 1, และ 1 ครั้งตามลำดับ ซึ่งสามารถคำนวณค่าความน่าจะเป็นในการส่งออกอักขระใดๆ ของ สถานะ `INSERTION(5)` และ `MATCH(2)` ได้ดังต่อไปนี้

$$emission_{INSERTION(5)}(A) = \frac{3}{5} \qquad emission_{MATCH(2)}(A) = 0$$

$$emission_{INSERTION(5)}(C) = \frac{1}{5} \qquad emission_{MATCH(2)}(C) = 2/4$$

$$emission_{INSERTION(5)}(D) = \frac{1}{5} \qquad emission_{MATCH(2)}(D) = 1/4$$

$$emission_{INSERTION(5)}(E) = 0 \qquad emission_{MATCH(2)}(E) = 0$$

$$emission_{INSERTION(5)}(F) = 0 \qquad emission_{MATCH(2)}(F) = 1/4$$

เมื่อเสร็จสิ้นการคำนวณค่าเมทริกซ์เปลี่ยนสถานะ (transition matrix) และ เมทริกซ์อิมิชชัน (emission matrix) จากผล multiple sequence alignment (*Alignment*) จะได้โพรไฟล์ HMM ที่สามารถนำไปใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายโปรตีนเข้าใหม่กับชุดของโปรตีนที่ใช้ในการสร้างโพรไฟล์

นิยามปัญหาที่ 6.5 ปัญหาโพรไฟล์ HMM

ปัญหาโพรไฟล์ HMM (Profile HMM Problem) สร้างโพรไฟล์ HMM จากผล multiple sequence alignment	
ข้อมูลเข้า	ผล multiple sequence alignment (*Alignment*) และค่า column removal threshold (θ)
ผลลัพธ์	`HMM(`$Alignment, \theta$`)`

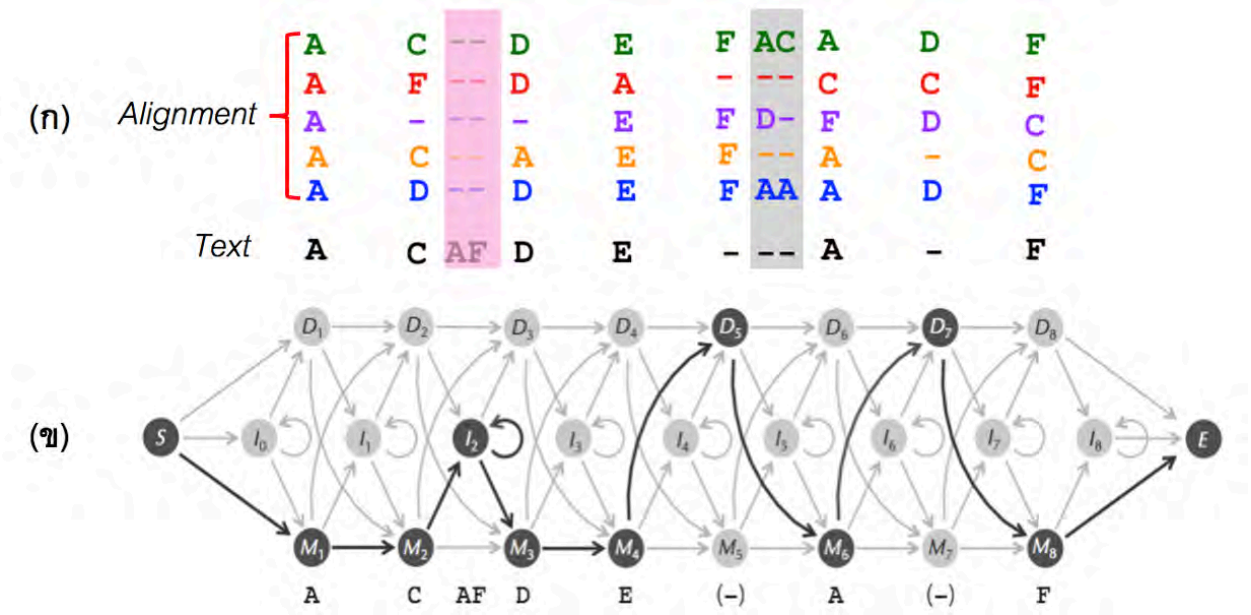
ฝึกหัด	สร้างโพรไฟล์ HMM โดยใช้ชุดของลำดับกรดแอมิโนในตำแหน่ง V3 loop ของโปรตีน gp120 จากเชื้อเอชไอวีในรูปที่ 6.2

การจำแนกโปรตีนโดยใช้โพรไฟล์ HMM

การเทียบสายโปรตีนกับโพรไฟล์ HMM

ผลการเปรียบเทียบความคล้ายคลึงกันของชุดโปรตีนในรูปแบบ *Alignment* สามารถนำมาสร้างโพรไฟล์ HMM หรือ `HMM(`$Alignment, \theta$`)` (นิยามปัญหาที่ 6.5) เพื่อใช้ในการตัดสินว่าสายโปรตีนเข้าใหม่ Text มีโอกาสเป็นสมาชิกของกลุ่มโปรตีนนี้หรือไม่ รูปที่ 6.15(ก) แสดงผลการเทียบสายโปรตีนเข้าใหม่ Text กับ *Alignment* รูปที่ 6.15(ข) แสดงเส้นทางลำดับของสถานะที่สอดคล้องกับผลในรูปที่ 6.15(ก) โดยสองอักขระแรกใน Text อยู่ในสถานะแมช สองอักขระถัดไปอยู่ในสถานะ insertion และมีคอลัมน์เป็นของตัวเอง (คอลัมน์พื้นหลังสีชมพู) อักขระ

‘-’ ในคอลัมน์ที่ 7 และ 11 แสดงสถานะ deletion ซึ่งไม่มีการแสดงอักขระใดโดย HMM ส่วนอักขระ ‘-’ ในคอลัมน์พื้นหลังสีเทาไม่ได้นำมาพิจารณาเนื่องจากถูกตัดออกตั้งแต่ตอนสร้าง HMM ตามเงื่อนไข column removal threshold (θ)



รูปที่ 6.15 (ก) ผลการเทียบสายโปรตีนเข้าใหม่ Text กับ Alignment (ข) เส้นทางลำดับสถานะใน HMM(Alignment, 0.35) ที่สอดคล้องกับผลการเทียบสายโปรตีนเข้าสายใหม่ Text กับ Alignment

(*ที่มา*: *ปรับจากรูปที่* 10.17 *ของ [52]*)

ในการเทียบ Text กับ *Alignment* เราสามารถใช้อัลกอริทึมวิเทอบิเพื่อหาเส้นทางที่ดีที่สุดที่ส่งออก Text จาก `HMM(Alignment, θ)` หรือหาผลคูณของค่าน้ำหนักคะแนนในเส้นทางหนึ่งๆ และเทียบกับเกณฑ์ที่กำหนด เพื่อตัดสินว่าโปรตีนเส้นใหม่ Text เป็นสมาชิกของกลุ่มโปรตีนในโพรไฟล์หรือไม่ ถ้า Text ถูกตัดสินว่าเป็นสมาชิกของกลุ่ม เราสามารถขยายจำนวนข้อมูลใน seed alignment โดยเพิ่มโปรตีนสายใหม่เข้ากลุ่มและทำการอัพเดตตัวแปรที่เกี่ยวข้องใน HMM การขยายจำนวนสมาชิกใน seed alignment ทำให้สามารถจำแนกโปรตีนในกลุ่ม (ที่อาจมีความหลากหลาย) ได้ครอบคลุมมากยิ่งขึ้น

ถึงจุดนี้ควรพบว่าโพรไฟล์ HMM สามารถใช้เป็นเครื่องมือในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายของโปรตีน โดยสามารถบรรลุเป้าหมายที่ต้องการให้แต่ละคอลัมน์ซึ่งเป็นผลของการทำ multiple sequence alignment สามารถมีคะแนนที่แตกต่างกันขึ้นอยู่กับความถี่ของแต่ละอักขระที่ถูกส่งออกในแต่ละคอลัมน์

หยุดคิด	ถ้าผลคูณของค่าน้ำหนักคะแนนเกินกว่าเกณฑ์ที่กำหนดในโปรตีนมากกว่า 1 กลุ่ม จะจำแนกโปรตีนสายใหม่นี้เข้ากลุ่มใด

สูโดเคาท์

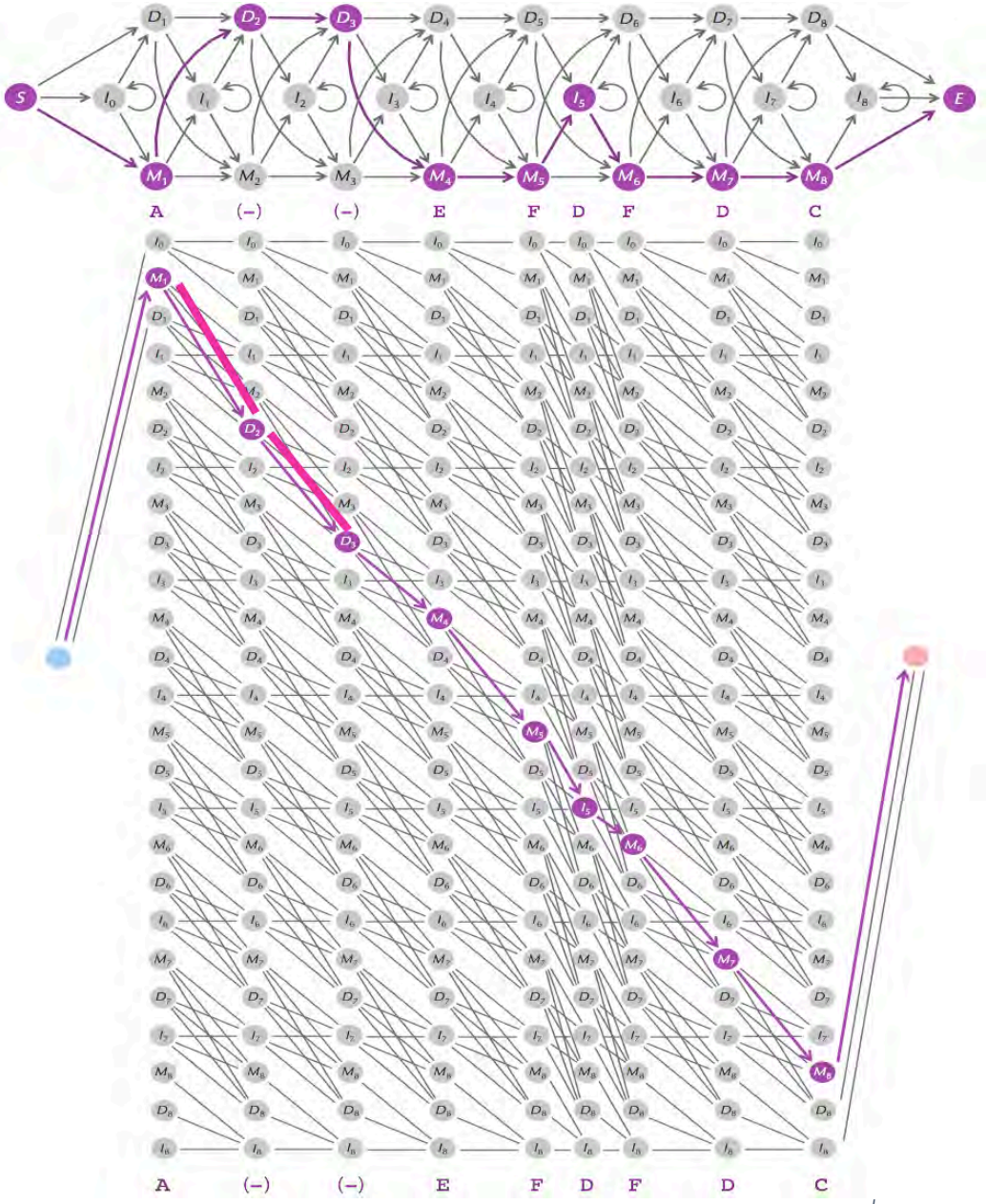
เมทริกซ์เปลี่ยนสถานะ (transition matrix) และ เมทริกซ์อิมิชชัน (emission matrix) ที่สร้างขึ้นจาก *Alignment* อาจมีหลายตำแหน่งที่มีค่าเป็น 0 ซึ่งทำให้เกิดปัญหาได้ ตัวอย่างเช่น เส้นทางในรูปที่ 6.15(ข) ควรเป็นเส้นทางที่ดีที่สุดของ Text = **ACAFDEAF** อย่างไรก็ตามค่า $\Pr(x,\pi)$ ที่คำนวณได้ กลับเป็น 0 เนื่องจากค่าความน่าจะเป็นในการเปลี่ยนจากสถานะ `MATCH(2)` ไปยังสถานะ `INSERTION(2)` ในโพรไฟล์ HMM นี้มีค่าเป็น 0 (ชุดของสายโปรตีนตั้งต้น 5 สายที่นำมาสร้างโพรไฟล์ HMM ไม่มีสายใดเลยที่มีการเปลี่ยนสถานะจาก `MATCH(2)` ไปเป็น `INSERTION(2)` จากบทที่ 4 ในการสร้างโพรไฟล์เพื่อใช้หาโมทิฟควบคุม (regulatory motif) ก็พบปัญหานี้เช่นกัน แนวทางในการแก้ปัญหาทำได้โดยเพิ่มสูโดเคาท์ (pseudo count) โดยในกรณีนี้จะบวกค่าตัวแปร σ (ซึ่งมีค่าน้อย) ให้กับเฉพาะช่องในเมทริกซ์ที่มีโอกาสเกิดขึ้นแต่ยังไม่พบในชุดข้อมูลที่มี เช่น จากสถานะ `MATCH(i)` ไปสถานะ `INSERTION(i)` จากสถานะ `MATCH(i)` ไปสถานะ `DELETION(i+1)` จากสถานะ `DELETION(i)` ไปยังสถานะ `INSETION(i)` จากสถานะ `INSERTION(i)` ไปยังสถานะ `DELETION(i+1)` หลังการเพิ่มค่า σ ให้กับช่องเหล่านี้ ต้องมีการปรับค่าให้เป็นบรรทัดฐาน (normalize) โดยที่ผลรวมของแต่ละคอลัมน์ต้องมีค่าเท่ากับ 1 และสามารถเพิ่มสูโดเคาท์ในเมทริกซ์ที่เก็บค่าความน่าจะเป็นในการส่งออกอักขระด้วยเช่นกัน โพรไฟล์ HMM ที่มีการใส่สูโดเคาท์และปรับค่าให้เป็นมาตรฐานแล้วแสดงด้วยฟังก์ชัน `HMM(Alignment,`θ`,` σ`)`

นิยามปัญหาที่ 6.6 ปัญหาโพรไฟล์ HMM ที่เพิ่มสูโดเคาท์

ปัญหาโพรไฟล์ HMM ที่เพิ่มสูโดเคาท์ (Profile HMM with Pseudo counts Problem) สร้างโพรไฟล์ HMM ที่มีการเพิ่มสูโดเคาท์ จากผล multiple sequence alignment	
ข้อมูลเข้า	ผล multiple sequence alignment (*Alignment*) ค่า column removal threshold (θ) และค่าสูโดเคาท์ σ
ผลลัพธ์	`HMM(`*`Alignment`*`,`θ`,` σ`)`

หยุดคิด	แผนภาพ HMM ในรูปที่ 6.15 มี 25 โหนด (ไม่รวมโหนดตั้งต้นและโหนดปลายทาง) ถ้าสร้างกราฟวิเทอบิจากแผนภาพนี้เพื่อการจำลองการส่งออกอักขระ จะประกอบด้วย 25 แถวและกี่คอลัมน์

ในการเทียบสายข้อมูลเข้า `Text(AEFDFDC)` กับโพรไฟล์ HMM เพื่อหาเส้นทางที่ดีที่สุดที่ส่งออก Text จาก `HMM(`*`Alignment`*`,`θ`)`เริ่มจากการสร้างกราฟวิเทอบิของสายข้อมูลเข้า (รูปที่ 6.16) และแก้ปัญหา Decoding เพื่อหาเส้นทางแสดงลำดับสถานะที่น่าจะเป็นมากที่สุด



รูปที่ 6.16 กราฟวิเทอบิของ $HMM(Alignment, \theta)$ *และเส้นทางในกราฟ (เส้นสีม่วง) ที่สอดคล้องกับสายอักขระที่ส่งออก* **AEFDFDC** *เส้นเชื่อมระหว่างคอลัมน์แสดงถึงการเปลี่ยนสถานะที่เป็นไปได้ซึ่งมีทิศทางมุ่งไปทางขวา เส้นสีชมพูเข้มแสดงส่วน deletion และด้านล่างสุดแสดงอักขระที่ส่งออกในแต่ละคอลัมน์*

(*ที่มา*: *รูปที่* 10.18 *ของ* [52])

หยุดคิด	ถ้าต้องหาเส้นทางแสดงลำดับสถานะผ่านกราฟวิเทอบิสำหรับเส้นทางสีเขียว แดง ส้ม และฟ้าในรูปที่ 6.14 จะเกิดอะไรขึ้น

ปัญหาของสถานะเงียบ

การหาเส้นทางแสดงลำดับสถานะที่ดีที่สุดโดยการประยุกต์ใช้วิธีการแก้ปัญหา Decoding ไม่ตรงไปตรงมาเหมือนตัวอย่างในตอนต้นบทเรียน เนื่องจากกราฟในรูปที่ 6.16 ไม่ใช่กราฟวิเทอบิ พิจารณาเส้นทางในรูปที่ 6.17 ซึ่งส่งออกสายอักขระเดียวกับที่ส่งออกในรูปที่ 6.16 แต่เส้นทางในรูปที่ 6.17 เดินผ่านโหนดที่เป็นสถานะเงียบ (silent state) หรือโหนดที่เป็นสถานะ deletion เพียงโหนดเดียว (ในรูปที่ 6.16 เดินผ่านสองโหนด) ทำให้จำนวนคอลัมน์ในรูปที่ 6.17 ลดลงไปหนึ่งคอลัมน์ (คอลัมน์ที่มีพื้นหลังสีเทาถูกตัดออกไป) อย่างไรก็ตาม เราไม่สามารถเปลี่ยนแปลงกราฟวิเทอบิให้สอดคล้องกับเส้นทางแสดงสถานะซ่อนเร้น π ใดๆ ได้ เนื่องจากเราไม่ทราบเส้นทางเหล่านี้ล่วงหน้า ในทางกลับกันจำนวนคอลัมน์ของกราฟวิเทอบิจะต้องเท่ากับความยาวของสายอักขระที่ส่งออก ซึ่งเงื่อนไขนี้ไม่เป็นจริงสำหรับกราฟวิเทอบิทั้งในรูปที่ 6.16 และ 6.17

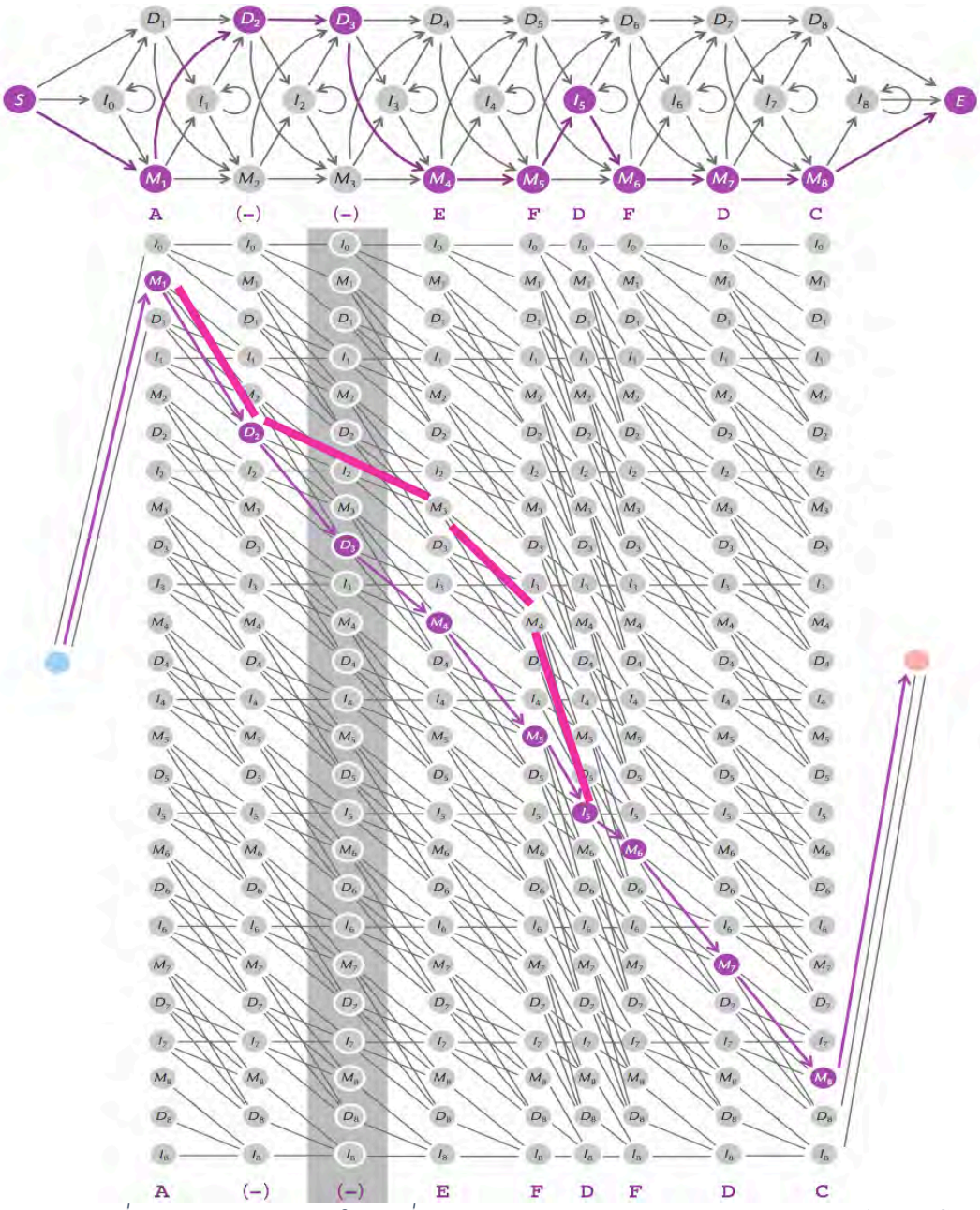
หยุดคิด	เราจะสามารถปรับแก้กราฟวิเทอบิสำหรับ HMM ที่มีสถานะเงียบได้อย่างไร

โดยทั่วไปอัลกอริทึมวิเทอบิไม่อนุญาตให้มีสถานะเงียบอื่นๆ นอกจากสถานะตั้งต้นและสถานะปลายทาง หรืออีกนัยหนึ่งคือตัวอัลกอริทึมมีสมมติฐานว่าโหนด (k,i) ในกราฟอธิบายเหตุการณ์ที่ HMM จะส่งออกอักขระ x_i เมื่ออยู่ในสถานะ k อย่างไรก็ตามถ้า k เป็นสถานะเงียบ บทบาทของโหนด (k,i) ในกราฟวิเทอบิจะไม่ชัดเจน เพราะไม่สามารถระบุค่าความน่าจะเป็นในการเปลี่ยนจากสถานะใดๆ มายังสถานะนี้ อย่างไรก็ตามในกรณีของโพรไฟล์ HMM เราสามารถแก้ปัญหานี้ โดยกำหนดกราฟวิเทอบิที่มีจำนวนแถวเท่ากับจำนวนสถานะ หรือ |States| และมีจำนวนคอลัมน์เท่ากับความยาวของ Text หรือ |Text| โดยทุกครั้งที่ HMM มีการเปลี่ยนจากสถานะใดๆ มายังสถานะ deletion จะไม่มีการขยับคอลัมน์ไปทางขวาในกราฟวิเทอบิแต่มีเส้นทางภายในคอลัมน์แทน แต่ถ้า HMM เปลี่ยนไปยังสถานะแมชหรือ insertion จะมีการขยับไปยังคอลัมน์ถัดไปทางขวา ผลที่ตามมาคือทุกคอลัมน์ในกราฟวิเทอบิจะส่งออกอักขระใดอักขระหนึ่งถึงแม้เส้นทางแสดงลำดับสถานะอาจผ่านมากกว่าหนึ่งสถานะในคอลัมน์หนึ่งๆ (รูปที่ 6.18)

หยุดคิด	จากกราฟวิเทอบิในรูปที่ 6.18 ยังมีประเด็นที่ต้องพิจารณาเพิ่มเติมอีกหรือไม่

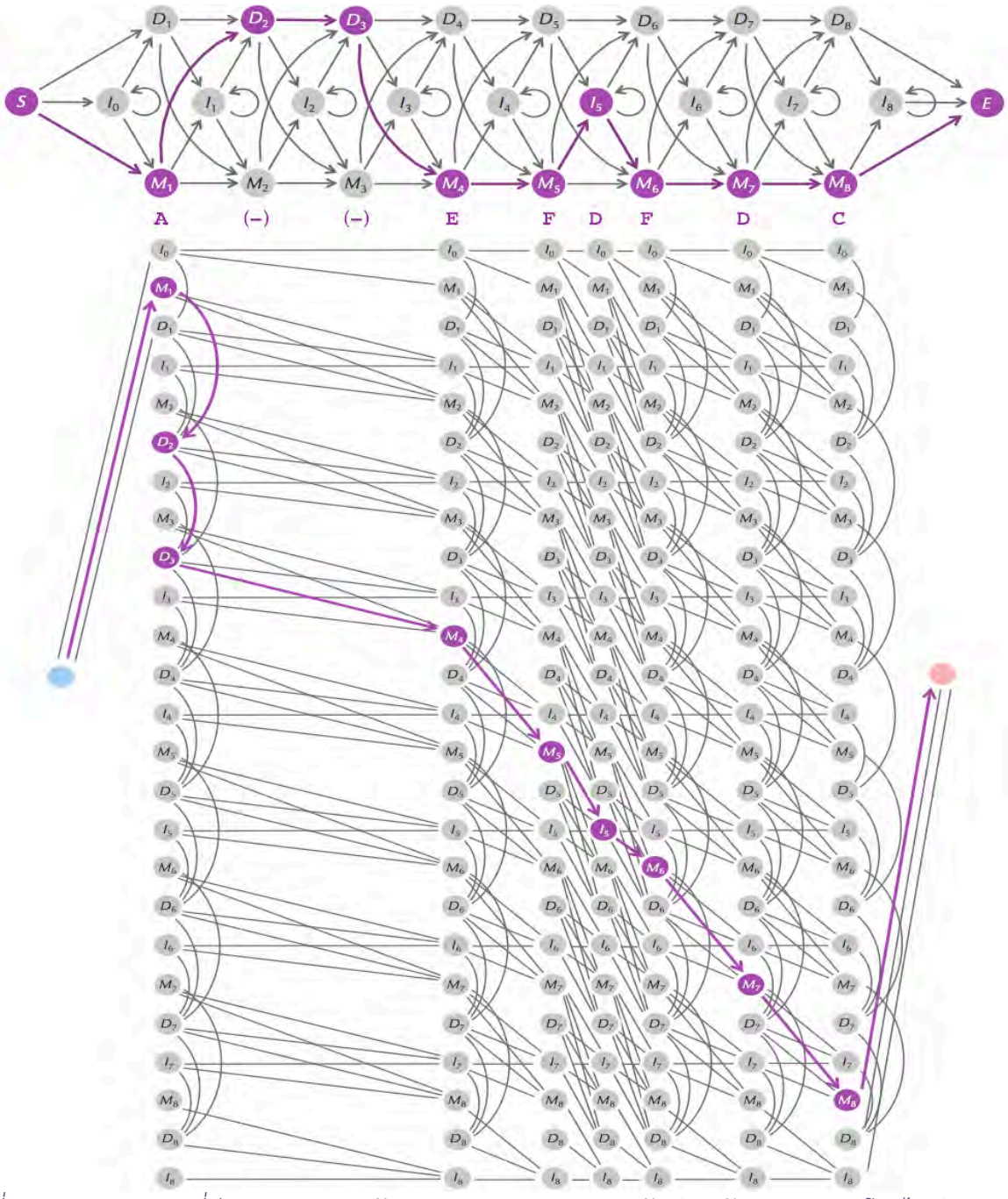
รูปที่ 6.18 ยังมีข้อจำกัดอยู่บ้าง โดยถ้า HMM มีการเปลี่ยนจากสถานะตั้งต้นไปยังสถานะ `DELETION(1)` จะไม่มีการส่งออกอักขระใดๆ ในคอลัมน์ที่ 1 ดังนั้นจึงต้องมีการเปลี่ยนรูปแบบของโหนดตั้งต้นไปเป็นคอลัมน์ของสถานะเงียบซึ่งประกอบด้วยโหนดตั้งต้นและสถานะ deletion ทั้งหมด (รูปที่ 6.19) ด้วยการปรับเปลี่ยนนี้ถ้า

HMM เข้าสู่สถานะ `DELETION(1)` ตั้งแต่เริ่มต้น ก็ยังสามารถผ่านสถานะ deletion อื่นๆ ก่อนเข้าสู่สถานะแมชหรือ insertion ในคอลัมน์ที่ 1 ต่อไป



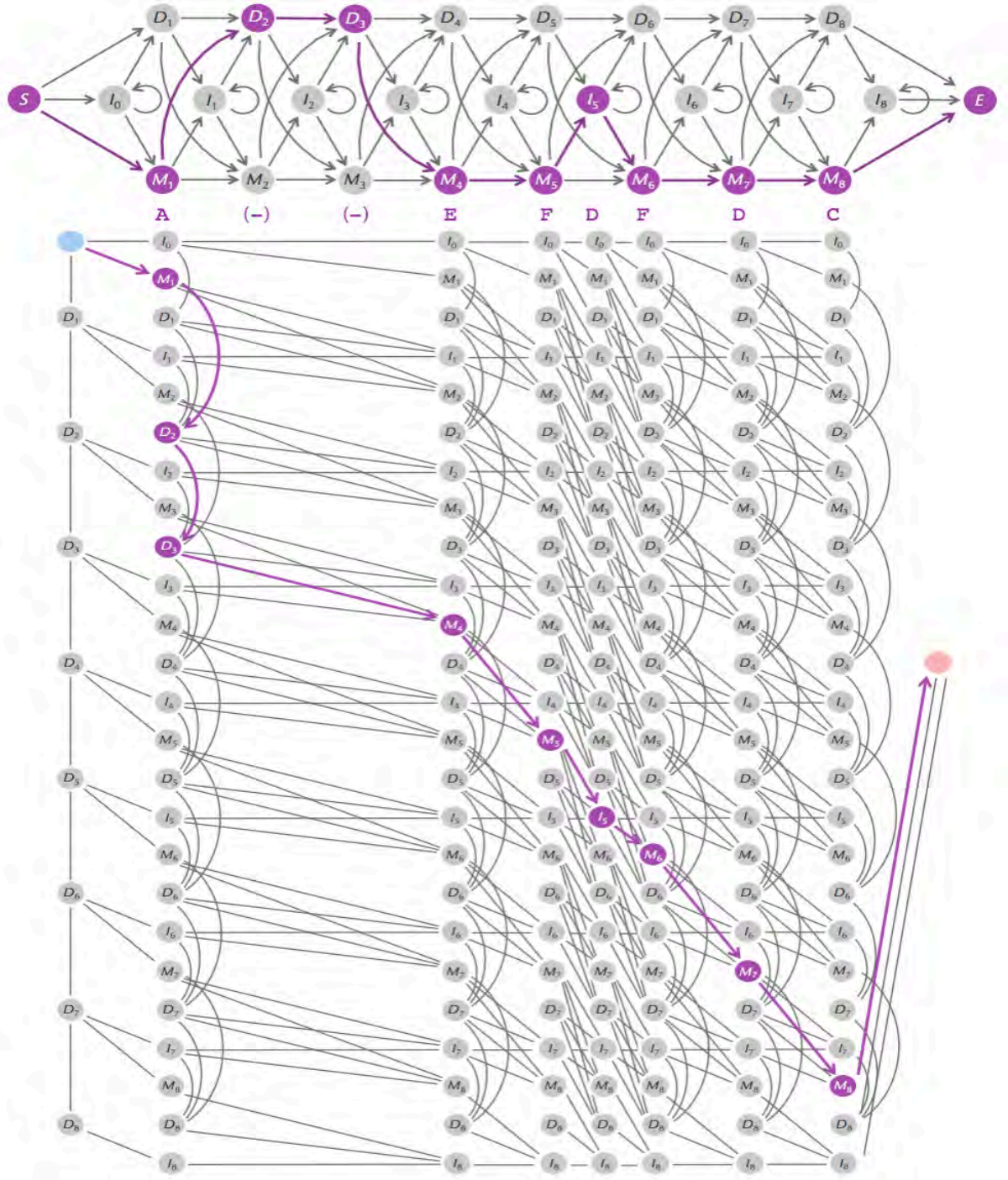
*รูปที่ 6.17 เส้นทางที่แตกต่างจากเส้นทางในรูปที่ 16.6 แต่ส่งออกสายอักขระ **AEFDFDC** เดียวกัน โดยตัดคอลัมน์ที่มีพื้นหลังสีเทาออก*

(*ที่มา*: *รูปที่* 10.18 *ของ [52]*)



รูปที่ 6.18 กราฟวิเทอบิที่มีจำนวนแถวเท่ากับ |States| และจำนวนคอลัมน์เท่ากับ |Text| ของโพรไฟล์ HMM ที่ส่งออกสายอักขระ **AEFDFDC** *โดยเส้นเชื่อมที่แสดงการเปลี่ยนจากสถานะใดๆ มายังสถานะ deletion จะอยู่ภายในคอลัมน์เดียวกัน*

(*ที่มา*: *รูปที่* 10.20 *ของ* [52])



*รูปที่ 6.19 กราฟวิเทอบิสุดท้ายของโพรไฟล์ HMM ที่ส่งออกอักขระจำนวน 7 ตัว โดยเส้นเชื่อมในคอลัมน์เดียวกันมีทิศทางชี้ลง ในขณะที่เส้นเชื่อมระหว่างคอลัมน์มีทิศทางชี้ไปทางขวามือ ทั้งนี้เส้นทางสีม่วงแสดงเส้นทางใน HMM ที่ส่งออกอักขระ **AEFDFDC***

(*ที่มา*: *รูปที่* 10.21 *ของ* [52])

ประโยชน์ของโพรไฟล์ HMM

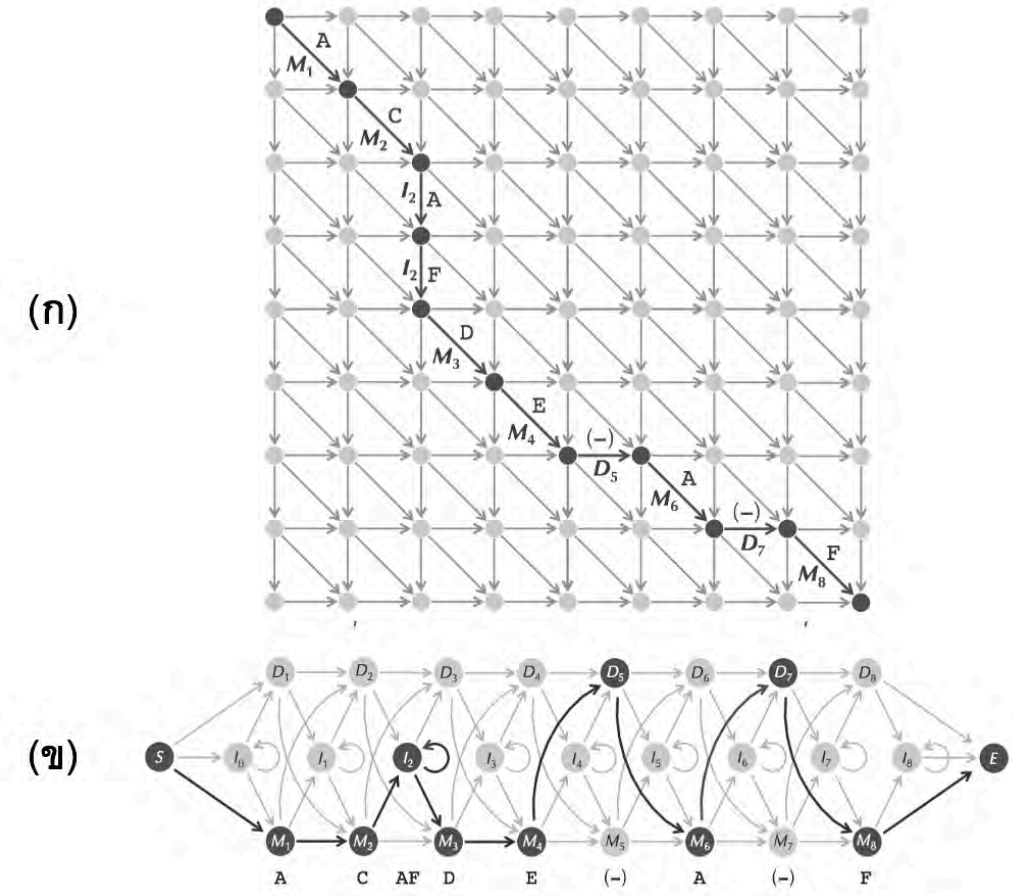
อัลกอริทึมวิเทอบิสามารถประยุกต์ใช้ได้กับ HMM ใดๆ สำหรับกรณีของโพรไฟล์ HMM ถ้ากำหนด $S_{MATCH(j),i}$ เป็นค่าความน่าจะเป็นของเส้นทางแสดงลำดับสถานะซ่อนเร้นที่ดีที่สุดของสายอักขระที่ส่งออก $x_1 \dots x_i$ ของ x โดยปิดท้ายด้วยสถานะซ่อนเร้น MATCH(j) และกำหนด $S_{INSERTION(j),i}$ และ $S_{DELETION(j),i}$ ในลักษณะเดียวกัน เนื่องจากมีเส้นเชื่อมเพียง 3 เส้นที่ชี้เข้าสู่สถานะแมช MATCH(j) ความสัมพันธ์เวียนเกิดของวิเทอบิสามารถแสดงได้ด้วยสมการต่อไปนี้

$$S_{MATCH(j),i} = max \begin{cases} S_{MATCH(j-1),i-1} \cdot WEIGHT_i(MATCH(j-1), MATCH(j)) \\ S_{INSERTION(j-1),i-1} \cdot WEIGHT_i(INSERTION(j-1), INSERTION(j)) \\ S_{DELETION(j-1),i-1} \cdot WEIGHT_i(DELETION(j-1), DELETION(j)) \end{cases}$$

ถ้าใส่ฟังก์ชันลอการิทึมทั้งสองฝั่งจะได้สมการชุดใหม่ดังต่อไปนี้ซึ่งมีความคล้ายคลึงกับชุดของสมการในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนสองเส้นแบบองค์รวม (global pairwise alignment)

$$\log(S_{MATCH(j),i}) = max \begin{cases} \log(S_{MATCH(j-1),i-1}) + \log(WEIGHT_i(MATCH(j-1), MATCH(j))) \\ \log(S_{INSERTION(j-1),i-1}) + \log(WEIGHT_i(INSERTION(j-1), INSERTION(j))) \\ \log(S_{DELETION(j-1),i-1}) + \log(WEIGHT_i(DELETION(j-1), DELETION(j))) \end{cases}$$

รูปที่ 6.20(ก) แสดงเส้นทางในกราฟที่มีลักษณะใกล้เคียงกับกราฟแมนฮัตตัน โดยเส้นทางนี้สอดคล้องกับเส้นทางเดียวกันในโพรไฟล์ HMM เส้นเชื่อมทแยงมุม เส้นเชื่อมแนวตั้งชี้ลง และเส้นเชื่อมแนวนอนชี้ไปทางขวาในกราฟนี้แสดงสถานะแมช insertion และ deletion ตามลำดับ รูปที่ 6.20 นี้อาจทำให้บางคนรู้สึกเสียเวลากับบทเรียนเรื่อง HMM และโพรไฟล์ HMM เนื่องจากเส้นทางแสดงสถานะซ่อนเร้นและเส้นทางในโพรไฟล์ HMM เป็นเส้นทางเดียวกับเส้นทางของการเปรียบเทียบความคล้ายคลึงกันของข้อมูลสองสายแบบองค์รวม (รูปที่ 6.20(ก)) อย่างไรก็ตามเมื่อพิจารณาในรายละเอียดจะพบว่าการเลือกเส้นเชื่อมในรูปที่ 6.20(ข) มีความแตกต่างกันไปตามค่าความน่าจะเป็นในการเปลี่ยนจากสถานะซ่อนเร้นหนึ่งไปยังอีกสถานะซ่อนเร้นหนึ่ง และค่าความน่าจะเป็นในการส่งออกแต่ละอักขระของสถานะซ่อนเร้นแต่ละลำดับของเส้นทางหนึ่งๆ โดยการได้มาซึ่งค่าพารามิเตอร์ของแต่ละคอลัมน์ในเมทริกซ์ที่แสดงผลการเปรียบเทียบความคล้ายคลึงกันของสายข้อมูล (alignment matrix) และสร้างเป็นโพรไฟล์ HMM ทำให้เราสามารถตรวจสอบความคล้ายคลึงกันของสายข้อมูลได้ละเอียดกว่า



รูปที่ 6.20 (ก) เส้นทางผ่านกราฟที่มีลักษณะใกล้เคียงกับกราฟแมนฮัตตันที่สอดคล้องกับเส้นทางแสดงลำดับสถานะซ่อนเร้นด้านล่าง (ข) เส้นทางแสดงลำดับสถานะซ่อนเร้นผ่านโพรไฟล์ HMM และส่งออกสายอักขระ ACAFDEAF

(*ที่มา*: *ปรับจากรูปที่* 10.23 *ของ* [52])

การเรียนรู้พารามิเตอร์ใน HMM

การประมาณค่าพารามิเตอร์ใน HMM โดยทราบวิถีซ่อนเร้น

สมมติฐานหลักของ HMM ในหัวข้อก่อนหน้าคือเราทราบค่าพารามิเตอร์ต่างๆ ของ HMM เช่นค่าความน่าจะเป็นในการเปลี่ยนสถานะซ่อนเร้น Transition และค่าความน่าจะเป็นในการส่งออกอักขระหนึ่งๆ ซึ่งในความเป็นจริงแล้วความซับซ้อนหลักในการประยุกต์ใช้ HMM เพื่อตอบโจทย์ทางชีววิทยาคือการประมาณค่าพารามิเตอร์เหล่านี้จากข้อมูลที่มีอยู่ ถ้าเปรียบเทียบกับปัญหาคาสิโนก่อนหน้า คือทราบว่าเจ้ามือมีทั้งเหรียญปกติและเหรียญถ่วงน้ำหนัก แต่ไม่ทราบว่าเหรียญถ่วงน้ำหนักนั้นจะถูกใช้ในการโยนรอบไหนบ้าง (ไม่ทราบโอกาสในการเปลี่ยนจากเหรียญปกติไปใช้เหรียญถ่วงน้ำหนักรวมทั้งในทางกลับกัน) และไม่ทราบค่าความน่าจะเป็นในการออกหน้าเหรียญหัวหรือก้อยของเหรียญถ่วงน้ำหนัก

หยุดคิด	ถ้าลำดับหน้าของเหรียญที่ปรากฏคือ $x =$ “HHTHHHTHHTTTH” จะสามารถบอกจำนวนครั้งที่ใช้เหรียญถ่วงน้ำหนักและความน่าจะเป็นในการเปลี่ยนเหรียญที่โยนได้หรือไม่ และถ้าทราบเส้นทางของสถานะซ่อนเร้น π = FFFBBFFFFFBBB จะเปลี่ยนผลคำตอบข้างต้นหรือไม่

เราไม่สามารถหาค่าของเมทริกซ์เปลี่ยนสถานะ (transition matrix) เมทริกซ์อิมิชชัน (emission matrix) และ π ถ้าทราบเพียงสายข้อมูล x ที่ส่งออกจาก HMM ทั้งนี้เพื่อให้สามารถหาค่าของเมทริกซ์ข้างต้นได้จำเป็นต้องมีสมมติฐานเพิ่มเติมว่านอกจากทราบสายข้อมูล x แล้วยังทราบค่าพารามิเตอร์ (เมทริกซ์) หรือ π อย่างใดอย่างหนึ่งด้วย ในหัวข้อที่ผ่านมาถ้าทราบสายข้อมูล x ที่ส่งออกและค่าพารามิเตอร์ เราจะสามารถหาเส้นทางสถานะซ่อนเร้น π ที่มีโอกาสเกิดมากสุดโดยใช้อัลกอริทึมวิเทอบิ คำถามคือถ้าทราบสายข้อมูล x และเส้นทางสถานะซ่อนเร้น π จะประมาณค่าพารามิเตอร์ต่างๆ ได้อย่างไร

นิยามปัญหาที่ 6.7 ปัญหาการประมาณค่าพารามิเตอร์ของ HMM

ปัญหาการประมาณค่าพารามิเตอร์ของ HMM (HMM Parameter Estimation Problem) หาชุดของค่าพารามิเตอร์ที่เหมาะที่สุดในการอธิบายการส่งออกสายอักขระ x และเส้นทางแสดงลำดับสถานะซ่อนเร้น π ของ HMM	
ข้อมูลเข้า	สายอักขระ $x = x_1 x_{2 \ldots} x_n$ ที่ถูกส่งออกและเส้นทางแสดงลำดับสถานะซ่อนเร้น $\pi = \pi_1 \ldots \pi_n$ ของ HMM โดยไม่ทราบค่าความน่าจะเป็นของเมทริกซ์เปลี่ยนสถานะและเมทริกซ์อิมิชชัน
ผลลัพธ์	ค่าความน่าจะเป็นที่ทำให้ $\Pr(x, \pi)$ มากสุด สำหรับทุกเมทริกซ์เปลี่ยนสถานะและเมทริกซ์อิมิชชัน

ถ้าทราบทั้ง x และ π จะสามารถประมาณค่าความน่าจะเป็นของการเปลี่ยนสถานะ (Transition) ได้จากการทดลองคำนวณ เช่นถ้า $T_{l,k}$ แสดงจำนวนการเปลี่ยนสถานะจากสถานะซ่อนเร้น l มาเป็นสถานะซ่อนเร้น k ในเส้นทางแสดงลำดับสถานะซ่อนเร้น π เราสามารถคำนวณค่าความน่าจะเป็น $transition_{l,k}$ โดยคำนวณอัตราส่วนของ $T_{l,k}$ เทียบกับจำนวนของการเปลี่ยนจากสถานะซ่อนเร้น l ไปยังสถานะซ่อนเร้นอื่นๆ ทั้งหมด ดังสมการต่อไปนี้

$$transition_{l,k} = \frac{T_{l,k}}{\sum_{all\ states\ j} T_{l,j}}$$

เช่นเดียวกัน สำหรับค่าความน่าจะเป็นอิมิชชัน ถ้า $E_k(b)$ แสดงจำนวนการส่งออกอักขระ b ในสถานะซ่อนเร้น k โดยมีเส้นทาง π จะสามารถประมาณค่าความน่าจะเป็น $emission_k(b)$ จากอัตราส่วนของ $E_k(b)$ เทียบกับจำนวนครั้งทั้งหมดที่ส่งออกอักขระใดๆ ในสถานะซ่อนเร้น k ดังสมการต่อไปนี้

$$emission_k(b) = \frac{E_k(b)}{\sum_{all\ symbols\ c\ in\ the\ alphabet} E_k(c)}$$

สมการทั้งสองนี้ทำให้เราสามารถประมาณค่าพารามิเตอร์เมทริกซ์เปลี่ยนสถานะและเมทริกซ์อิมิชชันได้

การเรียนรู้วิเทอบิ

จากความรู้ที่ว่าถ้าทราบสายข้อมูล x และชุดข้อมูลพารามิเตอร์ (Parameter) จะสามารถหาเส้นทางแสดงสถานะซ่อนเร้นที่มีโอกาสเกิดมากสุด π ได้ โดยใช้อัลกอริทึมวิเทอบิในการแก้ปัญหา Decoding

$$(x, ?, Parameters) \rightarrow \pi$$

ในทางกลับกันถ้าทราบ x และ π จะสามารถประมาณค่าพารามิเตอร์ได้

$$(x, \pi, ?) \rightarrow Parameters$$

หยุดคิด	การแสดงค่า $(x, \pi, ?) \rightarrow Parameters$ และ $(x, ?, Parameters) \rightarrow \pi$ ทำให้นึกถึงอะไร

นิยามปัญหาที่ 6.8 ปัญหาการเรียนรู้ค่าพารามิเตอร์ของ HMM

ปัญหาการเรียนรู้ค่าพารามิเตอร์ของ HMM (HMM Parameter Learning Problem) ประมาณค่าพารามิเตอร์ของ HMM เพื่ออธิบายการส่งออกสายอักขระ x	
ข้อมูลเข้า	สายอักขระ $x = x_1x_2\ldots x_n$ ที่ถูกส่งออกโดย HMM โดยไม่ทราบค่าความน่าจะเป็นในเมทริกซ์เปลี่ยนสถานะและเมทริกซ์อิมิชชัน
ผลลัพธ์	ค่าความน่าจะเป็นในเมทริกซ์เปลี่ยนสถานะและเมทริกซ์อิมิชชัน ที่ทำให้ $\Pr(x, \pi)$ มีค่ามากสุดสำหรับทุกเมทริกซ์เปลี่ยนสถานะ เมทริกซ์อิมิชชันและ π ที่เป็นไปได้

ปัญหาการเรียนรู้พารามิเตอร์ของ HMM เป็นปัญหาที่ยากในการหาคำตอบ ทั้งนี้จึงมีการนำฮิวริสติกมาช่วยโดยข้อมูลพารามิเตอร์ในรอบแรกมาจากการสุ่ม และใช้ข้อมูลพารามิเตอร์นี้กับ x ในการหา π และเมื่อได้ π มาแล้ว ย้อนกลับมาพิจารณาค่าพารามิเตอร์ โดยใช้ค่า x และ π และวนซ้ำระหว่างสองขั้นตอนนี้ โดยหวังว่าค่าประมาณพารามิเตอร์จะเข้าใกล้คำตอบ โดยแนวทางเรียนรู้ค่าพารามิเตอร์ของ HMM นี้เรียกว่า การเรียนรู้วิเทอบิ (Viterbi learning)

หยุดคิด	มีโอกาสไหมที่ระหว่างรอบของการเรียนรู้วิเทอบิ ค่า $\Pr(x, \pi)$ จะลดลง และใช้เงื่อนไขใดในการหยุดการวนซ้ำ

ทั้งนี้ยังไม่มีการระบุเงื่อนไขการหยุดของการเรียนรู้วิเทอบิ ในทางปฏิบัติมีกฎในการหยุดหลายแนวทาง เช่น หยุดการทำงานเมื่อจำนวนรอบในการวนซ้ำเกินค่าที่กำหนดไว้ หรือหยุดเมื่อค่าความน่าจะเป็น $\Pr(x, \pi)$ มีการเปลี่ยนแปลงระหว่างรอบน้อยกว่าค่าที่กำหนด นอกจากนี้เนื่องจากการเรียนรู้วิเทอบิขึ้นอยู่กับค่าพารามิเตอร์ที่เกิดจากการเดาสุ่มในรอบแรก ผลการเรียนรู้วิเทอบิอาจติดอยู่ในค่าต่ำสุดเฉพาะที่ (local optimum) เช่นเดียวกับการแก้ปัญหาอื่นๆ โดยการใช้ฮิวริสติกจำเป็นต้องมีการรันอัลกอริทึมซ้ำหลายๆ ครั้ง เพื่อหาชุดค่าพารามิเตอร์ที่ดีที่สุด

ฝึกหัด	ประยุกต์ใช้การเรียนรู้วิเทอบิในการเรียนรู้พารามิเตอร์ของ HMM สำหรับ CG-island และโพรไฟล์ HMM สำหรับ gp120 ของเชื้อเอชไอวี

การประมาณค่าพารามิเตอร์ของ HMM แบบยืดหยุ่น

ปัญหา Soft Decoding

เส้นทางแสดงลำดับสถานะซ่อนเร้นที่ดีที่สุดโดยใช้อัลกอริทึมวิเทอบิจะให้คำตอบเพียงใช่หรือไม่ใช่สำหรับคำถามว่าที่เวลา i นั้นสถานะซ่อนเร้นคือ k ใช่หรือไม่ ประเด็นคือเราสามารถมั่นใจกับคำตอบนี้ได้เพียงใด ลองพิจารณาปัญหาคาสิโนอีกครั้ง สมมติว่าการโยนเหรียญรอบที่ i ออกหัว ถ้าหัวที่ออกนี้อยู่ในลำดับตรงกลางของการออกหัว 10 ครั้งติดกันอาจมีความมั่นใจมากขึ้นว่าเจ้ามือน่าจะใช้เหรียญถ่วงน้ำหนักในรอบการโยนเหรียญเหล่านั้น อย่างไรก็ตาม ถ้าผลการออกหน้าเหรียญใน 10 ครั้งนั้น ออกหัว 6 ครั้งและออกก้อย 4 ครั้ง ในกรณีนี้เราอาจมีความมั่นใจลดลงว่าเจ้ามือใช้เหรียญถ่วงน้ำหนักในรอบการโยนเหล่านั้น

ในกรณีของ HMM ใดๆ เราต้องการคำนวณค่าความน่าจะเป็นแบบมีเงื่อนไข (conditional probability) $\Pr(\pi_i = k|x)$ โดยที่ HMM อยู่ในสถานะซ่อนเร้น k ที่เวลา i และมีการส่งออกสายอักขระ x

นิยามปัญหาที่ 6.9 ปัญหา Soft Decoding

ปัญหา Soft Decoding หาค่าความน่าจะเป็นที่ HMM อยู่ในสถานะซ่อนเร้นและเวลาที่จำเพาะ โดยทราบสายอักขระส่งออก x	
ข้อมูลเข้า	สายอักขระ $x = x_1 x_{2\ldots} x_n$ ที่ถูกส่งออกโดย HMM
ผลลัพธ์	ค่าความน่าจะเป็นแบบมีเงื่อนไข $\Pr(\pi_i = k|x)$ ที่ HMM อยู่ในสถานะซ่อนเร้น k ที่เวลาหรือรอบที่ i โดยมีการส่งออกสายอักขระ x

ค่าความน่าจะเป็นแบบ *ไม่มี* เงื่อนไขที่เส้นทางแสดงลำดับสถานะซ่อนเร้นผ่านสถานะ k ที่เวลา i และส่งออกสายอักขระ x สามารถคำนวณจากสมการผลรวมต่อไปนี้

$$\Pr(\pi_i = k, x) = \sum_{all\ paths\ \pi\ with\ \pi_i=k} \Pr(x, \pi)$$

และค่าความน่าจะเป็นแบบมีเงื่อนไข $\Pr(\pi_i = k|x)$ เท่ากับสัดส่วนของเส้นทางที่ผ่านสถานะ k ที่เวลา i และส่งออกสายอักขระ x เทียบกับจำนวนเส้นทางทั้งหมดที่สามารถส่งออกสายอักขระ x

$$\Pr(\pi_i = k|x) = \frac{\Pr(\pi_i = k, x)}{\Pr(x)} = \frac{\sum_{all\ paths\ \pi\ with\ \pi_i=k} \Pr(x, \pi)}{\sum_{all\ paths\ \pi} \Pr(x, \pi)}$$

หยุดคิด	ถ้าอัลกอริทึมวิเทอบิในปัญหาคาสิโนใช้เส้นทาง $\pi = \pi_1\pi_2 \ldots \pi_n$ และ π_i = B คำถามคือมีโอกาสที่เจ้ามือจะใช้เหรียญถ่วงน้ำหนักมากกว่าเหรียญปกติในการโยนเหรียญรอบที่ i หรือไม่ และมีความเป็นไปได้ไหมที่ π_i = B แต่ค่าความน่าจะเป็นแบบมีเงื่อนไข $\Pr(\pi_i = B|x)$ มีค่าน้อยกว่า $\Pr(\pi_i = F|x)$

อัลกอริทึมฟอร์เวิร์ด-แบคเวิร์ด

จากหัวข้อที่แล้วเรากำหนด $\Pr(\pi_i = k, x)$ มีค่าเท่ากับผลรวมของผลคูณค่าน้ำหนักคะแนน $\Pr(\pi, x)$ ของทุกเส้นทาง π ในกราฟวิเทอบิ ที่ผ่านโหนด (k,i) และแสดงออกสายอักขระ x ดังแสดงในรูปที่ 6.21(ก) เราสามารถแบ่งแต่ละเส้นทางออกเป็นเส้นทางย่อยสีฟ้าเริ่มจากโหนดต้นทาง (source) ไปยังโหนด (k,i) ซึ่งแสดงโดยสัญลักษณ์ π_{blue} และเส้นทางย่อยสีชมพูจากโหนด (k,i) ไปยังโหนดปลายทาง (sink) แสดงโดยสัญลักษณ์ π_{pink} โดยที่ $WEIGHT(\pi_{blue})$ และ $WEIGHT(\pi_{pink})$ เป็นผลคูณค่าน้ำหนักคะแนนของเส้นทางย่อยดังแสดงในสมการต่อไปนี้

$$\begin{aligned}\Pr(\pi_i = k, x) &= \sum_{all\ paths\ \pi\ with\ \pi_i=k} \Pr(x, \pi) \\ &= \sum_{all\ paths\ \pi_{blue}} \sum_{all\ paths\ \pi_{pink}} WEIGHT(\pi_{blue}) \cdot WEIGHT(\pi_{pink}) \\ &= \sum_{all\ paths\ \pi_{blue}} WEIGHT(\pi_{blue}) \cdot \sum_{all\ paths\ \pi_{pink}} WEIGHT(\pi_{pink})\end{aligned}$$

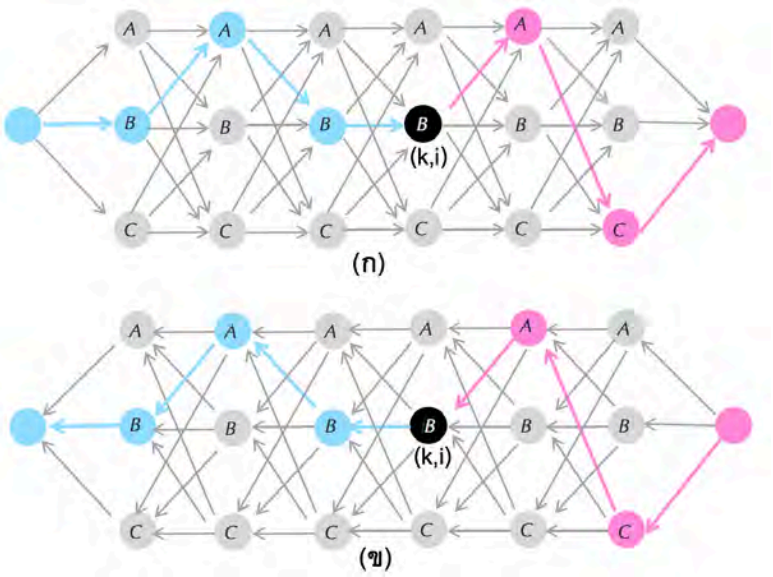
ผลบวกของผลคูณค่าน้ำหนักคะแนนเส้นทางย่อยสีฟ้าทั้งหมดหรือ $forward_{k,i}$ เป็นค่าที่มีการกล่าวถึงมาก่อนหน้าในการแก้ปัญหา Outcome Likelihood สำหรับหัวข้อนี้นอกจาก $forward_{k,i}$ แล้ว ยังต้องคำนวณ

ผลบวกของผลคูณค่าน้ำหนักคะแนนเส้นทางย่อยสีชมพูทั้งหมดหรือ $backward_{k,i}$ ดังนั้นสมการข้างต้นสามารถเขียนใหม่ได้เป็น

$$\Pr(\pi_i = k, x) \quad = forward_{k,i} \cdot backward_{k,i}$$

โดย $backward_{k,i}$ ได้จากการคำนวณค่าโดยกลับทิศทางกราฟวิเทอบิ (รูปที่ 6.21(ข)) และประยุกต์ใช้อัลกอริทึมกำหนดการพลวัตเช่นเดียวกับการหาค่า $forward_{k,i}$ เนื่องจากการกลับทิศทางเส้นเชื่อมจากโหนด (l, i+1) มายัง (k,i) มีค่าน้ำหนักคะแนนเป็น $WEIGHT_i(k,l) = transition_{k,l} \cdot emission_l(x_{i+1})$ และเขียนสมการได้ดังนี้

$$backward_{k,i} = \sum_{all\ states\ l} backward_{l,i+1} \cdot WEIGHT_i(k,l)$$



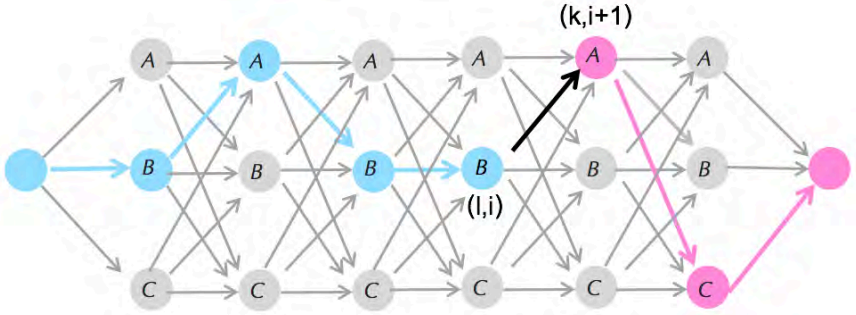
รูปที่ 6.21 (ก) เส้นทางจากโหนดต้นทาง (source) ไปยังโหนดปลายทาง (sink) โดยผ่านโหนดสีดำ (k,i) ในกราฟวิเทอบิ โดยแบ่งออกเป็นเส้นทางย่อยสีฟ้าจากโหนดต้นทางมายังโหนด (k,i) และเส้นทางย่อยสีชมพูจากโหนด (k,i) ไปยังโหนดปลายทาง (ข) กราฟวิเทอบิกลับด้าน (reversed Viterbi graph) โดยเส้นเชื่อมทุกเส้นถูกกลับทิศทางและมีเส้นทางจากโหนดปลายทางมายังโหนด (k,i)

การใช้กำหนดการพลวัตในการคำนวณค่าความน่าจะเป็น $\Pr(\pi_i = k, x)$ มีชื่อจำเพาะว่าอัลกอริทึมฟอร์เวิร์ด- แบคเวิร์ด (forward-backward algorithm) ถ้ารวมอัลกอริทึมฟอร์เวิร์ด-แบคเวิร์ดเข้ากับคำตอบของปัญหา Outcome Likelihood ที่ใช้ในการคำนวณ $\Pr(x)$ จะได้สมการที่ใช้ในการหาคำตอบของปัญหา Soft Decoding ดังต่อไปนี้

$$\Pr(\pi_i = k|x) = \frac{\Pr(\pi_i = k, x)}{\Pr(x)} = \frac{forward_{k,i} \cdot backward_{k,i}}{forward(sink)}$$

ฝึกหัด	พิจารณาปัญหาต่อไปนี้ • จาก HMM ในปัญหาคาสิโน จงคำนวณ $\Pr(\pi_i = k, x)$ โดย $x =$ "THTHHHTHTTH" สำหรับแต่ละ i และคำตอบต่างไปอย่างไรถ้าเปลี่ยนเป็น $x =$ "HHHHHHHHHH" • ประยุกต์ใช้วิธีการแก้ปัญหา Soft Decoding ในการหา CG-island ใน 1 ล้านนิวคลีโอไทด์แรกของโครโมโซม X ในมนุษย์ คำตอบนี้แตกต่างจากคำตอบที่ได้จากอัลกอริทึมวิเทอบิอย่างไร

โจทย์ข้างต้นเป็นการคำนวณค่าความน่าจะเป็นแบบมีเงื่อนไข $\Pr(\pi_i = k|x)$ ที่ HMM จะผ่านโหนด (k,i) ในกราฟวิเทอบิโดยมีเงื่อนไขว่า HMM ส่งออกสายอักขระ x คำถามถัดไปคือ ค่าความน่าจะเป็นแบบมีเงื่อนไข $\Pr(\pi_i = l, \pi_{i+1} = k|x)$ เป็นเท่าใด ถ้า HMM ผ่านเส้นเชื่อมระหว่างโหนด (l,i) ไปยังโหนด (k,i+1) โดยมีเงื่อนไขว่า HMM ส่งออกสายอักขระ x หากประยุกต์ใช้อัลกอริทึมฟอร์เวิร์ด-แบคเวิร์ด ในกรณีนี้ เราสามารถแบ่งเส้นทางเป็นเส้นทางย่อยสีฟ้าที่รวมทุกเส้นทางที่ผ่านเส้นเชื่อม (l,i) -> (k, i+1) (เส้นหนาสีดำ) จากโหนดต้นทางมายังเส้นเชื่อม และเส้นทางย่อยสีชมพูจากเส้นเชื่อมนี้ไปยังโหนดปลายทาง (รูปที่ 6.22)



รูปที่ 6.22 เส้นทางในกราฟวิเทอบิจากโหนดต้นทางไปยังโหนดปลายทางโดยผ่านเส้นเชื่อม (l,i) -> (k, i+1)

ฝึกหัด	จงพิสูจน์ว่า $\Pr(\pi_i = l, \pi_{i+1} = k|x)$ เท่ากับ $forward_{l,i} \cdot WEIGHT_i(l,k) \cdot backward_{k,i+1} / forward_{sink}$

ค่าความน่าจะเป็น $\Pr(\pi_i = k|x)$ สามารถเก็บในรูปแบบของเมทริกซ์ขนาด |States| x n เรียกว่าเมทริกซ์ responsibility Π^* โดยที่ $\Pi^*_{k,i}$ แสดงโหนดในกราฟวิเทอบิที่มีค่าความน่าจะเป็นเท่ากับ $\Pr(\pi_i = k|x)$ รูปที่ 6.23(ก) แสดงเมทริกซ์ responsibility Π^* สำหรับปัญหาคาสิโน สำหรับค่าความน่าจะเป็น $\Pr(\pi_i = l, \pi_{i+1} = k|x)$ สามารถเก็บในรูปแบบของเมทริกซ์ขนาด |States| x |States| x (n-1) เรียกว่าเมทริกซ์ responsibility Π^{**} โดย $\Pi^{**}_{l,k,i}$ แสดงเส้นเชื่อมในกราฟวิเทอบิและมีค่าความน่าจะเป็นเท่ากับ $\Pr(\pi_i = l, \pi_{i+1} = k|x)$ ดังแสดงในรูปที่ 6.23(ข) เพื่อความกระชับเราสามารถใช้ Π ในการอ้างอิงถึงทั้งเมทริกซ์ Π^* และ Π^{**}

(ก)

		T	H	T	H	H	H	T	H	T	T	H
	F	0.636	0.593	0.600	0.533	0.515	0.544	0.627	0.633	0.692	0.686	0.609
	B	0.364	0.407	0.400	0.467	0.485	0.456	0.373	0.367	0.308	0.314	0.391

(ข)

	1	2	3	4	5	6	7	8	9	10
FF	0.562	0.548	0.507	0.473	0.478	0.523	0.582	0.608	0.643	0.588
FB	0.074	0.045	0.093	0.059	0.037	0.022	0.045	0.025	0.049	0.098
BF	0.031	0.053	0.025	0.042	0.066	0.104	0.051	0.084	0.043	0.022
BB	0.333	0.354	0.374	0.426	0.418	0.351	0.322	0.282	0.265	0.293

รูปที่ 6.23 เมทริกซ์ responsibility จากปัญหาคาสิโน (ก) Π^* เก็บค่าความน่าจะเป็น $Pr(\pi_i = k|x)$ และ (ข) Π^{**} เก็บค่าความน่าจะเป็น $Pr(\pi_i = l, \pi_{i+1} = k|x)$

(ที่มา: รูปที่ 10.26 ของ [21])

การเรียนรู้บอม-เวลช์

อัลกอริทึม Expectation Maximization ที่ใช้ในการประมาณค่าพารามิเตอร์เรียกว่าการเรียนรู้บอม-เวลช์ (Baum-Welch learning) มีการทำงานระหว่าง 2 ขั้นตอนสลับกัน โดยขั้นตอน E (E-step) ทำการประมาณค่าเมทริกซ์ responsibility Π โดยใช้ค่าพารามิเตอร์ปัจจุบันตามสมการต่อไปนี้

$$(x, ?, Parameters) \rightarrow \Pi$$

และในขั้นตอน M (M-step) ทำการประมาณค่าพารามิเตอร์ใหม่โดยใช้เมทริกซ์ responsibility Π ที่เป็นผลลัพธ์จากขั้นตอน E ตามสมการต่อไปนี้

$$(x, \Pi, ?) \rightarrow Parameters$$

ทั้งนี้เราทราบวิธีการประมาณค่า π ในขั้นตอน E ของอัลกอริทึม Expectation Maximization แต่ยังไม่ทราบวิธีการประมาณค่าพารามิเตอร์ในขั้นตอน M การทราบเส้นทางลำดับสถานะซ่อนเร้น π ช่วยให้สามารถประมาณค่าพารามิเตอร์ที่ดีที่สุดสำหรับเส้นทาง π หนึ่งๆ ได้ดังสมการต่อไปนี้

$$transition_{l,k} = \frac{T_{l,k}}{\sum_{all\ states\ j} T_{l,j}} \qquad emission_k(b) = \frac{E_k(b)}{\sum_{all\ symbols\ c\ in\ the\ alphabet} E_k(c)}$$

โดยที่ $T_{l,k}$ คือจำนวนครั้งของการเปลี่ยนจากสถานะซ่อนเร้น l ไปยังสถานะซ่อนเร้น k ในเส้นทาง π และ $E_k(b)$ คือจำนวนครั้งที่มีการส่งออกอักขระ b เมื่อใช้เส้นทาง π และอยู่ในสถานะซ่อนเร้น k

ฝึกหัด	ถ้าไม่ทราบเส้นทาง π จะสามารถปรับสมการข้างต้นให้ประมาณค่าพารามิเตอร์ได้อย่างไร

พิจารณาการคำนวณค่า $T_{l,k}$ และ $E_k(b)$ โดยทราบเส้นทาง π แต่เปลี่ยนวิธีการคำนวณเล็กน้อย ดังต่อไปนี้

$$T_{l,k}^{i} = \begin{cases} 1 \; if \; \pi_i = l \; and \; \pi_{i+1} = k \\ 0 \; otherwise \end{cases} \qquad E_k^i(b) = \begin{cases} 1 \; if \; \pi_i = k \; and \; x_i = b \\ 0 \; otherwise \end{cases}$$

จากสมการข้างต้นเราสามารถคำนวณค่า $T_{l,k}$ และ $E_k(b)$ และสามารถเขียนใหม่ได้เป็น

$$T_{l,k} = \sum_{i=1}^{n-1} T_{l,k}^{i} \qquad E_k(b) = \sum_{i=1}^{n} E_k^i(b)$$

ซึ่งในกรณีที่ไม่ทราบเส้นทาง π สามารถจะแทนค่าตัวแปร $T_{l,k}$ และ $E_k(b)$ โดยใช้ตัวแปร $T_{l,k}^{i}$ และ $E_k^i(b)$ ตามลำดับ ซึ่งคำนวณได้จากค่าความน่าจะเป็นแบบมีเงื่อนไขของเส้นทางแสดงสถานะซ่อนเร้น π ผ่านโหนดหรือเส้นเชื่อมที่กำหนดในกราฟวิเทอบิ

$$T_{l,k}^{i} = \Pr(\pi_i = l, \pi_{i+1} = k|x) \qquad E_k^i(b) = \Pr(\pi_i = k|x)$$
$$= \Pi_{l,k,i}^{**} \qquad = \Pi_{k,i}^{*} \; if \; x_i = b \; and \; 0 \; otherwise$$

ซึ่งค่าความน่าจะเป็นเหล่านี้มีการอธิบายในหัวข้อที่ผ่านมาทำให้สามารถประมาณค่าพารามิเตอร์ใหม่โดยใช้สมการต่อไปนี้ (หมายเหตุ: การประมาณค่าโดยวิธีการนี้ส่วนใหญ่ให้ผลดีกว่าการประมาณค่าพารามิเตอร์โดยการเรียนรู้วิเทอบิ)

$$transition_{l,k} = \sum_{i=1}^{n-1} \Pi_{l,k,i}^{**} \qquad emission_k(b) = \sum_{i=1}^{n} \Pi_{k,i}^{*}$$

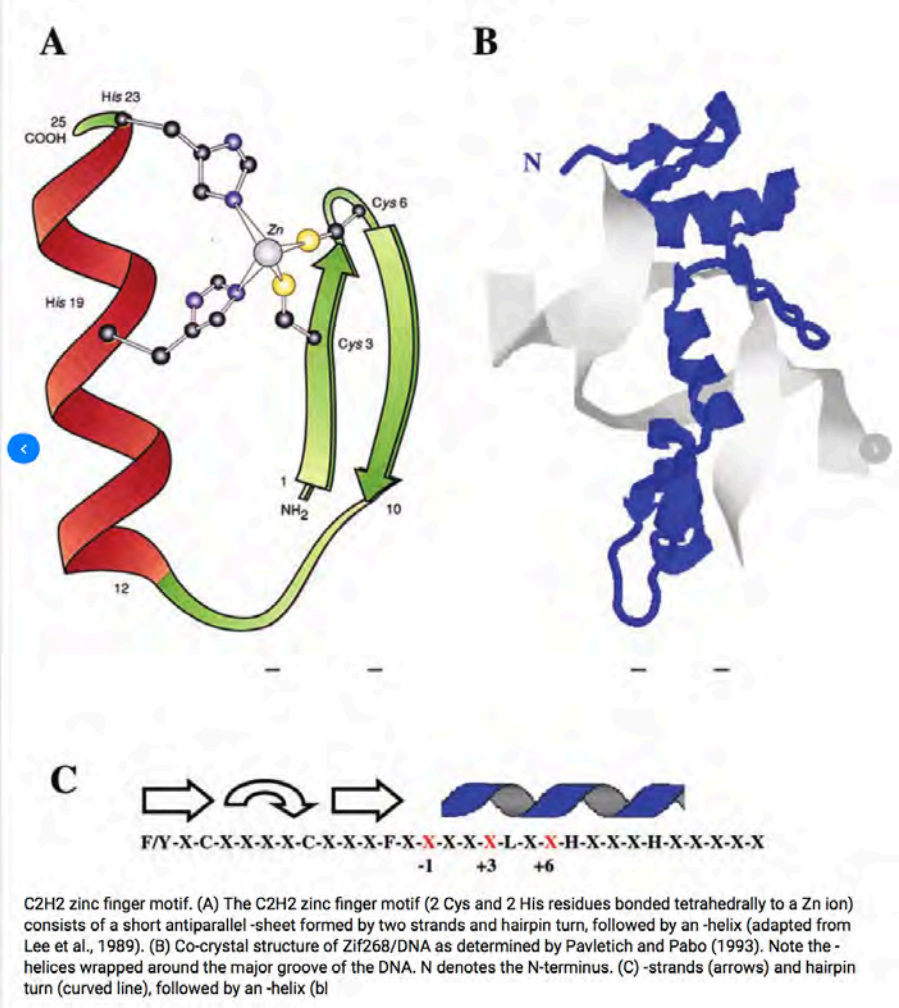
หยุดคิด	เราควรทำการนอร์มัลไลซ์ค่าความน่าจะเป็น transition และ emission ในสมการข้างต้นหรือไม่ หรืออีกนัยหนึ่งคือในสมการข้างต้นอนุมานได้ว่าผลรวมของค่าความน่าจะเป็นของการเปลี่ยนจากสถานะ (transition) ต้องเป็น 1 หรือไม่

ฝึกหัด	ใช้บอม-เวลช์ในการเรียนรู้ค่าพารามิเตอร์สำหรับ HMM ของ CG-island และโพรไฟล์ HMM ของเชื้อเอชไอวี เปรียบเทียบค่าของพารามิเตอร์เหล่านี้กับค่าพารามิเตอร์ที่ได้จากการเรียนรู้วิเทอบิ

บทส่งท้าย

ธรรมชาติในฐานะนักประกอบ

ส่วนของลำดับกรดแอมิโนในสายโปรตีนสะท้อนโครงสร้างสามมิติและฟังก์ชันการทำงานของโปรตีนนั้น ตัวอย่างเช่น โดเมนซิงค์ฟิงเกอร์ (zinc finger domain) หรือโมทิฟซิงค์ฟิงเกอร์ (รูปที่ 6.24) เป็นส่วนประกอบในโครงสร้างสามมิติของโปรตีนซิงค์ฟิงเกอร์ กรดแอมิโนซิสเทอีน (cysteine) และฮิสทิดีน (histidine) อย่างละ 2 ตัวที่อยู่ใกล้กันในโดเมนซิงค์ฟิงเกอร์ ทำให้โปรตีนสามารถจับได้กับซิงค์ไอออน (zinc ion) และสามารถม้วนรอบตัวไอออนได้แน่นหนา โดเมนซิงค์ฟิงเกอร์เป็นส่วนประกอบของสายโปรตีนในมนุษย์หลายพันโปรตีน นอกจากนี้โปรตีนซิงค์ฟิงเกอร์ยังสามารถจับกับไอออนอื่นทั้งที่เป็นโลหะ (metal) และไม่ใช่โลหะ (non-metal) ได้ด้วย



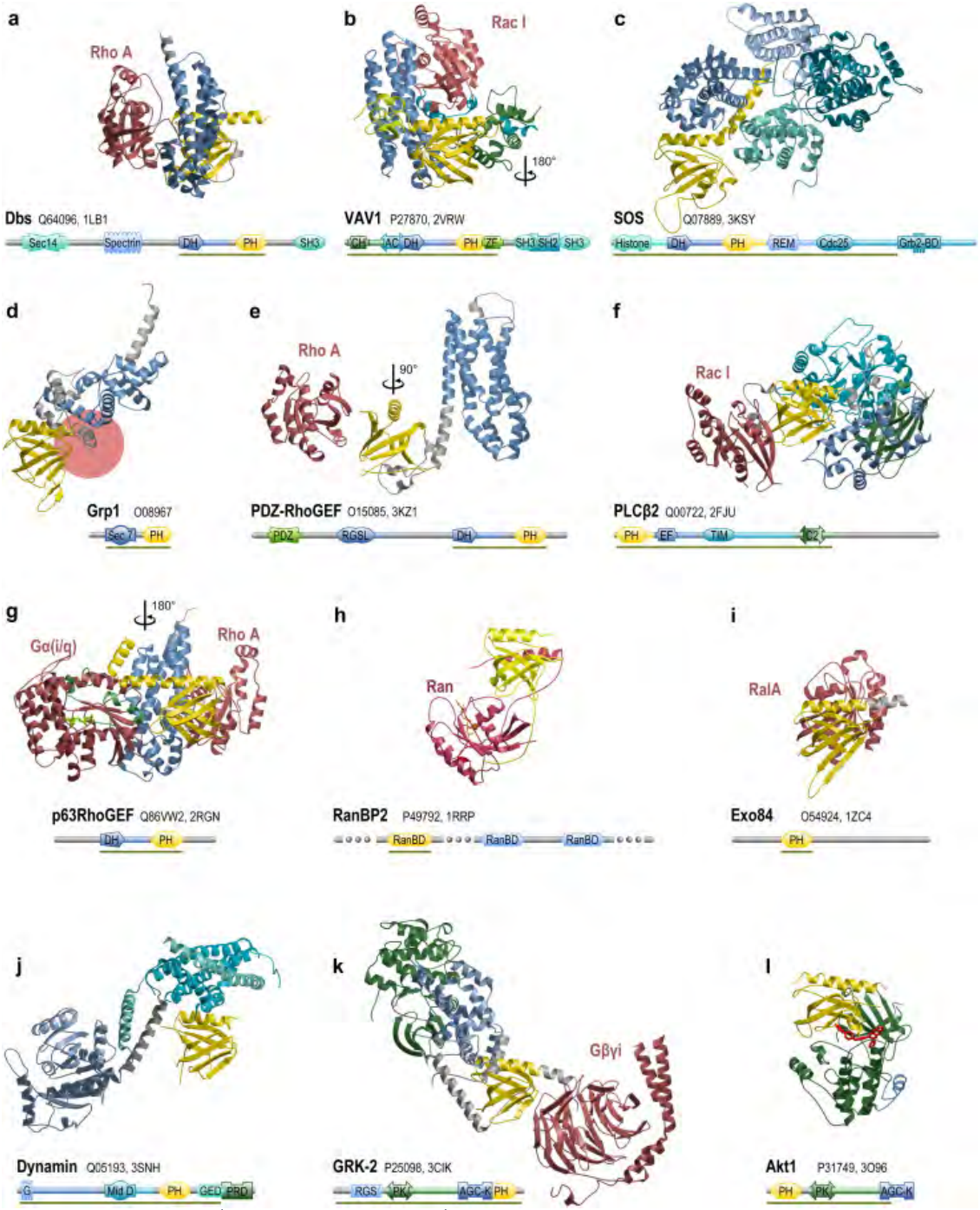
C2H2 zinc finger motif. (A) The C2H2 zinc finger motif (2 Cys and 2 His residues bonded tetrahedrally to a Zn ion) consists of a short antiparallel -sheet formed by two strands and hairpin turn, followed by an -helix (adapted from Lee et al., 1989). (B) Co-crystal structure of Zif268/DNA as determined by Pavletich and Pabo (1993). Note the - helices wrapped around the major groove of the DNA. N denotes the N-terminus. (C) -strands (arrows) and hairpin turn (curved line), followed by an -helix (bl

รูปที่ 6.24 โปรตีนโดเมน C2H2 zinc finger

(ที่มา: รูปที่ 3 ของ [151])

จากการทดลองในห้องปฏิบัติการมากกว่า 100,000 การทดลองเพื่อศึกษาโครงสร้างของโปรตีน พบว่าโปรตีนจำนวนมากมีโครงสร้างหรือส่วนของโครงสร้างที่มีความคล้ายคลึงกันมาก โดยโปรตีนโดเมน (protein domain) เป็นส่วนของสายโปรตีนที่มีความอนุรักษ์ร่วมกันระหว่างโปรตีน มีฟังก์ชันการทำงานจำเพาะ และเป็นอิสระจากส่วนอื่นๆ ของสายโปรตีน ความยาวของโปรตีนโดเมนมีความหลากหลายแต่ความยาวโดยเฉลี่ยอยู่ที่ 100 กรดแอมิโน (โดเมนซิงค์ฟิงเกอร์มีความยาวโดยเฉลี่ยประมาณ 20-30 กรดแอมิโน) โปรตีนจำนวนมากประกอบด้วยหลายโดเมนและแต่ละโปรตีนโดเมนมักปรากฏอยู่ในหลายโปรตีน

ฟร็องซัว เจคอบ (François Jacob) ผู้ได้รับรางวัลโนเบลสาขาการแพทย์ ได้กล่าวไว้ในปี ค.ศ. 1977 ว่า “ธรรมชาติเป็นเสมือนนักประกอบแต่ไม่ใช่นักประดิษฐ์” ธรรมชาติของโปรตีนมีโปรตีนโดเมนเป็นหน่วยโครงสร้างพื้นฐาน (building block) โดยแต่ละโปรตีนมักเกิดจากการนำโปรตีนโดเมนต่างๆ มาประกอบเข้าด้วยกันในลำดับที่แตกต่างกันไป ดังตัวอย่างโปรตีนที่มีโปรตีนโดเมน PH เป็นส่วนประกอบในรูปที่ 6.25



รูปที่ 6.25 ตัวอย่างโปรตีนที่มีโปรตีนโดเมน PH เป็นส่วนประกอบ

(ที่มา: รูปที่ 2 ของ [152])

ในการศึกษาที่ผ่านมาพบว่าโปรตีนโดเมนหลายชนิดปรากฏอยู่ในโปรตีนที่ประกอบด้วยหลายโดเมนในมนุษย์ แต่โดเมนเหล่านี้กลับปรากฏเป็นโดเมนเดี่ยวในหลายโปรตีนของแบคทีเรีย ในธรรมชาติการเกิดโปรตีนที่ประกอบด้วยหลายโดเมนสามารถเกิดจากเรียงลำดับเบสใหม่ของจีโนม (genome rearrangement) ซึ่งอาจทำให้

เกิดยีนเดิมมากกว่า 1 สำเนามาต่อกัน การประกอบสองโดเมนเป็น 1 โปรตีนมักแสดงถึงขบวนการวิวัฒนาการที่เป็นประโยชน์ เช่นการประกอบสองโดเมนที่เป็นเอนไซม์อาจช่วยให้เซลล์สามารถสนับสนุนการทำงานระหว่างสองเอนไซม์ได้ดีขึ้น เป็นต้น

เนื่องจากโปรตีนมักถูกสร้างขึ้นจากการประกอบโปรตีนโดเมนที่มีโครงสร้างและฟังก์ชันการทำงานที่แตกต่างกันเข้าด้วยกัน นักชีววิทยามักวิเคราะห์โปรตีนโดเมนเดี่ยวแทนการศึกษาโปรตีนทั้งสายในการศึกษาความสัมพันธ์ในเชิงวิวัฒนาการ ฐานข้อมูลพีแฟม (Pfam Database: http://pfam.xfam.org) มี HMM มากกว่า 10,000 แบบจำลอง โดยสร้างจากการเปรียบเทียบความคล้ายคลึงกันระหว่างสายโปรตีนในกลุ่ม (multiple sequence alignment) ซึ่งสามารถใช้ในการวิเคราะห์สายโปรตีนใหม่ว่าประกอบด้วยโปรตีนโดเมนใดบ้าง ในโครงการชีวสารสนเทศเพื่อการศึกษาและวิเคราะห์ระบบจังหวะหนึ่งวันนาฬิกาและความสัมพันธ์ของระบบต่อการออกดอกในพืช ผู้เขียนและคณะได้ออกแบบและพัฒนา d-Omix [153] ในรูปแบบตัวบริการเว็บ (web server) เพื่อให้ผู้ใช้สามารถวิเคราะห์และเปรียบเทียบความสัมพันธ์ระหว่างโปรตีนภายในชุดหรือระหว่างชุด จากองค์ประกอบของโปรตีนโดเมน เช่น ชุดของโปรตีนที่เกี่ยวข้องกับการออกดอกของ *Arabidopsis thaliana* ซึ่งเป็นพืชใบเลี้ยงคู่ หรือของข้าว (*Oryza sativa* และ *Oryza japonica*) ซึ่งเป็นพืชใบเลี้ยงเดี่ยว โดยผู้ใช้สามารถ (1) เปรียบเทียบความคล้ายคลึงกันระหว่างโปรตีนจากลำดับโปรตีนโดเมนที่ปรากฏในแต่ละโปรตีน (2) สร้างต้นไม้แสดงความสัมพันธ์ระหว่างโปรตีนภายในชุดโดยใช้เมทริกซ์ระยะทางที่คำนวณจากผลการเปรียบเทียบความคล้ายคลึงกันของลำดับโปรตีนโดเมน (3) พิจารณาความถี่ในการปรากฏของแต่ละโปรตีนโดเมน ความสามารถในการประกอบร่างร่วมกับโปรตีนโดเมนอื่นๆ และแสดงผลผ่านโดเมนกราฟ และ (4) แสดงโอกาสในการเกิดปฏิสัมพันธ์ระหว่างโปรตีนโดยใช้ข้อมูลการเกิดปฏิสัมพันธ์ระหว่างโปรตีนโดเมนที่นำเข้าเป็นส่วนหนึ่งของระบบจาก DOMINE [154] โดยข้อมูลเข้าจากผู้ใช้เป็นไฟล์ผลลัพธ์จาก InterProScan [155, 156] ที่สามารถปรับแต่งหรือคัดเลือกข้อมูลเข้าเฉพาะส่วนเช่น เลือกเฉพาะโปรตีนโดเมนจาก hmmpfam และ superfamily เป็นต้น

ฝึกหัด	สำรวจข้อมูลโปรตีนโดเมน Piwi (PF02171) ในฐานข้อมูลพีแฟม (Pfam) และดาวน์โหลดชุดสายโปรตีนตั้งต้น (seed sequences) มาสร้าง HMM ของกลุ่มโปรตีนโดเมน Piwi โดยใช้วิธีการที่ศึกษาในบทเรียนนี้

การประยุกต์ใช้ HMMs ในโจทย์ทางชีวสารสนเทศอื่นๆ

โพรไฟล์ HMM สำหรับการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนหลายเส้น หรือ HMM สำหรับการหา CG-island เป็นเพียงตัวอย่างเบื้องต้นในการประยุกต์ใช้ HMM เพื่อแก้โจทย์ทางชีวสารสนเทศ ตัวอย่างอื่นในการประยุกต์ใช้ HMM เช่น การทำนายบริเวณที่เป็นยีนในจีโนม [157] การทำนายโครงสร้างสองมิติของโปรตีน [158-160] การทำนายตำแหน่งโอเมก้า (ω) ในการศึกษา anchored proteins [161] การหาตำแหน่งที่เป็นโมทิฟ [162, 163] และหาโมทิฟในสายอาร์เอ็นเอที่เป็นตำแหน่งจับของโปรตีน [163, 164] การ

ประยุกต์ใช้ HMM อนันต์ (infinite HMM) เพื่อวิเคราะห์ข้อมูลโมเลกุลเดี่ยว (single molecule) [165] การประยุกต์ใช้ sparsely correlated HMM ในการเชื่อมโยงและศึกษาความสัมพันธ์ระหว่างบริเวณต่างๆ ในจีโนม [166] การประยุกต์ใช้ HMM ในการอนุมานการเกิดการแปรผันของนิวคลีโอไทด์เดี่ยว (single nucleotide variant) ในจีโนม [167] การหาบริเวณที่เกิดการแปรผันจำนวนซ้ำ (copy number variation; CNV) ในจีโนม [168, 169] การประยุกต์ใช้ HMM ในการประเมินความน่าเชื่อถือของผลการประกอบร่างจีโนม [170] การประยุกต์ใช้ HMM หลายตัวแปร (multivariate HMM) ในการระบุสถานะของโครมาทิน (chromatin state) ในบริเวณต่างๆ ของจีโนม [171] การประยุกต์ใช้ HMM ในการทำนายโปรตีนทรานสเมมเบรน (transmembrane protein) ในกลุ่มที่เป็นเกลียวแอลฟา (alpha helix) [172] และกลุ่มที่เป็นเบต้าบาร์เรล (beta barrel) [173] การประยุกต์ใช้ ensemble HMMs ในการจำแนกกลุ่มโปรตีน [174] การประยุกต์ใช้ HMM ในการจำแนกการพับของโปรตีน [175] การประยุกต์ใช้ HMM ในการวิเคราะห์และตัดสินความเหมือนหรือคล้ายคลึงกันระหว่างโปรตีนโดเมนหรือฮอโมโลยีของโปรตีนโดเมน (protein domain homology) [176] การประยุกต์ใช้โพรไฟล์ HMM ของกลุ่มโปรตีนในการสร้างแผนภูมิวิวัฒนาการชาติพันธุ์ (phylogenetic tree) [177] การประยุกต์ใช้ HMM ในการทำนายบริเวณที่เป็น MoRF (molecular recognition features) ซึ่งอยู่ในสายของโปรตีนที่มีความผิดปกติในตัว (intrinsically disordered proteins: IDPs) [178] การประยุกต์ใช้ HMM ในการหาบริเวณที่มีฟังก์ชันของโปรตีนในการจับกับอาร์เอ็นเอ (RNA-binding proteins: RBPs) [179] ตัวบริการเว็บ HMMER [180-182] สำหรับสืบค้นโปรตีนในฐานข้อมูลที่มีความคล้ายคลึงกับสายโปรตีนเข้าโดยใช้โพรไฟล์ HMM นอกจากนี้ยังมีการประยุกต์ใช้ self-organizing HMM map ในการจัดกลุ่มและแสดงผลการจัดกลุ่มของสายข้อมูล [183] การประยุกต์ใช้ HMM ในการศึกษาศาสตร์จีโนมระดับประชากร (population genomics) [184] การประยุกต์ใช้ HMM ในการจำลองกระบวนการทางชีววิทยา [185] ตัวอย่างเพิ่มเติมการประยุกต์ใช้ HMM กับโจทย์ทางชีววิทยาสามารถศึกษาได้จากบทปริทัศน์ [186, 187] เป็นต้น

แบบฝึกหัดบทที่ 6

เขียนโปรแกรมเพื่อแก้ปัญหาที่เกี่ยวข้องกับ HMM โดยใช้โจทย์ที่โรซาลินด์ต่อไปนี้

1) Implementing the Viterbi Algorithm (http://rosalind.info/problems/ba10c/)
2) Solve the Soft Decoding Problem (http://rosalind.info/problems/ba10j/)
3) Implement Baum-Welch Learning (http://rosalind.info/problems/ba10k/)

บทที่ 7 การวิเคราะห์การแสดงออกของยีน (Gene expression analysis)

วัตถุประสงค์

- เพื่อให้นิสิตเห็นความสำคัญของการวัดการแสดงออกของยีน รวมทั้งความเกี่ยวข้องระหว่างการแสดงออกของยีนกับความเชื่อตามหลักชีววิทยาระดับโมเลกุล (central dogma of molecular biology)
- เพื่อให้นิสิตเห็นตัวอย่างของการประยุกต์ใช้การวัดการแสดงออกของยีนในการตอบโจทย์ทางชีววิทยา เช่น การศึกษากลุ่มของยีนที่มีผลต่อการเปลี่ยนเอทานอลเป็นน้ำตาลในยีสต์ เป็นต้น
- เพื่อให้นิสิตคุ้นเคยกับเทคโนโลยีที่เกี่ยวข้องกับการวัดการแสดงออกของยีนรวมทั้งลักษณะข้อมูลการแสดงออกของยีนที่มาจากเทคโนโลยีที่แตกต่างกัน
- เพื่อให้นิสิตคุ้นเคยกับแนวทางการวิเคราะห์การแสดงออกของยีน
- เพื่อให้นิสิตเห็นตัวอย่างงานวิจัย รวมทั้งตัวอย่างโปรแกรมที่ใช้ในการวิเคราะห์การแสดงออกยีน
- เพื่อให้นิสิตเห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทาย รวมทั้งงานวิจัยอื่นๆ ที่เกี่ยวข้อง

ผลลัพธ์ที่คาดหวัง

- นิสิตสามารถอธิบายความสำคัญของการวัดการแสดงออกของยีนรวมทั้งสามารถยกตัวอย่างการประยุกต์ใช้การวัดการแสดงออกของยีนเพื่อตอบโจทย์ทางชีววิทยา
- นิสิตสามารถยกตัวอย่างเทคโนโลยีที่ใช้ในการวัดการแสดงออกของยีน รวมทั้งสามารถอธิบายความแตกต่างระหว่างเทคโนโลยี
- นิสิตเข้าใจลักษณะข้อมูลการแสดงออกของยีนที่มาจากเทคโนโลยีที่แตกต่าง รวมทั้งสามารถอธิบายแนวทางในการวิเคราะห์การแสดงออกของยีนและอัลกอริทึมพื้นฐานที่เกี่ยวข้อง
- นิสิตสามารถอธิบายการทำงานของอัลกอริทึมพื้นฐานที่ใช้ในการจัดกลุ่มการแสดงออกของยีน
- นิสิตสามารถเขียนโปรแกรมที่ใช้ในการจัดกลุ่มการแสดงออกของยีนแบบง่ายได้
- นิสิตสามารถยกตัวอย่างโปรแกรมที่ใช้ในการวิเคราะห์การแสดงออกของยีนรวมทั้งการจัดกลุ่มยีนที่มีลักษณะการแสดงออกร่วมกันได้

- นิสิตสามารถยกตัวอย่างความท้าทายที่มีอยู่และสามารถนำเสนอแนวทางในการพัฒนาวิธีการแก้ปัญหาเหล่านี้ รวมทั้งสามารถประยุกต์องค์ความรู้จากบทเรียนเพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้

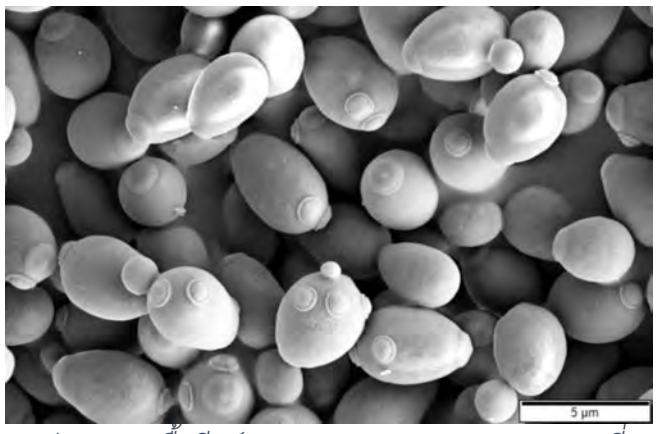
เนื้อหาโดยสรุป

การวัดการแสดงออกของยีน คือการวัดจำนวนอาร์เอ็นเอที่ถูกถอดรหัสจากดีเอ็นเอในขบวนการความเชื่อตามหลักชีววิทยาระดับโมเลกุล (central dogma of molecular biology) การวัดการแสดงออกของยีนมีความสำคัญเป็นอย่างมากในการศึกษาฟังก์ชันการทำงานของยีนในขบวนการต่างๆ ทางชีววิทยา ทั้งนี้สมมติฐานที่สำคัญอย่างหนึ่งคือยีนที่มีความเกี่ยวเนื่องกันมักมีการแสดงออกในรูปแบบเดียวกัน เทคโนโลยีที่สามารถวัดการแสดงออกของยีนจำนวนมากหรือของยีนทั้งจีโนม ประกอบด้วยเทคโนโลยีไมโครอาร์เรย์ (microarray) และการหาลำดับเบสของอาร์เอ็นเอทั้งหมด (RNA sequencing; transcriptome sequencing) หรืออาร์เอ็นเอซีค (RNA-seq) ซึ่งมีแนวทางในการวัดการแสดงออกและรูปแบบของข้อมูลที่ได้จากการทดลองมีความแตกต่างกัน เนื่องจากข้อมูลการแสดงออกของยีนที่ได้จากเทคโนโลยีทั้งสองกลุ่มมีความแตกต่างกันมาก การวิเคราะห์ข้อมูลเบื้องต้นจึงมีความจำเพาะตามเทคโนโลยีที่ใช้ในการผลิตข้อมูล อย่างไรก็ตามเป้าหมายหลักของการวัดการแสดงออกของยีน คือการจัดกลุ่มยีนที่มีการแสดงออกในรูปแบบเดียวกัน ซึ่งนำไปสู่การอนุมานหรือวิเคราะห์ฟังก์ชันการทำงานเพิ่มเติม เช่น การหาโมทิฟควบคุม (regulatory motif) ของยีนที่อยู่ในกลุ่มเดียวกัน (บทที่ 4) ชุดของยีนที่มีการตอบสนองต่อยาในรูปแบบเดียวกัน หรือชุดของยีนที่แสดงออกร่วมกันในผู้ป่วยมะเร็งชนิดหนึ่งๆ ในระดับความรุนแรงหรือในสถานะที่แตกต่างกัน เป็นต้น บทเรียนนี้เกี่ยวข้องกับอัลกอริทึมพื้นฐานที่ใช้ในการจัดกลุ่ม (clustering algorithm) เช่น อัลกอริทึม เค-เซ็นเตอร์ (K-Center) และ เค-มีนส์ (K-Means) แนวคิดเรื่อง soft clustering กระบวนการ Expectation Maximization (EM) และการจัดกลุ่มเชิงลำดับชั้น (hierarchical clustering) เป็นต้น รวมทั้งตัวอย่างโปรแกรมที่ใช้ในการวิเคราะห์การแสดงออกของยีน การประยุกต์ใช้องค์ความรู้เหล่านี้ในการแก้ปัญหาอื่นๆ และโจทย์วิจัยที่เกี่ยวข้อง

บทที่ 7 การวิเคราะห์การแสดงออกของยีน (Gene expression analysis)

การทำไวน์โดยใช้ยีสต์

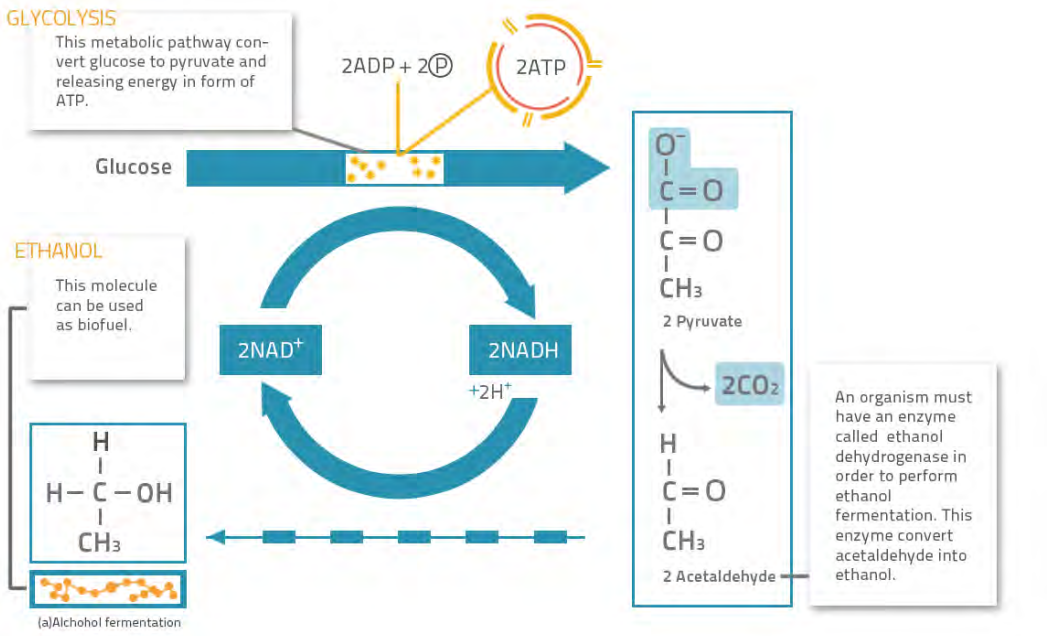
มีการนำยีสต์ (*Saccharomyces cerevisiae* หรือ *S. cerevisiae*) (รูปที่ 7.1) มาใช้ในการผลิตไวน์ เนื่องจากยีสต์สามารถเปลี่ยนกลูโคส (glucose) ที่อยู่ในผลไม้ให้เป็นเอทานอล (ethanol) ได้ ดังกระบวนการในรูปที่ 7.2 คำถามคือ ถ้ามียีสต์อยู่ที่ต้นองุ่นอยู่แล้ว ทำไมเวลาทำไวน์ต้องนำองุ่นมาบดและบรรจุในถังไม้ที่มีการปิดถังอย่างแน่นหนา คำตอบคือยีสต์ใช้กลูโคสเป็นอาหารและผลิตเอทานอลในกระบวนการหมัก (fermentation) เมื่อกลูโคสหมดยีสต์จะปรับกระบวนการภายในให้สามารถอยู่รอด โดยกลับด้านกระบวนการเมแทบอลิซึม (metabolism) เข้าสู่ diauxic shift ที่ใช้เอทานอลและออกซิเจน ถ้าไม่มีออกซิเจนยีสต์จะจำศีล (hibernate) และจะกลับมาทำงานอีกครั้งเมื่อมีออกซิเจนหรือกลูโคส ดังนั้นถ้าผู้ผลิตไวน์ไม่ได้ปิดฝาถังบ่มไวน์ให้แน่นหนา เมื่อยีสต์ใช้กลูโคสหมดแล้วยีสต์จะใช้เอทานอลผ่านกระบวนการ diauxic shift ทำให้สูญเสียเอทานอลที่ผลิตมาในกระบวนการหมัก



รูปที่ 7.1 ภาพถ่ายขยายเชื้อยีสต์ (Saccharomyces cerevisiae) ที่ 5 นาโนเมตร

(*ที่มา:* Murtey, M.D. and Ramasamy P., Public domain, via Wikimedia Commons. 2016. *Saccharomyces cerevisiae, SEM image*. [*เข้าถึงออนไลน์เมื่อวันที่* 11 *ก.ค. พ.ศ.* 2564])

กระบวนการ diauxic shift มีความซับซ้อนและมีผลกระทบต่อการแสดงออกของยีนจำนวนมาก รวมทั้งเกี่ยวข้องกับวิวัฒนาการในการเอาตัวรอดของยีสต์ เนื่องจากเอทานอลที่ผลิตได้เป็นพิษกับเชื้อแบคทีเรียและยีสต์อื่นๆ หลายชนิด นอกจากนี้ยีสต์ยังสามารถใช้เอทานอลเป็นแหล่งพลังงาน คำถามคือกระบวนการ diauxic shift พัฒนามาอย่างไรและมียีนอะไรบ้างที่เกี่ยวข้อง



รูปที่ 7.2 กระบวนการผลิตไวน์จากยีสต์โดยการเปลี่ยนกลูโคสในผลไม้เป็นเอทานอล

(ที่มา: Labster theory. *Ethanol Fermentation.* [ONLINE] Available at: https://theory.labster.com [เข้าถึงออนไลน์เมื่อวันที่ 11 ก.ค. พ.ศ.2564]*)*

การจัดกลุ่มของยีน

การวิเคราะห์การแสดงออกของยีน

ในปี ค.ศ. 1997 โจเซฟ เยริซิ (Joseph DeRisi) และคณะ [188] ออกแบบการทดลองให้สามารถวัดการแสดงออกของ 6,400 ยีนในยีสต์ *S. cerevisiae* ที่เพาะเลี้ยงไว้ใน 7 ช่วงเวลาประกอบด้วย -6, -4, -2, 0, +2, +4, และ +6 ชั่วโมง โดยชั่วโมงที่ 0 คือเวลาที่เกิด diauxic shift ผลการแสดงออกอยู่ในรูปแบบของเมทริกซ์ 6,400 x 7 ตามจำนวนยีนและช่วงเวลาที่วัดการแสดงออก

หยุดคิด	เทคโนโลยีใดที่สามารถใช้ในการวัดการแสดงออกของยีนในตัวอย่างการทดลองข้างต้น

ในปี ค.ศ. 1997 เยริซิและคณะใช้เทคโนโลยีไมโครอาร์เรย์ (microarray) (รูปที่ 7.3) ซึ่งปัจจุบันมีการใช้งานน้อยลงมาก เนื่องจากมีเทคโนโลยีการหาลำดับเบสของอาร์เอ็นเอทั้งหมด (RNA sequencing; transcriptome sequencing) หรือที่เรียกว่าอาร์เอ็นเอซีค (RNA-seq) โดยประยุกต์ใช้เทคโนโลยีเอ็นจีเอส (NGS) เข้ามาแทนที่ อย่างไรก็ตามระเบียบวิธีวิเคราะห์ข้อมูลในเชิงอัลกอริทึมที่เยริซิและคณะใช้ยังสามารถนำมาประยุกต์ใช้ในการจัดกลุ่มข้อมูลอาร์เอ็นเอซีคและข้อมูลอื่นๆ ได้

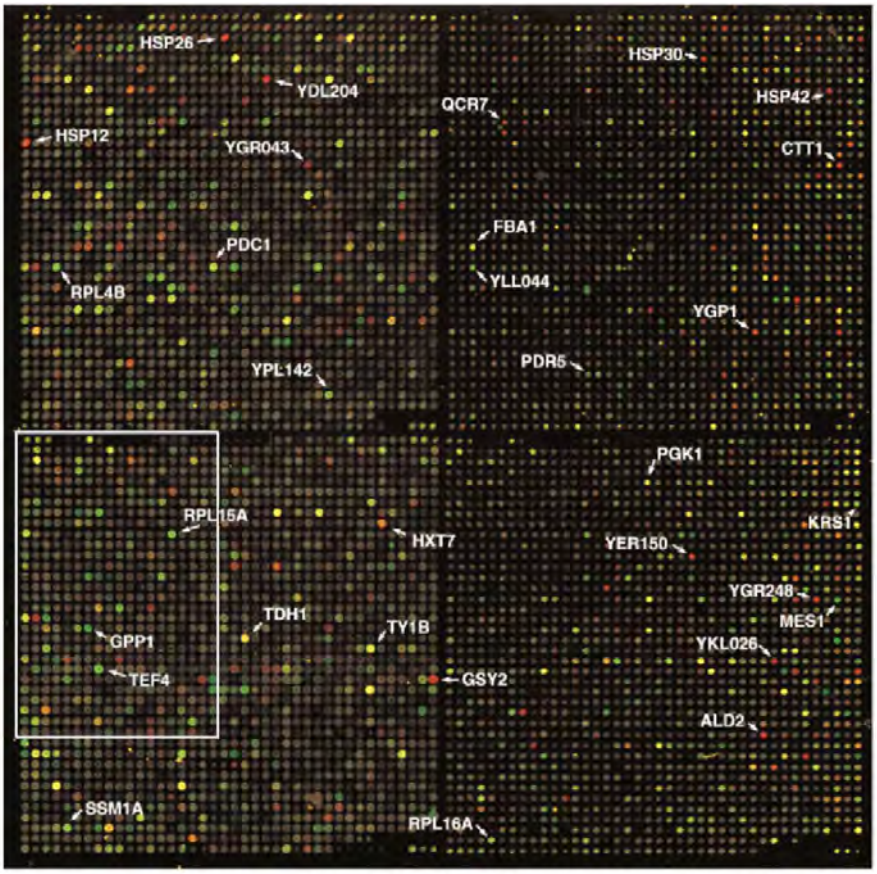


Fig. 1. Yeast genome microarray. The actual size of the microarray is 18 mm by 18 mm. The microarray was printed as described (*9*). This image was obtained with the same fluorescent scanning confocal microscope used to collect all the data we report (*49*). A fluorescently labeled cDNA probe was prepared from mRNA isolated from cells harvested shortly after inoculation (culture density of $<5 \times 10^6$ cells/ml and media glucose level of 19 g/liter) by reverse transcription in the presence of Cy3-dUTP. Similarly, a second probe was prepared from mRNA isolated from cells taken from the same culture 9.5 hours later (culture density of $\sim 2 \times 10^8$ cells/ml, with a glucose level of <0.2 g/liter) by reverse transcription in the presence of Cy5-dUTP. In this image, hybridization of the Cy3-dUTP–labeled cDNA (that is, mRNA expression at the initial timepoint) is represented as a green signal, and hybridization of Cy5-dUTP–labeled cDNA (that is, mRNA expression at 9.5 hours) is represented as a red signal. Thus, genes induced or repressed after the diauxic shift appear in this image as red and green spots, respectively. Genes expressed at roughly equal levels before and after the diauxic shift appear in this image as yellow spots.

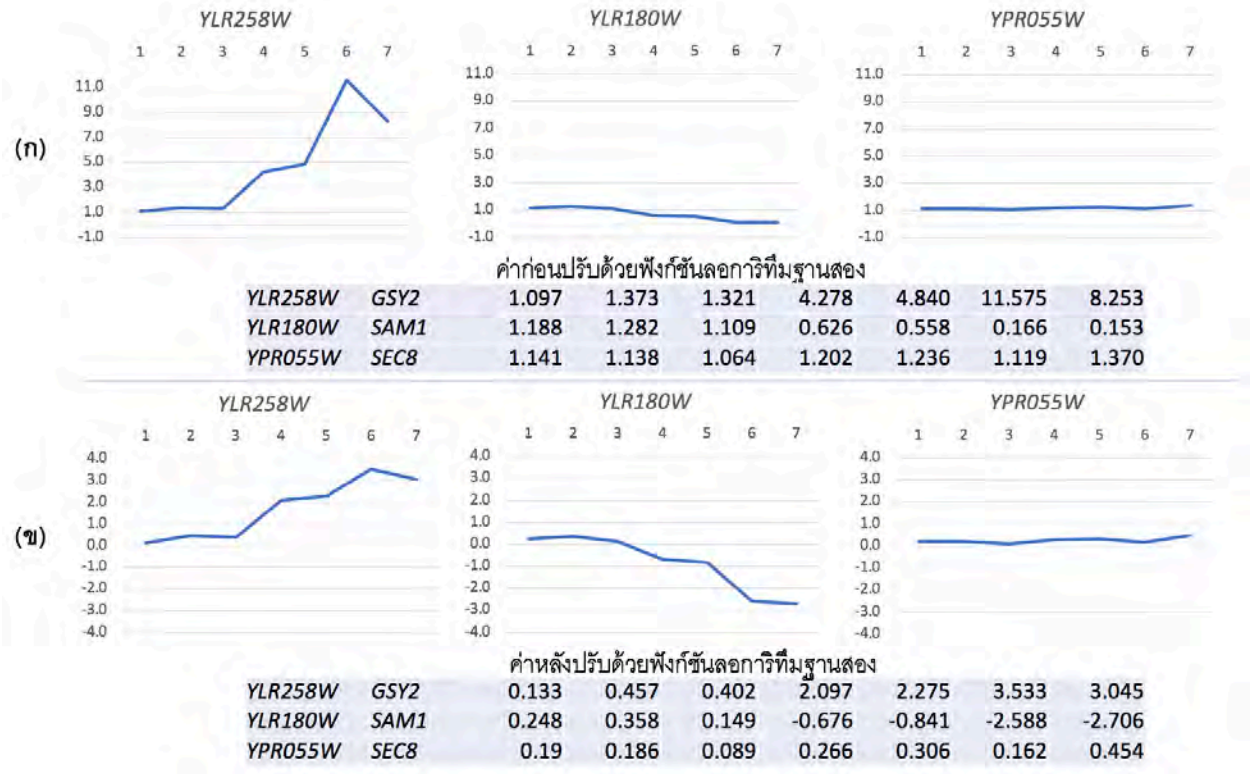
รูปที่ 7.3 ไมโครอาร์เรย์ของจีโนมยีสต์จากการทดลองของเยริซิและคณะ ค.ศ.1997

(*ที่มา*: *รูปที่* 1 *ของ* [188])

หยุดคิด	รูปที่ 7.4 แสดงรูปแบบการแสดงออกของยีนในยีสต์ *S. cerevisiae* จำนวน 3 ยีนในช่วงเวลาที่แตกต่างกัน คำถามคือใน 3 ยีนนี้ ยีนใดบ้างที่น่าจะเกี่ยวข้องกับ diauxic shift

ค่าการแสดงออกของยีน *YPR055W* ในรูปที่ 7.4 มีค่าคงที่ตลอด 7 ช่วงเวลา ดังนั้นเราสามารถสรุปได้ว่ายีนนี้ไม่เกี่ยวข้องกับ diauxic shift ในทางกลับกันการแสดงออกของยีน *YLR258W* มีการเปลี่ยนแปลงอย่างชัดเจนในช่วงเวลาที่ 5 และนำไปสู่สมมติฐานว่ายีน *YLR258W* น่าจะเกี่ยวข้องกับ diauxic shift ซึ่งถ้าไปตรวจสอบกับ

ฐานข้อมูล Saccharomyces Genome Database (SGD) (https://www.yeastgenome.org/) จะพบว่ายีน *YLR258W* คือ glycogen synthase ซึ่งเป็นเอนไซม์ที่ควบคุมการผลิตไกลโคเจน (glycogen) หรือกลูโคสพอลิแซกคาไรด์ (gluclose polysaccharide) ซึ่งเป็นแหล่งเก็บกลูโคสในเซลล์ของยีสต์



รูปที่ 7.4 ค่าการแสดงออกของยีน YLR258W, YLR180W, และ YPR055W (ก) ค่าเดิม (ข) ผลของการปรับค่าโดยใช้ฟังก์ชันลอการิทึมฐานสอง

ชื่อยีน		เวกเตอร์ค่าการแสดงออกของยีน						
YLR258W	GSY2	0.133	0.457	0.402	2.097	2.275	3.533	3.045
YKL026C	GPX1	-0.131	-0.015	-0.07	0.483	0.878	3.843	3.191
YFL014W	HSP12	-0.078	0.5	0.561	1.766	2.508	3.716	3.849
YMR290C	HAS1	0.191	-0.086	-0.33	-1.002	-1.319	-2.735	-2.712
YNL141W	AAH1	-0.11	0.097	-0.19	-1.057	-1.132	-2.774	-2.682
YLR180W	SAM1	0.248	0.358	0.149	-0.676	-0.841	-2.588	-2.706
YPR147C	YPR147C	0.149	0.152	0.094	0.324	0.311	0.236	0.373
YPR055W	SEC8	0.19	0.186	0.089	0.266	0.306	0.162	0.454
YGL133W	ITC1	0.117	0.282	0.128	0.284	0.196	0.154	0.444

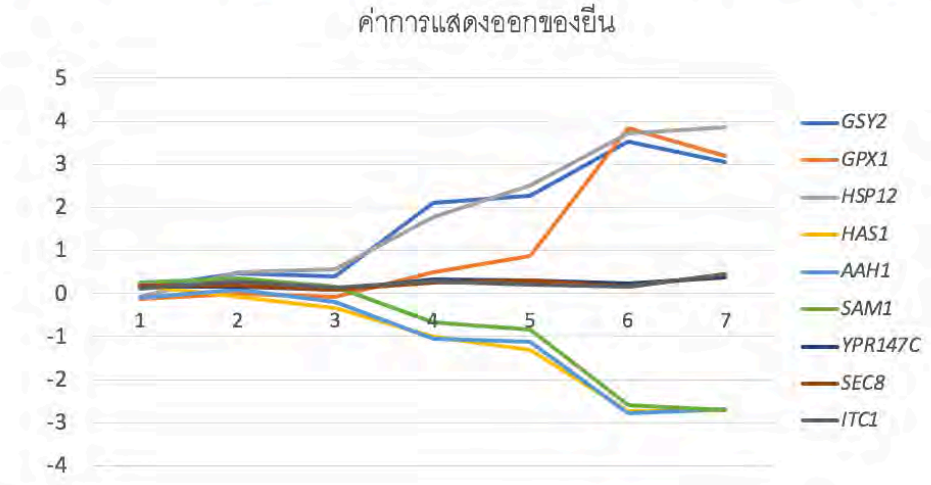
รูปที่ 7.5 เมทริกซ์ย่อยจากการทดลองของเยริซิและคณะที่ผ่านการปรับค่าโดยใช้ลอการิทึมฐาน 2 โดยบรรทัดที่ 1, 6, 8 แสดงค่าการแสดงออกของยีน YLR258W, YLR180W, YPR055W

ในทางปฏิบัตินักชีววิทยามักใช้ลอการิทึมในการปรับค่าการแสดงออกของยีน (รูปที่ 7.4 (ข)) โดยค่าที่เป็นบวกหลังปรับค่าแล้วหมายถึงการแสดงออกที่เพิ่มขึ้น ในขณะที่ค่าที่เป็นลบหมายถึงการแสดงออกที่ลดลง รูปที่ 7.5 แสดงเมทริกซ์ค่าการแสดงออกของยีนในยีสต์จำนวน 9 ยีนหลังปรับค่าแล้ว

การจัดกลุ่มยีน

จากข้อมูลข้างต้นที่ได้จากการทดลองของเยริชิและคณะ เป้าหมายคือการแบ่งกลุ่มยีนออกเป็น k กลุ่มตามรูปแบบการแสดงออก โดยแต่ละยีนต้องอยู่ในกลุ่มใดกลุ่มหนึ่งเท่านั้น ในทางปฏิบัติจะไม่ทราบจำนวนกลุ่มหรือค่า k ที่ควรจะเป็น ทั้งนี้นักชีววิทยามักลองจัดกลุ่มโดยใช้ค่า k ที่แตกต่าง และเลือกค่า k ที่ให้ผลการจัดกลุ่มที่มีความหมายในเชิงชีววิทยา ในเนื้อหาต่อไปนี้เพื่อลดความซับซ้อนของคำอธิบาย ค่า k จะถูกกำหนดไว้ล่วงหน้า รูปที่ 7.6 แสดงผลการจัดกลุ่มยีนในรูปที่ 7.5 ออกเป็น 3 กลุ่มประกอบด้วย กลุ่มที่มีการแสดงออกเพิ่ม ลด และไม่เปลี่ยนแปลงในช่วงก่อนและหลังเกิด diauxic shift

ชื่อยีน		เวกเตอร์ค่าการแสดงออกของยีน						
YLR258W	GSY2	0.133	0.457	0.402	2.097	2.275	3.533	3.045
YKL026C	GPX1	-0.131	-0.015	-0.07	0.483	0.878	3.843	3.191
YFL014W	HSP12	-0.078	0.5	0.561	1.766	2.508	3.716	3.849
YMR290C	HAS1	0.191	-0.086	-0.33	-1.002	-1.319	-2.735	-2.712
YNL141W	AAH1	-0.11	0.097	-0.19	-1.057	-1.132	-2.774	-2.682
YLR180W	SAM1	0.248	0.358	0.149	-0.676	-0.841	-2.588	-2.706
YPR147C	YPR147C	0.149	0.152	0.094	0.324	0.311	0.236	0.373
YPR055W	SEC8	0.19	0.186	0.089	0.266	0.306	0.162	0.454
YGL133W	ITC1	0.117	0.282	0.128	0.284	0.196	0.154	0.444



รูปที่ 7.6 ผลการแบ่งกลุ่มยีนในรูปที่ 7.5 ออกเป็น 3 กลุ่มตามรูปแบบการแสดงออกของยีนที่แตกต่างกัน

diauxic shift เป็นกระบวนการสำคัญในยีสต์ แต่ diauxic shift ไม่มีความเกี่ยวข้องกับฟังก์ชันหลักอื่นๆ ของยีสต์ซึ่งสอดคล้องกับผลการจัดกลุ่มยีนที่ในระหว่างที่เกิด diauxic shift ยีนส่วนใหญ่ไม่มีการเปลี่ยนแปลง

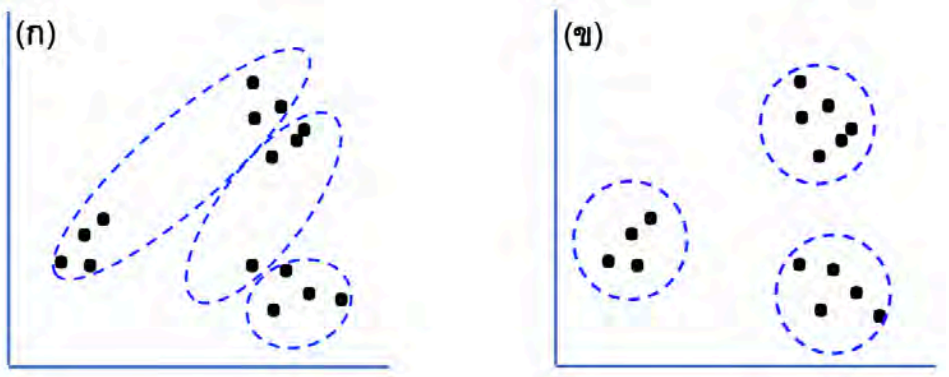
ระดับการแสดงออก และในการวิเคราะห์เพิ่มเติมยีนที่ไม่เกี่ยวข้องเหล่านี้จะถูกนำออกจากเมทริกซ์เพื่อเป็นการลดขนาดของเมทริกซ์

หลักเกณฑ์พื้นฐานในการจัดกลุ่มที่ดี

ในการจัดกลุ่มยีนที่มีรูปแบบการแสดงออกคล้ายคลึงกัน เวกเตอร์ของค่าระดับการแสดงออกของยีนหนึ่งๆ จำนวน m ค่า สามารถแทนด้วยจุด 1 จุดในปริภูมิ (space) ที่มีขนาด m มิติ และยีนที่มีค่าเวกเตอร์ใกล้เคียงกันควรอยู่ใกล้กันหรือเกาะกลุ่มกันภายในปริภูมิ m มิติ ในอุดมคติแต่ละคลัสเตอร์ (cluster) หรือกลุ่มของยีนควรมีลักษณะตรงตามเงื่อนไขต่อไปนี้ (และตามรูปที่ 7.7)

หลักเกณฑ์พื้นฐานในการจัดกลุ่มที่ดี : ทุกคู่ของจุดที่อยู่ในคลัสเตอร์เดียวกันควรมีความใกล้กันกว่าจุดที่อยู่ในคลัสเตอร์อื่น

หลักเกณฑ์ข้างต้นถูกนำมาใช้ในการวิเคราะห์การแสดงออกของยีนเพื่อจัดกลุ่มยีนตามรูปแบบการแสดงออก โดยแบ่ง n จุดที่อยู่ในปริภูมิ m มิติออกเป็น k กลุ่ม (คลัสเตอร์)



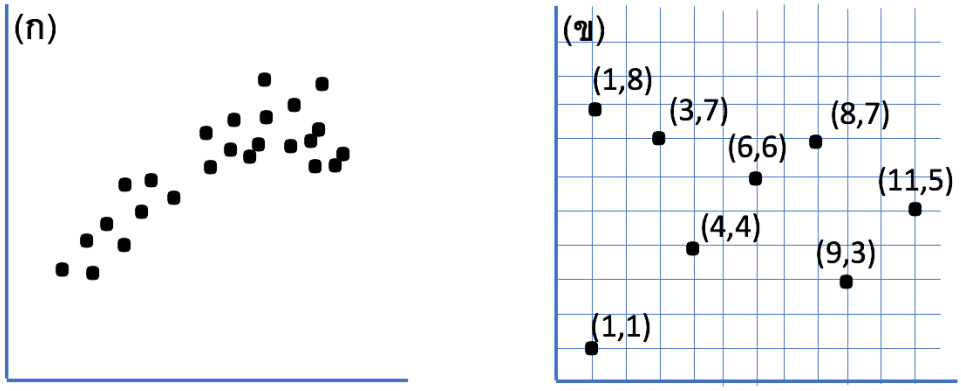
รูปที่ 7.7 (ก) การแบ่งจุดข้อมูล 15 จุดออกเป็น 3 กลุ่มโดยไม่เป็นไปตามเกณฑ์การจัดกลุ่มที่ดี (ข) ตัวอย่างการแบ่งกลุ่มที่เป็นไปตามเกณฑ์การจัดกลุ่มที่ดี

นิยามปัญหาที่ 7.1 ปัญหาการจัดกลุ่มที่ดี

ทำการแบ่งชุดของจุดออกเป็นกลุ่มตามเกณฑ์การจัดกลุ่มที่ดี (Good Clustering Problem)	
ข้อมูลเข้า	ชุดของ n จุด ในปริภูมิ m มิติ และเลขจำนวนเต็ม k แสดงจำนวนกลุ่มที่ต้องการ
ผลลัพธ์	ชุดของจุดที่ถูกแบ่งออกเป็น k กลุ่มและเป็นไปตามเกณฑ์การจัดกลุ่มที่ดี

ฝึกหัด	ข้อมูลการแสดงออกยีนในปริภูมิ 2 มิติจากคอลัมน์ที่ 4 และ 7 ของเมทริกซ์ในรูปที่ 7.5 สามารถแบ่ง 10 ยีนออกเป็น 3 คลัสเตอร์ได้อย่างไร

จากชุดของจุดในรูปที่ 7.8(ก) ต่อไปนี้ ถ้าพิจารณาด้วยตาเปล่าสามารถแบ่งได้เป็น 2 กลุ่ม อย่างไรก็ตาม 2 กลุ่มนี้ไม่เป็นไปตามเกณฑ์การจัดกลุ่มที่ดี และในความเป็นจริงแล้วเราไม่สามารถแบ่งชุดของจุดในตัวอย่างนี้ออกเป็น 2 กลุ่มที่เป็นไปตามเกณฑ์การจัดกลุ่มที่ดีได้



รูปที่ 7.8 (ก) ชุดของจุดที่สามารถแบ่งด้วยตาเปล่าได้เป็น 2 กลุ่ม แต่ไม่สามารถแบ่งตามเกณฑ์การจัดกลุ่มที่ดีได้ (ข) ตัวอย่างจุด 8 จุดในปริภูมิ 2 มิติ

ฝึกหัด	จงออกแบบอัลกอริทึมเพื่อตรวจสอบว่า สามารถแบ่งชุดข้อมูลออกเป็น k กลุ่มและเป็นไปตามเกณฑ์การจัดกลุ่มที่ดี โดยอัลกอริทึมต้องทำงานในเวลาพหุนาม (polynomial time)

หยุดคิด	จากข้อมูล 8 จุดในปริภูมิ 2 มิติของรูปที่ 7.8(ข) เราสามารถแบ่ง 8 จุดนี้ออกเป็น 3 กลุ่มได้อย่างไรบ้าง และสามารถแปลงปัญหาการจัดกลุ่มที่ดี (Good Clustering Problem) ให้อยู่ในรูปแบบของปัญหาเชิงคำนวณที่ชัดเจนมากขึ้นได้อย่างไร

แปลงปัญหาการแบ่งกลุ่มข้อมูลเป็นปัญหาการหาค่าที่เหมาะสมที่สุด

จากปัญหาการแบ่งกลุ่มข้อมูลที่ดีข้างต้นที่พยายามแบ่งข้อมูล n จุด (*Data*) ออกเป็น k กลุ่ม สามารถเปลี่ยนแนวทางในการแก้ปัญหาเป็นการเลือกชุดของจุดศูนย์กลาง (*Centers*) ของคลัสเตอร์จำนวน k จุด โดยต้องหา Centers ที่ทำให้ระยะทาง (distance) รวมระหว่าง Centers ใดๆ ไปยังจุดต่างๆ ใน *Data* มีค่าน้อยที่สุด คำถามที่สำคัญคำถามหนึ่งคือเรากำหนดฟังก์ชันที่ใช้ในการวัดระยะทางอย่างไร

ขั้นแรก กำหนดระยะทางยูคลิเดียน (Euclidian distance) ระหว่างจุด $v = (v_1,\ldots,v_m)$ และ $w = (w_1,\ldots w_m)$ ในปริภูมิ m มิติ แสดงโดย $d(v,w)$ เป็นความยาวของเส้นที่เชื่อมระว่างสองจุด $d(v,w)$ สามารถคำนวณได้โดยใช้สมการต่อไปนี้

$$d(v,w) = \sqrt{\sum_{i=1}^{m}(v_i - w_i)^2}$$

ขั้นที่สอง ในปริภูมิ m มิติ ที่มีจุดศูนย์กลาง *Centers* จำนวน k จุด เรากำหนด `d(DataPoint, Centers)` เป็นระยะทางยูคลิเดียนของจุดข้อมูล *DataPoint* ไปยัง Centers ที่ใกล้ที่สุด ดังนั้น

$$d(DataPoint, Centers) = \min_{all\ points\ x\ from\ Centers} d(DataPoint, x)$$

และกำหนด `MAXDISTANCE(Data, Centers)` เป็นระยะทางระหว่างทุกจุดใน *Data* กับ จุดศูนย์กลางใน *Centers* ซึ่งเป็นค่าที่มากที่สุดของ `d(DataPoint, Centers)` ระหว่างทุกจุดข้อมูล

$$MAXDISTANCE(Data, Centers) = \max_{all\ point\ DataPoint\ from\ Data} d(DataPoint, Centers)$$

นิยามปัญหาที่ 7.2 ปัญหาการจัดกลุ่มข้อมูลแบบเค-เซ็นเตอร์

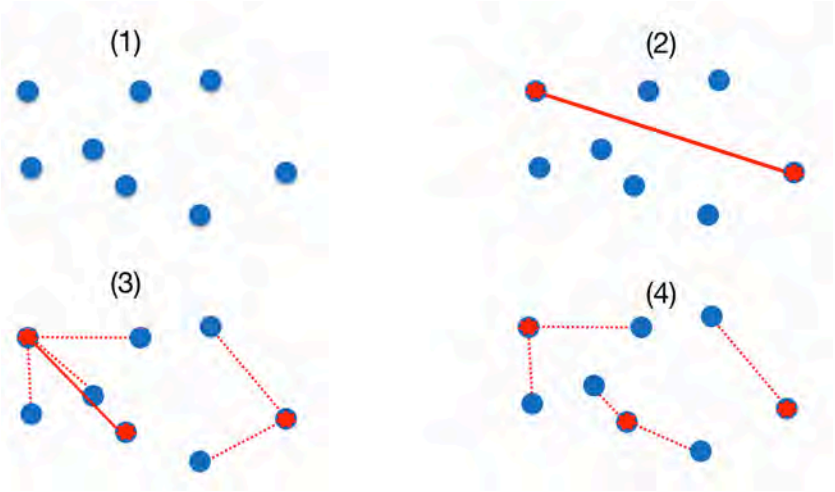
รับข้อมูลเข้าเป็นชุดของจุด หาจุดศูนย์กลาง k จุดที่ทำให้ค่า `MAXDISTANCE()` มีค่าน้อยที่สุด	
ข้อมูลเข้า	ชุดของจุดข้อมูล *Data* และค่าจำนวนเต็ม k
ผลลัพธ์	ชุดของ Centers จำนวน k จุด ที่ทำให้ค่า `MAXDISTANCE(DataPoint, Centers)` มีค่าน้อยที่สุดสำหรับทุกค่า k Centers ที่เป็นไปได้

การเลือกจุดที่ห่างที่สุดก่อน

ถึงแม้ปัญหาการจัดกลุ่มเค-เซ็นเตอร์ (K-Center) ตรงไปตรงมา ความซับซ้อนของการแก้ปัญหากลับอยู่ในระดับ NP-Hard เพื่อให้การแก้ปัญหามีความซับซ้อนน้อยลงจึงได้มีการเพิ่มฮิวริสติก (heuristic) โดยสุ่มเลือกจุดจาก *Data* (แทนการเลือกจากจุดในปริภูมิ m มิติ) และเพิ่มจุดนั้นเข้าไปในชุดของจุดศูนย์กลาง *Centers* วนซ้ำโดยการเลือกจุดศูนย์กลางถัดไปจาก *Data* ที่มีระยะห่างที่สุดจากจุดศูนย์กลางทุกจุดที่ถูกเลือกมาก่อนหน้า ดังแสดงในรหัสเทียมที่ 7.1 `FarthestFirstTraversal()`และรูปที่ 7.9

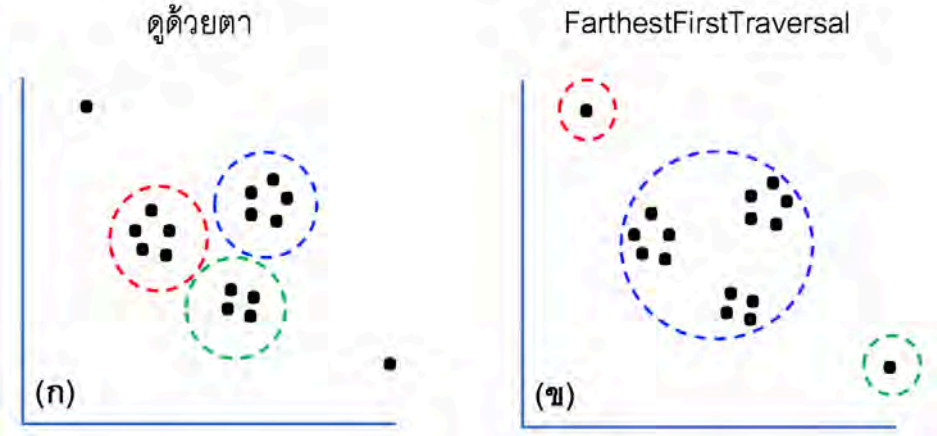
รหัสเทียมที่ 7.1 FarthestFirstTraversal

```
1  FartheseFirstTraversal(Data, k)
2      Centers <- 1 จุดที่เลือกมาแบบสุ่มจาก Data
3      while |Centers| < k
4          DataPoint <- จุดใน Data ที่ทำให้ d(DataPoint,Centers)มีค่ามากที่สุด
5          เพิ่ม DataPoint นี้ใน Centers
6      ส่งกลับ Centers
```



รูปที่ 7.9 การประยุกต์ใช้วิธี `FarthestFirstTraversal()` ในการจัดกลุ่มข้อมูล โดยจุดสีแดงในขั้นตอนที่ (2), (3) และ (4) เป็นจุดศูนย์กลางที่ถูกเลือกและเพิ่มเข้าชุดจุดศูนย์กลาง Centers ในแต่ละรอบ

วิธีการ `FarthestFirstTraversal()` เร็วและผลของการจัดกลุ่มมีความใกล้เคียงกับผลลัพธ์ที่ดีที่สุดของการจัดกลุ่มแบบเค-เซ็นเตอร์ อย่างไรก็ตามการจัดกลุ่มยืนตามรูปแบบการแสดงออกจะไม่ใช้อัลกอริทึมนี้ การจัดกลุ่มแบบเค-เซ็นเตอร์จะเลือกชุดของจุดศูนย์กลาง *Centers* ที่ทำให้ `MAXDISTANCE(Data, Centers)` มีค่าน้อยสุด ซึ่งค่า MAXDISTANCE คือระยะทางที่มากที่สุดระหว่างจุดใดๆ กับจุดศูนย์กลางที่ใกล้ที่สุด อย่างไรก็ตามนักชีววิทยามักสนใจความแตกต่างโดยรวมมากกว่าความแตกต่างที่มากที่สุด เนื่องจากในบางกรณีความแตกต่างที่มากที่สุดอาจเป็นเพียงข้อมูลที่มีสัญญาณรบกวน รูปที่ 7.10(ข) ไม่ว่าจุดใดจะถูกเลือกเป็นจุดศูนย์กลางแรก จากผลของ `MAXDISTNACE()` จุดที่เป็นสัญญาณรบกวนตำแหน่งซ้ายบนและขวาล่างอย่างละจุดจะถูกเลือกเป็นจุดศูนย์กลางของกลุ่มที่ 2 และ 3 ที่มีสมาชิกจุดเดียวคือจุดศูนย์กลางเอง



รูปที่ 7.10 (ก) ชุดของจุดข้อมูลที่เห็นได้ชัดด้วยตาเปล่าว่าสามารถแบ่งได้เป็น 3 กลุ่มและมีจุดข้อมูล 2 จุดที่เป็นสัญญาณรบกวน (ข) ปัญหาของการใช้วิธี `MAXDISTANCE()` ในการหา Centers จุดซ้ายบนและขวาล่างจะถูกเลือกมาเป็นจุดศูนย์กลางของกลุ่มที่ 2 และ 3 และมีสมาชิกเพียงจุดเดียว

หยุดคิด	จากปัญหาของ `FarthestFirstTraversal()` ข้างต้น จะเปลี่ยนแปลงฟังก์ชันการให้คะแนนจาก `MAXDISTANCE()` เป็นฟังก์ชันอื่นที่สามารถพิจารณาความแตกต่างของข้อมูลในกลุ่มแบบรวมซึ่งสะท้อนมุมมองของนักชีววิทยาได้อย่างไร

การจัดกลุ่มข้อมูลแบบเค-มีนส์

Squared error distortion

เพื่อเป็นการลดข้อจำกัดของการใช้ MAXDISTANCE มีการเสนอฟังก์ชัน ***`DISTORTION(Data, Centers)`*** การให้คะแนนใหม่ระหว่างชุดข้อมูล n จุดใน *Data* และชุดของจุดศูนย์กลาง k จุดใน *Centers* เพื่อวัดค่า squared error distortion ซึ่งคำนวณจากผลรวมเฉลี่ยของระยะทางระหว่างแต่ละ *DataPoint* ไปยังจุดศูนย์กลางที่ใกล้ที่สุดยกกำลังสอง ดังสมการต่อไปนี้

$$DISTORTION(Data, Centers) = \frac{1}{n} \sum_{all\ points\ DataPoint\ in\ Data} d(DataPoint, Centers)^2$$

ในขณะที่ `MAXDISTANCE(`***`Data,Centers`***`)` ใช้ระยะทางที่ยาวที่สุดจากจุดข้อมูลใดๆ ไปยังจุดศูนย์กลางที่ใกล้ที่สุด ในกรณีของ squared error distortion ใช้ระยะทางเฉลี่ยของทุกจุดข้อมูลไปยังจุดศูนย์กลางที่ใกล้ที่สุดของจุดเหล่านั้น

นิยามปัญหาที่ 7.3 ปัญหา Squared Error Distortion

คำนวณค่า squared error distortion จากชุดข้อมูล และ ชุดของจุดศูนย์กลาง	
ข้อมูลเข้า	ชุดของจุดข้อมูล *Data* และชุดของจุดศูนย์กลาง *Centers*
ผลลัพธ์	ค่า squared error distortion

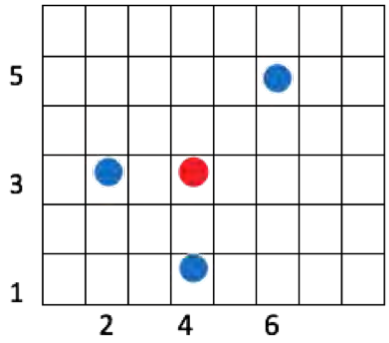
ค่า squared error distortion นำไปสู่การปรับปรุงการแก้ปัญหาการจัดกลุ่มข้อมูลแบบเค-เซ็นเตอร์

นิยามปัญหาที่ 7.4 ปัญหาการจัดกลุ่มข้อมูลแบบเค-มีนส์

รับข้อมูลเข้าเป็นชุดของจุด หาจุดศูนย์กลาง k จุดที่ทำให้ค่า squared error distortion มีค่าน้อยที่สุด	
ข้อมูลเข้า	ชุดของจุดข้อมูล *Data* และค่าจำนวนเต็ม k
ผลลัพธ์	ชุดของ Centers จำนวน k จุด ที่ทำให้ค่า `DISTORTION(`***`Data,Centers`***`)` มีค่าน้อยสุดสำหรับทุกค่า k Centers ที่เป็นไปได้

การจัดกลุ่มข้อมูลแบบเค-มีนส์และจุดศูนย์ถ่วง

ปัญหาการจัดกลุ่มข้อมูลแบบเค-มีนส์ (K-Means) เป็นปัญหา NP-Hard เมื่อ k > 1 สำหรับกรณีที่ k = 1 จะเท่ากับการหาจุดศูนย์กลาง x ที่ทำให้ค่า squared error distortion มีค่าน้อยสุด ถึงแม้การแบ่งชุดของข้อมูลโดยค่า k = 1 จะตรงไปตรงมา คำถามคือจะหาจุดศูนย์กลางที่ทำให้ค่า squared error distortion น้อยสุดได้อย่างไร คำตอบของคำถามนี้สามารถนำไปใช้ออกแบบวิธีการหาคำตอบในกรณีที่ k > 1 ได้ด้วยหรือไม่



รูปที่ 7.11 จุดข้อมูล (สีน้ำเงิน) และจุดศูนย์ถ่วง (สีแดง) ที่คำนวณจากทั้ง 3 จุดข้อมูล

เรากำหนดจุดศูนย์ถ่วง (center of gravity) ของชุดข้อมูล *Data* โดยค่าในเวกเตอร์ลำดับที่ i ของจุดศูนย์ถ่วงคำนวณจากผลรวมเฉลี่ยของค่าลำดับที่ i ของทุกจุดข้อมูลใน *Data* จากตัวอย่างในรูปที่ 7.11 ค่าจุดศูนย์ถ่วงของจุด (2,3), (4,1) และ (6,5) มีค่าเท่ากับ

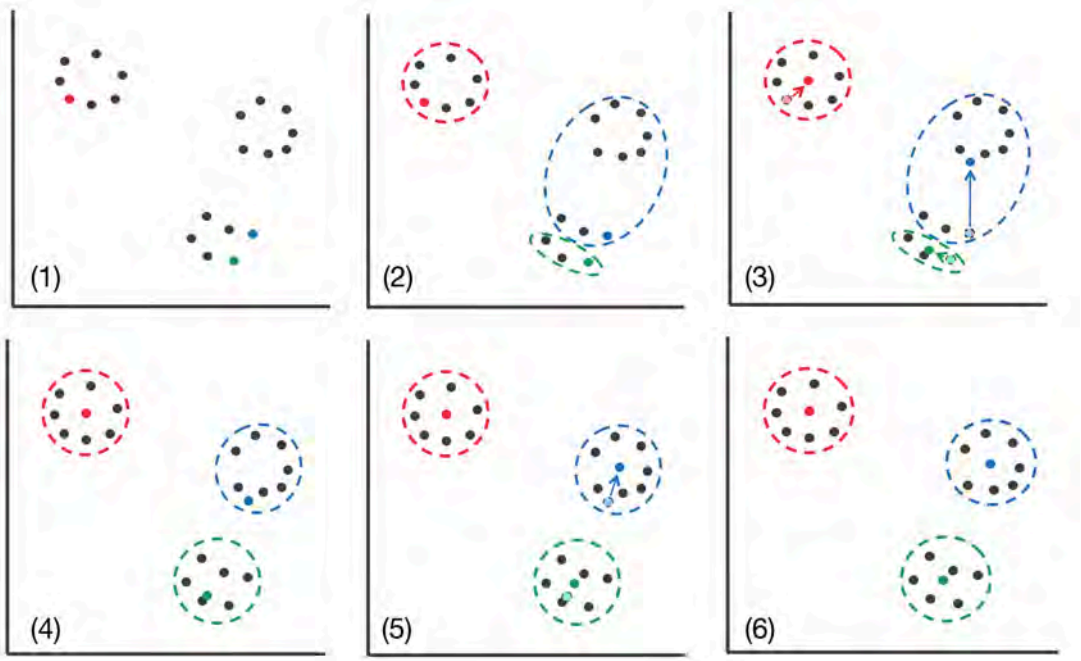
$$\left(\frac{2+4+6}{3}, \frac{3+1+5}{3}\right) = (4,3)$$

ทฤษฎีจุดศูนย์ถ่วง (Center of Gravity Theorem) **:** จุดศูนย์ถ่วงของชุดข้อมูลใน *Data* จะมีเพียงจุดเดียวซึ่งสามารถใช้ในการแก้ปัญหาการจัดกลุ่มแบบเค-มีนส์ในกรณีที่ k = 1

อัลกอริทึม Lloyd

อัลกอริทึม Lloyd (รูปที่ 7.12) เป็นฮิวริสติสที่ถูกใช้ในการจัดกลุ่มข้อมูลแบบเค-มีนส์ที่มีการใช้งานอย่างแพร่หลาย โดยในขั้นตอนแรกทำการสุ่มเลือกจุดข้อมูลจาก *Data* จำนวน k จุดเพื่อเป็น *Centers* และทำการวนซ้ำ 2 ขั้นตอนต่อไปนี้

- **จากชุดของจุดศูนย์กลางไปยังชุดของคลัสเตอร์** จากชุดของจุดศูนย์กลางที่เลือกไว้ ทำการกำหนดกลุ่มหรือคลัสเตอร์ให้กับจุดข้อมูลอื่นๆ โดยจุดเหล่านี้จะเป็นสมาชิกของกลุ่มที่มีระยะห่างระหว่างจุดข้อมูลและจุดศูนย์กลางของกลุ่มน้อยสุด
- **จากชุดของคลัสเตอร์ไปยังชุดของจุดศูนย์กลาง** หลังจากจุดข้อมูลทั้งหมดมีกลุ่มแล้ว ทำการคำนวณจุดศูนย์ถ่วงใหม่สำหรับแต่ละกลุ่มเพื่อใช้เป็นจุดศูนย์กลางในการกำหนดสมาชิกให้ในรอบถัดไป



รูปที่ 7.12 การทำงานของอัลกอริทึม Lloyd ในแต่ละขั้นตอนโดย k = 3

(ที่มา: รูปที่ 8.12 ของ [52])

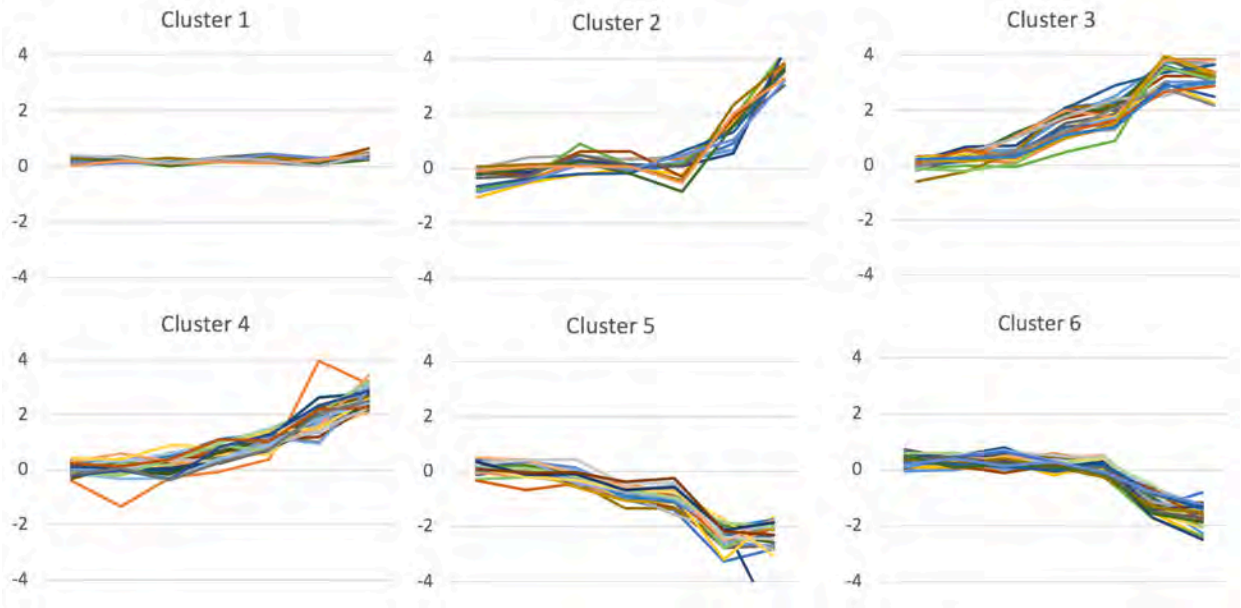
การจัดกลุ่มยีนตามรูปแบบการแสดงออกนำไปสู่ยีนที่เกี่ยวข้องกับ diauxic shift

รูปที่ 7.13 แสดงผลการจัดกลุ่ม 196 ยีนของยีสต์ออกเป็น 6 กลุ่ม (k = 6) ซึ่งแต่ละกลุ่มประกอบด้วย 15, 14, 23, 45, 32, และ 67 ยีนตามลำดับ โดยกราฟของแต่ละกลุ่มแสดงรูปแบบระดับการแสดงออกของยีนที่แตกต่างกันในช่วงก่อนและหลังการเกิด diauxic shift ผลการจัดกลุ่มนี้เป็นจุดเริ่มต้นของคำถามหรือสมมติฐานทางชีววิทยาอื่นๆ เพิ่มเติม เช่น ชุดของยีนในกลุ่มเดียวกันที่มีรูปแบบการแสดงออกของยีนคล้ายคลึงกันอาจถูกควบคุมโดยแฟกเตอร์ถอดรหัส (transcription factor: TF) เดียวกัน ซึ่งหมายถึงสายดีเอ็นเอในส่วนหน้าของยีนเหล่านี้อาจมีโมทิฟควบคุม (regulatory motif) ที่มีรูปแบบเดียวกันหรือใกล้เคียงกัน (ตัวอย่างอัลกอริทึมที่ใช้ในการหาโมทิฟควบคุมอยู่ในบทที่ 4) คำถามทางชีววิทยาอื่นๆ เช่น กระบวนการใดอยู่เบื้องหลังการเพิ่มระดับการแสดงออกของยีนในคลัสเตอร์ที่ 2 กระบวนการใดอยู่เบื้องหลังการลดการแสดงออกของยีนในคลัสเตอร์ที่ 5 และการเปลี่ยนแปลงระดับการแสดงออกของยีนเหล่านี้เกี่ยวข้องกับ diauxic shift อย่างไร

ข้อจำกัดของการจัดกลุ่มข้อมูลแบบเค-มีนส์

จากขั้นตอนการทำงานของอัลกอริทึม Lloyd การจัดกลุ่มข้อมูลดูเป็นเรื่องง่าย อย่างไรก็ตาม อัลกอริทึม Lloyd ไม่สามารถจัดกลุ่มข้อมูลที่แสดงในรูปที่ 7.14 ได้ถูกต้อง ดังผลที่แสดงในรูปที่ 7.15(ล่าง) อัลกอริทึม Lloyd มีข้อจำกัดในการจัดกลุ่มข้อมูลในกรณีที่การกระจายของจุดข้อมูลในกลุ่มอยู่ในลักษณะที่เป็นชั้นดังรูปที่ 7.15(ล่างซ้าย) การ

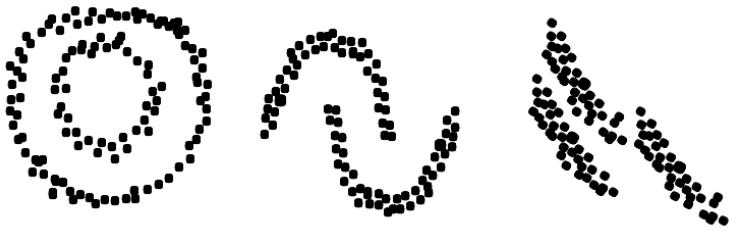
กระจายของจุดข้อมูลที่มีความคาบเกี่ยวระหว่างกลุ่ม 7.15(ล่างกลาง) และ 7.15(ล่างขวา) ซึ่งต่างจากผลของการจัดกลุ่มด้วยตาเปล่าในรูปที่ 7.15(บน)



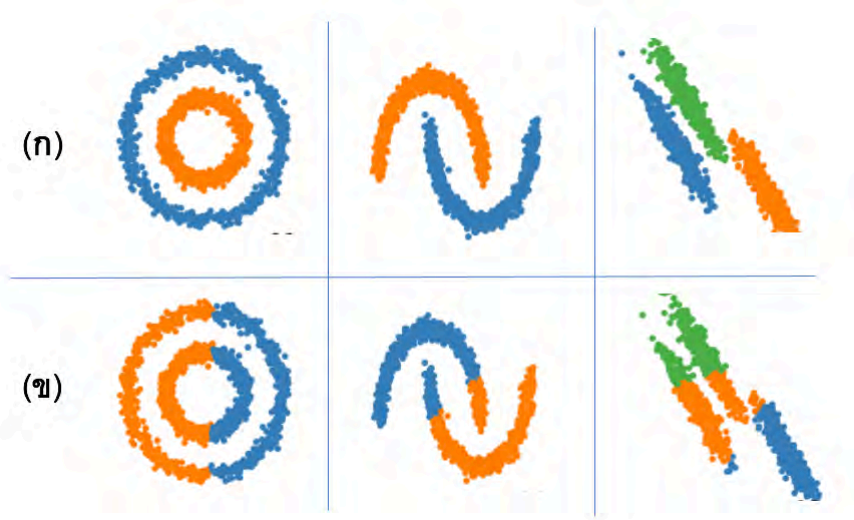
รูปที่ 7.13 ผลของการใช้อัลกอริทึม Lloyd ในการจัดกลุ่มยีน 196 ยีนของยีสต์ออกเป็น 6 กลุ่ม

หยุดคิด	เราจะจัดกลุ่มของจุดในรูปที่ 7.14 อย่างไร

นอกจากนี้ข้อจำกัดของการจัดกลุ่มแบบเค-มีนส์ตามที่อธิบายไปในหัวข้อก่อนหน้าคือแต่ละจุดข้อมูลจะเป็นสมาชิกของกลุ่มใดกลุ่มหนึ่งเท่านั้น ไม่สามารถอยู่ได้มากกว่า 1 กลุ่ม เงื่อนไขนี้เป็นเงื่อนไขของการจัดกลุ่มแบบ hard clustering ซึ่งไม่เหมาะสมกับจุดข้อมูลที่เป็น midpoint หรือจุดข้อมูลที่มีความใกล้เคียงกับกลุ่มข้อมูลมากกว่า 1 กลุ่ม วิธีการจัดการปัญหานี้ทำได้โดยเปลี่ยนวิธีการกำหนดกลุ่มให้กับจุดข้อมูลใดๆ โดยแต่ละจุดข้อมูลสามารถถูกกำหนดให้อยู่ได้มากกว่า 1 กลุ่ม และกำกับด้วยค่าความน่าจะเป็นในการเป็นสมาชิกของแต่ละกลุ่ม (รูปที่ 7.16)

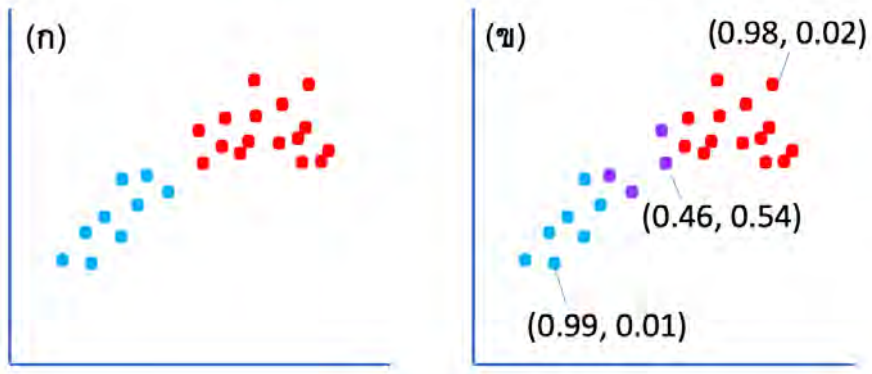


รูปที่ 7.14 ความท้าท้ายของปัญหาการจัดกลุ่มข้อมูล เมื่อ k = 2 สำหรับชุดข้อมูลรูปซ้ายและรูปตรงกลาง และ k = 3 สำหรับชุดข้อมูลรูปขวา



รูปที่ 7.15 (ก) ผลการจัดกลุ่มจุดข้อมูลโดยใช้สายตาหรือโดยอัลกอริทึมกลุ่มอื่น (ข) ผลการจัดกลุ่มจุดข้อมูลโดยใช้อัลกอริทึม Lloyd

(*ที่มา*: *ปรับจาก* https://scikit-learn.org/stable/modules/clustering.html)



รูปที่ 7.16 (ก) ชุดของจุดข้อมูลจากรูปที่ 7.8(ก) ที่ถูกแบ่งออกเป็น 2 กลุ่มโดยใช้อัลกอริทึม Lloyd (ข) ผลการจัดกลุ่มข้อมูลแบบซอฟต์เค-มีนส์โดยใช้ข้อมูลชุดเดียวกัน

การจัดกลุ่มข้อมูลแบบซอฟต์เค-มีนส์

การประยุกต์ใช้ Expectation Maximization ในการจัดกลุ่มข้อมูล

หัวข้อนี้แสดงการประยุกต์ใช้อัลกอริทึม Expectation Maximization (EM) กับการทำงานของอัลกอริทึม Lloyd เพื่อให้สามารถจัดกลุ่มข้อมูลแบบซอฟต์เค-มีนส์ (soft K-Means) ได้ อัลกอริทึมใหม่นี้เริ่มการทำงานโดยการสุ่มเลือกชุดของจุดศูนย์กลางจำนวน k จุดและทำการวนซ้ำ 2 ขั้นตอนต่อไปนี้

- **จากชุดของจุดศูนย์กลางไปยังชุดของซอฟต์คลัสเตอร์ (E-step):** จากชุดของจุดศูนย์กลางที่เลือกไว้ กำหนดค่าความน่าจะเป็นให้กับจุดข้อมูลที่จะมาเป็นสมาชิก โดยค่าความน่าจะเป็นที่มากกว่าหมายถึงจุดนั้นมีโอกาสอยู่ในกลุ่มมากกว่า

- **จากชุดของซอฟต์คลัสเตอร์ไปยังชุดของจุดศูนย์กลาง (M-step):** หลังจากทุกจุดข้อมูล ถูกกำกับด้วยชุดค่าความน่าจะเป็นของการเป็นสมาชิกแต่ละกลุ่มแล้ว ทำการคำนวณชุดของจุดศูนย์ถ่วงใหม่ เพื่อใช้กำหนดค่าความน่าจะเป็นของการเป็นสมาชิกให้กับจุดข้อมูลต่างๆ ในรอบถัดไป

จากชุดของจุดศูนย์กลางไปยังการจัดกลุ่มแบบซอฟต์

ในหัวข้อที่ผ่าน **"จุดศูนย์ถ่วง"** เป็นตัวแทนจุดศูนย์กลางที่ใช้ในการกำหนดจุดสมาชิกให้กับกลุ่ม ถ้าจินตนาการว่าจุดศูนย์กลางเหล่านี้คือดวงดาว ในขณะที่จุดข้อมูลอื่นๆ คือดาวบริวาร จุดข้อมูลที่อยู่ใกล้จุดศูนย์กลางจะมีแรงดึงดูดมากกว่า ถ้ากำหนดให้มีจุดศูนย์กลาง k จุด *Centers* = (x_1,...,x_k) และมีชุดของจุดข้อมูล n จุด *Data* = ($Data_1$,...,$Data_n$) เราสามารถสร้างเมทริกซ์ความรับผิดชอบ *HiddenMatrix* ขนาด k x n โดยที่ $HiddenMatrix_{i,j}$ เก็บค่าแรงดึงดูดระหว่างจุดศูนย์กลาง i กับจุดข้อมูล j โดยค่าแรงดึงดูดนี้คำนวณจากกฎกำลังสองผกผันของนิวตัน (Newtonian inverse-square law) ดังสมการต่อไปนี้

$$HiddenMatrix_{i,j} = \frac{\frac{1}{d(Data_j, x_i)^2}}{\sum_{all\ centers\ x_i} \frac{1}{d(Data_j, x_i)^2}}$$

อย่างไรก็ตาม พาร์ทิชันฟังก์ชัน (partition function) จากฟิสิกส์สถิติต่อไปนี้ ใช้งานได้ดีกว่าในเชิงปฏิบัติ

$$HiddenMatrix_{i,j} = \frac{e^{-\beta \cdot d(Data_j, x_i)}}{\sum_{all\ centers\ x_i} e^{-\beta \cdot d(Data_j, x_i)}}$$

โดยในสมการนี้ e เป็นค่าฐานของลอการิทึมธรรมชาติ $(e \approx 2.718)$ และ β เป็นค่าพารามิเตอร์ที่สะท้อนความยืดหยุ่นในการกำหนดค่าความน่าจะเป็นหรือ stiffness parameter รูปที่ 7.17 แสดงตัวอย่าง *HiddenMatrix* ของ 8 จุดข้อมูลที่ถูกแบ่งออกเป็น 3 กลุ่ม

HiddenMatrix

	A	B	C	D	E	F	G	H
1	0.70	0.15	0.73	0.40	0.15	0.80	0.05	0.05
2	0.20	0.80	0.17	0.20	0.80	0.10	0.05	0.20
3	0.10	0.05	0.10	0.40	0.05	0.10	0.90	0.75

รูปที่ 7.17 HiddenMatrix ของ 8 จุดข้อมูลที่ถูกแบ่งออกเป็น 3 กลุ่ม

จากชุดของซอฟต์คลัสเตอร์ไปยังชุดของจุดศูนย์กลาง

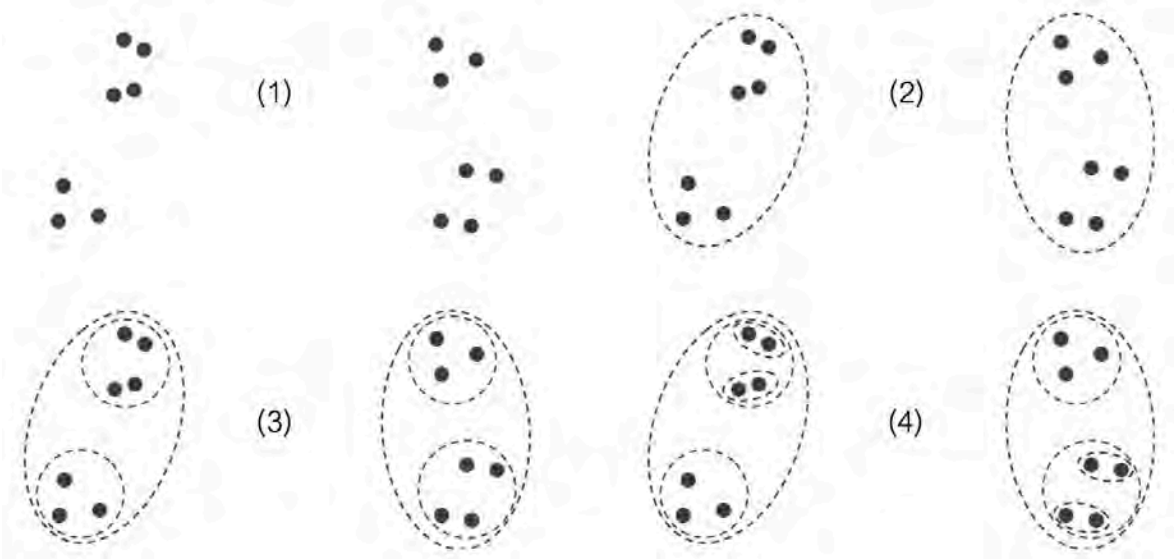
ในการจัดกลุ่มข้อมูลแบบซอฟต์เค-มีนส์ถ้ากำหนดให้ $HiddenMatrix_i$ แสดงบรรทัดที่ i ของ *HiddenMatrix* ดังนั้นเราสามารถปรับค่าจุดศูนย์กลาง x_i โดยใช้สมการต่อไปนี้

$$x_{i,j} = \frac{HiddenMatrix_i \cdot Data^j}{HiddenMatrix_i \cdot \vec{1}}$$

โดย $Data^j$ เป็นเวกเตอร์ขนาด n มิติ ที่เก็บค่าพิกัด (coordinate) ลำดับที่ j ของจุดข้อมูล n จุด ทั้งนี้จุดศูนย์กลาง x_i ที่มีการปรับข้อมูลแล้วเรียกว่า weighted center of gravity ของจุดข้อมูลใน *Data*

การจัดกลุ่มข้อมูลเชิงลำดับชั้น

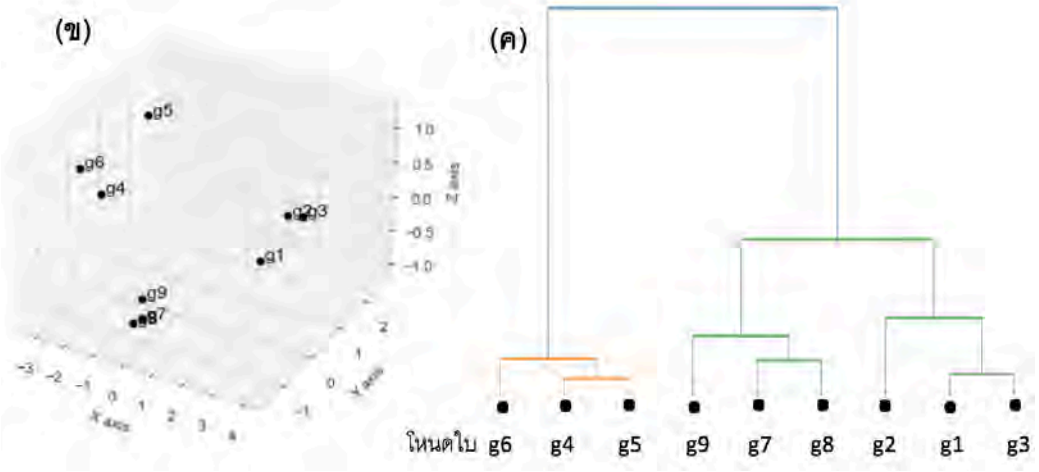
ในหัวข้อที่ผ่านมา สมมติฐานหลักในการจัดกลุ่มข้อมูลคือเราทราบจำนวนกลุ่มข้อมูล (ทราบค่า k) อย่างไรก็ตามในทางปฏิบัติ กลุ่มข้อมูลมักสามารถแบ่งกลุ่มย่อยลงไปได้อีกเป็นลำดับชั้นดังแสดงในรูปที่ 7.18 เพื่อให้เห็นความสัมพันธ์ของกลุ่มข้อมูลย่อยเหล่านี้ อัลกอริทึมในการจัดกลุ่มข้อมูลเชิงลำดับชั้น (hierarchical clustering) ใช้เมทริกซ์ระยะทาง *D* ขนาด n x n ในการสร้างลำดับชั้นระยะห่างระหว่างจุดข้อมูล โดยเริ่มสร้างจากจุดข้อมูลสองจุดที่มีความใกล้กันที่สุดก่อนและรวมกลุ่มขึ้นมาเป็นลำดับชั้นจนถึงโหนดราก (root) โดยผลการจัดกลุ่มข้อมูลเชิงลำดับชั้นอยู่ในรูปแบบของต้นไม้ (tree) ทั้งนี้โหนดใบ (leaf) ในต้นไม้คือยีน ส่วนโหนดภายในแสดงคลัสเตอร์หรือกลุ่มข้อมูล (รูปที่ 7.19)



รูปที่ 7.18 การแบ่งจุดข้อมูลเป็นกลุ่มย่อยเป็นลำดับชั้นตามความใกล้กันของจุด

(ก) เมทริกซ์ระยะทาง

	g1	g2	g3	g4	g5	g6	g7	g8	g9
g1	0.00	2.28	0.96	9.79	9.73	9.38	5.02	5.05	5.09
g2	2.28	0.00	2.33	9.24	9.20	8.99	4.63	4.65	4.67
g3	0.96	2.33	0.00	10.40	10.33	10.01	5.61	5.63	5.67
g4	9.79	9.24	10.40	0.00	0.43	0.89	4.80	4.79	4.75
g5	9.73	9.20	10.33	0.43	0.00	0.76	4.75	4.74	4.70
g6	9.38	8.99	10.01	0.89	0.76	0.00	4.45	4.45	4.41
g7	5.02	4.63	5.61	4.80	4.75	4.45	0.00	0.14	0.21
g8	5.05	4.65	5.63	4.79	4.74	4.45	0.14	0.00	0.17
g9	5.09	4.67	5.67	4.75	4.70	4.41	0.21	0.17	0.00



รูปที่ 7.19 (ก) เมทริกซ์ระยะทางสร้างจากระยะทางยูคลิด (Euclidian distance) (ข) เวกเตอร์ระดับการแสดงออกของยีนที่แสดงด้วยจุดข้อมูลใน 3 มิติ (ค) ต้นไม้ที่เป็นผลของการจัดกลุ่มข้อมูลเชิงลำดับชั้นโดยใช้ข้อมูลเมทริกซ์ระยะทางด้านบน

อัลกอริทึมในการจัดกลุ่มข้อมูลเชิงลำดับชั้น

อัลกอริทึมในการจัดกลุ่มข้อมูลเชิงลำดับชั้น (hierarchical clustering algorithm) พิจารณาข้อมูลเข้า n จุด เป็นข้อมูล n กลุ่ม จากนั้นจะทำการรวมข้อมูล 2 จุดที่มีระยะทางใกล้กันที่สุดเป็นกลุ่มใหม่ ทำการคำนวณหาตัวแทนข้อมูลของกลุ่มใหม่ วนซ้ำเพื่อทำการรวมสองจุดที่ใกล้กันที่สุดในรอบถัดไปจนกระทั่งข้อมูลทั้ง n จุดรวมเป็นกลุ่มเดียวดังแสดงในรหัสเทียมที่ 7.2 `HierarchicalClustering()` ทั้งนี้รหัสเทียมนี้ยังไม่มีการกำหนดวิธีการคำนวณค่าระยะทางระหว่างกลุ่มใหม่ที่เกิดขึ้นกับกลุ่มเดิมทั้งหมดที่มีอยู่ $D(C_{new}, C)$ ในทางปฏิบัติแต่ละอัลกอริทึมอาจมีวิธีการคำนวณค่าระยะทางแตกต่างกันไปซึ่งอาจทำให้ได้ผลลัพธ์ของการจัดกลุ่มที่แตกต่างกันอย่างมาก

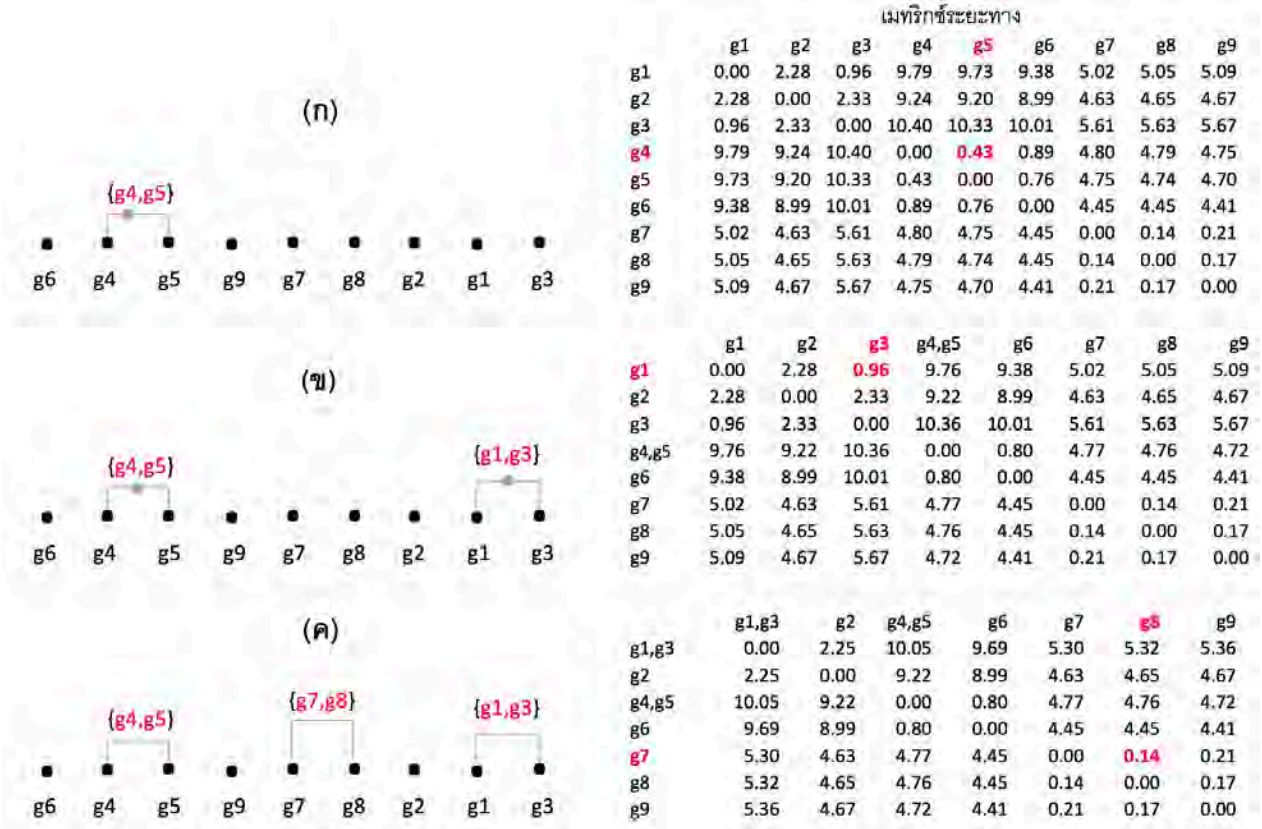
วิธีการพื้นฐานในการคำนวณค่าระยะทางระหว่างคลัสเตอร์ C_1 และ C_2 ที่มีการใช้งานกันอย่างแพร่หลายคือการหาระยะทางที่สั้นที่สุดระหว่างทุกคู่ของสมาชิกระหว่างสองคลัสเตอร์ ดังสมการต่อไปนี้

$$D_{min}(C_1, C_2) = \min_{all\ points\ i\ in\ cluster\ C_1, all\ points\ j\ in\ cluster\ C_2} D_{i,j}$$

รหัสเทียมที่ 7.2 HierarchicalClustering

```
HierarchicalClustering(D,n)
    Clusters <- ข้อมูล n จุดแยกจากกัน
    สร้างกราฟ T ที่มี n โหนด
    while ยังมีจำนวนคลัสเตอร์มากกว่า 1 คลัสเตอร์
        หาคลัสเตอร์ Ci และ Cj ที่มีระยะทางใกล้กันที่สุดจาก เมทริกซ์ระยะทาง D
        รวมคลัสเตอร์ Ci และ Cj เข้าเป็นคลัสเตอร์ใหม่ Cnew ที่มีจำนวนสมาชิกเท่ากับ |Ci|+|Cj|
        เพิ่มโหนดใหม่ Cnew เข้าใน T
        เพิ่มเส้นเชื่อมชี้จากโหนด Cnew ไปยัง Ci และ Cj
        ลบบรรทัดและคอลัมน์ที่เป็นข้อมูล Ci และ Cj เดิม
        ลบ Ci และ Cj ออกจาก Clusters
        เพิ่มบรรทัดและคอลัมน์ของ Cnew ไปยัง D โดยคำนวณ D(Cnew,C) สำหรับทุก C ใน Clusters
        เพิ่ม Cnew ไปยัง Clusters
    root <-โหนดที่เหลือใน T
    ส่งกลับ T
```

รูปที่ 7.20 (ก) แสดงเมทริกซ์ระยะทางจากรูปที่ 7.19 โดยตัวเลขสีแดงแสดงค่าระยะทางที่น้อยที่สุดระหว่างยีน g4 และ g5 รูปที่ 7.20 (ข) แสดงการรวม g4 และ g5 เข้าเป็นกลุ่มเดียวกัน อัพเดตค่าในเมทริกซ์ระยะทางระหว่างกลุ่มใหม่ {g4,g5} กับกลุ่มเดิมทั้งหมด และรวม g1 และ g3 เข้าเป็นกลุ่มเดียวกันเพราะมีระยะทางสั้นที่สุด รูปที่ 7.20 (ค) แสดงการอัพเดตค่าในเมทริกซ์ระยะทางระหว่างกลุ่มใหม่ {g1,g3} กับทุกกลุ่มที่เหลือ และรวม g7 และ g8 เข้าด้วยกันเพราะมีระยะทางสั้นที่สุด และวนทำซ้ำจนกว่าทุกยีนจะอยู่ในกลุ่มเดียวกัน



เมทริกซ์ระยะทาง

	g1	g2	g3	g4	g5	g6	g7	g8	g9
g1	0.00	2.28	0.96	9.79	9.73	9.38	5.02	5.05	5.09
g2	2.28	0.00	2.33	9.24	9.20	8.99	4.63	4.65	4.67
g3	0.96	2.33	0.00	10.40	10.33	10.01	5.61	5.63	5.67
g4	9.79	9.24	10.40	0.00	0.43	0.89	4.80	4.79	4.75
g5	9.73	9.20	10.33	0.43	0.00	0.76	4.75	4.74	4.70
g6	9.38	8.99	10.01	0.89	0.76	0.00	4.45	4.45	4.41
g7	5.02	4.63	5.61	4.80	4.75	4.45	0.00	0.14	0.21
g8	5.05	4.65	5.63	4.79	4.74	4.45	0.14	0.00	0.17
g9	5.09	4.67	5.67	4.75	4.70	4.41	0.21	0.17	0.00

	g1	g2	g3	g4,g5	g6	g7	g8	g9
g1	0.00	2.28	0.96	9.76	9.38	5.02	5.05	5.09
g2	2.28	0.00	2.33	9.22	8.99	4.63	4.65	4.67
g3	0.96	2.33	0.00	10.36	10.01	5.61	5.63	5.67
g4,g5	9.76	9.22	10.36	0.00	0.80	4.77	4.76	4.72
g6	9.38	8.99	10.01	0.80	0.00	4.45	4.45	4.41
g7	5.02	4.63	5.61	4.77	4.45	0.00	0.14	0.21
g8	5.05	4.65	5.63	4.76	4.45	0.14	0.00	0.17
g9	5.09	4.67	5.67	4.72	4.41	0.21	0.17	0.00

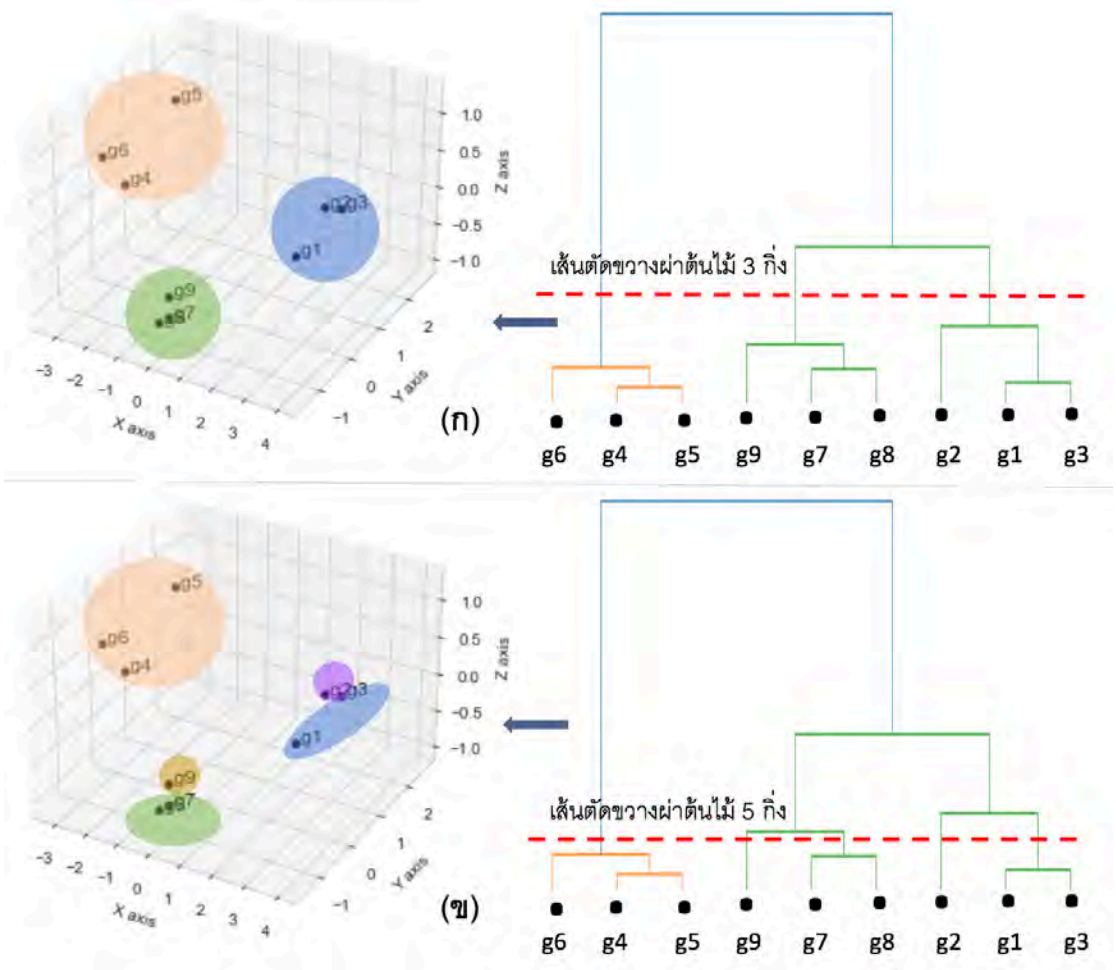
	g1,g3	g2	g4,g5	g6	g7	g8	g9
g1,g3	0.00	2.25	10.05	9.69	5.30	5.32	5.36
g2	2.25	0.00	9.22	8.99	4.63	4.65	4.67
g4,g5	10.05	9.22	0.00	0.80	4.77	4.76	4.72
g6	9.69	8.99	0.80	0.00	4.45	4.45	4.41
g7	5.30	4.63	4.77	4.45	0.00	0.14	0.21
g8	5.32	4.65	4.76	4.45	0.14	0.00	0.17
g9	5.36	4.67	4.72	4.41	0.21	0.17	0.00

รูปที่ 7.20 ขั้นตอนการจัดกลุ่มข้อมูลเชิงลำดับชั้น (Hierarchical clustering)

สำหรับวิธีการ UPGMA (Unweighted Pair Group Method with Arithmetic Mean) ที่ใช้ในการสร้างแผนภูมิวิวัฒนาการชาติพันธุ์ใช้ข้อมูลเข้าเป็นเมทริกซ์ระยะทางเช่นกัน โดยคำนวณค่าระยะทางระหว่างสองคลัสเตอร์ดังสมการต่อไปนี้

$$D_{avg}(C_1, C_2) = \frac{\sum_{all\ points\ i\ in\ cluster\ C_1} \sum_{all\ points\ j\ in\ cluster\ C_2} D_{i,j}}{|C_1| \cdot |C_2|}$$

สำหรับรูปที่ 7.21 เส้นแนวขวางที่ตัดผ่านต้นไม้กลุ่มข้อมูล i กิ่งจะแบ่ง n ยีนในต้นไม้ออกเป็น i กลุ่ม



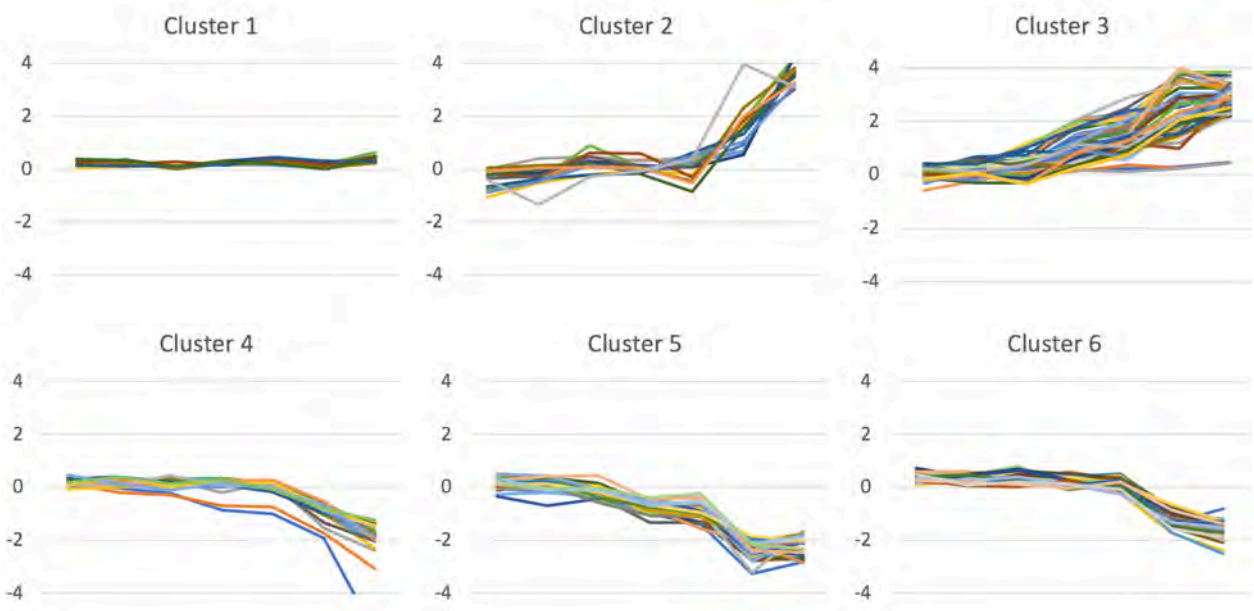
รูปที่ 7.21 (ก) การตัดผ่าน 3 กิ่งของต้นไม้กลุ่มข้อมูลทำให้แบ่งกลุ่มของยีนออกเป็น 3 กลุ่ม (ข) การตัดกิ่งลึกลงมาผ่าน 5 กิ่งทำให้แบ่งกลุ่มของยีนออกเป็น 5 กลุ่ม

การวิเคราะห์ diauxic shift จากผลการจัดกลุ่มยีนเชิงลำดับชั้น

รูปที่ 7.22 แสดงผลการจัดกลุ่มยีนโดยใช้วิธีการจัดกลุ่มเชิงลำดับชั้น (hierarchical clustering) เมื่อเปรียบเทียบผลของการจัดกลุ่มด้วยวิธีการนี้กับการจัดกลุ่มโดยอัลกอริทึม Lloyd ในรูปที่ 7.13 พบว่ามีความแตกต่างกัน

หยุดคิด	ควรกังวลมากน้อยแค่ไหนกับผลการจัดกลุ่มที่แตกต่างกันระหว่างการจัดกลุ่มเชิงลำดับชั้นกับการจัดกลุ่มโดยอัลกอริทึม Lloyd

ฝึกหัด	จงเขียนโค้ดเพื่อแสดงการจัดกลุ่มเชิงลำดับชั้น (hierarchical clustering) โดยใช้ D_{min} และใช้โค้ดนี้เพื่อจัดกลุ่ม 196 ยีนในยีสต์ออกเป็น 6 กลุ่ม ผลของการจัดกลุ่มมีความแตกต่างจาก 6 กลุ่มในรูปที่ 7.22 อย่างไร



รูปที่ 7.22 ผลการใช้การจัดกลุ่มเชิงลำดับชั้นในการจัดกลุ่มยีน 196 ยีนของยีสต์ออกเป็น 6 กลุ่ม

นักชีววิทยาไม่กังวลกับผลการจัดกลุ่มเมื่อใช้วิธีการจัดกลุ่มที่แตกต่างกัน เนื่องจากผลการจัดกลุ่มนี้มักเป็นเพียงจุดเริ่มต้นของการศึกษาเพิ่มเติมในมิติต่างๆ ทางชีววิทยา โดยต้องมีการทดลองเพิ่มเติมในห้องปฏิบัติการเพื่อยืนยันผลการจัดกลุ่มว่าชุดของยีนในแต่ละกลุ่มหรือบางกลุ่มที่เป็นที่สนใจนั้นมีความสัมพันธ์หรือเกี่ยวข้องกันในเชิงชีววิทยาจริง ตัวอย่างเช่น แต่ละคลัสเตอร์ในรูปที่ 7.22 แสดงผลการจัดกลุ่มเชิงลำดับชั้นของ 196 ยีนในยีสต์ออกเป็น 6 กลุ่ม ซึ่งประกอบด้วย 12, 15, 70, 18, 30 และ 51 ยีนตามลำดับ ทั้งนี้สามารถวิเคราะห์เพิ่มเติมเพื่อหายีนกลุ่มย่อยภายในคลัสเตอร์ซึ่งอาจมีรูปแบบระดับการแสดงออกของยีนที่ชัดเจนมากขึ้นหรือเฉพาะกลุ่มมากขึ้น เช่นคลัสเตอร์ที่ 2 มี 7 ยีนที่มีการเปลี่ยนแปลงระดับการแสดงออกไม่มากนักใน 6 ช่วงเวลาแรกแต่กลับมีระดับการแสดงออกเพิ่มขึ้นชัดเจนในช่วงเวลาสุดท้ายของการทดลอง ซึ่งนักชีววิทยาค้นพบว่า 6 ใน 7 ยีนในกลุ่มนี้มีโมทิฟที่เรียกว่า carbon source response element (CSRE) ซึ่งมีรูปแบบของลำดับเบสเป็น **CATTCATCCG** ในสายดีเอ็นเอส่วนหน้าของยีน และเมื่อทำการค้นหาโมทิฟนี้ในดีเอ็นเอส่วนหน้าของยีนของยีสต์ทั้งจีโนม พบว่ามี

อีกเพียง 4 ยีนที่ลำดับเบสส่วนหน้ามีโมทิฟนี้เช่นกัน ซึ่งเป็นข้อสนับสนุนว่าการจัดกลุ่มย่อยในคลัสเตอร์ที่ 2 โดยมี 6 ยีนนี้เป็นสมาชิกในกลุ่มย่อยนั้นเป็นความคิดที่ดี ทั้งนี้ยีสต์ชอบใช้กลูโคสเป็นแหล่งพลังงานมากกว่าเอทานอล ดังนั้นในกรณีที่มีกลูโคสอยู่ ชุดของยีนที่รับผิดชอบต่อกระบวนการเมแทบอไลต์ของสารอื่น เช่น เอทานอล จะถูกปิดสวิตช์หรือถูกกดไว้ไม่ให้แสดงออกมาเป็นเอ็มอาร์เอ็นเอเพื่อทำงาน นักวิจัยสรุปว่าโมทิฟ CSRE ใช้วิธีการบางอย่างในการส่งสัญญาณให้ยีสต์ทราบว่ามีกลูโคสและจะ *เปิด* การทำงานของ 6 ยีนนี้ก็ต่อเมื่อยีสต์ไม่มีกลูโคสเป็นแหล่งอาหาร ดังนั้นทั้งโมทิฟ CSRE และ 6 ยีนเป็นส่วนประกอบที่สำคัญของกระบวนการ diauxic shift

ถึงจุดนี้หลายคนอาจคิดว่าได้ศึกษาวิธีการจัดกลุ่มข้อมูลครอบคลุมแล้ว อย่างไรก็ตามถ้ากลับไปพิจารณาลักษณะข้อมูลในรูปที่ 7.14 และ 7.15 จะเห็นว่ายังไม่มีอัลกอริทึมใดในบทเรียนนี้สามารถจัดกลุ่มข้อมูลดังกล่าวได้ถูกต้อง

การจัดกลุ่มผู้ป่วยโรคมะเร็ง

มีการนำการวัดระดับการแสดงออกของยีนไปประยุกต์ใช้ในการตอบโจทย์ทางชีววิทยาที่หลากหลาย รวมทั้งการศึกษาเกี่ยวกับโรคมะเร็ง ในปี ค.ศ. 1999 ยูรี อาลอน (Uri Alon) และคณะได้ทำการวิเคราะห์ข้อมูลระดับการแสดงออกของ 2,000 ยีน จากเนื้อเยื่อมะเร็งลำไส้ 40 ตัวอย่าง และทำการเปรียบเทียบกับข้อมูลระดับการแสดงออกของยีนจากเนื้อเยื่อลำไส้ปกติ 20 ตัวอย่าง ชุดข้อมูลของอาลอนสามารถแสดงในรูปแบบเมทริกซ์การแสดงออกของยีนขนาด 2,000 x 60 โดยที่ 40 คอลัมน์แรกเป็นข้อมูลจากเนื้อเยื่อมะเร็งในขณะที่ 20 คอลัมน์หลังเป็นข้อมูลจากเนื้อเยื่อปกติ ถ้ามีการวัดการแสดงออกของยีนของผู้ป่วยใหม่และนำมาเป็นข้อมูลคอลัมน์ที่ 61 เป้าหมายคือการทำนายว่าผู้ป่วยใหม่นี้เป็นมะเร็งลำไส้หรือไม่ เนื่องจากเราทราบประเภทของเนื้อเยื่อ (มะเร็ง และปกติ) อยู่แล้ว การจำแนกว่าตัวอย่างเนื้อเยื่อของผู้ป่วยใหม่เป็นเนื้อเยื่อมะเร็งหรือเนื้อเยื่อปกติดูจะเป็นเรื่องง่าย ทั้งนี้ข้อมูลผู้ป่วยแต่ละคนคือหนึ่งจุดในปริภูมิ 2,000 มิติ โดยสามารถคำนวณค่าจุดศูนย์ถ่วงของชุดข้อมูลที่เป็นเนื้อเยื่อมะเร็งและเนื้อเยื่อปกติได้และสามารถตรวจสอบต่อไปได้ว่าจุดตัวอย่างใหม่อยู่ใกล้จุดศูนย์ถ่วงไหนมากกว่ากัน

อีกทางเลือกหนึ่งที่เป็นไปได้คือทำการวิเคราะห์ข้อมูลเสมือนว่าไม่มีความรู้เรื่องกลุ่มของเนื้อเยื่อมาก่อน โดยทำการจัดกลุ่มข้อมูลในเมทริกซ์ 2,000 x 61 โดยกำหนดค่า k = 2 ถ้ากลุ่มข้อมูลใดมีเนื้อเยื่อมะเร็งเป็นส่วนใหญ่ กลุ่มข้อมูลนั้นอาจนำมาช่วยในการวินิจฉัยมะเร็งลำไส้ได้

ปัญหาท้าทาย	วิธีการข้างต้นตรงไปตรงมาในการระบุเนื้อเยื่อของผู้ป่วยใหม่ว่าเป็นมะเร็งลำไส้หรือไม่ อย่างไรก็ตามทั้งสองวิธียังมีความน่าเชื่อถือไม่เพียงพอในการวินิจฉัยผู้ป่วยใหม่ คำถามคือเพราะอะไร จากข้อมูลเมทริกซ์การแสดงออกของยีนขนาด 2,000 x 60 ของอาลอนและข้อมูลยีนของผู้ป่วยใหม่ จงเสนอวิธีการประเมินว่าผู้ป่วยคนนี้มีโอกาสเป็นมะเร็งลำไส้หรือไม่

อาร์เอ็นเอซีค

การวัดการแสดงออกของยีนคือการวัดปริมาณเอ็มอาร์เอ็นเอที่ถูกถอดรหัส (transcribed) มาจากยีนในระดับดีเอ็นเอที่เป็นส่วนหนึ่งของจีโนม การศึกษาการแสดงออกของยีนทั้งจีโนมเรียกว่าทรานสคริปโทมิกส์ (transcriptomics) และผลการแสดงออกของยีนทั้งจีโนมเรียกว่าทรานสคริปโทม (transcriptome) ปัจจุบันอาร์เอ็นเอซีค (RNA-seq) เป็นเทคโนโลยีหลักที่ใช้วัดปริมาณอาร์เอ็นเอที่แสดงออกจากทั้งยีนที่สามารถแปลรหัสไปเป็นโปรตีน (protein-coding gene) และยีนที่สามารถถอดรหัสเป็นอาร์เอ็นเอไม่กำหนดรหัส (noncoding RNA) ซึ่งไม่แปลรหัสต่อไปเป็นโปรตีน อาร์เอ็นเอซีคกลายเป็นเทคโนโลยีหลักที่ใช้ในการวัดการแสดงออกของยีนเนื่องจากเทคโนโลยีอาร์เอ็นเอซีคอยู่บนพื้นฐานเดียวกับเทคโนโลยีการหาลำดับเบสยุคใหม่ (next generation sequencing) หรือเอ็นจีเอส (NGS) โดยอาร์เอ็นเอที่แสดงออกจะถูกแปลงให้เป็นซีดีเอ็นเอ (cDNA) หรือดีเอ็นเอคู่สม (complementary DNA) จากนั้นถูกทำให้เป็นเส้นสั้น (short read) และทำการถอดรหัสจากเครื่องเอ็นจีเอส ในลักษณะเดียวกับการหาลำดับเบสจีโนม ซึ่งไม่ต้องมีการเตรียมอาร์เรย์ชิปและออกแบบโพรบ (probe) เพื่อให้สามารถจับกับซีดีเอ็นเอของแต่ละยีนที่สนใจวัดการแสดงออก นอกจากนี้ข้อมูลเอ็นเอซีคยังสามารถนำไปใช้วิเคราะห์การเกิด spliced site และไอโซฟอร์ม (isoform) แบบต่างๆ ที่เป็นไปได้ที่ถูกถอดรหัสมาจากยีนเดียวกัน วิธีการวิเคราะห์ข้อมูลเอ็นเอซีคพื้นฐานที่มีการอ้างอิงแพร่หลายตีพิมพ์ใน [189] ทั้งนี้ในการจัดกลุ่มข้อมูลเอ็นเอซีคยังไม่มีหลักเกณฑ์แน่นอน ยาซโคเวียค พี.เอ. (Jaskowiak P.A.) และคณะ ทำการเปรียบเทียบวิธีการจัดกลุ่มและฟังก์ชันวัดระยะทางที่เหมาะสมไว้ใน [190]

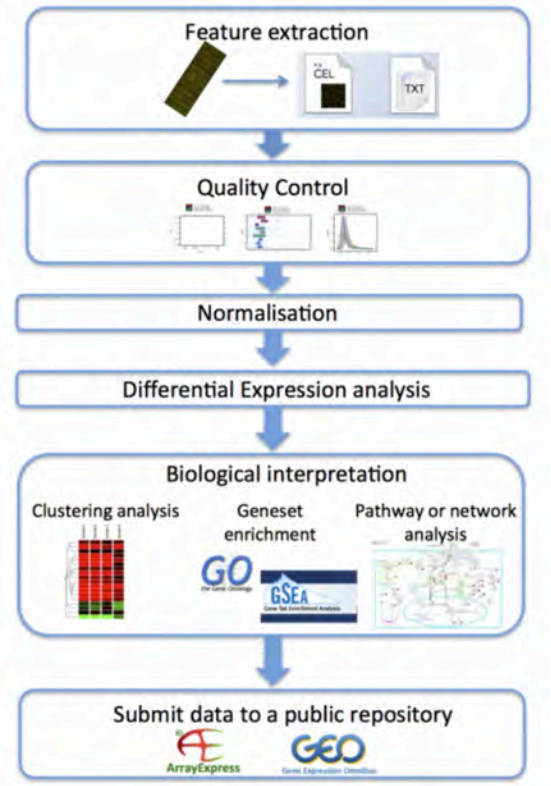
บทส่งท้าย

ปัญหาการจัดกลุ่มข้อมูลหรือการแบ่งกลุ่มข้อมูลโดยวิธีการทางคอมพิวเตอร์เป็นกลุ่มปัญหาสำคัญกลุ่มหนึ่งของการวิจัยและพัฒนาการเรียนรู้ของเครื่อง (machine learning: ML) ภายใต้หัวข้อการเรียนรู้ของเครื่องแบบไม่มีผู้สอน (unsupervised learning) การออกแบบและพัฒนาวิธีการจัดกลุ่มข้อมูลนอกเหนือจากที่มีการศึกษาในบทเรียนนี้ ในเชิงวิจัยมีการนำเสนอวิธีการในการจัดกลุ่มข้อมูลอีกหลากหลายวิธีเช่น การนำ nonnegative matrix factorization (NMF) และวิธีการอื่นที่เป็นส่วนขยายมาประยุกต์ใช้ในการจัดกลุ่มข้อมูลการแสดงออกของยีน [191] ข้อมูลการแสดงออกของยีนและข้อมูลจุลชีวนิเวศหรือไมโครไบโอม (microbiome) [192] ข้อมูลการเกิดปฏิสัมพันธ์ระหว่างอาร์เอ็นเอและโปรตีนที่มาจับผ่านข้อมูลการหาลำดับเบสแบบคลิป (CLIP-seq) โดยเป็นการจัดกลุ่มข้อมูลแบบ soft clustering [193] นอกจากนี้ด้วยข้อมูลทางชีวสารสนเทศที่เกิดจากเทคโนโลยีใหม่มีมิติของข้อมูลจำนวนมาก วิธีการจัดกลุ่มข้อมูลเพื่อรองรับข้อมูลเหล่านี้ยังเป็นที่ต้องการและเป็นปัญหาที่ท้าทาย เช่น การจัดกลุ่มข้อมูลอาร์เอ็นเอซีคจากเซลล์เดี่ยว (single-cell RNA-seq clustering) [194-198] และการจัดกลุ่มข้อมูลแมสไซโตเมทรี (mass cytometry) จากเซลล์เดี่ยว [199, 200] เป็นต้น ซึ่งมีตัวอย่างผลงานวิจัยและพัฒนาเช่น การประยุกต์ใช้ nonnegative matrix factorization ในการทำ coupled clustering กับข้อมูลอาร์เอ็นเอซีคของ

เซลล์เดี่ยวและข้อมูล ATAC-seq [201] การประยุกต์ใช้ nonnegative matrix factorization ในการจัดกลุ่มข้อมูลอาร์เอ็นเอซีคในเซลล์เดี่ยว [202, 203] และที่นำเสนอเป็นไลบรารีภาษา R ชื่อ ccfindR ซึ่งเป็นส่วนหนึ่งของ Bioconductor (https://www.bioconductor.org/packages/release/bioc/html/ccfindR.html) เป็นต้น

ตัวอย่างโปรแกรมที่มีการใช้งานอย่างแพร่หลาย

ขั้นตอนพื้นฐานในการวิเคราะห์ข้อมูลไมโครอาร์เรย์ (รูปที่ 7.23) ประกอบด้วย (1) การสกัดคุณลักษณะ (extract feature) (2) การตรวจสอบและควบคุมคุณภาพของข้อมูล (quality control) (3) การปรับค่าข้อมูลให้เป็นมาตรฐาน (normalization) (4) การวิเคราะห์ระดับการแสดงออกของยีน (differential expression analysis) (เป็นเนื้อหาหลักในบทเรียนนี้ โดยบทเรียนนี้เน้นวิธีการเชิงอัลกอริทึมในการจัดกลุ่มยีน) และ (5) การแปลความหมายหรือการตีความผลในเชิงชีววิทยา โดยขั้นตอนเหล่านี้มีตัวอย่างซอฟต์แวร์หรือไลบรารีที่สามารถใช้งานได้แตกต่างกันไป



รูปที่ 7.23 ผังงานมาตรฐานในการวิเคราะห์ข้อมูลไมโครอาร์เรย์

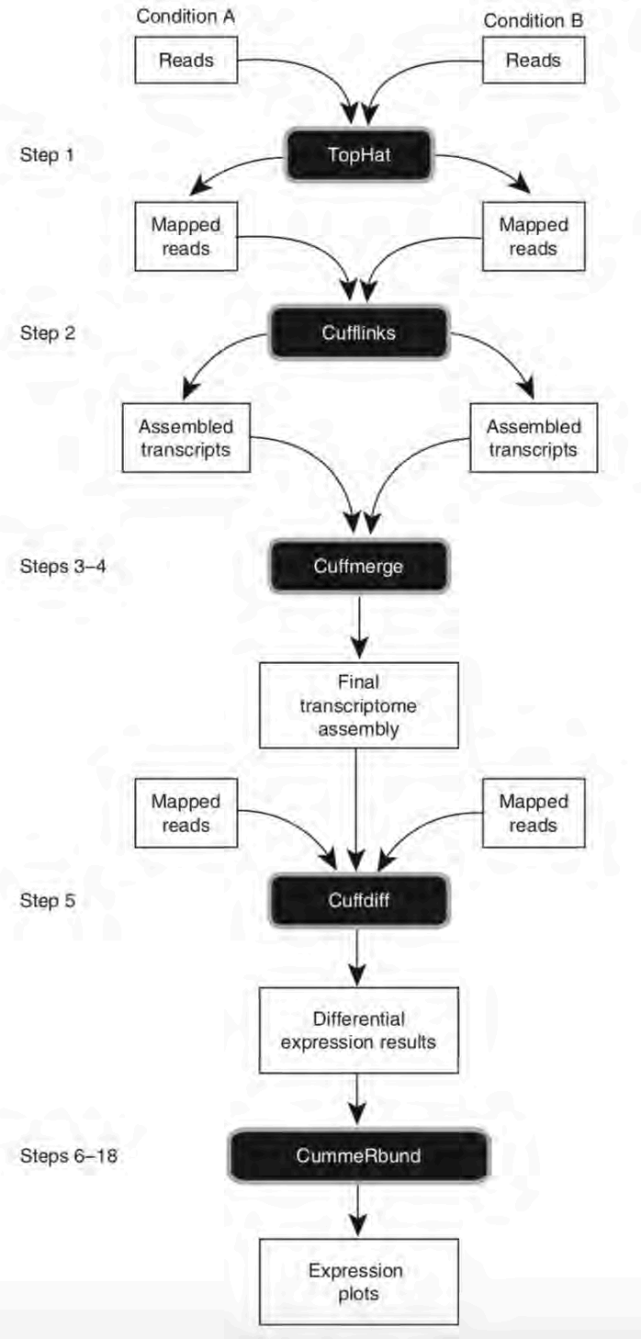
(*ที่มา:* EMBL-EBI Training. *Functional genomics II, Analysis of microarray data*. [ONLINE] Available at: https://www.ebi.ac.uk/training [เข้าถึงออนไลน์เมื่อวันที่ 14 ก.ค. พ.ศ. 2564])

การปรับค่าข้อมูลให้เป็นมาตรฐาน (normalization) สามารถใช้วิธีการ rma (Robust Multi-array Average) [204-206] ในชุดโปรแกรมภาษา R ชื่อ oligo ส่วนการวิเคราะห์ระดับการแสดงออกของยีนที่แตกต่างกันสามารถใช้ชุดโปรแกรมภาษา R ชื่อ limma [207] เป็นต้น ตัวอย่างซอฟต์แวร์และไลบรารีอื่นๆ ทั้งแบบที่เปิดให้ดาวน์โหลดโดยสาธารณะและแบบที่เป็นเชิงพาณิชย์สามารถศึกษาเพิ่มเติมได้จาก [208]

จาก [189] โปรแกรมหลักที่ใช้ในการวิเคราะห์ข้อมูลอาร์เอ็นเอซีคประกอบด้วย TopHat [209] ซึ่งใช้ Bowtie [79] ในการเทียบซีดีเอ็นเอ (cDNA) สายสั้นกับจีโนมอ้างอิง จากนั้น TopHat จะพยายามหา splice site ที่เป็นไปได้ระหว่างเอกซอนที่อยู่ติดกัน สำหรับรีดที่ไม่สามารถแมพ (map) ได้โดย Bowtie จะถูกนำมาเทียบอีกครั้งโดย TopHat เมื่อได้ผลการแมพซีดีเอ็นเอกับจีโนมอ้างอิงแล้ว ใช้โปรแกรม Cufflinks [210] ในการประกอบร่างทรานสคริปต์ (transcripts) (หรือซีดีเอ็นเอที่แปลงมาจากอาร์เอ็นเอและหาลำดับเบสโดยเครื่องเอ็นจีเอส) และใช้โปรแกรม Cuffcompare ในการเปรียบเทียบทรานสคริปต์ที่ประกอบร่างแล้วกับยีนในจีโนม ใช้โปรแกรม Cuffmerge ในการรวมทรานสคริปต์ที่ประกอบร่างแล้วเข้าด้วยกัน และใช้โปรแกรม Cuffdiff ในการหาชุดของยีนที่มีระดับการแสดงออกของอาร์เอ็นเอที่แตกต่างกัน รายละเอียดของชุดโปรแกรม Cufflinks สามารถศึกษาเพิ่มเติมได้ที่ http://cole-trapnell-lab.github.io/cufflinks/ รูปที่ 7.24 แสดงขั้นตอนการวิเคราะห์ข้อมูลอาร์เอ็นเอซีคเพื่อเปรียบเทียบผลระหว่าง 2 เงื่อนไข [189]

แบบฝึกหัดบทที่ 7

1. เขียนโปรแกรมเพื่อแก้ปัญหาที่เกี่ยวข้องกับการจัดกลุ่มข้อมูลโดยใช้โจทย์ที่โรซาลินด์ต่อไปนี้
   1) Implement the Lloyd Algorithm for K-Means Clustering (http://rosalind.info/problems/ba8c/)
   2) Implement the Soft K-Means Clustering Algorithm (http://rosalind.info/problems/ba8d/)
   3) Implement Hierarchical Clustering (http://rosalind.info/problems/ba8e/)
2. ศึกษาวิธีการวิเคราะห์ข้อมูลไมโครอาร์เรย์เพิ่มเติมจาก EMBL-EBI ออนไลน์คอร์สที่ https://www.ebi.ac.uk/training/online/course/functional-genomics-ii-common-technologies-and-data-analysis-methods/analysis-microarray-data
3. ศึกษาวิธีการพื้นฐานเกี่ยวกับการหาลำดับเบสของอาร์เอ็นเอทั้งหมด (RNA sequencing) จาก EMBL-EBI ออนไลน์คอร์สที่ https://www.ebi.ac.uk/training/online/course/functional-genomics-ii-common-technologies-and-data-analysis-methods/rna-sequencing



รูปที่ 7.24 โปรโตคอลที่นำเสนอใน [162] เพื่อใช้วิเคราะห์ข้อมูลอาร์เอ็นเอซีคจากเงื่อนไขการทดลอง 2 เงื่อนไข (*ที่มา*: *รูปที่* 2 *ของ* [189])

บทที่ 8 การวิเคราะห์การแสดงออกของโปรตีน (Protein expression analysis)

วัตถุประสงค์

- เพื่อให้นิสิตเห็นตัวอย่างการประยุกต์ใช้เทคโนโลยีแมสสเปกโทรเมตรีกับงานวิจัยทางชีววิทยาและชีวการแพทย์
- เพื่อให้นิสิตคุ้นเคยกับลักษณะข้อมูลของแมสสเปกโทรเมตรี อัลกอริทึมพื้นฐานที่ใช้ในการหาลำดับกรดแอมิโนสายสั้นหรือเพปไทด์ รวมทั้งตัวอย่างวิธีการทางสถิติที่ใช้ในการประเมินความสำคัญของเพปไทด์ที่เป็นผลลัพธ์ของการสืบค้นฐานข้อมูลโปรตีโอมโดยใช้ข้อมูลสเปกตรัม
- เพื่อให้นิสิตเห็นตัวอย่างงานวิจัยและผลงานวิจัยที่ใช้ในการอนุมานหรือหาลำดับกรดแอมิโนของสายเพปไทด์จากข้อมูลสเปกตรัม รวมทั้งการระบุสายเพปไทด์โดยเทียบกับฐานข้อมูลโปรตีโอม
- เพื่อให้นิสิตเห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทาย รวมทั้งงานวิจัยอื่นๆ ที่เกี่ยวข้อง

ผลลัพธ์ที่คาดหวัง

- นิสิตเข้าใจลักษณะการทำงานพื้นฐานของเครื่องแมสสเปกโทรเมตรี และการประยุกต์ใช้ในงานทางด้านโพรทีโอมิกส์ (proteomics) เพื่อการวิจัยทางชีววิทยาและชีวการแพทย์
- นิสิตเข้าใจคุณลักษณะของข้อมูลแมสสเปกโทรเมตรี
- นิสิตสามารถอธิบายการทำงานของอัลกอริทึมพื้นฐานที่ใช้ในการหาลำดับกรดแอมิโนในสายเพปไทด์จากข้อมูลสเปกตรัมของเครื่องแมสสเปกโทรเมตรี รวมทั้งสามารถแสดงการคำนวณการประเมินนัยสำคัญทางสถิติของชุดสายเพปไทด์ที่เป็นผลจากการสืบค้นฐานข้อมูลโปรตีโอมโดยใช้ลำดับกรดแอมิโนจากการหาลำดับ
- นิสิตสามารถเขียนโปรแกรมที่ใช้ในการหาลำดับกรดแอมิโนอย่างง่ายได้
- นิสิตสามารถยกตัวอย่างโปรแกรมที่ใช้ในการหาลำดับกรดแอมิโนจากข้อมูลสเปกตรัมที่มีการใช้งานกันอย่างแพร่หลายได้
- นิสิตสามารถยกตัวอย่างความท้าทายที่ยังมีอยู่และสามารถนำเสนอแนวทางในการพัฒนาวิธีการแก้ปัญหาเหล่านี้ รวมทั้งสามารถประยุกต์องค์ความรู้จากบทเรียนเพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้

เนื้อหาโดยสรุป

งานวิจัยทางด้านโพรทีโอมิกส์เน้นการศึกษาการแสดงออกของโปรตีนจำนวนมากในเงื่อนไขที่แตกต่างกันของเนื้อเยื่อหนึ่งๆ โดยมีการประยุกต์ใช้เทคนิคแมสสเปกโทรเมตรีในการศึกษาการแสดงออกของโปรตีนอย่างต่อเนื่อง ข้อมูลที่เป็นผลลัพธ์จากเครื่องแมสสเปกโทรมิเตอร์อยู่ในรูปแบบสเปกตรัมที่แสดงค่าน้ำหนักชิ้นส่วนย่อยของโปรตีนที่เรียกว่าเพปไทด์ งานทางด้านอัลกอริทึมมีเป้าหมายเพื่อหาลำดับกรดแอมิโนที่ประกอบเป็นโปรตีนจากข้อมูลสเปกตรัมเหล่านี้ หรืออีกนัยหนึ่งคือหาลำดับกรดแอมิโนของโปรตีนที่มีโอกาสทำให้เกิดข้อมูลสเปกตรัมเหล่านี้ การประยุกต์ใช้ทฤษฎีกราฟเป็นวิธีการพื้นฐานในการอนุมานลำดับกรดแอมิโน โดยแต่ละโหนดแสดงค่าน้ำหนักของสายเพปไทด์ เส้นเชื่อมระหว่างโหนดแสดงค่าน้ำหนักที่แตกต่างกันระหว่าง 2 โหนดซึ่งเท่ากับค่าน้ำหนักของกรดแอมิโนใดแอมิโนหนึ่ง และเส้นทางที่เชื่อมทุกโหนดเข้าด้วยกันแสดงลำดับกรดแอมิโนที่อนุมานได้ ลำดับกรดแอมิโนของโปรตีนที่อนุมานได้นี้มักมีจำนวนมาก วิธีการพื้นฐานที่ใช้คัดเลือกลำดับกรดแอมิโนที่มีความสำคัญคือการนำไปสืบค้นกับฐานข้อมูลโปรตีโอมที่มีอยู่ อย่างไรก็ตามสิ่งที่ต้องคำนึงถึงคือลำดับกรดแอมิโนเหล่านี้มีนัยสำคัญเชิงสถิติมากน้อยเพียงใด บทเรียนนี้เราจะศึกษาหัวข้อเหล่านี้ รวมทั้งตัวอย่างโปรแกรมที่มีการใช้งานอย่างแพร่หลาย งานวิจัยที่เกี่ยวข้อง และความท้าทายที่มีอยู่

บทที่ 8 การวิเคราะห์การแสดงออกของโปรตีน (Protein expression analysis)

เมื่อบรรพชีวินวิทยาพบกับการคำนวณ

แจ็ค ฮอนเนอร์ (Jack Horner) เกิดที่รัฐมอนตานาในปี ค.ศ. 1946 และเติบโตที่นั่น เขาเป็นเด็กขี้อาย ค่อนข้างเก็บตัวและมีพัฒนาการในการอ่านและการคำนวณทางคณิตศาสตร์ช้ากว่าเด็กคนอื่นในวัยเดียวกัน อย่างไรก็ตามโครงงานในสมัยมัธยมเกี่ยวกับไดโนเสาร์ของเขาเป็นหนึ่งในโครงงานที่ได้รับรางวัลในงานประกวดโครงงานวิทยาศาสตร์ในพื้นที่ และได้รับความสนใจจากศาสตราจารย์ที่มหาวิทยาลัยมอนตานา ซึ่งเป็นผู้ช่วยให้ฮอนเนอร์ได้รับคัดเลือกมาศึกษาต่อที่มหาวิทยาลัย

รูปที่ 8.1 แจ็ค ฮอนเนอร์ในปี ค.ศ. 2015

(*ที่มา:* Jonathunder, Public domain, via Wikimedia Commons. 2015. *Paleontologist Jack Horner in Rochester, Minnesota, United States.* [*เข้าถึงออนไลน์เมื่อวันที่* 11 *ก.ค. พ.ศ.* 2564])

ฮอนเนอร์ทำคะแนนได้ไม่ดีและสอบไม่ผ่านติดกัน 5 ควอเทอร์ทำให้เขาต้องออกจากการเรียน หลายปีต่อมาจึงทราบว่าอาการของฮอนเนอร์นั้น เกิดจากภาวะบกพร่องในการอ่านและเขียนหรือดีย์สเลกเซีย (dyslexia) ทำให้ไม่สามารถอ่านได้รู้ความแม้ผู้ป่วยจะมีระดับสติปัญญาปกติหรือสูงกว่าปกติก็ตาม หลังถูกเกณฑ์ไปเป็นทหารในสงครามเวียดนามและทำงานเป็นคนขับรถบรรทุก เขาได้งานใหม่เป็นเจ้าหน้าที่พิพิธภัณฑ์ประวัติศาสตร์ธรรมชาติแห่งมหาวิทยาลัยพรินซ์ตัน (Princeton's Natural History Museum) ที่นี่เป็นที่ที่เขาได้รับชื่อเสียงและการยอมรับจากเพื่อนร่วมงานว่าเป็นนักวิจัยที่มีความหลักแหลมอย่างมากและกลายมาเป็นนักบรรพชีวินวิทยา (paleontologist) ที่มีชื่อเสียงเป็นที่รู้จักระดับโลก เป็นผู้สร้างแรงบันดาลใจให้กับตัวเอกในนวนิยายที่มีชื่อเสียงเรื่องจูแรสซิกพาร์ค (Jurassic Park) และเป็นที่ปรึกษาให้กับผู้กำกับภาพยนตร์ชื่อดัง สตีเว่น สปีลเบิร์ก (Steven Spielberg) ในการสร้างภาพยนตร์เรื่องดังกล่าว

ฮอนเนอร์ประสบความสำเร็จอย่างมากถึงแม้จะมีภาวะดีย์สเลกเซีย ส่วนหนึ่งเป็นเพราะงานด้านบรรพชีวินวิทยาไม่ต้องการการคำนวณทางคณิตศาสตร์มากมาย อย่างไรก็ตามลูกศิษย์ของฮอนเนอร์แสดงให้เห็นว่างานด้าน

บรรพชีวินวิทยานั้นมีการใช้คณิตศาสตร์เช่นกัน ในปี ค.ศ. 2000 ฮอนเนอร์ค้นพบสุสานไดโนเสาร์สายพันธุ์ที่เขาสนใจอย่างมากในรัฐมอนตานาและขุดพบฟอสซิลกระดูกขาของทีเร็กซ์ (*Tyrannosaurus rex*) ที่มีอายุถึง 68 ล้านปี สามปีหลังจากนั้น ฮอนเนอร์มอบส่วนของฟอสซิลให้กับลูกศิษย์เขาชื่อ แมรี่ ชไวท์เซอร์ (Mary Schweitzer) ซึ่งทำการสลายกระดูกเพื่อศึกษาองค์ประกอบ แต่เนื่องจากแช่ใน demineralizing bath นานเกินไป จึงเหลือเพียงส่วนที่เป็นเนื้อเยื่อเส้นใยเท่านั้น เธอส่งส่วนที่เหลือเหล่านี้ไปให้จอห์น แอสรา (John Asara) ซึ่งเป็นผู้เชี่ยวชาญทางด้านแมสสเปกโทรเมตรี (mass spectrometry) โดยหวังว่าจะตรวจพบเพปไทด์หรือส่วนของโปรตีนสายสั้นของทีเร็กซ์ (*T. rex*) ซึ่งอาจหลงเหลืออยู่ในกระดูก

ในปี ค.ศ. 2007 หลังจากวิเคราะห์ข้อมูลสเปกตรัมหลายพันข้อมูล แอสราและชไวท์เซอร์ตีพิมพ์ผลงานวิจัยในวารสาร *Science* [211] โดยรายงานการค้นพบสายเพปไทด์ของทีเร็กซ์ที่มีความใกล้เคียงกับสายเพปไทด์ที่พบในไก่ (chicken) มาก ผลงานตีพิมพ์นี้เป็นผลงานแรกในเชิงอณูชีววิทยาที่สนับสนุนสมมติฐาน (ที่เป็นที่โต้เถียงกัน) ว่าสัตว์ปีกมีวิวัฒนาการมาจากไดโนเสาร์

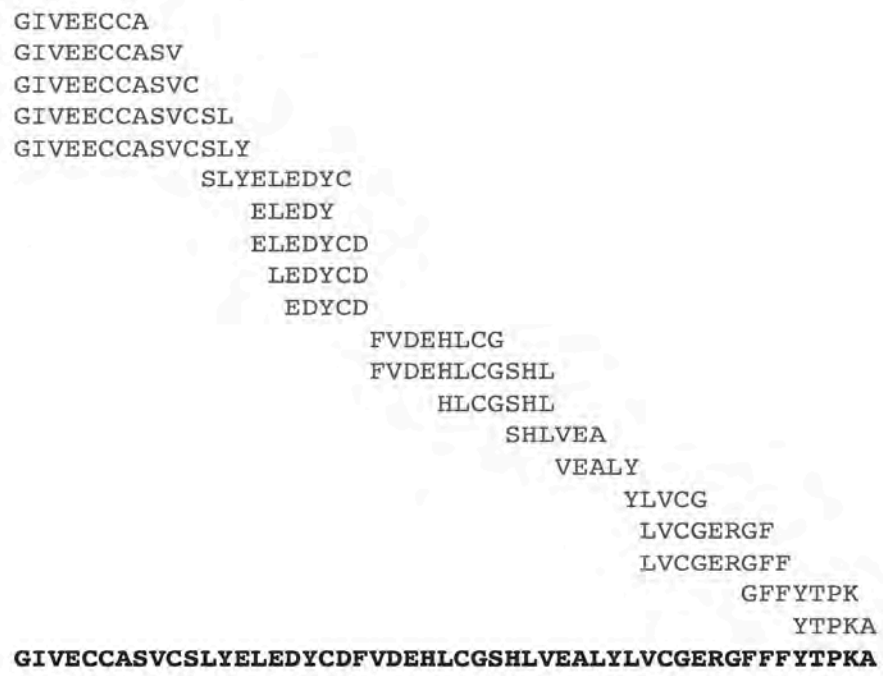
ความจริงที่ว่าโปรตีนสามารถคงอยู่เป็นเวลากว่าล้านปีเป็นเรื่องอัศจรรย์และนำไปสู่หัวข้อวิจัยต่างๆ ที่ยิ่งใหญ่ เช่น นักบรรพชีวินวิทยาฮานส์ ลาร์สสัน (Hans Larsson) เสนอว่าการศึกษาข้อมูลเชิงอณูชีววิทยาของไดโนเสาร์จะเป็นเส้นเชื่อมโยงงานทางบรรพชีวินวิทยากับอณูชีววิทยาและวิทยาศาสตร์สมัยใหม่ ในขณะที่ *The Guardian* คาดการณ์ว่า ในอนาคตนักวิทยาศาสตร์จะสามารถจำลองจูแรสซิกพาร์คได้โดยการโคลนไดโนเสาร์ ฮอนเนอร์เขียนหนังสือชื่อ “*How to Build a Dinosaur*” โดยมีรายละเอียดเกี่ยวกับแผนของเขาในการสร้างไดโนเสาร์จากการดัดแปลงพันธุกรรมของจีโนมไก่

อย่างไรก็ตามนักวิทยาศาสตร์ส่วนหนึ่งยังเคลือบแคลงกับสิ่งเหล่านี้ ในขณะที่การศึกษาในอดีตเกี่ยวกับไดโนเสาร์ไม่ต้องการการคำนวณมากนัก หลังการวิเคราะห์ข้อมูลเพปไทด์ของทีเร็กซ์โดยแอสรา ที่ใช้อัลกอริทึมบนพื้นฐานทางสถิติที่ซับซ้อน ในปี ค.ศ. 2008 มีการตีพิมพ์ผลงานวิจัยในวารสาร *Science* [212, 213] ออกมาโต้แย้งแอสราและชไวท์เซอร์ โดยระบุว่าสายเพปไทด์บางส่วนที่นำเสนอไว้ในปี ค.ศ. 2007 นั้นเป็นเพียงผลข้างเคียงที่เกิดขึ้นจากวิธีการทางสถิติ คำถามคือ จะทราบได้อย่างไรว่าฝ่ายไหนที่ถูกต้อง ในบทเรียนนี้จะพิจารณาว่าสายเพปไทด์ที่มีการโต้แย้งกันนั้นเป็นของทีเร็กซ์จริงหรือไม่โดยการศึกษาอัลกอริทึมที่เกี่ยวข้องกับการวิเคราะห์ข้อมูลสเปกตรัม

ตัวอย่างนี้มีโปรตีนอะไรบ้าง

มีนักวิทยาศาสตร์เพียง 4 คนที่เคยได้รับรางวัลโนเบล 2 ครั้ง หนึ่งในนั้นคือ เฟรดเดอริก แซงเกอร์ (Frederick Sanger) ที่นำเสนอการประกอบร่างจีโนมแรกในปี ค.ศ. 1977 โดยแซงเกอร์ได้รับรางวัลโนเบลครั้งแรกเมื่อ 20 ปีก่อนหน้าโดยนำเสนออินซูลินที่ประกอบด้วย 52 ลำดับกรดแอมิโน โดยอินซูลินเป็นโปรตีนที่จำเป็นต่อการดูดซึมกลูโคสในกระแสเลือด แซงเกอร์ทราบลำดับกรดแอมิโนของอินซูลินโดยใช้วิธีการที่คล้ายคลึงกับการหาลำดับเบส

จีโนมในปัจจุบัน แซงเกอร์ทำการแยกโมเลกุลของอินซูลินให้เป็นเพปไทด์ย่อยและทำการพิจารณาองค์ประกอบของกรดแอมิโนในสายเพปไทด์เหล่านั้น จากนั้นจึงต่อลำดับกรดแอมิโนสายสั้นเข้าด้วยกันดังแสดงในรูปที่ 8.2



รูปที่ 8.2 การประกอบร่างสายเพปไทด์ที่แฟรดเดอริก แซงเกอร์ ใช้ในการหาลำดับกรดแอมิโนของอินซูลิน
(*ที่มา*: *รูปที่* 11.1 *ของ [52]*)

ในช่วงคริสต์ทศวรรษ 1950 การหาลำดับกรดแอมิโนยังเป็นเรื่องที่ซับซ้อน ในขณะที่ยังไม่มีวิธีในการวิเคราะห์หาลำดับเบสของดีเอ็นเอ แต่ในปัจจุบันการหาลำดับเบสดีเอ็นเอผ่านเทคโนโลยีเอ็นจีเอสมีการใช้งานสะดวกและกว้างขวาง ในขณะที่การหาลำดับกรดแอมิโนในสายโปรตีนยังเป็นเรื่องยาก ด้วยเหตุนี้โปรตีนส่วนใหญ่มักถูกค้นพบจากการหาลำดับเบสจีโนม และทำนายยีนในจีโนมที่สามารถแปลรหัสต่อไปเป็นโปรตีน ซึ่งทำให้สามารถอนุมานโปรตีโอม (proteome) หรือชุดของโปรตีนทั้งหมดในเซลล์ของสิ่งมีชีวิตหนึ่งๆ ได้

อย่างไรก็ตามเซลล์แต่ละประเภทและในเงื่อนไขที่แตกต่างกันมีชุดของโปรตีนที่แสดงออกแตกต่างกัน (เช่นเดียวกับการแสดงออกของอาร์เอ็นเอ) ตัวอย่างเช่น โปรตีนที่แสดงออกในเซลล์สมองจะเป็นกลุ่มที่เพิ่มจำนวนนิวโรเพปไทด์ (neuropeptide) ในขณะที่เซลล์อื่นจะไม่มีการแสดงออกของโปรตีนกลุ่มนี้ การศึกษาโปรตีนที่แสดงออกในองค์รวมในเนื้อเยื่อและเงื่อนไขที่จำเพาะ เป็นสาขาหนึ่งในงานวิจัยเชิงโพรทีโอมิกส์ (proteomics) และเป็นแนวทางหนึ่งที่สำคัญในการศึกษากระบวนการต่างๆ ทางชีววิทยา รวมทั้งการวินิจฉัยและรักษาโรค ตัวอย่างเช่น การศึกษาไรโบโซมของไก่ ซึ่งไรโบโซมเป็นโมเลกุลที่มีความซับซ้อนประกอบด้วยหลายโปรตีน การทราบข้อมูลโปรตีโอมของไก่ไม่สามารถบอกได้ว่ามีโปรตีนอะไรบ้างที่ประกอบกันเป็นไรโบโซม ในทางกลับกัน เราสามารถแยกไรโบโซมออกมา ทำการแตกโมเลกุลไรโบโซมให้เป็นส่วนย่อย และตรวจสอบว่าประกอบด้วยโปรตีนอะไรบ้าง ในทาง

ปฏิบัติการตรวจพบเพปไทด์ขนาด 10 ลำดับกรดแอมิโนที่ทราบว่าเป็นส่วนของโปรตีนของไก่เพียงพอในการยืนยันว่ามีโปรตีนของไก่ประกอบอยู่ กระบวนการในการตรวจสอบว่ามีสายเพปไทด์จากโปรตีโอมที่ทราบข้อมูล อยู่ในตัวอย่างที่ทดสอบหรือไม่เรียกว่า peptide identification คำถามคือ จะทราบหรือสร้างโปรตีโอมของไดโนเสาร์ทีเร็กซ์ได้อย่างไร

ถึงแม้การศึกษาส่วนใหญ่ในระดับโพรทีโอมิกส์ในปัจจุบันเน้นการทำ peptide identification สิ่งมีชีวิตจำนวนมากรวมทั้งสิ่งมีชีวิตที่สูญพันธุ์ไปแล้ว เช่น ทีเร็กซ์ ไม่มีข้อมูลโปรตีโอม ในกรณีนี้นักชีววิทยาจำเป็นต้องอาศัยการทดลองแบบ *de novo* peptide sequencing หรือการหาลำดับกรดแอมิโนของสายเพปไทด์ใหม่ซึ่งเป็นเนื้อหาของบทเรียนนี้

การหาลำดับกรดแอมิโนจากสเปกตรัมในอุดมคติ

ถ้ามีสายเพปไทด์เดียวกันจำนวนมากในตัวอย่างทดสอบซึ่งมักประกอบด้วยหลายล้านเซลล์ เครื่องแมสสเปกโทรมิเตอร์จะทำการแตกสายเพปไทด์แต่ละเส้นเป็นสองส่วนที่สั้นลง ซึ่งเพปไทด์เดียวกันแต่ละเส้นอาจถูกแตกออกเป็นสองส่วนที่แตกต่างกันไป เช่น เพปไทด์ `PINKA` แรกอาจแตกเป็น `PI` และ `NKA` ในขณะที่ เพปไทด์ `PINKA` ที่สองอาจแตกเป็น `PIN` และ `KA` เป็นต้น ส่วนย่อยด้านหน้า `PI` และ `PIN` เรียกว่าพรีฟิกซ์ (prefix) ของ `PINKA` ในขณะที่ส่วนย่อยด้านหลัง `NKA` และ `KA` เรียกว่าซัฟฟิกซ์ของ `PINKA` รูปที่ 8.3 แสดงค่าน้ำหนักของกรดแอมิโนมาตรฐาน (เพื่อลดความซับซ้อนการอ้างถึงค่าน้ำหนักของกรดแอมิโนในเนื้อหาส่วนถัดไปจะเป็นเลขจำนวนเต็ม)

คำถามในเชิงอัลกอริทึมคำถามแรกคือจะหาลำดับกรดแอมิโนจากชุดค่าน้ำหนักพรีฟิกซ์และซัฟฟิกซ์ได้อย่างไร สเปกตรัมในอุดมคติของสายเพปไทด์แสดงโดย IDEALSPECTRUM(Peptide) ประกอบด้วยชุดค่าน้ำหนักของพรีฟิกซ์และซัฟฟิกซ์ที่เกิดขึ้นทั้งหมด ดังตัวอย่างในรูปที่ 8.4(ก) โดยค่าข้อมูลสเปกตรัมในอุดมคติอาจมีค่าซ้ำได้ เช่น IDEALSPECTRUM(GPG) = {0, 57, 57, 154, 154, 211} เป็นต้น และกล่าวได้ว่าลำดับกรดแอมิโนของเพปไทด์อธิบายชุดค่าตัวเลขสเปกตรัมถ้า IDEALSPECTRUM(Peptide) = Spectrum

นิยามปัญหาที่ 8.1 ปัญหาการหาลำดับกรดแอมิโนจากสเปกตรัมในอุดมคติ

สร้างสายเพปไทด์จากชุดข้อมูลสเปกตรัมในอุดมคติ	
ข้อมูลเข้า	ชุดค่าน้ำหนักพรีฟิกซ์และซัฟฟิกซ์ที่แสดงสเปกตรัมในอุดมคติ
ผลลัพธ์	ลำดับกรดแอมิโนของเพปไทด์ที่อธิบายสเปกตรัม

รูปที่ 8.4(ข) แสดงกราฟแบบมีทิศทางและไม่มีลูป (directed acyclic graph: DAG) ผ่านฟังก์ชัน `GRAPH(IDEALSPECTRUM(PINKA))` โดยค่าน้ำหนักแต่ละค่าเป็นโหนดและเส้นเชื่อมที่ชี้จากโหนด A ไปโหนด B แสดงส่วนต่างของค่าน้ำหนักระหว่างสองโหนดที่มีค่าเท่ากับค่าน้ำหนักของกรดแอมิโนใดแอมิโนหนึ่งและ

ใช้เป็นป้าย (label) กำกับเส้นเชื่อม โดยกรดแอมิโนจะเรียงลำดับตามป้ายที่กำกับแต่ละเส้นเชื่อม

Name	3-letter code	1-letter code	Residue Mass	Immonium ion	Related ions	Composition
Alanine	Ala	A	71.03711	44		C_3H_5NO
Arginine	Arg	R	156.10111	129	59,70,73,87,100,112	$C_6H_{12}N_4O$
Asparagine	Asn	N	114.04293	87	70	$C_4H_6N_2O_2$
Aspartic Acid	Asp	D	115.02694	88	70	$C_4H_5NO_3$
Cysteine	Cys	C	103.00919	76		C_3H_5NOS
Glutamic Acid	Glu	E	129.04259	102		$C_5H_7NO_3$
Glutamine	Gln	Q	128.05858	101	56,84,129	$C_5H_8N_2O_2$
Glycine	Gly	G	57.02146	30		C_2H_3NO
Histidine	His	H	137.05891	110	82,121,123,138,166	$C_6H_7N_3O$
Isoleucine	Ile	I	113.08406	86	44,72	$C_6H_{11}NO$
Leucine	Leu	L	113.08406	86	44,72	$C_6H_{11}NO$
Lysine	Lys	K	128.09496	101	70,84,112,129	$C_6H_{12}N_2O$
Methionine	Met	M	131.04049	104	61	C_5H_9NOS
Phenyalanine	Phe	F	147.06841	120	91	C_9H_9NO
Proline	Pro	P	97.05276	70		C_5H_7NO
Serine	Ser	S	87.03203	60		$C_3H_5NO_2$
Threonine	Thr	T	101.04768	74		$C_4H_7NO_2$
Tryptophan	Trp	W	186.07931	159	11,117,130,132,170,100	$C_{11}H_{10}N_2O$
Tyrosine	Tyr	Y	163.06333	136	91,107	$C_9H_9NO_2$
Valine	Val	V	99.06841	72	44,55,69	C_5H_9NO

รูปที่ 8.3 ค่าน้ำหนักของกรดแอมิโนมาตรฐาน

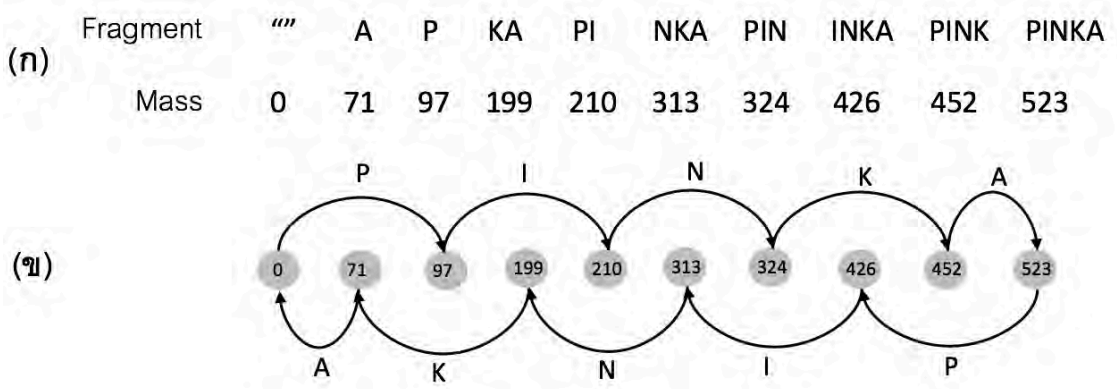
(*ที่มา:* Huimin Zhong, Public domain, via Wikimedia Commons. 2015. *Mass of amino acid fragment ions.* [*เข้าถึงออนไลน์เมื่อวันที่* 11 *ก.ค. พ.ศ.* 2564])

จากรูปที่ 8.4(ข) เส้นทางด้านบนจากซ้ายไปขวาแสดงลำดับกรดแอมิโนที่อนุมานได้ส่วนเส้นทางด้านล่างของกราฟเป็นเส้นทางกลับด้านของเพปไทด์ (รหัสเทียมที่ 8.1 `DecodingIdealSpectrum()`)

รหัสเทียมที่ 8.1 DecodingIdealSpectrum

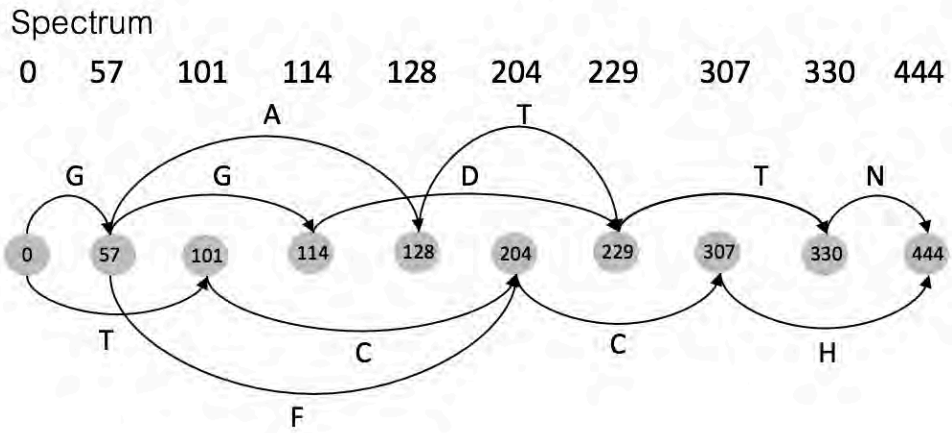
```
DecodingIdealSpectrum()
    สร้าง GRAPH(Spectrum) แบบมีทิศทาง
    หาเส้นทาง Path จากจุดตั้งต้น source ไปยังจุดสิ้นสุด sink ใน GRAPH(Spectrum)
    ส่งกลับ สตริงแสดงลำดับกรดแอมิโนจากฉลากที่กำกับแต่ละเส้นเชื่อมใน Path
```

ฝึกหัด	จงหาลำดับกรดแอมิโนของสายเพปไทด์โดยใช้ข้อมูลสเปกตรัมในอุดมคติต่อไปนี้ {0, 57, 114, 128, 215, 229, 316, 330, 387, 444}



รูปที่ 8.4 (ก) ค่าน้ำหนักของพรีฟิกซ์และซัฟฟิกซ์ของ **PINKA** *ซึ่งประกอบกันเป็น IDEALSPECTRUM(***PINKA***) = {0, 71, 97, 199, 210, 313, 324, 426, 452, 523} (ข) กราฟแบบมีทิศทางโดยเส้นทางด้านบนจากซ้ายไปขวาแสดงลำดับกรดแอมิโนที่หาได้*

จากแบบฝึกหัดข้างต้น อาจพบเส้นทางจากจุดตั้งต้นไปยังจุดปลายทางได้มากกว่า 1 เส้นทาง ซึ่งเส้นทางอื่นๆ ที่หาได้หลายเส้นทางไม่ได้แสดงลำดับกรดแอมิโนที่ถูกต้องของเพปไทด์ (รูปที่ 8.5) ดังนั้นจึงต้องมีการแก้ไขรหัสเทียมข้างต้นเป็นรหัสเทียมที่ 8.2 `DecodingIdealSpectrumV2()`



รูปที่ 8.5 **GRAPH(Sepctrum)** *แบบมีทิศทางของสเปกตรัม {0, 57, 101, 114, 128, 204, 229, 307, 330, 444} โดยมีเพียงบางเส้นทางจากจุดเริ่มต้นไปยังจุดสิ้นสุดที่สอดคล้องกับชุดของเพปไทด์ที่อธิบายสเปกตรัม*

รหัสเทียมที่ 8.2 DecodingIdealSpectrumV2

```
1  DecodingIdealSpectrum()
2      สร้าง GRAPH(Spectrum) แบบมีทิศทาง
3      for แต่ละเส้นทาง Path จากจุดตั้งต้น source ไปยังจุดสิ้นสุด sink ใน GRAPH(Spectrum)
4          Peptide <- ลำดับกรดแอมิโนจากฉลากที่กำกับแต่ละเส้นเชื่อมใน Path
5          if IdealSpectrum(Peptide) เท่ากับ Spectrum
6              ส่งกลับ Pedtide
```

ถึงแม้รหัสเทียมที่ 8.2 แก้ปัญหาการหาลำดับกรดแอมิโนจากสเปกตรัมในอุดมคติได้ การหาทุกเส้นทางที่เป็นไปได้จากจุดเริ่มต้นไปยังจุดสิ้นสุดใช้เวลามาก

การหาลำดับกรดแอมิโนจากสเปกตรัมที่วัดได้จริง

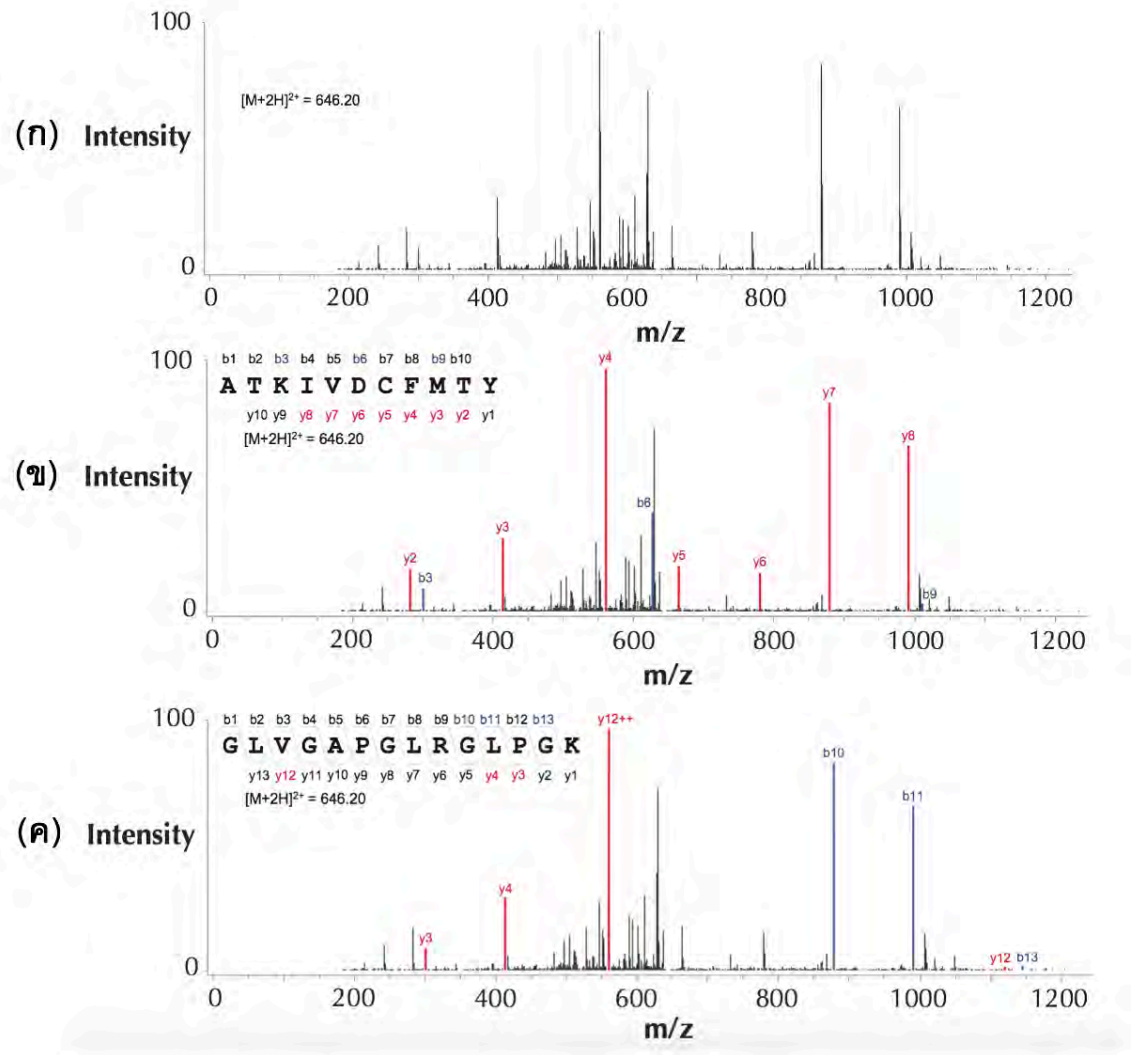
หลังจากเครื่องแมสสเปกโทรมิเตอร์ ทำการแยกสายเพปไทด์หนึ่งๆ ออกเป็นสองส่วนย่อย สายเพปไทด์ย่อยแต่ละส่วนจะถูกไอออไนซ์ กลายเป็นส่วนย่อยที่มีประจุ fragment ions ซึ่งเครื่องแมสสเปกโทรมิเตอร์จะทำการวัดค่า mass-to-charge ratio (m/z) ของแต่ละเพปไทด์ย่อย รวมทั้งค่า intensity หรือจำนวน fragment ions ที่มีค่า m/z เดียวกัน (เพปไทด์หนึ่งๆ อาจมีโอกาสถูกแยกเป็นสองส่วนในรูปแบบใดรูปแบบหนึ่งมากกว่ารูปแบบอื่นๆ เนื่องจากบางพันธะในสายเพปไทด์อาจถูกแยกได้ง่ายกว่า) ผลที่ตามมาคือสเปกตรัมจะประกอบด้วยชุดของพีค (peak) ในแผนภูมิโดยแต่ละพีคบนแกน x แสดงค่า mass-to-charge ratio ในขณะที่ความสูงของพีคแสดงค่า intensity (รูปที่ 8.6(ก))

เครื่องแมสสเปกโทรมิเตอร์สมัยใหม่ยังมีข้อจำกัดในช่วงของค่า mass-to-chare ratio ที่สามารถวัดได้ ดังนั้นจึงเป็นการยากที่จะวัดสายโปรตีนทั้งสายโดยใช้เทคนิคแมสสเปกโทรเมตรี ผลที่ตามมาคือในการวิเคราะห์โปรตีน สายของโปรตีนจะถูกตัดออกเป็นเพปไทด์สั้นๆ โดยใช้เอนไซม์กลุ่มโพรทีเอส (protease) โดยโพรทีเอสที่ใช้อย่างแพร่หลายในการศึกษาโพรทีโอมิกส์และในการศึกษาเพปไทด์ของทีเร็กซ์ด้วย คือ ทริปซิน (trypsin) โดยทริปซินจะแตกสายโปรตีนหลังกรดแอมิโนอาร์จินีน (arginine: R) และไลซีน (lysine: K) ได้เป็นเพปไทด์สายสั้นมีความยาวเฉลี่ยประมาณ 14 กรดแอมิโน

รูปที่ 8.6(ก) แสดงตัวอย่างแมสสเปกตรัมของทีเร็กซ์ และตัวอย่างของการตีความในรูปที่ 8.6(ข) และรูปที่ 8.6(ค) ว่ามีเพปไทด์ ATKIVDCFMTY และ GLVGAPGLRGLPGK ตามลำดับ เพปไทด์เหล่านี้สามารถเชื่อมโยงกลับไปยังพีคของสเปกตรัมในแผนภูมิที่แสดงพรีฟิกซ์และซัฟฟิกซ์ ทั้งนี้เพื่อให้เป็นไปตามมาตรฐานของนิยามศัพท์ของแมสสเปกโทรเมตรี พีคที่ถูกระบุว่าเป็นพรีฟิกซ์ความยาว i จะถูกกำหนดป้ายเป็น b_i และพีคที่ถูกระบุว่าเป็นซัฟฟิกซ์ความยาว i จะถูกกำหนดเป็น y_i

ฝึกหัด	เพปไทด์ใดในรูปที่ 8.6 อธิบายสเปกตรัมของทีเร็กซ์ได้ดีกว่ากัน

การหาลำดับกรดแอมิโนของเพปไทด์จากสเปกตรัมจริงซับซ้อนกว่าในกรณีของสเปกตรัมในอุดมคติ เนื่องจากข้อมูลจากเครื่องแมสสเปกโทรมิเตอร์มักมีพีคที่เป็นสัญญาณรบกวน (noise) ซึ่งเป็นค่า m/z ที่ไม่ถูกต้อง นอกจากนี้บางพันธะอาจแตกได้ยาก ทำให้ค่า m/z ของแต่ละ intensity มีความแตกต่างกันไป และในสเปกตรัมอาจไม่เกิดบางพรีฟิกซ์หรือซัฟฟิกซ์เลย เช่น รูปที่ 8.6(ค) ไม่มีพีคที่ติดป้าย b_5 และ y_9 เป็นต้น



*รูปที่ 8.6 (ก) ตัวอย่างสเปกตรัมของทีเร็กซ์ (ข) สเปกตรัมเดียวกันที่มีการระบุเพปไทด์ **ATKIVDCFMTY** (ค) สเปกตรัมเดียวกันที่มีการระบุเพปไทด์ **GLVGAPGLRGLPGK***

(*ที่มา*: *รูปที่* 11.5 *ของ [52]*)

การหาลำดับเพปไทด์

การให้คะแนนเพปไทด์เมื่อเทียบกับสเปกตรัม

กำหนดให้เพปไทด์เส้นหนึ่งประกอบด้วยกรดแอมิโน 2 แบบคือ X และ Z ซึ่งมีค่าน้ำหนัก 4 และ 5 ตามลำดับ ถ้ามีเพปไทด์ XZZXX แล้วจะมีค่าน้ำหนักของพรีฟิกซ์เป็น 4, 9, 14, 18, 22 ในขณะที่ค่าน้ำหนักของซัฟฟิกซ์เป็น 22, 18, 13, 8 , 4 และเวกเตอร์ของสเปกตรัมที่วัดการแตกตัวของเพปไทด์ข้างต้นมีค่าต่อไปนี้

(0, 0, 0, 3, 8, 7, 2, 1, 100, 0 ,1, 4, 3, 500, 2, 1, 3, 9, 1, 2, 2, 0)

โดยตำแหน่งที่ i ของเวกเตอร์เป็นค่า intensity ที่วัดได้ของน้ำหนัก i ในตัวอย่างนี้พรีฟิกซ์ของ XZZXX มีค่า intensity เป็น 3, 100, 500, 9, และ 0 ในขณะที่ซัฟฟิกซ์มีค่า intensity เป็น 0, 8, 0, 2, และ 1 เป้าหมายคือหา

วิธีการให้คะแนนเพปไทด์ที่สอดคล้องกับข้อมูลจากสเปกตรัม โดยหวังว่าเพปไทด์ที่สอดคล้องกับสเปกตรัมจะมีค่าคะแนนสูงสุด

หยุดคิด	ควรให้คะแนนเพปไทด์หนึ่งๆ โดยเทียบกับสเปกตรัมอย่างไร

แนวทางแรกที่เป็นไปได้คือการนับผลรวมของ intensity หรือ intensity count ของทุกพีคที่เป็นค่าน้ำหนักพรีฟิกซ์และซัฟฟิกซ์โดยในตัวอย่างข้างต้น intensity count ของเพปไทด์ **XZZXX** มีค่าเท่ากับ 3+100+500+9+0+0+8+0+2+1 อย่างไรก็ตามวิธีนี้มีข้อจำกัดเนื่องจากในข้อมูลจริงค่า intensity ของพีคต่างๆ มีความแตกต่างกันมาก ทำให้พีคที่มีค่า intensity สูงมาก (ซึ่งอาจเป็นสัญญาณรบกวน) มีผลกระทบต่อคะแนนโดยรวมมาก ในขณะที่พีคที่ถูกต้องแต่มีค่า intensity น้อยจะถูกลดความสำคัญลงไป

เพื่อเป็นการแก้ไขข้อจำกัดข้างต้นจึงมีการนำเสนอวิธีการ shared peaks count โดยนับจำนวนของพีคที่มีค่า intensity สูงกว่าค่าขีดแบ่ง (threshold) ค่าหนึ่ง สมมติค่าขีดแบ่ง คือ 5 จากตัวอย่างข้างต้น จะได้ค่า shared peaks count = 4 โดยมาจาก 3 พีค **100, 500, และ 9** ของพรีฟิกซ์และ 1 พีคที่มี intensity **8** ของซัฟฟิกซ์ ตัวอย่างในรูปที่ 8.6 (ข) มีจำนวน shared peaks count เท่ากับ 10 ในขณะที่รูปที่ 8.6(ค) มีจำนวน shared peaks count เท่ากับ 6 เป็นต้น ถึงแม้ shared peaks count จะใช้งานได้ดีกว่า intensity count แต่ยังให้ผลไกลจากอุดมคติมาก แนวทางที่ดีกว่าควรจะเป็นแนวทางที่สามารถใช้ intensity แต่พีคที่มีค่า intensity สูงไม่ควรเป็นตัวชี้นำผลของความถูกต้องของเพปไทด์มากกว่าพีคที่ถูกต้องอื่นๆ ที่มีค่า intensity น้อยกว่า เพื่อให้ได้วิธีการที่ดีกว่าเราสามารถแปลงเพปไทด์และสเปกตรัมให้อยู่ในรูปแบบเวกเตอร์และกำหนดฟังก์ชันการให้คะแนนเป็น dot product ของสองเวกเตอร์

การแปลงเพปไทด์และสเปกตรัมให้อยู่ในรูปแบบเวกเตอร์

จากสายอักขระที่แสดงลำดับกรดแอมิโน *Peptide* = $a_1,\ldots a_n$ ความยาว n กำหนดเพปไทด์เวกเตอร์ (peptide vector) $\overrightarrow{Peptide}$ เพื่อเก็บค่าน้ำหนักของพรีฟิกซ์ โดยตำแหน่งที่มีค่าน้ำหนักพรีฟิกซ์มีค่าเป็น 1 และตำแหน่งที่เหลือทั้งหมดเป็น 0 ในกรณีของเพปไทด์ตัวอย่าง **XZZXX** ข้างต้น ชุดค่าน้ำหนักพรีฟิกซ์ประกอบด้วย **4, 9, 14, 18, 22** ซึ่งสอดคล้องกับเพปไทด์เวกเตอร์ (**0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1**) ที่มีความยาว 22 ตำแหน่ง

นิยามปัญหาที่ 8.2 ปัญหาการแปลงเพปไทด์เป็นเพปไทด์เวกเตอร์

แปลงสายเพปไทด์ให้เป็นเพปไทด์เวกเตอร์	
ข้อมูลเข้า	สายอักขระของลำดับกรดแอมิโน *Peptide*
ผลลัพธ์	เพปไทด์เวกเตอร์ $\overrightarrow{Peptide}$ ของ *Peptide*

เนื่องจากเพปไทด์เวกเตอร์เป็นตัวแทนของเพปไทด์ ในส่วนต่อไปนี้คำว่าเพปไทด์เวกเตอร์และเพปไทด์สามารถใช้แทนกันได้

นิยามปัญหาที่ 8.3 ปัญหาการแปลงเพปไทด์เวกเตอร์เป็นเพปไทด์

แปลงเพปไทด์เวกเตอร์ให้เป็นสายเพปไทด์	
ข้อมูลเข้า	เพปไทด์เวกเตอร์ *P*
ผลลัพธ์	เพปไทด์ที่มีเพปไทด์เวกเตอร์เท่ากับ *P* (ถ้ามีเพปไทด์นั้นอยู่จริง)

ซัฟฟิกซ์เพปไทด์หายไปไหน

ในความเป็นจริงแล้วซัฟฟิกซ์เพปไทด์ไม่ได้หายไปไหน เนื่องจากแต่ละพีคที่แสดงค่าน้ำหนักนั้นอาจเป็นค่าน้ำหนักของพรีฟิกซ์หรือซัฟฟิกซ์ก็ได้ จากความไม่แน่นอนนี้ผู้เชี่ยวชาญเทคนิคแมสสเปกโทรเมตรีจะทำการแปลงสเปกตรัม *Spectrum* ให้อยู่ในรูปแบบเวกเตอร์ $\overrightarrow{Spectrum}$ ที่มีการรวมข้อมูลเกี่ยวกับค่า intensity ของแต่ละพีคและพีคฝาแฝด (twin peak: พีคที่แสดงค่า MASS(*Peptide*)-s) ของมันเข้าเป็นค่าเดียวเรียกว่า แอมพลิจูด (amplitude) ทั้งนี้ค่าแอมพลิจูดที่เป็นลบมักแสดงถึงตำแหน่งในสเปกตรัมที่ไม่มีพีคหรือเป็นพีคที่มีค่า intensity ต่ำ ทั้งนี้ค่าแอมพลิจูดที่ค่าน้ำหนัก i สะท้อนความเป็นไปได้ (*likelihood)* ที่สายเพปไทด์ (ที่ไม่ทราบลำดับกรดแอมิโน) จะทำให้เกิดสเปกตรัมมีค่าน้ำหนักพรีฟิกซ์เท่ากับ i

หลังจากสายเพปไทด์ *Peptide* ได้ถูกแปลงไปเป็นเพปไทด์เวกเตอร์ $\overrightarrow{Peptide} = (p_1,\ldots,p_m)$ และสเปกตรัม *Spectrum* ได้ถูกแปลงไปเป็นสเปกตรัมเวกเตอร์ $\overrightarrow{Spectrum} = (s_1,\ldots,s_m)$ ที่มีจำนวนตำแหน่งเท่ากัน เราสามารถกำหนดฟังก์ชันการให้คะแนนสายเพปไทด์ $\mathrm{SCORE}(Peptide, Spectrum) = SCORE(\overrightarrow{Peptide}, \overrightarrow{Spectrum})$ ซึ่งเท่ากับ dot product ของ $\overrightarrow{Peptide}$ และ $\overrightarrow{Spectrum}$ ดังต่อไปนี้

$$SCORE(Peptide, Spectrum) = p_1 \cdot s_1 + \cdots + p_m \cdot s_m$$

ซึ่งในความเป็นจริงแล้ว $SCORE(\overrightarrow{Peptide}, \overrightarrow{Spectrum})$ คือผลรวมค่าแอมพลิจูดหรือ amplitude count ที่สามารถเป็นตัวแทนของแต่ละพีคของสายเพปไทด์นั่นเอง อย่างไรก็ตามในกรณีนี้จะไม่พบปัญหาเหมือนในกรณี intensity count เนื่องจากเรามีการแปลงค่า intensity เป็นค่าแอมพลิจูด ซึ่งพีคที่มีค่า intensity สูงจะไม่ชี้นำการให้คะแนนของสายเพปไทด์มากเกินไป

ส่วนที่เหลือของบทเรียนนี้จะใช้สเปกตรัมเวกเตอร์แทนสเปกตรัม โดยถ้ามีข้อมูลเข้าเป็นสเปกตรัมเวกเตอร์ $\overrightarrow{Spectrum}$ เป้าหมายคือหาเพปไทด์ *Peptide* ที่ให้ค่าคะแนน $SCORE(\overrightarrow{Peptide}, \overrightarrow{Spectrum})$ สูงสุด

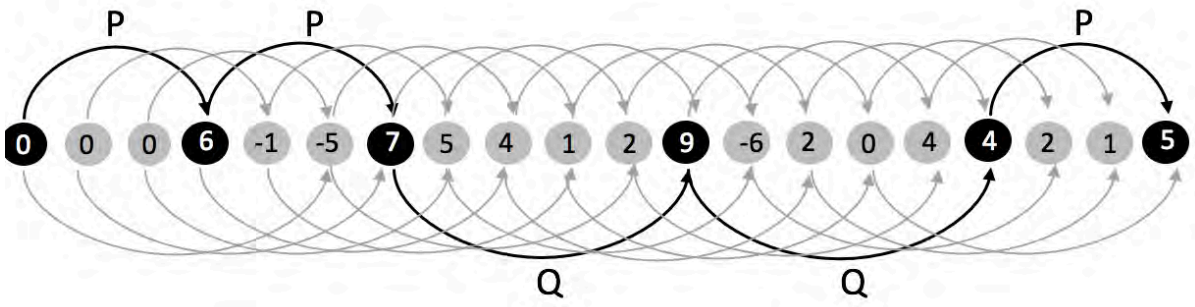
อัลกอริทึมหาลำดับกรดแอมิโนของเพปไทด์

ถ้าข้อมูลเข้าเป็นสเปกตรัมเวกเตอร์ $\overrightarrow{Spectrum} = (s_1, \ldots, s_m)$ เราสามารถสร้างกราฟแบบมีทิศทางที่ประกอบด้วย m+1 โหนด โดยแต่ละโหนดมีป้ายแสดงค่าน้ำหนักจากโหนด 0 (โหนดเริ่มต้น) ถึงโหนด m (โหนดปลายทาง)

และเพิ่มเส้นเชื่อมจากโหนด i ไปโหนด j ถ้า j – i มีค่าเท่ากับค่าน้ำหนักของกรดแอมิโนใดแอมิโนหนึ่ง รูปที่ 8.7 แสดงกราฟแบบมีทิศทางเชื่อมต่อโหนดในสเปกตรัมเวกเตอร์ที่มีจำนวนโหนดทั้งหมด 20 โหนด (โดยเพิ่มโหนดเริ่มต้นทางซ้ายเข้าไป 1 โหนด) และมีกรดแอมิโนสองตัว P และ Q ที่มีค่าน้ำหนัก 3 และ 5 ตามลำดับ เส้นทางจาก 0 ถึง m ที่แสดงเพปไทด์ **PPQQP** ที่มีค่าเพปไทด์เวกเตอร์เป็น (0, 0, **1**, 0, 0, **1**, 0, 0, 0, 0, **1**, 0, 0, 0, 0, **1**, 0, 0, **1**) โดยมีค่าคะแนนเท่ากับ 0 + 6 + 7 + 9 + 4 + 5

นิยามปัญหาที่ 8.4 ปัญหาการหาลำดับกรดแอมิโนของเพปไทด์

จากสเปกตรัมเวกเตอร์ที่เป็นข้อมูลเข้า หาเพปไทด์ที่ให้ค่าคะแนนมากสุดเมื่อเทียบกับสเปกตรัมเวกเตอร์	
ข้อมูลเข้า	สเปกตรัมเวกเตอร์ $\overrightarrow{Spectrum}$
ผลลัพธ์	ลำดับกรดแอมิโนในเพปไทด์ *Peptide* ที่ทำให้ค่าคะแนน $SCORE(\overrightarrow{Peptide}, \overrightarrow{Spectrum})$ สูงสุด จากลำดับกรดแอมิโนที่เป็นไปได้ทั้งหมด



*รูปที่ 8.7 กราฟแบบมีทิศทางจำนวน 19 โหนดแสดงสเปกตรัมเวกเตอร์ที่มีกรดแอมิโนสองตัวคือ **P** และ **Q** ซึ่งมีค่าน้ำหนัก 3 และ 5 เป็นส่วนประกอบ*

หยุดคิด	กราฟแบบมีทิศทางในรูปที่ 8.7 นี้ แตกต่างจากกราฟในรูปที่ 8.4 อย่างไร

จากกราฟแบบมีทิศทางในรูปที่ 8.7 ทุกเส้นทางจากโหนดเริ่มต้นไปยังโหนดสิ้นสุดเป็นตัวแทนของลำดับกรดแอมิโนของเพปไทด์และน้ำหนักรวมของทุกโหนดในเส้นทางหนึ่งๆ มีค่าเท่ากับ $SCORE(\overrightarrow{Peptide}, \overrightarrow{Spectrum})$ ดังนั้นความซับซ้อนของปัญหาการหาลำดับกรดแอมิโนกลายเป็นปัญหาการหาเส้นทางที่มีค่าน้ำหนักรวมมากสุดจากโหนดเริ่มต้นถึงโหนดปลายทาง

เมื่อใช้วิธีการข้างต้นกับสเปกตรัมเวกเตอร์ของทีเร็กซ์ เพปไทด์ **ATKIVDCFMTY** ได้คะแนน 96 (รูปที่ 8.6(ข)) อย่างไรก็ตามแอสราได้เสนอเพปไทด์อีกสายคือ **GLVGAPGLRGLPGK** ที่มีคะแนนจากวิธีการข้างต้นเป็น -19 (รูปที่ 8.6(ค)) โดยระบุว่าเป็นทีเร็กซ์เพปไทด์ เพปไทด์ที่เสนอโดยแอสรานี้มีคะแนนรวมต่ำกว่า

ATKIVDCFMTY มาก และในความเป็นจริงแล้วมีเพปไทด์หลายพันล้านเส้นได้คะแนนมากกว่าทีเร็กซ์เพปไทด์ข้างต้น

หยุดคิด	เราสามารถดัดแปลงอัลกอริทึมในบทที่ 5 เพื่อนำมาใช้ในการหาเส้นทางที่มีค่าน้ำหนักรวมมากสุดจากโหนดเริ่มต้นถึงโหนดปลายทางในกราฟแบบมีทิศทางข้างต้นได้อย่างไร

หยุดคิด	ทำไมแอสราถึงไม่นำเสนอเพปไทด์ ATKIVDCFMTY เป็นทีเร็กซ์เพปไทด์ทั้งที่มีคะแนนสูงกว่ามาก

การระบุเพปไทด์

ปัญหาการระบุเพปไทด์

ถึงแม้มีความพยายามสร้างฟังก์ชันการให้คะแนนที่สอดคล้องกับความถูกต้องของเพปไทด์ โดยเพปไทด์ที่ทำให้เกิดสเปกตรัมควรเป็นเพปไทด์ที่มีคะแนนสูงสุด แต่ฟังก์ชันการให้คะแนนที่มีอยู่ยังไม่ดีพอ อย่างไรก็ตาม ถึงแม้เพปไทด์ที่ถูกต้องมักไม่ได้คะแนนสูงสุดเมื่อเทียบกับเพปไทด์ทั้งหมด แต่เพปไทด์ที่ถูกต้องมักมีคะแนนสูงสุดเมื่อนำไปเปรียบเทียบกับเพปไทด์ที่เป็นส่วนประกอบของโปรตีโอมในสิ่งมีชีวิตหนึ่งๆ ผลที่ตามมาคือเราสามารถเปลี่ยนจากการหาลำดับกรดแอมิโนในเพปไทด์ผ่านชุดของค่าสเปกตรัม ไปเป็นการระบุเพปไทด์โดยเทียบกับข้อมูลเพปไทด์ที่ปรากฏในโปรตีโอมของสิ่งมีชีวิต ทั้งนี้ให้สมมติว่าโปรตีโอมเกิดจากการนำเพปไทด์ทั้งหมดที่พบในสิ่งมีชีวิตหนึ่งๆ มาต่อกันเป็นอักขระสายยาว

นิยามปัญหาที่ 8.5 ปัญหาการระบุเพปไทด์

หาเพปไทด์จากโปรตีโอมโดยเป็นเพปไทด์ที่ให้ค่าคะแนนสูงสุดเมื่อนำไปเทียบกับสเปกตรัม	
ข้อมูลเข้า	สเปกตรัมเวกเตอร์ $\overrightarrow{Spectrum}$ และลำดับกรดแอมิโนโปรตีโอม *Proteome*
ผลลัพธ์	ลำดับกรดแอมิโนในเพปไทด์ *Peptide* ที่ทำให้ค่าคะแนน $SCORE(\overrightarrow{Peptide}, \overrightarrow{Spectrum})$ สูงสุดและเป็นลำดับกรดแอมิโนที่อยู่ในโปรตีโอม

หยุดคิด	ในทางปฏิบัติข้อมูลเข้าสำหรับการระบุเพปไทด์จะเป็นชุดของสายโปรตีนมากกว่าเป็นโปรตีโอมยาวสายเดียว เมื่อเราใช้ข้อมูลเข้าเป็นโปรตีโอมที่เกิดจากการนำสายโปรตีนมาเรียงต่อกัน มีข้อพึงระวังอะไรบ้าง

การระบุเพปไทด์ในโปรตีโอมของทีเร็กซ์

อาจมีข้อสงสัยว่าทำไมจึงพิจารณาปัญหาการระบุเพปไทด์ ในเมื่อไม่มีข้อมูลโปรตีโอมของทีเร็กซ์ ซึ่งหมายความว่าเราไม่สามารถระบุเพปไทด์ของทีเร็กซ์โดยใช้โปรตีโอมได้ อย่างไรก็ตามประมาณ 90% ของโปรตีนกระดูกสัตว์เป็นคอลลาเจน (collagen) ดังนั้นกระดูกไดโนเสาร์อาจประกอบด้วยคอลลาเจนเช่นกัน และมีโอกาสน้อยที่โปรตีนอื่นๆ จะคงสภาพในฟอสซิลมากว่าหลายล้านปี แอสรามีสมมติฐานว่าโปรตีนใดก็ตามที่พบในฟอสซิลของทีเร็กซ์น่าจะมีความคล้ายคลึงกับคอลลาเจนที่พบในสิ่งมีชีวิตในปัจจุบัน

เพื่อเป็นการตรวจสอบสมมติฐานนี้แอสราทำการเปรียบเทียบสเปกตรัมของทีเร็กซ์กับโปรตีนทั้งหมดที่อยู่ในฐานข้อมูลโปรตีนยูนิพรอต (UniProt) ซึ่งมีข้อมูลสายโปรตีนจากสิ่งมีชีวิตหลายร้อยสปีชีส์ และมีข้อมูลอย่างน้อย 200 ล้านกรดแอมิโน นอกจากนี้แอสรายังทำการเพิ่มข้อมูลสายโปรตีนคอลลาเจนของสิ่งมีชีวิตในปัจจุบันที่มีการแปรผันเข้าไปเป็นส่วนหนึ่งของฐานข้อมูลเพื่อเป็นการจำลองความแตกต่างที่เป็นไปได้ระหว่างคอลลาเจนเหล่านี้กับคอลลาเจนของทีเร็กซ์ด้วย (ฐานข้อมูลนี้ว่า UniProt+) ซึ่งพบว่าเพปไทด์ส่วนใหญ่ในฐานข้อมูลที่ตรงกับสเปกตรัมและมีคะแนนสูงนั้นเป็นคอลลาเจนของไก่ ซึ่งสนับสนุนสมมติฐานว่าสัตว์ปีกมีวิวัฒนาการมาจากไดโนเสาร์ ซึ่งทีเร็กซ์เพปไทด์ต่างจากเพปไทด์คอลลาเจนของไก่เพียง 1 กรดแอมิโน อย่างไรก็ตาม คำถามที่ตามมาคือจะตรวจสอบได้อย่างไรว่าการตีความเกี่ยวกับทีเร็กซ์เพปไทด์นี้เป็นการตีความที่ถูกต้องเกี่ยวกับข้อมูลสเปกตรัมของไดโน-เสาร์

การระบุเพปไทด์กับทฤษฎีลิงพิมพ์ดีด

ถ้ากำหนดให้ PSM (Peptide-Spectrum Match) เป็นปัญหาที่ต่อยอดมาจากปัญหาที่ 8.5 โดยมีการเปลี่ยนข้อมูลเข้าจากสเปกตรัมเวกเตอร์เดียวไปเป็นชุดของสเปกตรัมเวกเตอร์ และมีตัวแปรเพิ่มอีกหนึ่งตัวคือค่าขีดแบ่ง (threshold) ผลลัพธ์ที่คาดหวังประกอบด้วยชุดของเพปไทด์ที่พบในโปรตีโอม ที่มีค่าคะแนนมากกว่าค่าขีดแบ่ง ผลของการใช้ PSM เพื่อระบุเพปไทด์จากชุดของสเปกตรัมเวกเตอร์ พบว่าทีเร็กซ์เพปไทด์ยังเป็นเพปไทด์ที่มีค่าคะแนนสูงสุดเมื่อเทียบกับเพปไทด์ที่ได้จากสเปกตรัมเวกเตอร์อื่นที่ผ่านค่าขีดแบ่งเช่นกัน เนื่องจากยังมีเพปไทด์อีกหลายพันล้านเส้นที่ไม่ได้อยู่ในฐานข้อมูล UniProt+ ที่อาจมีคะแนนด์สูงกว่าคะแนนของทีเร็กซ์เพปไทด์มาก คำถามคือฐานข้อมูล UniProt+ ของแอสรามีความสมบูรณ์หรือเพียงพอสำหรับการได้มาซึ่งทีเร็กซ์เพปไทด์ที่ถูกต้องหรือไม่ รวมทั้งความถูกต้องของ PSM คำตอบคือมีความเป็นไปได้ที่มีเพปไทด์จำนวนมากมีน้ำหนักเท่ากับทีเร็กซ์เพปไทด์แต่ได้คะแนนมากกว่าตอนหาลำดับกรดแอมิโนถ้า PSM ทำงานถูกต้อง แต่สิ่งที่ต้องพิจารณาเพิ่มเติมคือชุดของเพปไทด์ที่เป็นผลลัพธ์จาก PSM นั้นมีความสำคัญทางสถิติมากน้อยเพียงใด

False discovery rate

ในการประมาณจำนวนเพปไทด์ที่ไม่มีความสำคัญทางสถิติจากชุดของเพปไทด์ที่เป็นผลลัพธ์ของ PSM สามารถทำได้โดยสร้างโปรตีโอมหลอกหรือดีคอยโปรตีโอม (decoy proteome) โดยแต่ละลำดับกรดแอมิโนในดีคอยโปรตี-

โอมนี้มาจากการสุ่มเลือกกรดแอมิโนที่เป็นไปได้ 20 ตัว ด้วยความน่าจะเป็นที่เท่ากัน (1/20) โดยจะสร้างดีคอยโปรตีโอมให้มีความยาวเท่ากับโปรตีโอมจริง จากนั้นลองใช้ PSM เพื่อหาชุดของเพปไทด์ในดีคอยโปรตีโอมที่มีคะแนนเกินค่าขีดแบ่ง และนำจำนวนเพปไทด์ที่เป็นผลลัพธ์ไปคำนวณค่า false discovery rate (FDR) ตามสมการต่อไปนี้

$$\frac{|PSM_{threshold}(DecoyProteome, SpectralVectors)|}{|PSM_{threshold}(Proteome, SpectralVectors)|}$$

ถ้าใช้ PSM ค้นหาเพปไทด์ในโปรตีโอมพบ 100 เพปไทด์ ในขณะที่ค้นหาในดีคอยโปรตีโอมพบ 5 เพปไทด์ เราสามารถสรุปได้ว่า 95% ของเพปไทด์ที่หาได้จาก PSM น่าจะเป็นเพปไทด์ที่ถูกต้อง ถ้าผลของการค้นหาในดีคอยโปรตีโอมพบ 100 เพปไทด์ ค่า FDR จะเข้าใกล้ 1 ซึ่งทำให้ตัดสินได้ยากว่าเพปไทด์ที่เป็นผลลัพธ์แต่ละเส้นนั้นมีความสำคัญหรือมีความหมายทางชีววิทยาหรือไม่

หยุดคิด	ถ้าค่า FDR เข้าใกล้ 1 สำหรับค่าขีดแบ่งหนึ่งๆ ยังสามารถหาเพปไทด์ที่มีความหมายทางชีววิทยาจาก PSM ได้หรือไม่

เราไม่ควรสรุปว่าข้อมูลสเปกตรัมที่มีนั้นไม่มีประโยชน์ถึงแม้ FDR มีค่าสูง หรือไม่ควรสรุปว่าเรากำลังใช้ฐานข้อมูลที่ไม่ถูกต้อง เพราะในความเป็นจริงแล้วค่า FDR ที่สูงนั้นอาจเป็นผลจากการเลือกค่าขีดแบ่งที่ไม่ถูกต้อง เนื่องจากค่า FDR อาจแปรผันมากตามค่าขีดแบ่ง

สำหรับข้อมูลสเปกตรัมของทีเร็กซ์พบ 27 เพปไทด์เมื่อใช้โปรตีโอม UniProt+ และพบเพียง 1 เพปไทด์ในดีคอยโปรตีโอมเมื่อใช้ค่าขีดแบ่ง = 100 และได้ค่า FDR = 3.7% อย่างไรก็ตาม ยังไม่สามารถสรุปได้ว่าทั้ง 26 เพปไทด์ที่เหลือนั้นเป็นเพปไทด์ของไดโนเสาร์ทั้งหมด เนื่องจากหลายเพปไทด์ที่พบนั้นอาจเกิดจากการปนเปื้อนที่พบเป็นปกติในกระบวนการทางห้องปฏิบัติการ ในขณะที่ FDR ช่วยประเมินคุณภาพโดยรวมของชุดเพปไทด์ที่เป็นผลลัพธ์จากการรัน PSM โดยใช้สเปกตรัมของทีเร็กซ์ คำถามคือเพปไทด์ที่เป็นผลลัพธ์นี้มีความสำคัญเชิงสถิติมากน้อยเพียงใด เป้าหมายถัดไปคือหาวิธีการประเมินว่า 26 เพปไทด์นี้เป็นเพปไทด์ของไดโนเสาร์จริง

ลิงกับเครื่องพิมพ์ดีด

ทฤษฎี infinite monkey กล่าวว่าถ้าปล่อยให้ลิงใช้เครื่องพิมพ์ดีดพิมพ์อะไรไปเรื่อยๆ คำที่พิมพ์ออกมาบางคำจะเป็นคำศัพท์ที่ถูกต้อง ถ้ามีชุดของคำศัพท์ *Dictionary* กำหนด E(*Dictionary, n*) เป็นจำนวนคำศัพท์ใน *Dictionary* ที่คาดหวังว่าจะพบในสายอักขระความยาว n ที่สร้างมาแบบสุ่มโดยแต่ละอักขระมีโอกาสที่จะถูกสุ่มเลือกมาเท่ากัน และกำหนด *EnglishDictionary* เป็นชุดคำศัพท์ทั้งหมดในภาษาอังกฤษ ถ้าปรากฏว่าหลังลิงพิมพ์ไป n อักขระ จำนวนคำศัพท์ภาษาอังกฤษที่ลิงพิมพ์ได้มีจำนวนมากกว่าค่า E(*Dictionary, n*) เราอาจสรุปได้ว่าลิงสะกดคำศัพท์ได้

นิยามปัญหาที่ 8.6 ปัญหาลิงกับเครื่องพิมพ์ดีด

ประมาณจำนวนคำศัพท์ในพจนานุกรมที่คาดหวังว่าจะพบในสายข้อมูล n อักขระที่สร้างมาแบบสุ่ม	
ข้อมูลเข้า	ชุดคำศัพท์ *Dictionary* และเลขจำนวนเต็ม n
ผลลัพธ์	E(*Dictionary*, *n*)

หยุดคิด	ลิงกับพิมพ์ดีดเกี่ยวข้องกับแมสสเปกโทรเมตรีอย่างไร

นัยสำคัญทางสถิติของ PSM

ถ้าเปลี่ยนจากการปล่อยให้ลิงพิมพ์อักขระไปเรื่อยๆ เป็นการใช้อัลกอริทึมในการสร้างชุดของเพปไทด์ที่มีคะแนนเกินค่าขีดแบ่งเมื่อเทียบกับสเปกตรัมเวกเตอร์ $\overrightarrow{Spectrum}$ และเรียกชุดของเพปไทด์นี้ว่าเป็นพจนานุกรมสเปกตรัม (spectral dictionary) ซึ่งแสดงโดย

$$DICTIONARY_{threshold}(\overrightarrow{Spectrum})$$

และพจนานุกรมสเปกตรัมของไดโนเสาร์หรือ *DinosaurPSM* จะถูกแสดง โดย

$$DICTIONARY_{-19}(\overrightarrow{DinosaurSpectrum})$$

แทนการตรวจสอบว่าคำศัพท์ที่ลิงพิมพ์ออกมามีคำใดบ้างที่อยู่ในพจนานุกรม เราจะตรวจสอบว่าเพปไทด์ในพจนานุกรมสเปกตรัมปรากฏในโปรตีโอมหรือไม่ และถ้าพบต้องตัดสินว่าเพปไทด์นั้นมีความหมายทางชีววิทยาหรือไม่ หรือเป็นเพียงผลพวงของค่าสถิติที่ใช้ เพื่อให้ตัดสินใจได้เราต้องพิจารณา

$$E(DICTIONARY_{threshold}(\overrightarrow{Spectrum}), n)$$

ซึ่งแทนจำนวนเพปไทด์ในดีคอยโปรตีโอมความยาว n อักขระ ที่พบใน $DICTIONARY_{threshold}(\overrightarrow{Spectrum})$ ถ้าค่านี้มากกว่า 1 ก็ไม่น่าแปลกใจที่จะพบเพปไทด์ที่มีคะแนนเกินค่าขีดแบ่ง เมื่อเทียบกับสเปกตรัมเวกเตอร์ ดังนั้นจึงต้องมีการกำหนดปัญหาการทดสอบความสำคัญทางสถิติให้ชัดเจนมากขึ้น

เพื่อหาจำนวนเพปไทด์นี้เราเริ่มจากพจนานุกรมสเปกตรัมที่มีเพปไทด์เพียงเส้นเดียวซึ่งจะนำไปหาในดีคอยโปรตีโอมที่มีความยาว n แอมิโน โดยค่าความน่าจะเป็นที่พบเพปไทด์สายนี้ในดีคอยโปรตีโอมที่ตำแหน่งจำเพาะมีค่าเท่ากับ $\frac{1}{20^{|Peptide|}}$ ดังนั้นจำนวนครั้งที่สามารถพบเพปไทด์นี้ในดีคอยโปรตีโอมมีค่าเท่ากับ

$$\frac{n - |Peptide| + 1}{20^{|Peptide|}} \approx n \cdot \frac{1}{20^{|Peptide|}}$$

ถ้าสมมติว่าเรามีชุดของเพปไทด์อยู่ใน *Dictionary* โดยแต่ละเพปไทด์อาจมีความยาวแตกต่างกันไป ถ้าใช้การประมาณค่าข้างต้น ค่าประมาณจำนวนครั้งที่สามารถพบเพปไทด์เหล่านี้ทั้งในดีคอยโปรตีโอมและโปรตีโอมปกติคำนวณได้จากสมการต่อไปนี้

$$E(Dictionary, n) \approx n \cdot \left(\sum_{each\ peptide\ Peptide\ in\ Dictionary} \frac{1}{20^{|Peptide|}} \right)$$

พจน์ผลรวมภายในวงเล็บเป็นค่าความน่าจะเป็นของ *Dictionary* แสดงโดย Pr(*Dictionary*) ซึ่งทำให้เขียนสมการข้างต้นได้ใหม่ดังต่อไปนี้

$$E(Dictionary, n) \approx n \cdot \Pr\left(Dictionary\right)$$

สมการนี้แสดงการวิเคราะห์ความสำคัญทางสถิติของเพปไทด์ในรูปแบบการคำนวณค่าความน่าจะเป็น จากสมการถ้าต้องการทดสอบความสำคัญทางสถิติของ *DinosaurPSM* ขั้นแรกสร้างพจนานุกรม $DICTIONARY(DinosaurPSM)$ และคำนวณค่า

$$n \cdot \Pr\left(DICTIONARY(DinosaurPSM)\right)$$

โดยที่ n คือความยาวของลำดับกรดแอมิโนที่เกิดจากการนำสายโปรตีนทั้งหมดในฐานข้อมูล UniProt+ มาต่อกัน ถ้าค่าที่คำนวณได้มีค่าน้อย เช่น 0.001 จะสามารถยืนยันได้ว่า *DinosaurPeptide* เป็นเพปไทด์ของทีเร็กซ์จริงไม่ใช่ผลพวงที่เกิดจากการคำนวณทางสถิติ อย่างไรก็ตาม $DICTIONARY(DinosaurPSM)$ มีจำนวนเพปไทด์มากกว่า 200 พันล้านเส้นซึ่งใช้เวลานานมากในการคำนวณ คำถามคือสามารถคำนวณค่าความน่าจะเป็นของพจนานุกรมนี้โดยไม่ต้องสร้างเพปไทด์ออกมาทั้งหมดได้หรือไม่

พจนานุกรมสเปกตรัม

ในขั้นแรกเราจะคำนวณจำนวนเพปไทด์ในพจนานุกรมสเปกตรัม ซึ่งเมื่อทราบจำนวนแล้วอาจได้แนวทางในการคำนวณค่าความน่าจะเป็นของพจนานุกรมสเปกตรัม

นิยามปัญหาที่ 8.7 ปัญหาการหาขนาดของพจนานุกรมสเปกตรัม

หาจำนวนเพปไทด์ของพจนานุกรมสเปกตรัมเมื่อมีข้อมูลเข้าเป็นสเปกตรัมเวกเตอร์ และค่าขีดแบ่ง	
ข้อมูลเข้า	สเปกตรัมเวกเตอร์ $\overrightarrow{Spectrum}$ และค่าขีดแบ่งที่เป็นจำนวนเต็ม
ผลลัพธ์	จำนวนของเพปไทด์ใน $DICTIONARY_{threshold}(\overrightarrow{Spectrum})$

เราสามารถใช้กำหนดการพลวัต (dynamic programming) ในการคำนวณหาขนาดของพจนานุกรมสเปกตรัม โดยถ้ามีข้อมูลเข้าเป็น สเปกตรัมเวกเตอร์ $\overrightarrow{Spectrum} = (s_1, \ldots, s_m)$ และกำหนด i-prefix (สำหรับ i ที่มีค่าระหว่าง 1 ถึง m) โดย $\overrightarrow{Spectrum_i} = (s_1, \ldots, s_i)$ และกำหนดตัวแปร $\mathrm{SIZE(i, t)}$ เป็นจำนวนเพปไทด์ *Peptides* ที่มีน้ำหนัก i และทำให้ $\mathrm{SCORE}(Peptide, \overrightarrow{Spectrum})$ มีค่าเท่ากับ t ตัวอย่างเช่น สเปกตรัมเวกเตอร์ $\overrightarrow{Spectrum} = (4, -3, -2, 3, 3, -4, 5, -3, -1, -1,\ 3, 4, 1, 3)$ มีจำนวน 14 ค่า ในขณะที่แต่ละเพปไทด์ประกอบด้วย 2 กรดแอมิโนคือ X และ Z ซึ่งมีน้ำหนัก 4 และ 5 ตามลำดับ และมีเพียง 3 เพปไทด์ที่มีน้ำหนักเท่ากับ 13 คือ XXZ, XZX, และ ZXX โดยที่เพปไทด์แรกมีค่าคะแนนเป็น 1 เมื่อเทียบกับ $\overrightarrow{Spectrum_{13}}$ ในขณะที่สองเพปไทด์หลังมีค่าคะแนนเท่ากับ 3 ดังนั้น $\mathrm{SIZE}(13,1) = 1$ และ $\mathrm{SIZE}(13,3) = 2$ และ $\mathrm{SIZE}(13, t) = 0$ สำหรับค่า t อื่นๆทั้งหมด

จากชุดตัวแปรข้างต้นสามารถหาจำนวนเพปไทด์ได้โดยใช้ความสัมพันธ์เวียนเกิดในการคำนวณค่า $\mathrm{SIZE(i, t)}$ โดยชุดของเพปไทด์ที่เป็นสมาชิกของ $\mathrm{SIZE(i, t)}$ นี้สามารถถูกแบ่งออกเป็น 20 กลุ่มย่อยขึ้นอยู่กับกรดแอมิโนตัวสุดท้าย และเพปไทด์ที่ลงท้ายด้วยกรดแอมิโน *a* จำเพาะนี้ จะเป็นเพปไทด์ที่สั้นลง 1 กรดแอมิโนและมีน้ำหนักเท่ากับ i - |a| (โดย |a| คือค่าน้ำหนักของกรดแอมิโน *a*) และมีคะแนนเท่ากับ t – s_i ถ้าเรานำกรดแอมิโน *a* ออกจากเพปไทด์ ความสัมพันธ์เวียนเกิดของ $\mathrm{SIZE(i, t)}$ แสดงโดยสมการต่อไปนี้

$$\mathrm{SIZE}(i, t) = \sum_{all\ amino\ acids\ a} \mathrm{SIZE}(i - |a|, t - s_i)$$

กำหนดค่าเพปไทด์ที่ “ว่าง” คือมีความยาวเป็น 0 แอมิโนด้วย $\mathrm{SIZE}(0,0) = 1$ และกำหนดค่า $\mathrm{SIZE(i, t)} = 0$ สำหรับค่า i ที่เป็นลบ โดยการใช้สมการเวียนเกิดข้างต้นเราสามารถคำนวณหาขนาดของพจนานุกรมสเปกตรัมของ $\overrightarrow{Spectrum} = (s_1, \ldots, s_m)$ ได้ดังนี้

$$\left|\mathrm{DICTIONARY}_{threshold}\left(\overrightarrow{Spectrum}\right)\right| = \sum_{t \geq threshold} \mathrm{SIZE(m, t)}$$

ทั้งนี้สมการ *ค่าความน่าจะเป็น* ของพจนานุกรมคือ

$$\Pr(Dictionary) = \sum_{each\ peptide\ Peptide\ in\ the\ Dictionary} \frac{1}{20^{|Peptide|}}$$

มีความคล้ายคลึงกับสมการที่ใช้หา *ขนาด* ของพจนานุกรมคือ

$$|Dictionary| = \sum_{each\ peptide\ Peptide\ in\ Dictionary} 1$$

จากความคล้ายคลึงกันนี้เราสามารถอนุมานสมการเวียนเกิดในการหาค่าความน่าจะเป็นของพจนานุกรมโดยใช้ตัวแปรที่คล้ายคลึงกับที่ใช้ในกรณีของการหาขนาดของพจนานุกรม

กำหนด $\Pr(i,t)$ เป็นผลรวมค่าความน่าจะเป็นของเพปไทด์ทั้งหมดที่มีน้ำหนักเท่ากับ i สำหรับ $\text{SCORE}(\overrightarrow{Peptide}, \overrightarrow{Spectrum})$ ที่มีค่าเท่ากับ t โดยชุดของเพปไทด์นี้สามารถแบ่งออกเป็น 20 กลุ่มย่อยตามกรดแอมิโนที่ลงท้าย กำหนดให้เพปไทด์ที่ลงท้ายด้วยกรดแอมิโน *a* แสดงโดย $Peptide_a$ และถ้านำ *a* ออก *Peptide* จะมีน้ำหนักเท่ากับ i - $|a|$ และมีคะแนนเท่ากับ t - s_i เนื่องจากค่าความน่าจะเป็นของ *Peptide* มีค่าน้อยกว่าค่าความน่าจะเป็นของ $Peptide_a$ 20 เท่า หรืออีกนัยหนึ่งคือ *Peptide* มีส่วนสนันสนุนค่า Pr(i, t) น้อยกว่าที่ $Peptide_a$ สนันสนุนค่า Pr(i-$|a|$, t- s_i) 20 เท่า ดังนั้นค่า Pr(i, t) สามารถคำนวณได้จากสมการต่อไปนี้

$$\Pr(i,t) = \sum_{all\ amino\ acids\ a} \frac{1}{20} \cdot \Pr\left(i - |a|, t - s_i\right)$$

ซึ่งต่างจากสมการเวียนเกิดในการคำนวณ $\text{SIZE}(\text{i},\text{t})$ เฉพาะการเพิ่มสัมประสิทธิ์ 1/20 มาเป็นตัวคูณ ดังนั้นค่าความน่าจะเป็นของพจนานุกรมสามารถคำนวณได้จากสมการต่อไปนี้

$$\Pr\left(\text{DICTIONARY}_{threshold}\overrightarrow{Spectrum}\right) = \sum_{t \geq threshold} \Pr\left(m,t\right)$$

โดย DICTIONARY(*DinosaurPSM*) ประกอบด้วย 219,136,251,374 เพปไทด์ และมีค่าความน่าจะเป็นเท่ากับ 0.00018 สำหรับการทดสอบความสำคัญเชิงสถิติของ *DinosuarPSM* ที่พบในฐานข้อมูล UniProt+ ที่มีความยาว n เท่ากับ 194,613,142 กรดแอมิโนจากจำนวนโปรตีนทั้งหมด 546,799 เส้น เป้าหมายคือการคำนวณค่า

$$n \cdot \Pr(\text{DICTIONARY}(DinosaurPSM))$$

ซึ่งเป็นค่าประมาณของจำนวนเพปไทด์จาก $\text{DICTIONARY}(DinosaurPSM)$ ที่คาดหวังว่าจะพบในดีคอยโปรตีโอมความยาว n กรดแอมิโน เนื่องจาก $\Pr(\text{DICTIONARY}(DinosaurPSM))$ = 0.00018 ดังนั้นจำนวนเพปไทด์ที่คาดหวังว่าจะพบในดีคอยโปรตีโอมเท่ากับ 35,030 ซึ่งหมายถึงสามารถพบมากกว่าหมื่นเพปไทด์ที่มีค่าคะแนนอย่างน้อยเท่ากับคะแนนของ *DinosaurPeptide* เมื่อนำไปเทียบกับ $\overrightarrow{DinosaurSpectrum}$ ในฐานข้อมูลดีคอย จึงไม่น่าแปลกใจถ้าพบ *DinosuarPSM* ในฐานข้อมูล UniProt+ และสรุปได้ว่า *DinosaurPeptide* เป็นเพียงผลข้างเคียงจากการคำนวณทางสถิติมากกว่าที่จะเป็นเพปไทด์ของทีเร็กซ์จริง คำถามถัดไปคือทีเร็กซ์เพปไทด์อื่นที่รายงานนั้นเป็นเพปไทด์ของทีเร็กซ์จริงหรือไม่

เพปไทด์ของทีเร็กซ์เป็นเพียงโปรตีนปนเปื้อนหรือขุมสมบัติล้านปี

ปริศนาฮีโมโกลบิน

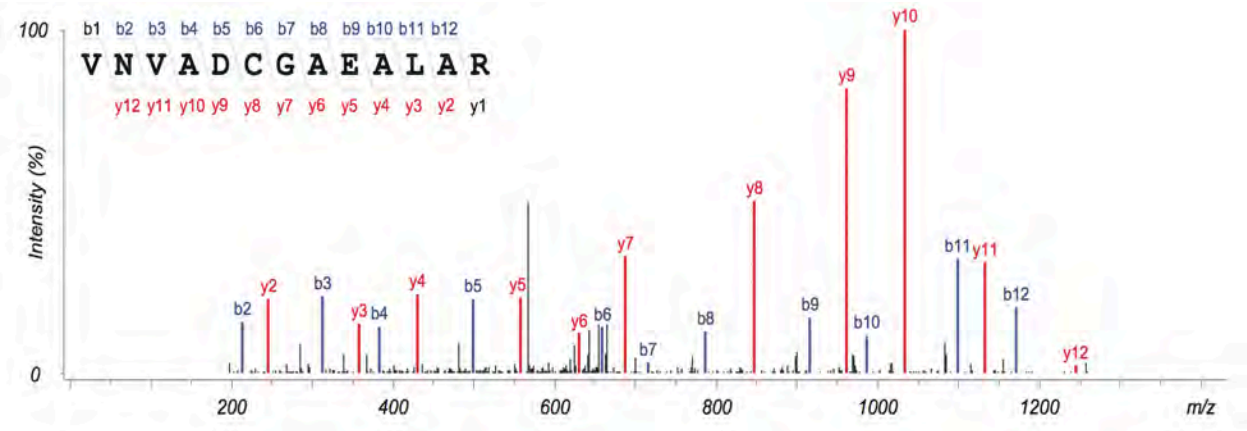
หลังจากได้รับคำวิจารณ์เกี่ยวกับค่าสถิติที่ใช้เบื้องหลังในการสรุปผลเกี่ยวกับเพปไทด์ของทีเร็กซ์แอสรายอมรับปัญหาบางส่วนที่เกิดจากการวิเคราะห์ข้อมูลและทำการถอน *DinosaurPeptide* จากการตีความว่าเป็นของเพปไทด์ที่แสดง *DinosaurSpectrum* และทำการปรับบางเพปไทด์ที่เคยเสนอไว้ว่าเป็นเพปไทด์ของทีเร็กซ์ออก รวม

ทั้งเปิดข้อมูลทั้งหมด 31,372 สเปกตรัมที่ได้จากฟอสซิลของทีเร็กซ์ หลังจากนั้นนักวิทยาศาสตร์คนอื่นได้ทำการวิเคราะห์ข้อมูลใหม่สำหรับทุกสเปกตรัมและยืนยันว่าถึงแม้บางเพปไทด์ที่ถูกรายงานโดยแอสราอาจไม่ถูกต้อง แต่ก็มีชุดเพปไทด์ที่มีความสำคัญจริงในเชิงสถิติ รูปที่ 8.8 แสดงเพปไทด์ 7 เส้น (P1-P7) รายงานโดยแอสราที่อาจเป็นตัวแทนคอลลาเจนเพปไทด์ของทีเร็กซ์รวมทั้งฮีโมโกลบินเพปไทด์ (P8) อักษรสีแดงแสดงกรดแอมิโนที่แตกต่างไปจากเพปไทด์ในฐานข้อมูลยูนิพรอตและกรดแอมิโน P_{oh} แสดงไฮดรอกซิโพรลีน (hydroxyproline) ซึ่งเป็นรูปแบบหนึ่งของโพรลีน (proline) ที่ถูกดัดแปลงและพบโดยทั่วไปในคอลลาเจน

ID	Peptide	Protein	Probability	*n* * Probability
P1	GLVGAPGLRGLPGK	Collagen α1t2	$1.8 * 10^{-4}$	36,000
P2	GVVGLP$_{oh}$GQR	Collagen α1t1	$7.6 * 10^{-8}$	16
P3	GVQGPP$_{oh}$GPQGPR	Collagen α1t1	$7.9 * 10^{-11}$	0.016
P4	GATGAP$_{oh}$GIAGAP$_{oh}$GFP$_{oh}$GAR	Collagen α1t1	$3.2 * 10^{-12}$	0.00064
P5	GLPGESGAVGPAGPIGSR	Collagen α2t1	$9.9 * 10^{-14}$	$2.0 * 10^{-5}$
P6	GSAGPP$_{oh}$GATGFP$_{oh}$GAAGR	Collagen α1t1	$3.2 * 10^{-14}$	$6.4 * 10^{-6}$
P7	GAPGPQGPSGAP$_{oh}$GPK	Collagen α1t1	$7.0 * 10^{-16}$	$1.4 * 10^{-7}$
P8	VNVADCGAEALAR	Hemoglobin β	$7.8 * 10^{-17}$	$1.6 * 10^{-8}$

รูปที่ 8.8 เพปไทด์ 7 เส้น (P1-P7) ที่อาจเป็นตัวแทนคอลลาเจนเพปไทด์ของทีเร็กซ์รายงานโดยแอสราและฮีโมโกลบินเพปไทด์ (P8) ที่ไม่ได้ถูกรายงาน

(*ที่มา*: *รูปที่* 11.13 *ของ [52]*)



รูปที่ 8.9 สเปกตรัมของทีเร็กซ์คุณภาพสูงที่ตรงกับฮีโมโกลบินเพปไทด์ของนกกระจอกเทศ VNVADCGGAEALAR ซึ่งพรีฟิกซ์และซัฟฟิกซ์ที่พบส่วนใหญ่ถูกแสดงโดยพีคที่มีค่า intensity สูง รวมทั้งผ่านการยืนยันผลโดยการระบุเพปไทด์จากสเปกตรัม

(*ที่มา*: *รูปที่* 11.14 *ของ [52]*)

ชุดสเปกตรัมที่แอสราเปิดเผยออกมานั้นทำให้เกิดคำถามเพิ่มเติม โดยแมทธิว ฟิตซ์กิบบอน (Matthew Fitzgibbon) และ มาร์ติน แมคอินทอช (Martin McIntosh) พบสเปกตรัม (รูปที่ 8.9) ซึ่งตรงกับฮีโมโกลบินเพปไทด์ของนกกระจอกเทศ ดังนั้นจึงมีการเพิ่มเพปไทด์ของทีเร็กซ์ในบรรทัดที่ 8 (P8) ของรูปที่ 8.8 โดยฮีโมโกลบิน PSM นี้ไม่ได้ถูกรายงานไว้โดยแอสรา ทั้งที่ในความเป็นจริงแล้วเพปไทด์เส้นนี้มีความสำคัญเชิงสถิติมากกว่าเพปไทด์ทุกเส้นที่แอสรารายงานไว้ว่าเป็นคอลลาเจนเพปไทด์ของทีเร็กซ์ และมีความน่าสนใจมากกว่าถ้าฮีโมโกลบินเพปไทด์นี้เป็นของทีเร็กซ์จริง เนื่องจากฮีโมโกลบินมีความอนุรักษ์ (conserved) ระหว่างสิ่งมีชีวิตต่างๆ น้อยกว่าคอลลาเจนมาก ตัวอย่างเช่น เบตาเชนฮีโมโกลบินของมนุษย์มีความยาว 146 กรดแอมิโน โดยยาวแตกต่างจากหนู (mouse) จิงโจ้ (kangaroo) และไก่ (chicken) เท่ากับ 27, 38, และ 45 กรดแอมิโน ตามลำดับ นอกจากนี้ยังไม่เคยพบฮีโมโกลบินเพปไทด์ที่สมบูรณ์ในฟอสซิลที่ขุดพบโดยทั่วไป แต่มักถูกพบในถ้ำแถวยุโรปโดยเป็นแหล่งฟอสเฟตเพื่อผลิตดินปืนระหว่างสงครามโลกครั้งที่ 1 เนื่องจากแอสราทำการวิเคราะห์ตัวอย่างนกกระจอกเทศหลายตัวอย่างก่อนมาวิเคราะห์ตัวอย่างทีเร็กซ์ ฟิตซ์กิบบอนและแมคอินทอชจึงตั้งข้อสงสัยว่าฮีโมโกลบินเพปไทด์ที่พบนั้นอาจเป็นผลจากการปนเปื้อนของตัวอย่างในลักษณะที่เรียกว่าแคร์รี่โอเวอร์ (carry over) โดยเพปไทด์ของการทดลองครั้งที่ผ่านมายังตกค้างอยู่ในเครื่องแมสสเปกโทรเมตรี และการปนเปื้อนนี้เป็นสิ่งที่เกิดขึ้นได้ในห้องปฏิบัติการทางโพรทีโอมิกส์ ผู้เชี่ยวชาญเทคนิคแมสสเปกโทรเมตรีไม่แปลกใจถ้าพบว่าตัวอย่างที่ทำการวิเคราะห์อยู่นั้นมีเคอราทิน (keratin) ของมนุษย์ปนอยู่ด้วย ทั้งนี้เพราะอากาศในห้องมักมีชิ้นส่วนผิวหนังมนุษย์ขนาดเล็กมากหลายล้านชิ้นเป็นส่วนประกอบ

ถ้าฮีโมโกลบินเพปไทด์เป็นแคร์รี่โอเวอร์หมายความว่าตัวอย่างทีเร็กซ์เกิดการปนเปื้อนและหมายถึงเพปไทด์ต่างๆ ที่มีการรายงานว่าเป็นของทีเร็กซ์ไม่มีความหมาย อย่างไรก็ตามแอสราสามารถแสดงได้ว่าไม่มีการปนเปื้อนในการทดลองของเขาและฮีโมโกลบินของนกกระจอกเทศเป็นหนึ่งในเพปไทด์ของทีเร็กซ์ ซึ่งทำให้เพิ่มจำนวนกลุ่มโปรตีนที่พบได้ในฟอสซิลโบราณที่มีอายุมากกว่า 68 ล้านปีนอกเหนือจากคอลลาเจน อย่างไรก็ตามถ้าฟอสซิลทีเร็กซ์ของฮอนเนอร์เป็นขุมสมบัติโปรตีนโบราณและเชื่อว่าเพปไทด์ฮีโมโกลบินมาจากทีเร็กซ์จริง ทำไมถึงจำกัดการสืบค้นเพปไทด์กับเฉพาะฐานข้อมูลกลุ่มคอลลาเจนเพปไทด์และคอลลาเจนเพปไทด์ที่มีการแปรผัน ทำไมถึงไม่สืบค้นกับฐานข้อมูลโปรตีนทั้งหมดของสัตว์มีกระดูกสันหลัง โดยยังสามารถใช้เงื่อนไขเดียวกับที่แอสราใช้ได้ เช่น เพปไทด์ที่สืบค้นกับเพปไทด์ในฐานข้อมูลแตกต่างกันได้อย่างมาก 1 กรดแอมิโน เป็นต้น ซึ่งถ้าใช้เงื่อนไขนี้ จะได้เพปไทด์เพิ่มเติมจากทั้งนกกระจอกเทศ ไก่ หนู และมนุษย์ ซึ่งเพปไทด์เหล่านี้ทำให้การสรุปผลของแอสราว่าสัตว์ปีกและไดโนเสาร์มีความเกี่ยวเนื่องกันในเชิงอณูชีววิทยานั้นมีน้ำหนักน้อยลง ในทางกลับกันถ้ายกเลิกเพปไทด์เพิ่มเติมเหล่านี้ทั้งหมดโดยระบุว่าเป็นผลข้างเคียงของค่าทางสถิติ ก็จำเป็นต้องยกเลิกเพปไทด์ของทีเร็กซ์ในรูปที่ 8.8

ข้อโต้เถียงเกี่ยวกับดีเอ็นเอของไดโนเสาร์

ถึงแม้ผลงานตีพิมพ์เกี่ยวกับเพปไทด์ของไดโนเสาร์ยังคงเป็นที่โต้แย้งกันอยู่ ในความเป็นจริงแล้วผลงานนี้ไม่ใช่งานวิจัยแรกที่รายงานเกี่ยวกับสารพันธุกรรมของไดโนเสาร์ ในปี ค.ศ. 1994 สก็อตต์ วู้ดวาร์ด (Scott Woodward)

ประกาศว่าได้ทำการหาลำดับเบสดีเอ็นเอของกระดูกไดโนเสาร์ที่มีอายุ 80 ล้านปี คำวิจารณ์ที่ร้อนแรงสุดต่อผลงานวิจัยนี้ คือ เชื่อหรือไม่ (Believe it or not) โดยมาร์ค ชไวท์เซอร์ (Mark Schweitzer) เป็นผู้พิสูจน์ว่าผลงานของวู้ดวาร์ดเป็นเพียงรหัสพันธุกรรมของมนุษย์ที่ปนเปื้อน

บทส่งท้าย

การเปลี่ยนแปลงสายเพปไทด์หลังการแปลรหัส

อัลกอริทึม PSM สามารถใช้ในการค้นหาสายเพปไทด์ที่มีอยู่ในฐานข้อมูลเท่านั้น ซึ่งฐานข้อมูลมักมีเฉพาะข้อมูลสายเพปไทด์ปกติโดยไม่รวมสายเพปไทด์ที่แปรผันไป

หยุดคิด	สามารถปรับแก้อัลกอริทึม PSM ให้ค้นหาเพปไทด์ที่มีการแปรผันได้อย่างไร

การค้นหาเพปไทด์ในฐานข้อมูลโดยอนุญาตให้เพปไทด์นั้นมีการแปรผันไปมากสุด n กรดแอมิโน วิธีการหนึ่งคือ สร้างสายเพปไทด์ทุกรูปแบบที่เป็นไปได้ที่มีการแปรผันไปอย่างมาก n กรดแอมิโน และนำเพปไทด์เหล่านี้เข้าไปรวมเป็นส่วนหนึ่งของโปรตีโอมตั้งต้น จากนั้นทำการรันอัลกอริทึม PSM กับโปรตีโอมใหม่นี้ อย่างไรก็ตามแนวทางนี้ไม่เหมาะสมในเชิงปฏิบัติเนื่องจากโปรตีโอมใหม่นี้จะมีชุดของเพปไทด์เพิ่มขึ้นเป็นจำนวนมากแม้ในกรณีอนุญาตให้มีการแปรผันได้เพียง 1 กรดแอมิโน

หยุดคิด	เพปไทด์ที่ต้องสร้างขึ้นใหม่มีจำนวนทั้งสิ้นกี่เส้น ถ้าสายเพปไทด์ยาว L กรดแอมิโน และมีการแปรผันได้อย่างมาก n กรดแอมิโน

นอกจากต้องค้นหาเพปไทด์ที่มีการแปรผันข้างต้นจากฐานข้อมูลโปรตีโอมแล้ว ยังต้องค้นหาเพปไทด์ที่มีการดัดแปรหลังการแปลรหัส (posttranslational modification) ซึ่งกรดแอมิโนบางตำแหน่งจะถูกเปลี่ยนแปลงไปหลังกระบวนการแปลรหัสจากเอ็มอาร์เอ็นเอมาเป็นโปรตีน ซึ่งในความเป็นจริงแล้วโปรตีนแทบทั้งหมดจะถูกดัดแปรหลังการแปลรหัส โดยลักษณะการดัดแปรที่ถูกค้นพบและรายงานมีจำนวนหลายร้อยประเภท เช่น ปฏิกิริยาเอนไซม์ของหลายโปรตีนถูกควบคุมผ่านการเติมหมู่ฟอสเฟต (phosphorylation) ที่ตำแหน่งกรดแอมิโนจำเพาะและกลับด้านได้ (reversible) โดยโปรตีนกลุ่มไคเนส (protein kinase) ทำหน้าที่เพิ่มกลุ่มฟอสเฟตในขณะที่โปรตีนกลุ่มฟอสฟาเทส (protein phosphatase) ทำหน้าที่ดึงกลุ่มฟอสเฟตออก จากรูปที่ 8.8 สังเกตได้ว่ากรดแอมิโนโพรลีน (proline น้ำหนัก 97) ในเพปไทด์ของทีเร็กซ์แทบทั้งหมดถูกดัดแปรไปเป็นไฮดรอกซิโพรลีน (hydroxyproline น้ำหนัก 113) ซึ่งเป็นองค์ประกอบหลักและมีความสำคัญในการเพิ่มความเสถียรให้กับคอลลาเจน

diphthamide เป็นอีกตัวอย่างของการดัดแปรหลังการแปลรหัสที่พบไม่มากนักแต่มีความสำคัญมากเช่นกัน ซึ่งเป็นผลจากการดัดแปรฮิสทิดีน (histidine) โดยพบเฉพาะในโปรตีน protein synthesis elongation factor-2 ซึ่งเป็นโปรตีนที่พบในยูแคริโอต (eukaryote) ทั้งหมด นักวิจัยพบว่า diphthamide นี้เป็นเป้าหมายของหลายสารพิษ (toxin) ที่ผลิตโดยแบคทีเรียก่อโรคหลายชนิด และนำไปสู่คำถามว่าทำไมสิ่งมีชีวิตในกลุ่มยูแคริโอตยังรักษาลักษณะการเปลี่ยนแปลงนี้ไว้ ไม่สูญหายไปในขบวนการวิวัฒนาการ เพราะลักษณะการเปลี่ยนแปลงนี้ทำให้ยูแคริโอตมีความเสี่ยงในการเกิดโรคจากแบคทีเรียก่อโรค ดังนั้นสมมติฐานคือ ลักษณะการเปลี่ยนแปลงนี้น่าจะมีความสำคัญเพียงแต่เรายังไม่ทราบฟังก์ชันการทำงาน

ตัวอย่างโปรแกรมที่มีการใช้งานอย่างแพร่หลาย

ปัจจุบันมีการประยุกต์ใช้เทคนิคแมสสเปกโทรเมตรีในงานวิจัยต่างๆ อย่างกว้างขวาง เช่น การศึกษาเกี่ยวกับโรคมะเร็ง [214, 215] การศึกษาเมแทบอลิซึมของมะเร็ง [216] การหาโปรตีนที่เป็นตัวชี้วัดทางชีวภาพของการเป็นโรคมะเร็งตับอ่อน [217] โรคเอสแอลอี [218] การศึกษาการแปรผันของโปรตีนกลุ่ม SPOP ในผู้ป่วยโรคมะเร็ง [219] การศึกษาปฏิสัมพันธ์ระหว่างโปรตีน [220] การวิเคราะห์โครงสร้างของโปรตีน [221] การศึกษาเกี่ยวกับการทำลายดีเอ็นเอและ protein ubiquitylation [222] เป็นต้น

โดยโปรแกรมที่มีการใช้งานกันอย่างแพร่หลายโปรแกรมหนึ่งคือ โปรแกรม Mascot ซึ่งมีการพัฒนาและตีพิมพ์ผลงานวิจัยครั้งแรกในปี ค.ศ. 1999 [223] Mascot เป็นโปรแกรมที่ใช้สืบค้นฐานข้อมูลของโปรตีนโดยใช้ข้อมูลแมสสเปกตรัม โดยในปี ค.ศ. 2008 มีการพัฒนาวิธีคำนวณค่าคะแนนที่มีความเสถียรมากขึ้น [224] โปรแกรม MaxQuant เป็นอีกโปรแกรมที่มีการใช้งานอย่างแพร่หลายโดย MaxQuant ตีพิมพ์ครั้งแรกในปี ค.ศ. 2008 [225] โดยเป็นการรวมชุดของอัลกอริทึมที่ใช้ในการวิเคราะห์ข้อมูลแมสสเปกโทรเมตรีที่มีความละเอียดสูง โดยใช้การวิเคราะห์สหสัมพันธ์และทฤษฎีกราฟเป็นเครื่องมือในการตรวจจับพีค ชุดของไอโซโทป (isotope cluster) และคู่ของเพปไทด์ที่มีการติดป้ายด้วยกรดแอมิโน (SILAC; stable isotope labelling by/with amino acids in cell culture) ในรูปแบบของอ็อบเจกต์ 3 มิติของค่า m/z ค่า elution time และค่า signal intensity ทั้งนี้ MaxQuant เวอร์ชันแรกใช้ Mascot ในการหาชุดของเพปไทด์ที่ทำให้เกิดสเปกตรัมหนึ่งๆ ในปี ค.ศ. 2016 โปรแกรม Max Quant [226] มีการพัฒนาเพิ่มเติมโดยสนับสนุนแพลตฟอร์มแมสสเปกโทรเมตรีที่หลากหลายมากขึ้น รวมทั้งมีการนำ Andromeda [227] เข้ามาแทน Mascot ซึ่งเป็นโปรแกรมเชิงพาณิชย์ นอกจาก Mascot และ Andromeda โปรแกรม SEQUEST [228] เป็นอีกโปรแกรมที่มีการใช้งานอย่างแพร่หลายโดยตีพิมพ์ครั้งแรกในปี ค.ศ. 1994 และมีการนำเสนอ Tide [229] ที่พัฒนาอัลกอริทึม SEQUEST โดยเน้นความเร็ว โปรแกรม MSFragger [230] เป็นอีกโปรแกรมที่ใช้ในการสืบค้นฐานข้อมูลโปรตีนโดยใช้ข้อมูลสเปกตรัม

นอกจากงานวิจัยข้างต้นที่เน้นอัลกอริทึมและวิธีการวิเคราะห์ข้อมูล ผลงานตีพิมพ์อีกลักษณะหนึ่งคือการนำเสนอการทำงานแบบสายท่อ (pipeline) หรือกระแสงาน (workflow) ที่ใช้ในการวิเคราะห์ข้อมูลแมสสเปกโทรเมตรี เช่น [231, 232] เป็นต้น เถา เฉิน (Tao Chen) และคณะ [233] ตีพิมพ์บทปริทัศน์เกี่ยวกับเว็บต่างๆ ที่

สนับสนุนงานวิจัยที่เกี่ยวข้องกับแมสสเปกโทรเมตรีสำหรับงานทางด้านโพรทีโอมิกส์ รายละเอียดเพิ่มเติมเกี่ยวกับเทคโนโลยีที่เกี่ยวข้องกับแมสสเปกโทรเมตรีและการประยุกต์ใช้สามารถศึกษาเพิ่มเติมได้จาก [234-236] และวิธีการเชิงคำนวณที่เกี่ยวข้องกับการวิเคราะห์ข้อมูลแมสสเปกโทรเมตรีสำหรับงานวิจัยเชิงโพรทีโอมิกส์สามารถศึกษาเพิ่มเติมได้จากบทในหนังสือโดยลี (Li) และคณะ [237] นอกจากนี้ทอมมี่ วาลิกันกัส (Tommi Välikangas) และคณะทำการเปรียบเทียบกระแสงานที่ใช้ในการวิเคราะห์ข้อมูลแมสสเปกโทรเมตรีของโพรทีโอมิกส์ [238]

แบบฝึกหัดบทที่ 8

เขียนโปรแกรมเพื่อแก้ปัญหาที่เกี่ยวข้องกับการวิเคราะห์ข้อมูลแมสสเปกโทรเมตรีโดยใช้โจทย์ที่โรซาลินด์ต่อไปนี้

1) Calculating Protein Mass (http://rosalind.info/problems/prtm/)
2) Inferring Protein from Spectrum (http://rosalind.info/problems/spec/)
3) Comparing Spectra with the Spectral Convolution (http://rosalind.info/problems/conv/)
4) Matching a Spectrum to a Protein (http://rosalind.info/problems/prsm/)
5) Using the Spectrum Graph to Infer Peptides (http://rosalind.info/problems/sgra/)
6) Inferring Peptide from Full Spectrum (http://rosalind.info/problems/full/)

เอกสารอ้างอิง

1. Altschul, S.F., et al., *Basic local alignment search tool.* J Mol Biol, 1990. 215(3): p. 403-10.
2. Lander, E.S., et al., *Initial sequencing and analysis of the human genome.* Nature, 2001. 409(6822): p. 860-921.
3. Hutchinson, J., *Congenital Absence of Hair and Mammary Glands with Atrophic Condition of the Skin and its Appendages, in a Boy whose Mother had been almost wholly Bald from Alopecia Areata from the age of Six.* Medico-Chirurgical Transactions, 1886. 69: p. 473-477.
4. De Sandre-Giovannoli, A., et al., *Lamin a truncation in Hutchinson-Gilford progeria.* Science, 2003. 300(5628): p. 2055.
5. Wendelin, D.S., D.N. Pope, and S.B. Mallory, *Hypertrichosis.* J Am Acad Dermatol, 2003. 48(2): p. 161-79; quiz 180-1.
6. Zhu, H., et al., *X-Linked Congenital Hypertrichosis Syndrome Is Associated with Interchromosomal Insertions Mediated by a Human-Specific Palindrome near SOX*3. American Journal of Human Genetics, 2011. 88(6): p. 819-826.
7. Sun, M., et al., *Copy-Number Mutations on Chromosome* 17*q*24.2-*q*24.3 *in Congenital Generalized Hypertrichosis Terminalis with or without Gingival Hyperplasia.* American Journal of Human Genetics, 2009. 84(6): p. 807-813.
8. Fantauzzo, K.A., et al., *A position effect on TRPS*1 *is associated with Ambras syndrome in humans and the Koala phenotype in mice.* Human Molecular Genetics, 2008. 17(22): p. 3539-3551.
9. Tadin, M., et al., *Complex cytogenetic rearrangement of chromosome* 8*q in a case of Ambras syndrome.* Am J Med Genet, 2001. 102(1): p. 100-4.
10. Scherer, S.W., et al., *Physical mapping of the split hand/split foot locus on chromosome* 7 *and implication in syndromic ectrodactyly.* Hum Mol Genet, 1994. 3(8): p. 1345-54.
11. Genomes Project, C., et al., *A map of human genome variation from population-scale sequencing.* Nature, 2010. 467(7319): p. 1061-73.
12. Genomes Project, C., et al., *An integrated map of genetic variation from* 1,092 *human genomes.* Nature, 2012. 491(7422): p. 56-65.
13. Genomes Project, C., et al., *A global reference for human genetic variation.* Nature, 2015. 526(7571): p. 68-74.
14. Sudmant, P.H., et al., *An integrated map of structural variation in* 2,504 *human genomes.* Nature, 2015. 526(7571): p. 75-81.

15. Siva, N., *UK gears up to decode* 100,000 *genomes from NHS patients.* Lancet, 2015. 385(9963): p. 103-4.
16. Heather, J.M. and B. Chain, *The sequence of sequencers: The history of sequencing DNA.* Genomics, 2016. 107(1): p. 1-8.
17. Goodwin, S., J.D. McPherson, and W.R. McCombie, *Coming of age: ten years of next-generation sequencing technologies.* Nat Rev Genet, 2016. 17(6): p. 333-51.
18. Miller, J.R., S. Koren, and G. Sutton, *Assembly algorithms for next-generation sequencing data.* Genomics, 2010. 95(6): p. 315-27.
19. Wajid, B. and E. Serpedin, *Review of General Algorithmic Features for Genome Assemblers for Next Generation Sequencers.* Genomics, Proteomics & Bioinformatics, 2012. 10(2): p. 58-73.
20. Compeau, P.E.C., P.A. Pevzner, and G. Tesler, *How to apply de Bruijn graphs to genome assembly.* Nature Biotechnology, 2011. 29: p. 987.
21. Mielczarek, M. and J. Szyda, *Review of alignment and SNP calling algorithms for next-generation sequencing data.* J Appl Genet, 2016. 57(1): p. 71-9.
22. Burge, C. and S. Karlin, *Prediction of complete gene structures in human genomic DNA.* J Mol Biol, 1997. 268(1): p. 78-94.
23. Tattini, L., R. D'Aurizio, and A. Magi, *Detection of Genomic Structural Variants from Next-Generation Sequencing Data.* Front Bioeng Biotechnol, 2015. 3: p. 92.
24. Guan, P. and W.-K. Sung, *Structural variation detection using next-generation sequencing data: A comparative technical review.* Methods, 2016. 102: p. 36-49.
25. Ruzzo, E.K., et al., *Genetics*, A.M. Husain, Editor., Springer Publishing Company: New York. p. 11-27.
26. James D. Watson, T.A.B., Stephen P. Bell, Alexander Gann, Michael Levine, Richard Losick *Molecular Biology of the Gene*. Books a la Carte. 2013: Pearson; 7 edition (March 2, 2013).
27. Jocelyn E. Krebs, E.S.G., Stephen T. Kilpatrick, *Lewin's genes XI* 11*st Edition*. Lewins Genes. 2014: Jones & Bartlett Learning; 11 edition (January 14, 2013). 940.
28. Thomas Shafee, R.L., *Eukaryotic and prokaryotic gene structure.* WikiJournal of Medicine, 2017. 4(1): p. 2.
29. Crick, F.H., *On protein synthesis.* Symp Soc Exp Biol, 1958. 12: p. 138-63.
30. Crick, F., *Central dogma of molecular biology.* Nature, 1970. 227(5258): p. 561-3.
31. Horgan, R.P. and L.C. Kenny, *'Omic' technologies: genomics, transcriptomics, proteomics and metabolomics.* The Obstetrician & Gynaecologist, 2011. 13(3): p. 189-195.

32. Fischer, H.P., *Mathematical modeling of complex biological systems: from parts lists to understanding systems behavior.* Alcohol Res Health, 2008. 31(1): p. 49-59.
33. Lau, J.W., et al., *The Cancer Genomics Cloud: Collaborative, Reproducible, and Democratized-A New Paradigm in Large-Scale Computational Research.* Cancer Res, 2017. 77(21): p. e3-e6.
34. Ewing, B. and P. Green, *Base-calling of automated sequencer traces using phred. II. Error probabilities.* Genome Res, 1998. 8(3): p. 186-94.
35. Mouse Genome Sequencing, C., et al., *Initial sequencing and comparative analysis of the mouse genome.* Nature, 2002. 420(6915): p. 520-62.
36. Gibbs, R.A., et al., *Genome sequence of the Brown Norway rat yields insights into mammalian evolution.* Nature, 2004. 428(6982): p. 493-521.
37. Ostrander, E.A. and R.K. Wayne, *The canine genome.* Genome Res, 2005. 15(12): p. 1706-16.
38. Chimpanzee, S. and C. Analysis, *Initial sequence of the chimpanzee genome and comparison with the human genome.* Nature, 2005. 437(7055): p. 69-87.
39. Rhesus Macaque Genome, S., et al., *Evolutionary and biomedical insights from the rhesus macaque genome.* Science, 2007. 316(5822): p. 222-34.
40. Wade, C.M., et al., *Genome sequence, comparative analysis, and population genetics of the domestic horse.* Science, 2009. 326(5954): p. 865-7.
41. Samollow, P.B., *The opossum genome: insights and opportunities from an alternative mammal.* Genome Res, 2008. 18(8): p. 1199-215.
42. Zimin, A.V., et al., *A whole-genome assembly of the domestic cow, Bos taurus.* Genome Biol, 2009. 10(4): p. R42.
43. Li, R., et al., *The sequence and de novo assembly of the giant panda genome.* Nature, 2010. 463(7279): p. 311-7.
44. Castoe, T.A., et al., *Sequencing the genome of the Burmese python (Python molurus bivittatus) as a model for studying extreme adaptations in snakes.* Genome Biol, 2011. 12(7): p. 406.
45. Goff, S.A., et al., *A draft sequence of the rice genome (Oryza sativa L. ssp. japonica).* Science, 2002. 296(5565): p. 92-100.
46. Rahman, A.Y., et al., *Draft genome sequence of the rubber tree Hevea brasiliensis.* BMC Genomics, 2013. 14: p. 75.
47. Tang, C., et al., *The rubber tree genome reveals new insights into rubber production and species adaptation.* Nat Plants, 2016. 2(6): p. 16073.
48. Singh, R., et al., *Oil palm genome sequence reveals divergence of interfertile species in Old and New worlds.* Nature, 2013. 500(7462): p. 335-9.

49. Teh, B.T., et al., *The draft genome of tropical fruit durian (Durio zibethinus).* Nat Genet, 2017. 49(11): p. 1633-1641.
50. Marx, V., *The DNA of a nation.* Nature, 2015. 524(7566): p. 503-5.
51. Head, S.R., et al., *Library construction for next-generation sequencing: overviews and challenges.* Biotechniques, 2014. 56(2): p. 61-4, 66, 68, passim.
52. Compeau, P. and P. Pevzner, *Bioinformatics algorithms : an active learning approach*. 2nd Edition. ed. 2015, La Jolla, CA: Active Learning Publishers. volumes.
53. Knierim, E., et al., *Systematic comparison of three methods for fragmentation of long-range PCR products for next generation sequencing.* PLoS One, 2011. 6(11): p. e28240.
54. Wichadakul, D., et al., *Insights from the genome of Ophiocordyceps polyrhachis-furcata to pathogenicity and host specificity in insect fungi.* BMC Genomics, 2015. 16: p. 881.
55. Boetzer, M., et al., *Scaffolding pre-assembled contigs using SSPACE.* Bioinformatics, 2011. 27(4): p. 578-9.
56. Luo, R., et al., *SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler.* Gigascience, 2012. 1(1): p. 18.
57. Luo, R., et al., *Erratum: SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler.* Gigascience, 2015. 4: p. 30.
58. Butler, J., et al., *ALLPATHS: de novo assembly of whole-genome shotgun microreads.* Genome Res, 2008. 18(5): p. 810-20.
59. Zerbino, D.R. and E. Birney, *Velvet: algorithms for de novo short read assembly using de Bruijn graphs.* Genome Res, 2008. 18(5): p. 821-9.
60. Simpson, J.T., et al., *ABySS: a parallel assembler for short read sequence data.* Genome Res, 2009. 19(6): p. 1117-23.
61. Nagarajan, N. and M. Pop, *Sequence assembly demystified.* Nat Rev Genet, 2013. 14(3): p. 157-67.
62. Warr, A., et al., *Exome Sequencing: Current and Future Perspectives.* G3 (Bethesda), 2015. 5(8): p. 1543-50.
63. Clayton-Smith, J., et al., *Whole-exome-sequencing identifies mutations in histone acetyltransferase gene KAT6B in individuals with the Say-Barber-Biesecker variant of Ohdo syndrome.* Am J Hum Genet, 2011. 89(5): p. 675-81.
64. Reinert, K., et al., *Alignment of Next-Generation Sequencing Reads.* Annu Rev Genomics Hum Genet, 2015. 16: p. 133-51.

65. Handelsman, J., *Metagenomics: application of genomics to uncultured microorganisms.* Microbiol Mol Biol Rev, 2004. 68(4): p. 669-85.

66. Thomas, T., J. Gilbert, and F. Meyer, *Metagenomics - a guide from sampling to data analysis.* Microb Inform Exp, 2012. 2(1): p. 3.

67. Schnable, P.S., et al., *The B73 maize genome: complexity, diversity, and dynamics.* Science, 2009. 326(5956): p. 1112-5.

68. Venter, J.C., et al., *The sequence of the human genome.* Science, 2001. 291(5507): p. 1304-51.

69. Montgomery, S.B., et al., *The origin, evolution, and functional impact of short insertion-deletion variants identified in 179 human genomes.* Genome Res, 2013. 23(5): p. 749-61.

70. Seo, J.S., et al., *De novo assembly and phasing of a Korean human genome.* Nature, 2016. 538(7624): p. 243-247.

71. Roach, J.C., et al., *Analysis of genetic inheritance in a family quartet by whole-genome sequencing.* Science, 2010. 328(5978): p. 636-9.

72. Ferragina, P. and B.B. Mishra, *Algorithms in Stringomics (I): Pattern-Matching against "Stringomes".* bioRxiv, 2014.

73. Mäkinen, V., et al., *Storage and Retrieval of Individual Genomes*, in *Research in Computational Molecular Biology: 13th Annual International Conference, RECOMB 2009, Tucson, AZ, USA, May 18-21, 2009. Proceedings*, S. Batzoglou, Editor. 2009, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 121-137.

74. Marcus, S., H. Lee, and M.C. Schatz, *SplitMEM: a graphical algorithm for pan-genome analysis with suffix skips.* Bioinformatics, 2014. 30(24): p. 3476-83.

75. Schneeberger, K., et al., *Simultaneous alignment of short reads against multiple genomes.* Genome Biol, 2009. 10(9): p. R98.

76. Dilthey, A., et al., *Improved genome inference in the MHC using a population reference graph.* Nat Genet, 2015. 47(6): p. 682-8.

77. Li, H. and R. Durbin, *Fast and accurate short read alignment with Burrows-Wheeler transform.* Bioinformatics, 2009. 25(14): p. 1754-60.

78. Li, H. and R. Durbin, *Fast and accurate long-read alignment with Burrows-Wheeler transform.* Bioinformatics, 2010. 26(5): p. 589-95.

79. Langmead, B., et al., *Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.* Genome Biol, 2009. 10(3): p. R25.

80. Langmead, B. and S.L. Salzberg, *Fast gapped-read alignment with Bowtie 2.* Nat Methods, 2012. 9(4): p. 357-9.

81. Liu, Y., B. Schmidt, and D.L. Maskell, *CUSHAW: a CUDA compatible short read aligner to large genomes based on the Burrows-Wheeler transform.* Bioinformatics, 2012. 28(14): p. 1830-7.
82. Li, R., et al., *SOAP*2: *an improved ultrafast tool for short read alignment.* Bioinformatics, 2009. 25(15): p. 1966-7.
83. Liu, C.M., et al., *SOAP*3: *ultra-fast GPU-based parallel alignment tool for short reads.* Bioinformatics, 2012. 28(6): p. 878-9.
84. Fonseca, N.A., et al., *Tools for mapping high-throughput sequencing data.* Bioinformatics, 2012. 28(24): p. 3169-77.
85. Thankaswamy-Kosalai, S., P. Sen, and I. Nookaew, *Evaluation and assessment of read-mapping by multiple next-generation sequencing aligners based on genome-wide characteristics.* Genomics, 2017. 109(3-4): p. 186-191.
86. Li, H., et al., *The Sequence Alignment/Map format and SAMtools.* Bioinformatics, 2009. 25(16): p. 2078-9.
87. Konopka, R.J. and S. Benzer, *Clock mutants of Drosophila melanogaster.* Proc Natl Acad Sci U S A, 1971. 68(9): p. 2112-6.
88. Stormo, G.D., et al., *Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli.* Nucleic Acids Res, 1982. 10(9): p. 2997-3011.
89. Wichadakul, D., et al., *A computational tool for the design of live attenuated virus vaccine based on microRNA-mediated gene silencing.* BMC Genomics, 2012. 13 **Suppl** 7: p. S15.
90. Huntzinger, E. and E. Izaurralde, *Gene silencing by microRNAs: contributions of translational repression and mRNA decay.* Nat Rev Genet, 2011. 12(2): p. 99-110.
91. Perez, J.T., et al., *MicroRNA-mediated species-specific attenuation of influenza A virus.* Nat Biotechnol, 2009. 27(6): p. 572-6.
92. Griffiths-Jones, S., *miRBase: the microRNA sequence database.* Methods Mol Biol, 2006. 342: p. 129-38.
93. Enright, A.J., et al., *MicroRNA targets in Drosophila.* Genome Biol, 2003. 5(1): p. R1.
94. John, B., et al., *Human MicroRNA targets.* PLoS Biol, 2004. 2(11): p. e363.
95. Bailey, T.L., et al., *MEME SUITE: tools for motif discovery and searching.* Nucleic Acids Res, 2009. 37(Web Server issue): p. W202-8.
96. Bailey, T.L., et al., *The MEME Suite.* Nucleic Acids Res, 2015. 43(W1): p. W39-49.
97. Das, M.K. and H.K. Dai, *A survey of DNA motif finding algorithms.* BMC Bioinformatics, 2007. 8 **Suppl** 7: p. S21.

98. Jayaram, N., D. Usvyat, and R.M. AC, *Evaluating tools for transcription factor binding site prediction.* BMC Bioinformatics, 2016.

99. Bryne, J.C., et al., *JASPAR, the open access database of transcription factor-binding profiles: new content and tools in the* 2008 *update.* Nucleic Acids Res, 2008. 36(Database issue): p. D102-6.

100. Khan, A., et al., *JASPAR* 2018: *update of the open-access database of transcription factor binding profiles and its web framework.* Nucleic Acids Res, 2018. 46(D1): p. D1284.

101. Khan, A., et al., *JASPAR* 2018: *update of the open-access database of transcription factor binding profiles and its web framework.* Nucleic Acids Res, 2018. 46(D1): p. D260-D266.

102. Khan, A. and A. Mathelier, *JASPAR RESTful API: accessing JASPAR data from any programming language.* Bioinformatics, 2017.

103. Mathelier, A., et al., *JASPAR* 2016: *a major expansion and update of the open-access database of transcription factor binding profiles.* Nucleic Acids Res, 2016. 44(D1): p. D110-5.

104. Mathelier, A., et al., *JASPAR* 2014: *an extensively expanded and updated open-access database of transcription factor binding profiles.* Nucleic Acids Res, 2014. 42(Database issue): p. D142-7.

105. Portales-Casamar, E., et al., *JASPAR* 2010: *the greatly expanded open-access database of transcription factor binding profiles.* Nucleic Acids Res, 2010. 38(Database issue): p. D105-10.

106. Sandelin, A., et al., *JASPAR: an open-access database for eukaryotic transcription factor binding profiles.* Nucleic Acids Res, 2004. 32(Database issue): p. D91-4.

107. Vlieghe, D., et al., *A new generation of JASPAR, the open-access repository for transcription factor binding site profiles.* Nucleic Acids Res, 2006. 34(Database issue): p. D95-7.

108. Weirauch, M.T., et al., *Determination and inference of eukaryotic transcription factor sequence specificity.* Cell, 2014. 158(6): p. 1431-1443.

109. Hume, M.A., et al., *UniPROBE, update* 2015: *new tools and content for the online database of protein-binding microarray data on protein-DNA interactions.* Nucleic Acids Res, 2015. 43(Database issue): p. D117-22.

110. Yang, L., et al., *TFBSshape: a motif database for DNA shape features of transcription factor binding sites.* Nucleic Acids Res, 2014. 42(Database issue): p. D148-55.

111. Heinemeyer, T., et al., *Databases on transcriptional regulation: TRANSFAC, TRRD and COMPEL.* Nucleic Acids Res, 1998. 26(1): p. 362-7.

112. Wingender, E., et al., *TRANSFAC: an integrated system for gene expression regulation.* Nucleic Acids Res, 2000. 28(1): p. 316-9.

113. Wingender, E., et al., *TRANSFAC: a database on transcription factors and their DNA binding sites.* Nucleic Acids Res, 1996. 24(1): p. 238-41.

114. Kulakovskiy, I.V., et al., *HOCOMOCO: a comprehensive collection of human transcription factor binding sites models.* Nucleic Acids Res, 2013. 41(Database issue): p. D195-202.
115. Kulakovskiy, I.V., et al., *HOCOMOCO: towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-Seq analysis.* Nucleic Acids Res, 2018. 46(D1): p. D252-D259.
116. Kulakovskiy, I.V., et al., *HOCOMOCO: expansion and enhancement of the collection of transcription factor binding sites models.* Nucleic Acids Res, 2016. 44(D1): p. D116-25.
117. Munch, R., et al., *PRODORIC: prokaryotic database of gene regulation.* Nucleic Acids Res, 2003. 31(1): p. 266-9.
118. Stormo, G.D., *DNA motif databases and their uses.* Curr. Protoc. Bioinform., 2015. 51(2.15.1-2.15.6).
119. Inukai, S., K.H. Kock, and M.L. Bulyk, *Transcription factor-DNA binding: beyond binding site motifs.* Curr Opin Genet Dev, 2017. 43: p. 110-119.
120. Mundade, R., et al., *Role of ChIP-seq in the discovery of transcription factor binding sites, differential gene regulation mechanism, epigenetic marks and beyond.* Cell Cycle, 2014. 13(18): p. 2847-52.
121. Schneider, T.D. and R.M. Stephens, *Sequence logos: a new way to display consensus sequences.* Nucleic Acids Res, 1990. 18(20): p. 6097-100.
122. Crooks, G.E., et al., *WebLogo: a sequence logo generator.* Genome Res, 2004. 14(6): p. 1188-90.
123. Gao, Z., L. Liu, and J. Ruan, *Logo2PWM: a tool to convert sequence logo to position weight matrix.* BMC Genomics, 2017. 18(Suppl 6): p. 709.
124. Doolittle, R.F., et al., *Simian sarcoma virus onc gene, v-sis, is derived from the gene (or genes) encoding a platelet-derived growth factor.* Science, 1983. 221(4607): p. 275.
125. D. Waterfield, M., et al., *Platelet-derived growth factor is structurally related to the putative transforming protein p28sis of simian sarcoma virus*. Vol. 304. 1983. 35-9.
126. Mark, M., F.M. Rijli, and P. Chambon, *Homeobox genes in embryogenesis and pathogenesis.* Pediatr Res, 1997. 42(4): p. 421-9.
127. Higgins, D.G. and P.M. Sharp, *CLUSTAL: a package for performing multiple sequence alignment on a microcomputer.* Gene, 1988. 73(1): p. 237-44.
128. Needleman, S.B. and C.D. Wunsch, *A general method applicable to the search for similarities in the amino acid sequence of two proteins.* J Mol Biol, 1970. 48(3): p. 443-53.
129. Xiong, J., *Essential Bioinformatics*. 2006, Cambridge: Cambridge University Press.
130. Dong, Q., et al., *Comparative plant genomics resources at PlantGDB.* Plant Physiol, 2005. 139(2): p. 610-8.

131. Duvick, J., et al., *PlantGDB: a resource for comparative plant genomics.* Nucleic Acids Res, 2008. 36(Database issue): p. D959-65.

132. Griffiths-Jones, S., et al., *Rfam: annotating non-coding RNAs in complete genomes.* Nucleic Acids Res, 2005. 33(Database issue): p. D121-4.

133. Markham, N.R. and M. Zuker, *UNAFold: software for nucleic acid folding and hybridization.* Methods Mol Biol, 2008. 453: p. 3-31.

134. Ambros, V., et al., *A uniform system for microRNA annotation.* RNA, 2003. 9(3): p. 277-9.

135. Jones-Rhoades, M.W. and D.P. Bartel, *Computational identification of plant microRNAs and their targets, including a stress-induced miRNA.* Mol Cell, 2004. 14(6): p. 787-99.

136. Rhoades, M.W., et al., *Prediction of plant microRNA targets.* Cell, 2002. 110(4): p. 513-20.

137. Mhuantong, W. and D. Wichadakul, *MicroPC (microPC): A comprehensive resource for predicting and comparing plant microRNAs.* BMC Genomics, 2009. 10: p. 366.

138. Numnark, S., et al., *C-mii: a tool for plant miRNA and target identification.* BMC Genomics, 2012. 13 **Suppl** 7: p. S16.

139. Smith, T.F. and M.S. Waterman, *Identification of common molecular subsequences.* J Mol Biol, 1981. 147(1): p. 195-7.

140. Thompson, J.D., D.G. Higgins, and T.J. Gibson, *CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.* Nucleic Acids Res, 1994. 22(22): p. 4673-80.

141. Notredame, C., D.G. Higgins, and J. Heringa, *T-Coffee: A novel method for fast and accurate multiple sequence alignment.* J Mol Biol, 2000. 302(1): p. 205-17.

142. Katoh, K., et al., *MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform.* Nucleic Acids Res, 2002. 30(14): p. 3059-66.

143. Edgar, R.C., *MUSCLE: multiple sequence alignment with high accuracy and high throughput.* Nucleic Acids Res, 2004. 32(5): p. 1792-7.

144. Sievers, F. and D.G. Higgins, *Clustal Omega for making accurate alignments of many protein sequences.* Protein Sci, 2018. 27(1): p. 135-145.

145. Sievers, F. and D.G. Higgins, *Clustal Omega, accurate alignment of very large numbers of sequences.* Methods Mol Biol, 2014. 1079: p. 105-16.

146. Choudhuri, S., *Chapter 6 - Sequence Alignment and Similarity Searching in Genomic Databases: BLAST and FASTA**, in *Bioinformatics for Beginners*. 2014, Academic Press: Oxford. p. 133-155.

147. Henikoff, S. and J.G. Henikoff, *Amino acid substitution matrices from protein blocks.* Proc Natl Acad Sci U S A, 1992. 89(22): p. 10915-9.

148. Henikoff, J.G. and S. Henikoff, *Blocks database and its applications.* Methods Enzymol, 1996. 266: p. 88-105.

149. Clamp, M., et al., *The Jalview Java alignment editor.* Bioinformatics, 2004. 20(3): p. 426-7.

150. Waterhouse, A.M., et al., *Jalview Version 2--a multiple sequence alignment editor and analysis workbench.* Bioinformatics, 2009. 25(9): p. 1189-91.

151. Ganss, B. and A. Jheon, *Zinc finger transcription factors in skeletal development.* Crit Rev Oral Biol Med, 2004. 15(5): p. 282-97.

152. Scheffzek, K. and S. Welti, *Pleckstrin homology (PH) like domains - versatile modules in protein-protein interaction platforms.* FEBS Lett, 2012. 586(17): p. 2662-73.

153. Wichadakul, D., S. Numnark, and S. Ingsriswang, *d-Omix: a mixer of generic protein domain analysis tools.* Nucleic Acids Res, 2009. 37(Web Server issue): p. W417-21.

154. Raghavachari, B., et al., *DOMINE: a database of protein domain interactions.* Nucleic Acids Res, 2008. 36(Database issue): p. D656-61.

155. Quevillon, E., et al., *InterProScan: protein domains identifier.* Nucleic Acids Res, 2005. 33(Web Server issue): p. W116-20.

156. Zdobnov, E.M. and R. Apweiler, *InterProScan--an integration platform for the signature-recognition methods in InterPro.* Bioinformatics, 2001. 17(9): p. 847-8.

157. Stanke, M. and S. Waack, *Gene prediction with a hidden Markov model and a new intron submodel.* Bioinformatics, 2003. 19 **Suppl** 2: p. ii215-25.

158. Bystroff, C. and A. Krogh, *Hidden Markov Models for prediction of protein features.* Methods Mol Biol, 2008. 413: p. 173-98.

159. Won, K.J., et al., *An evolutionary method for learning HMM structure: prediction of protein secondary structure.* BMC Bioinformatics, 2007. 8: p. 357.

160. Martin, J., J.F. Gibrat, and F. Rodolphe, *Analysis of an optimal hidden Markov model for secondary structure prediction.* BMC Struct Biol, 2006. 6: p. 25.

161. Pierleoni, A., P.L. Martelli, and R. Casadio, *PredGPI: a GPI-anchor predictor.* BMC Bioinformatics, 2008. 9: p. 392.

162. Wu, J. and J. Xie, *Hidden Markov model and its applications in motif findings.* Methods Mol Biol, 2010. 620: p. 405-16.

163. Heller, D., et al., *ssHMM: extracting intuitive sequence-structure motifs from high-throughput RNA-binding protein data.* Nucleic Acids Res, 2017. 45(19): p. 11004-11018.

164. Wang, T., et al., *Finding RNA-Protein Interaction Sites Using HMMs.* Methods Mol Biol, 2017. 1552: p. 177-184.

165. Sgouralis, I. and S. Presse, *An Introduction to Infinite HMMs for Single-Molecule Data Analysis.* Biophys J, 2017. 112(10): p. 2021-2029.

166. Choi, H., et al., *Sparsely correlated hidden Markov models with application to genome-wide location studies.* Bioinformatics, 2013. 29(5): p. 533-41.

167. Bian, J. and X. Zhou, *Hidden Markov Models in Bioinformatics: SNV Inference from Next Generation Sequence.* Methods Mol Biol, 2017. 1552: p. 123-133.

168. Malekpour, S.A., H. Pezeshk, and M. Sadeghi, *PSE-HMM: genome-wide CNV detection from NGS data using an HMM with Position-Specific Emission probabilities.* BMC Bioinformatics, 2016. 18(1): p. 30.

169. Malekpour, S.A., H. Pezeshk, and M. Sadeghi, *MGP-HMM: Detecting genome-wide CNVs using an HMM for modeling mate pair insertion sizes and read counts.* Math Biosci, 2016. 279: p. 53-62.

170. Abante, J., et al., *HiMMe: using genetic patterns as a proxy for genome assembly reliability assessment.* BMC Genomics, 2017. 18(1): p. 694.

171. Ernst, J. and M. Kellis, *Chromatin-state discovery and genome annotation with ChromHMM.* Nat Protoc, 2017. 12(12): p. 2478-2492.

172. Tsaousis, G.N., et al., *Predicting Alpha Helical Transmembrane Proteins Using HMMs*, in *Hidden Markov Models: Methods and Protocols*, D.R. Westhead and M.S. Vijayabaskar, Editors. 2017, Springer New York: New York, NY. p. 63-82.

173. Tsaousis, G.N., S.J. Hamodrakas, and P.G. Bagos, *Predicting Beta Barrel Transmembrane Proteins Using HMMs*, in *Hidden Markov Models: Methods and Protocols*, D.R. Westhead and M.S. Vijayabaskar, Editors. 2017, Springer New York: New York, NY. p. 43-61.

174. Nguyen, N.P., et al., *HIPPI: highly accurate protein family classification with ensembles of HMMs.* BMC Genomics, 2016. 17(Suppl 10): p. 765.

175. Lampros, C., et al., *HMMs in Protein Fold Classification.* Methods Mol Biol, 2017. 1552: p. 13-27.

176. Jablonowski, K., *Hidden Markov Models for Protein Domain Homology Identification and Analysis.* Methods Mol Biol, 2017. 1555: p. 47-58.

177. Huo, L., et al., *pHMM-tree: phylogeny of profile hidden Markov models.* Bioinformatics, 2017. 33(7): p. 1093-1095.

178. Sharma, R., et al., *Predicting MoRFs in protein sequences using HMM profiles.* BMC Bioinformatics, 2016. 17(Suppl 19): p. 504.

179. Sharan, M., et al., *APRICOT: an integrated computational pipeline for the sequence-based identification and characterization of RNA-binding proteins.* Nucleic Acids Res, 2017. 45(11): p. e96.

180. Finn, R.D., et al., *HMMER web server:* 2015 *update.* Nucleic Acids Res, 2015. 43(W1): p. W30-8.

181. Finn, R.D., J. Clements, and S.R. Eddy, *HMMER web server: interactive sequence similarity searching.* Nucleic Acids Res, 2011. 39(Web Server issue): p. W29-37.

182. Prakash, A., et al., *The HMMER Web Server for Protein Sequence Similarity Search.* Curr Protoc Bioinformatics, 2017. 60: p. 3 15 1-3 15 23.

183. Ferles, C., W.S. Beaufort, and V. Ferle, *Self-Organizing Hidden Markov Model Map (SOHMMM): Biological Sequence Clustering and Cluster Visualization.* Methods Mol Biol, 2017. 1552: p. 83-101.

184. Dutheil, J.Y., *Hidden Markov Models in Population Genomics.* Methods Mol Biol, 2017. 1552: p. 149-164.

185. Shukla, S., et al., *Application of Hidden Markov Models in Biomolecular Simulations*, in *Hidden Markov Models: Methods and Protocols*, D.R. Westhead and M.S. Vijayabaskar, Editors. 2017, Springer New York: New York, NY. p. 29-41.

186. Vogl, C. and A. Futschik, *Hidden Markov models in biology.* Methods Mol Biol, 2010. 609: p. 241-53.

187. Vijayabaskar, M.S., *Introduction to Hidden Markov Models and Its Applications in Biology.* Methods Mol Biol, 2017. 1552: p. 1-12.

188. DeRisi, J.L., V.R. Iyer, and P.O. Brown, *Exploring the metabolic and genetic control of gene expression on a genomic scale.* Science, 1997. 278(5338): p. 680-6.

189. Trapnell, C., et al., *Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks.* Nat Protoc, 2012. 7(3): p. 562-78.

190. Jaskowiak, P.A., I.G. Costa, and R. Campello, *Clustering of RNA-Seq samples: Comparison study on cancer data.* Methods, 2018. 132: p. 42-49.

191. Zhu, R., et al., *A Robust Manifold Graph Regularized Nonnegative Matrix Factorization Algorithm for Cancer Gene Clustering.* Molecules, 2017. 22(12).

192. Ma, Y., et al., *Hessian regularization based symmetric nonnegative matrix factorization for clustering gene expression and microbiome data.* Methods, 2016. 111: p. 80-84.

193. Li, Y.E., et al., *Identification of high-confidence RNA regulatory elements by combinatorial classification of RNA-protein binding sites.* Genome Biol, 2017. 18(1): p. 169.

194. Menon, V., *Clustering single cells: a review of approaches on high-and low-depth single-cell RNA-seq data.* Brief Funct Genomics, 2018.

195. Menon, V., *Clustering single cells: a review of approaches on high-and low-depth single-cell RNA-seq data.* Brief Funct Genomics, 2017.

196. Yang, L., et al., *SAIC: an iterative clustering approach for analysis of single cell RNA-seq data.* BMC Genomics, 2017. 18(Suppl 6): p. 689.

197. Zurauskiene, J. and C. Yau, *pcaReduce: hierarchical clustering of single cell transcriptional profiles.* BMC Bioinformatics, 2016. 17: p. 140.

198. Lin, P., M. Troup, and J.W.K. Ho, *CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data.* Genome Biology, 2017. 18(1): p. 59.

199. Newell, E.W. and Y. Cheng, *Mass cytometry: blessed with the curse of dimensionality.* Nat Immunol, 2016. 17(8): p. 890-5.

200. Irish, J.M. and D.B. Doxie, *High-dimensional single-cell cancer biology.* Curr Top Microbiol Immunol, 2014. 377: p. 1-21.

201. Duren, Z., et al., *Integrative analysis of single-cell genomics data by coupled nonnegative matrix factorizations.* Proc Natl Acad Sci U S A, 2018. 115(30): p. 7723-7728.

202. Zhu, X., et al., *Detecting heterogeneity in single-cell RNA-Seq data by non-negative matrix factorization.* PeerJ, 2017. 5: p. e2888.

203. Mukherjee, S., et al., *Scalable preprocessing for sparse scRNA-seq data exploiting prior knowledge.* Bioinformatics, 2018. 34(13): p. i124-i132.

204. Irizarry, R.A., et al., *Exploration, normalization, and summaries of high density oligonucleotide array probe level data.* Biostatistics, 2003. 4(2): p. 249-64.

205. Bolstad, B.M., et al., *A comparison of normalization methods for high density oligonucleotide array data based on variance and bias.* Bioinformatics, 2003. 19(2): p. 185-93.

206. Irizarry, R.A., et al., *Summaries of Affymetrix GeneChip probe level data.* Nucleic Acids Res, 2003. 31(4): p. e15.

207. Ritchie, M.E., et al., *limma powers differential expression analyses for RNA-sequencing and microarray studies.* Nucleic Acids Res, 2015. 43(7): p. e47.

208. Mehta, J.P. and S. Rani, *Software and Tools for Microarray Data Analysis*, in *Gene Expression Profiling: Methods and Protocols*, L. O'Driscoll, Editor. 2011, Humana Press: Totowa, NJ. p. 41-53.

209. Trapnell, C., L. Pachter, and S.L. Salzberg, *TopHat: discovering splice junctions with RNA-Seq.* Bioinformatics, 2009. 25(9): p. 1105-11.

210. Trapnell, C., et al., *Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation.* Nat Biotechnol, 2010. 28(5): p. 511-5.

211. Asara, J.M., et al., *Protein sequences from mastodon and Tyrannosaurus rex revealed by mass spectrometry.* Science, 2007. 316(5822): p. 280-5.

212. Buckley, M., et al., *Comment on "Protein sequences from mastodon and Tyrannosaurus rex revealed by mass spectrometry".* Science, 2008. 319(5859): p. 33; author reply 33.

213. Pevzner, P.A., S. Kim, and J. Ng, *Comment on "Protein sequences from mastodon and Tyrannosaurus rex revealed by mass spectrometry".* Science, 2008. 321(5892): p. 1040; author reply 1040.

214. Cho, W.C., *Mass spectrometry-based proteomics in cancer research.* Expert Rev Proteomics, 2017. 14(9): p. 725-727.

215. Timms, J.F., O.J. Hale, and R. Cramer, *Advances in mass spectrometry-based cancer research and analysis: from cancer proteomics to clinical diagnostics.* Expert Rev Proteomics, 2016. 13(6): p. 593-607.

216. Zhou, W., L.A. Liotta, and E.F. Petricoin, *Cancer metabolism and mass spectrometry-based proteomics.* Cancer Lett, 2015. 356(2 Pt A): p. 176-83.

217. Park, J., et al., *Large-scale clinical validation of biomarkers for pancreatic cancer using a mass spectrometry-based proteomics approach.* Oncotarget, 2017. 8(26): p. 42761-42771.

218. Nicolaou, O., et al., *Biomarkers of systemic lupus erythematosus identified using mass spectrometry-based proteomics: a systematic review.* J Cell Mol Med, 2017. 21(5): p. 993-1012.

219. Wang, H., et al., *Quantification of mutant SPOP proteins in prostate cancer using mass spectrometry-based targeted proteomics.* J Transl Med, 2017. 15(1): p. 175.

220. Turriziani, B., A. von Kriegsheim, and S.R. Pennington, *Protein-Protein Interaction Detection Via Mass Spectrometry-Based Proteomics.* Adv Exp Med Biol, 2016. 919: p. 383-396.

221. Artigues, A., et al., *Protein Structural Analysis via Mass Spectrometry-Based Proteomics.* Adv Exp Med Biol, 2016. 919: p. 397-431.

222. Heidelberger, J.B., S.A. Wagner, and P. Beli, *Mass Spectrometry-Based Proteomics for Investigating DNA Damage-Associated Protein Ubiquitylation.* Front Genet, 2016. 7: p. 109.

223. Perkins, D.N., et al., *Probability-based protein identification by searching sequence databases using mass spectrometry data.* Electrophoresis, 1999. 20(18): p. 3551-3567.

224. Koenig, T., et al., *Robust prediction of the MASCOT score for an improved quality assessment in mass spectrometric proteomics.* J Proteome Res, 2008. 7(9): p. 3708-17.

225. Cox, J. and M. Mann, *MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification.* Nat Biotechnol, 2008. 26(12): p. 1367-72.

226. Tyanova, S., T. Temu, and J. Cox, *The MaxQuant computational platform for mass spectrometry-based shotgun proteomics.* Nat Protoc, 2016. 11(12): p. 2301-2319.

227. Cox, J., et al., *Andromeda: a peptide search engine integrated into the MaxQuant environment.* J Proteome Res, 2011. 10(4): p. 1794-805.

228. Eng, J.K., A.L. McCormack, and J.R. Yates, *An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database.* J Am Soc Mass Spectrom, 1994. 5(11): p. 976-89.

229. Diament, B.J. and W.S. Noble, *Faster SEQUEST searching for peptide identification from tandem mass spectra.* J Proteome Res, 2011. 10(9): p. 3871-9.

230. Kong, A.T., et al., *MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics.* Nat Methods, 2017. 14(5): p. 513-520.

231. Lavallee-Adam, M., et al., *From raw data to biological discoveries: a computational analysis pipeline for mass spectrometry-based proteomics.* J Am Soc Mass Spectrom, 2015. 26(11): p. 1820-6.

232. Colangelo, C.M., et al., *YPED: an integrated bioinformatics suite and database for mass spectrometry-based proteomics research.* Genomics Proteomics Bioinformatics, 2015. 13(1): p. 25-35.

233. Chen, T., et al., *Web resources for mass spectrometry-based proteomics.* Genomics Proteomics Bioinformatics, 2015. 13(1): p. 36-9.

234. Aebersold, R. and M. Mann, *Mass spectrometry-based proteomics.* Nature, 2003. 422(6928): p. 198-207.

235. Domon, B. and R. Aebersold, *Mass spectrometry and protein analysis.* Science, 2006. 312(5771): p. 212-7.

236. Yates, J.R., C.I. Ruse, and A. Nakorchevsky, *Proteomics by mass spectrometry: approaches, advances, and applications.* Annu Rev Biomed Eng, 2009. 11: p. 49-79.

237. Li, S. and H. Tang, *Computational Methods in Mass Spectrometry-Based Proteomics.* Adv Exp Med Biol, 2016. 939: p. 63-89.

238. Välikangas, T., T. Suomi, and L.L. Elo, *A comprehensive evaluation of popular proteomics software workflows for label-free proteome quantification and imputation.* Brief Bioinform, 2018. 19(6): p. 1344-1355.

ดัชนี

G

H

I

J

K

L

M

N

O