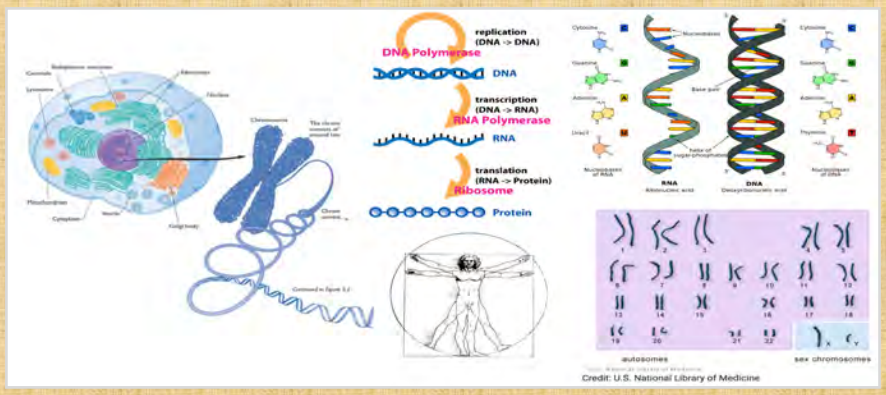
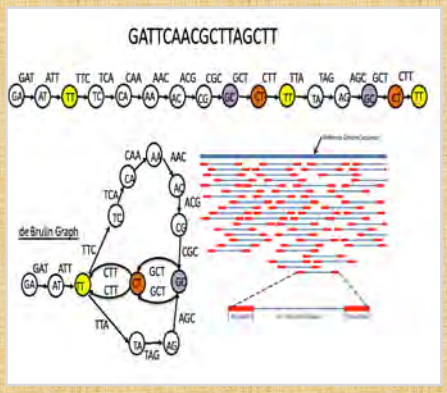
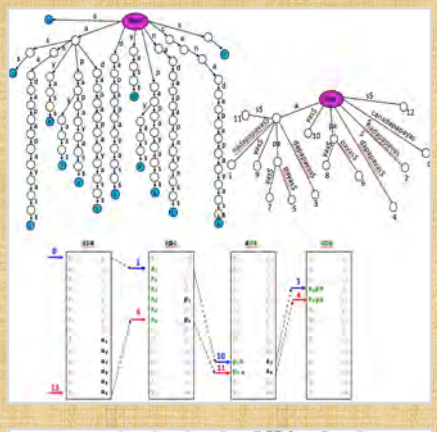
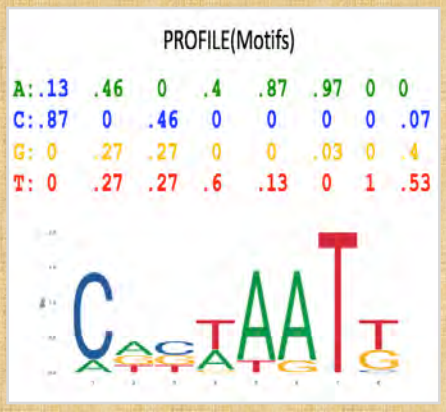
ชีวสารสนเทศ 1

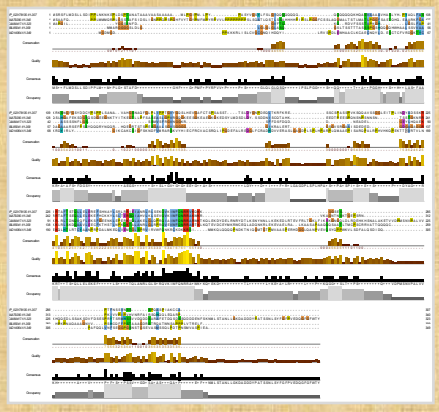
แนวทางอัลกอริทึม

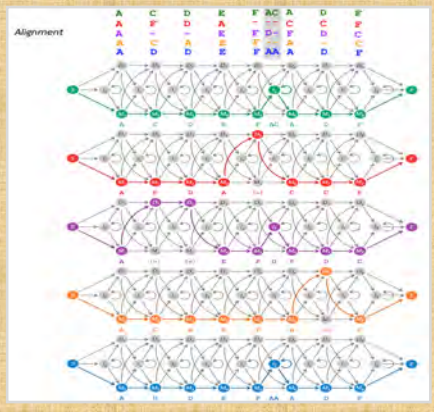


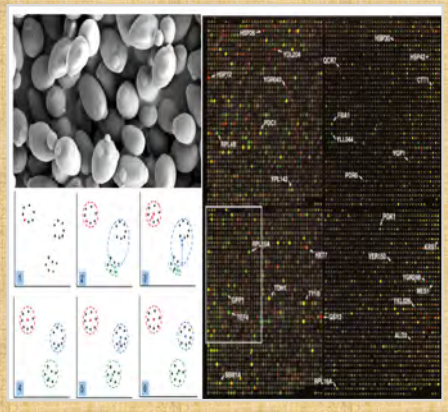


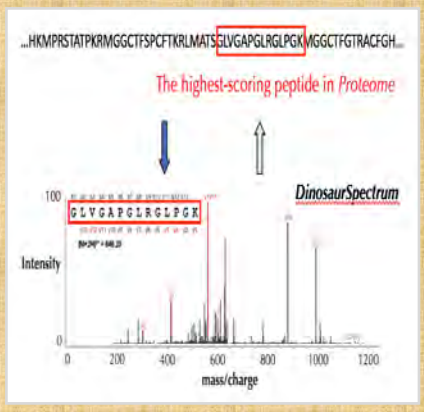












ดวงดาว วิชาดากุล

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

ชีวสารสนเทศ 1

แนวทางอัลกอริทึม

ดวงดาว วิชาดากุล
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

ดวงดาว วิชาดากุล
ชีวสารสนเทศ 1 แนวทางอัลกอริทึม / ดวงดาว วิชาดากุล
1. ชีวสารสนเทศ
2. Bioinformatics

พิมพ์ครั้งที่ 1 จำนวน 100 เล่ม พ.ศ. 2561

จัดพิมพ์โดย
ดวงดาว วิชาดากุล
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย
พญาไท กรุงเทพฯ 10330

ออกแบบปก : ดวงดาว วิชาดากุล
ออกแบบรูปเล่ม : ดวงดาว วิชาดากุล

พิมพ์ที่ ศูนย์ถ่ายเอกสาร ก๊อปปี้วัน โทรศัพท์ 0-2215-2570, 0-2215-4237
198 อาคารยูเซ็นเตอร์ 1 ห้อง A06 ซอยจุฬา 42 ถนน จุฬาลงกรณ์ 42 ปทุมวัน กรุงเทพฯ 10330

คำนำ

ตำราเรื่อง ชีวสารสนเทศ 1 แนวทางอัลกอริทึม นี้จัดทำขึ้นเพื่อเผยแพร่ความรู้ให้กับนิสิต นักศึกษา และบุคคลทั่วไปที่สนใจศาสตร์ในสหสาขาวิชาชีวสารสนเทศซึ่งเกี่ยวข้องกับการประยุกต์ใช้องค์ความรู้ในสาขาวิชาวิศวกรรมคอมพิวเตอร์ วิทยาศาสตร์คอมพิวเตอร์ คณิตศาสตร์และสถิติ ในการแก้ปัญหาในเชิงอัลกอริทึมกับโจทย์ทางชีววิทยาและอณูวิทยาทั้งทางการแพทย์และเทคโนโลยีชีวภาพ โดยตำรานี้ถูกใช้ประกอบการเรียนการสอนในรายวิชา 2110495 Advanced Topics in Computer Engineering I (Bioinformatics I) ในหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เนื้อหาในตำรานี้ประกอบด้วยความรู้เกี่ยวกับอณูวิทยาพื้นฐาน เช่น ดีเอ็นเอ อาร์เอ็นเอ โปรตีน ยีน จีโนม กระบวนการเซ็นทรัลดอกมา เทคโนโลยีโอมิกส์ การถอดรหัสพันธุกรรม เป็นต้น ที่จำเป็นต่อการทำความเข้าใจลักษณะและรูปแบบของข้อมูลแบบต่างๆ ที่สามารถนำมาวิเคราะห์หรือสร้างอัลกอริทึมเพื่อวิเคราะห์โดยกระบวนการทางคอมพิวเตอร์ได้ องค์ความรู้ในตำรานี้เกิดจากการเรียบเรียงและรวบรวมจากแหล่งความรู้ต่างๆ โดยเฉพาะหนังสือ *Bioinformatics algorithms: an active learning approach* 2nd edition ทั้งเล่ม Volume 1 และ 2 ของ Compeau, P.E.C, & Pevzner, P.A. (2015) ตัวอย่างโจทย์ทางอัลกอริทึมจาก Rosalind (http://rosalind.info) ตัวอย่างโจทย์วิจัยร่วมสมัยและผลงานวิจัยต่างๆ ที่ตีพิมพ์ในวารสารวิชาการนานาชาติ คอร์สออนไลน์แบบสั้นที่เน้นการให้ความรู้เกี่ยวกับเทคโนโลยีต่างๆที่เกี่ยวข้อง เช่น EMBL-EBI Train Online (https://www.ebi.ac.uk/training/online/) เป็นต้น รวมทั้งตัวอย่างผลงานวิจัยและหัวข้อวิจัยที่ทางผู้จัดทำดำเนินการอยู่ โดยผู้จัดทำหวังเป็นอย่างยิ่งว่าตำราเล่มนี้จะเป็นประโยชน์ต่อนิสิตและบุคคลทั่วไปที่มีความสนใจหรือเริ่มต้นสนใจศาสตร์ในสาขานี้

อ.ดร. ดวงดาว วิชาดากุล
ผู้จัดทำ

สารบัญ

สารบัญรูปภาพ

บทที่ 1 ทำความรู้จักชีวสารสนเทศ

วัตถุประสงค์

- เพื่อให้นิสิตเห็นที่มา เข้าใจความหมาย ความสำคัญ และองค์ความรู้ที่เกี่ยวข้องกับชีวสารสนเทศ
- เพื่อปูพื้นความรู้ทางอณูชีววิทยาและเทคโนโลยีที่เกี่ยวข้อง ที่จำเป็นต่อความเข้าใจโจทย์ทางชีววิทยา ชีวการแพทย์ และเทคโนโลยีชีวภาพ
- เพื่อให้นิสิตได้เห็นตัวอย่างโจทย์ทางชีววิทยา ชีวการแพทย์ และเทคโนโลยีชีวภาพ และแนวทางในการแก้ปัญหาโจทย์เหล่านี้จากมุมมองของวิศวกรรมและวิทยาศาสตร์คอมพิวเตอร์ คณิตศาสตร์ และสถิติ
- เพื่อให้นิสิตได้ทำความรู้จักกับฐานข้อมูลสาธารณะ ตัวอย่างข้อมูลทางชีวสารสนเทศ ลักษณะการเข้าถึงข้อมูล และการประมวลผลข้อมูลพื้นฐาน

ผลลัพธ์ที่คาดหวัง

- นิสิตสามารถอธิบายที่มา ความหมาย ความสำคัญและองค์ความรู้ที่เกี่ยวข้องกับชีวสารสนเทศได้
- นิสิตสามารถอธิบายหลักการเซ็นทรัลดอกมา และศัพท์พื้นฐานทางชีววิทยาและอณูชีววิทยา เช่น จีโนม โครโมโซม ยีน ดีเอ็นเอ นิวคลีโอไทด์ อาร์เอ็นเอ โปรตีน กรดอะมิโน โคดอน เป็นต้น และศัพท์เทคนิคที่เกี่ยวข้องเช่น เทคโนโลยีโอมิกส์ จีโนมิกส์ ทรานสคริปโทมิกส์ โปรตีโอมิกส์ เมตาโบโลมิกส์ เป็นต้น
- นิสิตสามารถยกตัวอย่างโจทย์ทางอณูชีววิทยา ชีวการแพทย์ และเทคโนโลยีชีวภาพ และแนวทางในการแก้ปัญหาโจทย์เหล่านี้จากมุมมองของวิศวกรรมและวิทยาศาสตร์คอมพิวเตอร์ คณิตศาสตร์ และสถิติ
- นิสิตสามารถเขียนโปรแกรมเพื่อประมวลผลตัวอย่างข้อมูลทางชีวสารสนเทศได้

เนื้อหาโดยสรุป

แนะนำเนื้อหาวิชา โดยเริ่มจากการอธิบายว่า จีโนมิกส์ (genomics) คืออะไร มีความสำคัญอย่างไร การถอดรหัสจีโนมมนุษย์ ขนาดของข้อมูลจีโนมมนุษย์ การถอดรหัสพันธุกรรมในระดับจีโนมกับการวินิจฉัยและรักษาโรค ตัวอย่างเทคโนโลยีที่เกี่ยวข้องกับการถอดรหัสจีโนม งานวิจัยที่เกี่ยวข้องในเชิงชีวสารสนเทศ เช่น อัลกอริทึมที่เกี่ยวกับการประกอบร่างจีโนม (genome assembly) การทำนายตำแหน่งของยีนในจีโนม (gene prediction) การหาโมทิฟ (regulatory motif finding) การหาลำดับสายนิวคลีโอไทด์ที่มีความเหมือนกัน เป็นต้น ตัวอย่างโปรแกรม หรือเครื่องมือทางชีวสารสนเทศที่มีการใช้งานกันอย่างกว้างขวาง เช่น โปรแกรม BLAST [1] งานวิจัยในแง่อื่นๆ ที่เกี่ยวข้อง เช่น การหาการแปรผันของลำดับเบส (variation) ในจีโนม และการอนุมานการเกิดโรค

โครงการ 1,000 จีโนม โดย EMBL-EBI (http://www.internationalgenome.org) โครงการ 100,000 genomes ของประเทศอังกฤษ (UK) (https://www.genomicsengland.co.uk/the-100000-genomes-project/) และโครงการ 100K Genome Asia (http://www.genomeasia100k .com) หลักการพื้นฐานทางชีววิทยาและอณูชีววิทยา จีโนม โครโมโซม ยีน ดีเอ็นเอ อาร์เอ็นเอ โปรตีน หลักการเซ็นทรัลดอกมา (central dogma) การเข้ารหัส การถอดรหัส และ การแปลรหัส ดีเอ็นเอ อาร์เอ็นเอ และโปรตีน เทคโนโลยีโอมิกส์ (omics) อื่นๆ นอกเหนือจากจีโนมิกส์ บิกดาต้ากับจีโนมิกส์ สถาปัตยกรรมคลาวด์ที่สนับสนุน งานวิจัยในเชิงชีวสารสนเทศ การประยุกต์ใช้ชีวสารสนเทศกับการแพทย์ การเกษตร เทคโนโลยีชีวภาพ

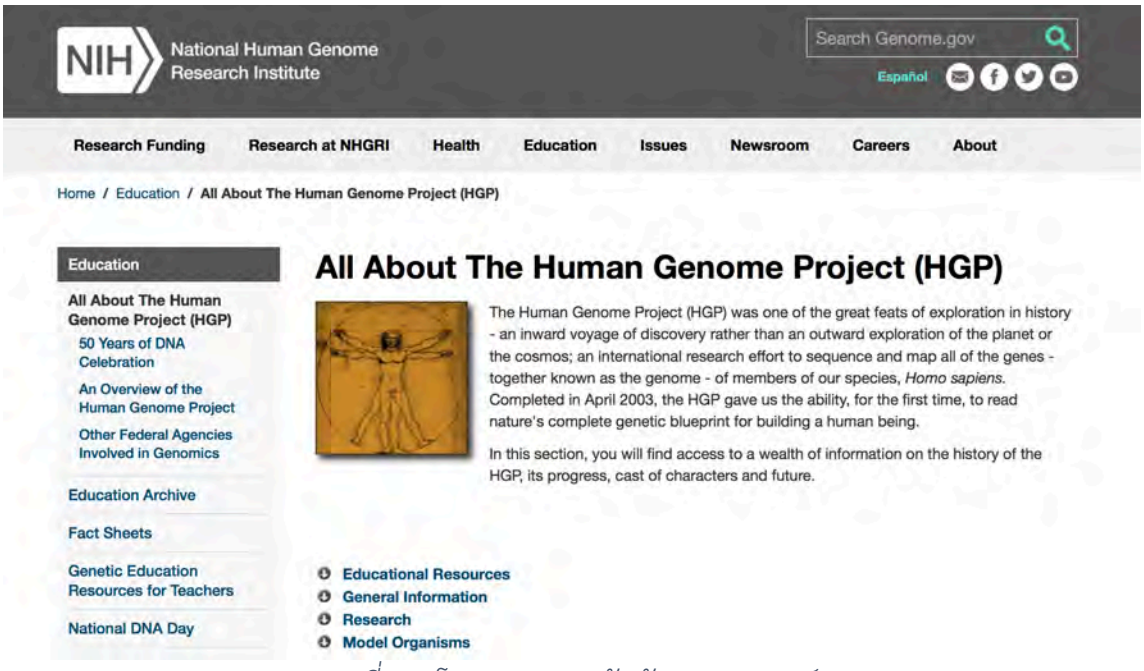
บทที่ 1 ทำความรู้จักกับชีวสารสนเทศ

บทนำเกี่ยวกับจีโนมิกส์และจีโนม

รูปที่ 1.1 แนะนำความหมายของจีโนมิกส์และจีโนม

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

คลิปวีดีโอในลิงค์รูปที่ 1.1 อธิบายความหมายของจีโนมิกส์ โดยเริ่มต้นอธิบายเกี่ยวกับจีโนม (genome) ซึ่งก็คือรหัสพันธุกรรมทั้งหมดของสิ่งมีชีวิตหนึ่งๆ โดยรหัสพันธุกรรมเหล่านี้เข้ารหัสอยู่ในแต่ละโครโมโซมในเซลล์ของสิ่งมีชีวิต ในรูปแบบของดีเอ็นเอ (**D**eoxyribo **n**ucleic **A**cid: DNA) สายคู่ (double strand) ซึ่งประกอบด้วยลำดับนิวคลีโอไทด์ (nucleotide) 4 ประเภท คือ อะดีนีน (Adenine) ไทมีน (Thymine) ไซโตซีน (Cytosine) และ กวานีน (Guanine) โดยในกรณีของจีโนมมนุษย์ลำดับเบสนิวคลีโอไทด์ที่ถอดรหัสได้นั้นมีจำนวนถึง 3 พันล้านตัวอักษร โดยเทียบเท่ากับจำนวนอักษรในดิกชันนารีมาตรฐานถึง 400 เล่มรวมกัน โดยในรหัสพันธุกรรมทั้งหมดนี้จะมีเฉพาะบางบริเวณที่เป็นยีนที่สามารถแปลรหัสต่อไปเป็นโปรตีน ที่มีฟังก์ชันการทำงานจำเพาะ เช่น โปรตีนที่เกี่ยวข้องกับกล้ามเนื้อ โปรตีนที่เกี่ยวข้องกับการย่อยอาหาร เป็นต้น โดยอัตราส่วนของบริเวณที่เป็นยีนเหล่านี้มีเพียง 2-3% ในจีโนม และอีก 97-98% นั้นเป็นส่วนที่เรียกว่า นอนโคดดิง ดีเอ็นเอ (non-coding DNA) โดยนอนโคดดิงก็คือไม่มีการเข้ารหัสที่สามารถแปลต่อไปเป็นโปรตีนได้นั่นเอง โดยจีโนมิกส์คือการศึกษาจีโนม เพื่อศึกษาบริเวณที่เป็นยีนและบริเวณที่เป็นนอนโคดดิงดีเอ็นเอ ความสำคัญของบริเวณเหล่านี้และฟังก์ชันการทำงานต่างๆ ที่เกี่ยวข้อง

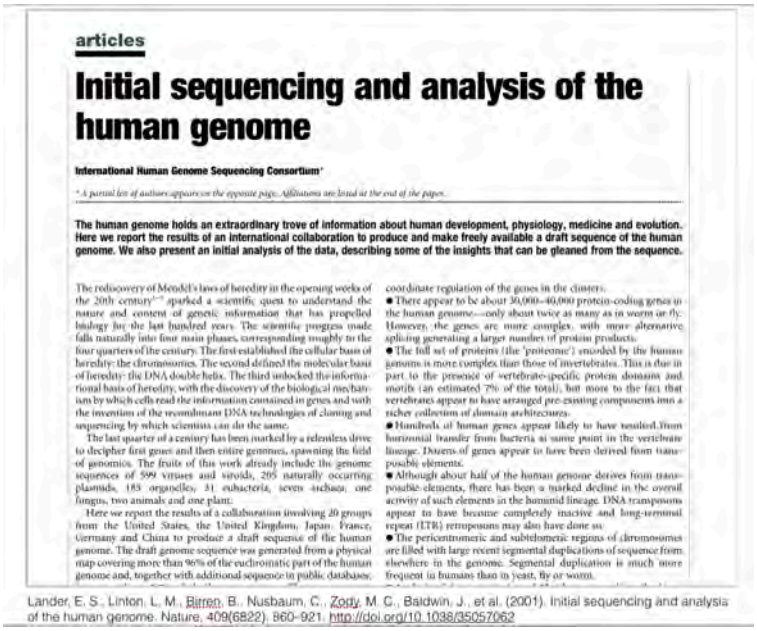


รูปที่ 1.2 โครงการถอดรหัสพันธุกรรมมนุษย์

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

โครงการถอดรหัสจีโนมมนุษย์ (Human Genome Project: HGP) (รูปที่ 1.2) เป็นความร่วมมือของหน่วยงานวิจัย ศูนย์วิจัยระหว่างประเทศ โดยมีเป้าหมายเพื่อถอดรหัสพันธุกรรมของมนุษย์ รู้ตำแหน่งบนโครโมโซมและเข้าใจฟังก์ชันการทำงานยีนทั้งหมดที่อยู่ในจีโนมของมนุษย์ โดย International Human Genome Sequencing Consortium ได้ตีพิมพ์โครงร่างแรกของจีโนมมนุษย์ในเดือนกุมภาพันธ์ปี ค.ศ. 2001 [2] ในนิตยสารเนเจอร์ (Nature) (รูปที่ 1.3) โดยรายงานถึงขนาดของจีโนมมนุษย์ที่ประกอบด้วยประมาณ 3 พันล้านเบสแพร์ (base pairs) โดยจำนวนยีนที่ได้จากการถอดรหัสอยู่ที่ประมาณ 20,000 ถึง 25,000 ยีน โดยโครงการถอดรหัสพันธุกรรมมนุษย์นี้ มีจุดเริ่มต้นมาจาก Alfred Sturtevant ทำการสร้างแผนที่ยีนของแมลงหวี่ (Drosophila) ในปี ค.ศ. 1911 และตามมาด้วยโดยผลการศึกษาจีโนมในระดับโมเลกุลและงานวิจัยทางด้านอณูชีววิทยาหรือชีววิทยาระดับโมเลกุล โดยเฉพาะการค้นพบโครงสร้างของดีเอ็นเอสายคู่ของฟรานซิส คริก (Francis Crick) และเจมส์ วัตสัน (James Watson) ในปี ค.ศ. 1953 และโดยนักวิทยาศาสตร์ทั้งสองท่านได้รับรางวัลโนเบลในปี ค.ศ. 1962 (ที่มา: https://www.genome.gov/12011239/a-brief-history-of-the-human-genome-project/)

ผลงานตีพิมพ์จีโนมมนุษย์โครงร่างแรกนี้เป็นจุดเริ่มต้นของมิติวิจัยและพัฒนาต่างๆที่เกี่ยวข้อง เช่น เครื่องมือและเทคโนโลยีในการทำถอดรหัสพันธุกรรม กลุ่มวิจัยใหม่ กฎหมายใหม่ที่จำเป็น จีโนมของยีสต์ จีโนมของเชื้อแบคทีเรียอีโคไล จีโนมของหนอน จีโนมของแมลงวันผลไม้ ของการศึกษาวิจัยเกี่ยวจีโนมของสิ่งมีชีวิตต่างๆ มากมาย รวมทั้งการลงทะเบียนข้อมูลจำนวนมหา ศาลในฐานข้อมูลออนไลน์ ซึ่งเป็นการพลิกโฉมการวิจัยและพัฒนาในเชิงชีววิทยา เทคโนโลยีชีวภาพ การเกษตร และทางการแพทย์ อย่างก้าวกระโดด



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]

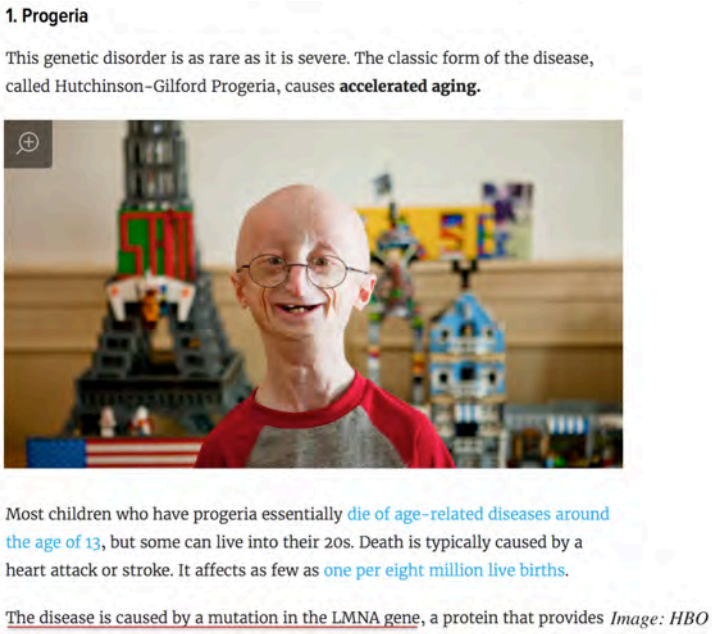
การประยุกต์ใช้จีโนมในการวิจัยและวินิจฉัยโรค

การแปรผันในระดับพันธุกรรมกับการเกิดโรค

โดยเฉลี่ยในแต่ละ 1000 เบสของจีโนมมนุษย์แต่ละคนจะมีความแตกต่างกันประมาณ 1 ตำแหน่ง ซึ่งตำแหน่งที่แตกต่างกันเหล่านั้น มีผลต่อลักษณะการแสดงออกหรือฟีโนไทป์ (phenotype) เช่น สีตา ความสูง ความเสี่ยงต่อการมีคอเลสเตอรอลสูง และการเกิดโรคทางพันธุกรรม เป็นต้น รูปที่ 1.4 แสดงตัวอย่างโรคชื่อ โรคชราในเด็ก หรือโรคโปรเจเรีย (Progeria) ซึ่งเป็นโรคที่พบได้น้อยในระดับ 1 ใน 8 ล้านคน โดยมีอาการคือเด็กจะมีรูปร่างแคระแกรนและผิวหนังเหี่ยวย่นเหมือนคนชรา โดยมักจะเสียชีวิตอายุประมาณ 13 ปี แต่มีบางรายที่มีอายุยืนกว่านั้น โดยมีการอธิบายถึงอาการของโรคนี้ครั้งแรกในงานวิจัยปีค.ศ. 1886 [3] และในปี ค.ศ. 2003 ได้มีการรายงานผลวิจัยเกี่ยวกับสาเหตุของโรคชราในเด็กว่าเกิดจากการแปรผันของเบสลำดับที่ 1824 ของยีน LMNA [4] โดยนิวคลีโอไทด์ไซโตซีนถูกแทนที่ด้วยนิวคลีโอไทด์ไทมีน ซึ่งการแปรผันในลำดับเบสเดียวที่ตำแหน่งนี้มีผลต่อการสร้างเมสเซ็นเจอร์อาร์เอ็นเอที่สั้นกว่าปกติ และแปลรหัสเป็นโปรตีนที่ผิดปกติ

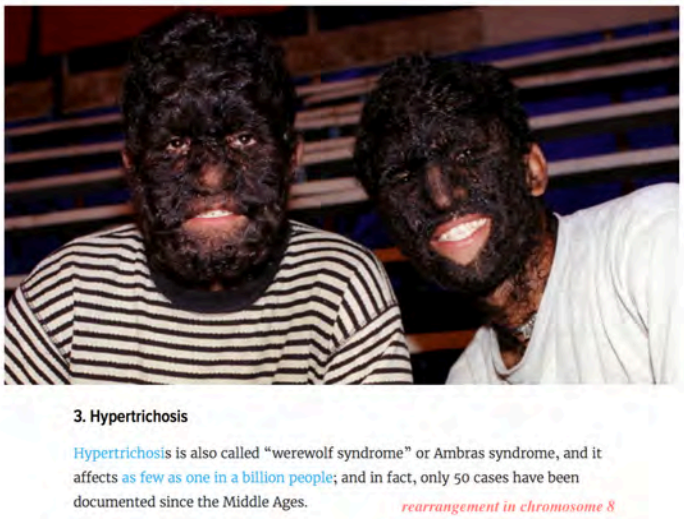
รูปที่ 1.5 แสดงตัวอย่างของโรคมนุษย์หมาป่า (Hypertrichosis) ที่ผู้ป่วยจะมีขนเยอะและยาวมากกว่าคนปกติโดยอาจครอบคลุมพื้นที่ทั้งร่างกายหรือเฉพาะบางส่วนของร่างกาย [5] โดยได้มีการระบุสาเหตุว่าเกิดจากการเปลี่ยน แปลงโครงสร้างการจัดเรียนตัวของลำดับเบสในบริเวณใกล้กับยีน SOX3 [6] การเกิดจำนวนชุดซ้ำของดีเอ็นเอในบริเวณ Chromosome 17q24.2-q24.3 [7] และการเกิดการแปลผันเชิงโครงสร้างในบริเวณที่ใกล้เคียงกับยีน TRPS1 ในโครโมโซมที่ 8 [8] และการเกิดการแปรผันเชิงโครงสร้างในโครโมโซมที่ 8 [9] เป็นต้น และรูปที่ 1.6 แสดงผลของการแปรผันเชิงโครงสร้างของโครโมโซมที่ 7 ที่มีผลต่อการเกิดโรคมือก้ามกุ้ง (Ectrodactyly) [10]

ในปีค.ศ. 2010 เด็กชาย Nicholas Volker เป็นมนุษย์คนแรกที่รอดชีวิตด้วยการถอดรหัสพันธุกรรมเพื่อหาสาเหตุของการเกิดโรค โดยเด็กชาย Nicholas มีอาการลำไส้อักเสบอย่างรุนแรงโดยไม่ทราบสาเหตุ วิธีการรักษาของแพทย์คือการทำการผ่าตัด เด็กชาย Nicholas หลายต่อหลายครั้งรวมทั้งการตัดส่วนของลำไส้ออก แต่ก็ไม่หายจากโรคและเกือบไม่รอดชีวิต ท้ายที่สุดคณะแพทย์ของโรงเรียนแพทย์วิสคอนซิน (Medical College of Medicine) ได้ทำการถอดรหัสดีเอ็นเอของ Nicholas และพบการแปรผันที่ไม่คาดคิดในยีน XIAP ซึ่งสัมพันธ์กับระบบ



รูปที่ 1.4 การแปรผันในลำดับเบสของยีน LMNA ที่ก่อให้เกิดโรคชราในเด็ก

(ที่มารูป: https://io9.gizmodo.com/10-unusual-genetic-mutations-in-humans-470843733)



รูปที่ 1.5 การแปรผันในบริเวณใกล้เคียงกับยีน SOX3 ที่มีผลต่อการเกิดโรคมนุษย์หมาป่า

(ที่มารูป: http://io9.gizmodo.com/10-unusual-genetic-mutations-in-humans-470843733)

ภูมิคุ้มกันของร่างกายที่ผิดปกติและน่าจะสามารถอธิบายอาการที่ปรากฏของ Nicholas รวมทั้งแนวทางในการรักษาที่ตรงจุด โดยแพทย์ได้เปลี่ยนแนวทางการรักษา Nicholas จากการผ่าตัดเป็นการรักษาด้วยภูมิคุ้มกันบำบัด (immunotherapy) โดยทำการเปลี่ยนถ่ายไขกระดูกโดยใช้เซลล์จากเลือดจากสายสะดือซึ่งมีเซลล์ต้นกำเนิด ซึ่งทำให้รักษาชีวิตเด็กชาย Nicholas ไว้ได้

รูปที่ 1.6 การแปรผันเชิงโครงสร้างของโครโมโมโซมที่ 7 ที่มีผลต่อการเกิดความผิดปกติของนิ้วมือ
(ที่มารูป: http://io9.gizmodo.com/10-unusual-genetic-mutations-in-humans-470843733)

โครงการถอดรหัสพันธุกรรมมนุษย์

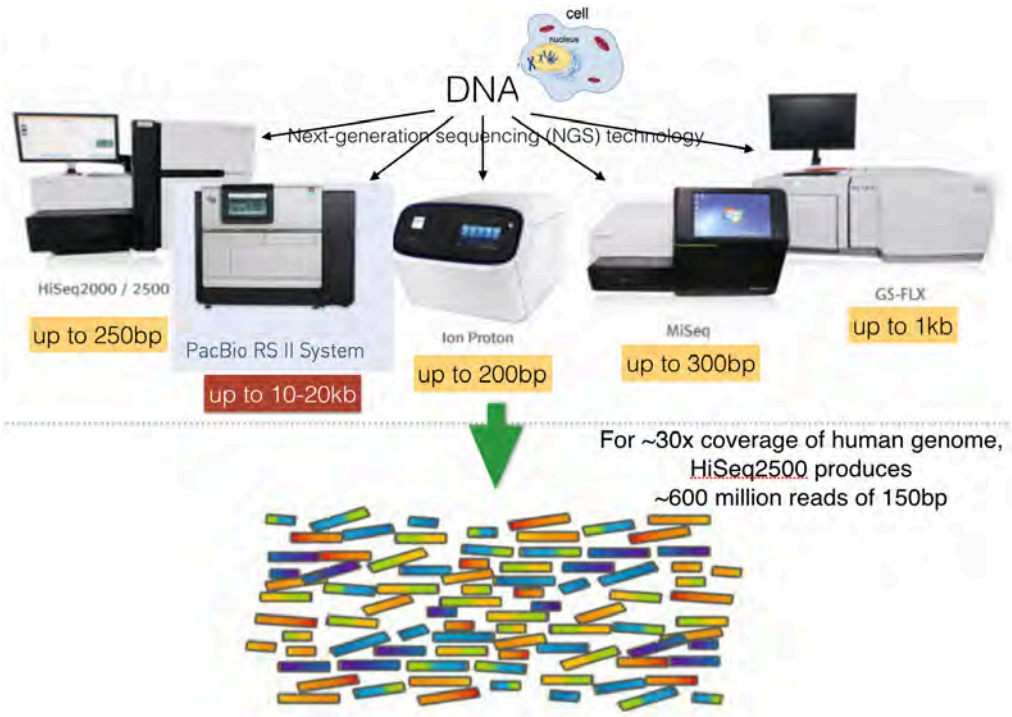
โครงการ 1000 จีโนมมนุษย์เป็นโครงการแรกของโลกที่มีการถอดรหัสพันธุกรรมของคนจำนวนมาก โดยโครงการนี้เป็นความร่วมมือของหน่วยงานวิจัยระหว่างประเทศอเมริกา อังกฤษ จีน และเยอรมัน เพื่อสร้างข้อมูลเกี่ยวกับการแปรผันในระดับพันธุกรรมเพื่อเป็นข้อมูลสนับสนุนงานวิจัยทางการแพทย์ในอนาคต โดยจะมีการขยายข้อมูลเพิ่มเติมจากโครงการ International HapMap ที่ถูกใช้ในการตรวจพบมากกว่า 100 บริเวณในจีโนมที่มีความเกี่ยวข้องกับโรคจำเพาะต่างๆ เช่น โรคหลอดเลือดหัวใจ (coronary artery disease) และโรคเบาหวาน (diabetes) เป็นต้น โดยเป้าหมายของโครงการ 1000 จีโนมนี้เพื่อเป็นแหล่งสนับสนุนข้อมูลการแปรผันในเชิงพันธุกรรมรูปแบบต่างๆ (ที่พบอย่างน้อย 1% ของกลุ่มประชาการที่ศึกษา) ทั้งการแปรผันในลำดับเบสเช่น สนิปส์ (SNPs) และ อินเดล และการแปรผันเชิงโครงสร้าง (structural variants) โดยในโครงการมีการถอดรหัสจีโนมมนุษย์ 1000 คน ที่เก็บตัวอย่างดีเอ็นเอมาจากทั่วโลก นำมาถอดรหัสโดยเทคโนโลยี NGS แบบต่างๆ และวิเคราะห์ข้อมูลการแปรผันที่พบ โดยในเฟสที่ 1 และ 3 มีการถอดรหัสพันธุกรรมจีโนมโดยข้อมูลในแต่ละเฟสได้ถูกวิเคราะห์และเปิดเผยสู่สาธารณะ โดยมีการตีพิมพ์ผลงานวิจัยหลักๆในโครงการนำร่อง [11] เฟสที่ 1 [12] และเฟสที่ 3 [13,

14] โครงการ 1000 จีโนมมนุษย์มีระยะเวลาโครงการระหว่างปีค.ศ. 2008 ถึง 2015 โดยในปัจจุบัน (ค.ศ. 2018) แม้โครงการจะสิ้นสุดไปแล้ว ข้อมูลต่างๆที่เป็นผลจากโครงการยังเปิดให้เข้าถึงได้โดยสาธารณะ ภายใต้การดูแลของศูนย์ข้อมูลที่ EMBL-EBI (The European Bioinformatics Institute) โดยได้รับเงินทุนสนับสนุนจากเวลคัมทรัส (Wellcome Trust) และมีชื่อโครงการต่อเนื่องเป็น IGSR: The International Genome Sample Resource

โครงการ 100,000 จีโนมของอังกฤษ [15] มีเป้าหมายหลักเพื่อถอดรหัสจีโนมของคนอังกฤษ 100,000 คน โดยเน้นผู้ป่วยของ NHS (National Health Service) ที่เป็น “โรคหายาก” (rare diseases) รวมทั้งสมาชิกในครอบครัวของผู้ป่วยรวมทั้งผู้ป่วยที่เป็นโรคมะเร็งประเภทที่พบได้บ่อย (common cancers) โครงการ 100K จีโนมเอเชีย มีเป้าหมายหลักเพื่อถอดรหัสจีโนมของคนเชื้อชาติเอเชีย 100,000 คน เพื่อสนับสนุนความก้าวหน้าในทางการแพทย์และการแพทย์แม่นยำ (precision medicine) ประเทศกาตาร์เป็นอีกประเทศที่มีโครงการแห่งชาติในการถอดรหัสจีโนมประชากรกาตาร์ 10,000 คน โดยเพื่อให้เป็นฐานข้อมูลอ้างอิงสำหรับประเทศในการวิจัยและพัฒนาทางการแพทย์โดยเฉพาะการแพทย์เฉพาะคน (personalized healthcare)

เทคโนโลยีในการถอดรหัสดีเอ็นเอ

เทคโนโลยีที่ใช้ในการถอดรหัสดีเอ็นเอมีประวัติที่ยาวนานกว่า 50 ปี [16] โดยแพลตฟอร์มหลักที่มีการใช้งานกันอย่างแพร่หลายในปัจจุบันนั้นคือแพลตฟอร์มอิลูมินา (Illumina) ด้วยราคาที่ถูกลงเป็นอย่างมากประมาณ 40,000 บาทต่อจีโนมมนุษย์ 1 คนหรือถูกกว่านั้นถ้ามีการถอดรหัสจีโนมคนจำนวนมาก ทั้งนี้ข้อมูลรหัสพันธุกรรมที่ได้จากแพลตฟอร์มอิลูมินามักเป็นคู่ของดีเอ็นเอสายสั้นยาวประมาณ 100-150 เบสต่อเส้น (เรียกว่ารีด: read) ขึ้นอยู่กับแพลตฟอร์มจำเพาะที่เลือกใช้ โดยจำนวนข้อมูลที่ส่งออกจากเครื่องถอดรหัสพันธุกรรมมีปริมาณมากโดยประมาณคือ 90 กิกะไบต์ต่อจีโนม 1 คน ทั้งนี้หลายคนอาจสงสัยว่าจีโนมมนุษย์มีขนาดประมาณ 3 กิกะไบต์แล้วทำไมจำนวนข้อมูลที่ผลิตจากเทคโนโลยีถอดรหัสอย่างอิลูมินาถึงมีจำนวนถึง 90 กิกะไบต์ต่อมนุษย์ 1 คน คำตอบคือเนื่องจากข้อมูลที่ผลิตนั้นเป็นดีเอ็นสั้นซึ่งถูกตัดแบบสุ่มจากสายดีเอ็นเอ ดังนั้นเพื่อให้สามารถถอดรหัสได้ครอบคลุมข้อมูลทั้ง 3 กิกะไบต์จึงจำเป็นต้องมีในการตัดแบบสุ่มของสายดีเอ็นเอหลายๆสำเนา โดยในตัวอย่างนี้ข้อมูลประมาณ 90 กิกะไบต์ แสดงจำนวน 30 สำเนาของสายดีเอ็นเอเดียวกันที่จะถูกนำไปตัดแบบสุ่มในกระบวนการถอดรหัส ซึ่งแสดงความครอบคลุมโดยเฉลี่ย (mean coverage) เท่ากับ 30x คำว่า sequence coverage ของแพลตฟอร์มหมายถึงจำนวนรีดโดยเฉลี่ยที่สามารถนำไปแมพ (map) ได้กับจีโนมอ้างอิงในบริเวณที่ที่มีลักษณะเบสเหมือนกันหรือใกล้เคียงกันรีดที่สุด เทคโนโลยี PacBio เป็นเทคโนโลยีที่สามารถถอดรหัสพันธุกรรมได้ยาวว่าอิลูมินามาก (โดยได้รีดยาวประมาณ 10k-20k) อย่างไรก็ตาม PacBio ยังมีข้อจำกัดในเรื่องของความผิดพลาดในการอ่านลำดับเบสซึ่งสูงถึงประมาณ 15% ของความยาวรีดและยังมีราคาสูงมากเมื่อเทียบกับแพลตฟอร์มอิลูมินา รูปที่ 1.7 แสดงตัวอย่างเครื่องมือที่ใช้ในการถอดรหัสพันธุกรรม



รูปที่ 1.7 ตัวอย่างแพลตฟอร์มและเครื่องมือที่ใช้ในการถอดรหัสพันธุกรรม

ตัวอย่างโจทย์ทางชีวสารสนเทศ

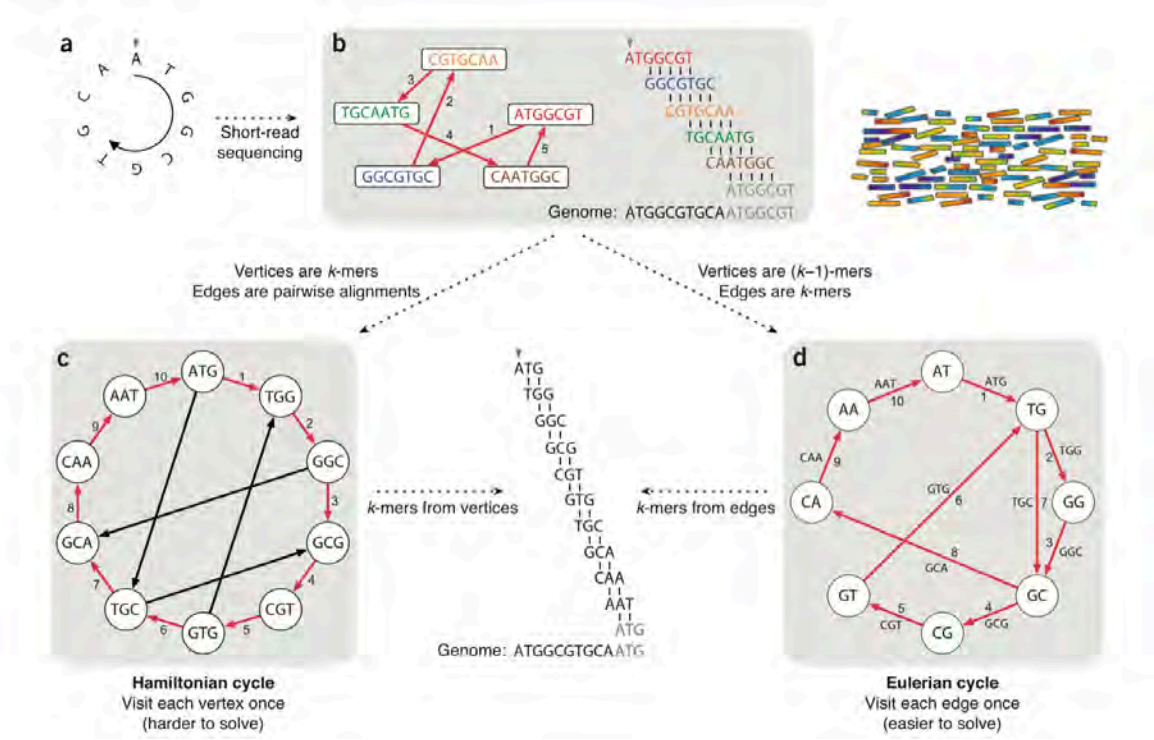
ปัญหาการประกอบร่างจีโนม

ปัญหาการประกอบร่างจีโนม (*de novo* genome assembly) เกิดจากข้อมูลดีเอ็นเอที่ได้จากเครื่องมือถอดรหัสพันธุกรรมส่วนใหญ่จะเป็นลำดับเบสสายสั้นซึ่งอาจเป็นสายคู่หรือสายเดี่ยวจำนวนมาก การจะได้มาซึ่งลำดับเบสที่มีความยาวในระดับโครโมโซม[1] จำเป็นต้องนำดีเอ็นสายสั้นเหล่านี้มาต่อกัน คำถามเชิงอัลกอริทึมและการคำนวณคือเราจะต่อดีเอ็นเอสายสั้นจำนวนใกล้พันล้านรีดเข้าด้วยกันให้เป็นดีเอ็นเอสายยาวซึ่งเป็นตัวแทนโครโมโซมที่ถูกต้อง ใช้ทรัพยากรการคำนวณอย่างมีประสิทธิภาพ และให้ผลลัพธ์ได้รวดเร็วได้อย่างไร รูปที่ 1.8 แสดงสองแนวคิดในการต่อดีเอ็นเอสายสั้นเข้าด้วยกัน โดยรูปทางซ้ายแสดงวิธีการต่อดีเอ็นเอโดยใช้แนวคิดของการหาเส้นทางฮามิลโทเนียนเพื่อเป็นตัวแทนของดีเอ็นเอสายยาวหลังการต่อโดยออก แบบให้แต่ละโหนดเป็นตัวแทนรีดและเส้นเชื่อมระหว่างโหนดแสดงการเกิดความทับซ้อน (overlapping) ของลำดับเบสระหว่างรีดสองโหนด ในขณะที่รูปทางขวาแสดงการหาเส้นทางออยเลอร์เพื่อเป็นตัวแทนของดีเอ็นเอสายยาวหลังการต่อโดยออกแบบให้แต่ละโหนดเป็นตัวแทนส่วนของรีดโดยเส้นเชื่อมระหว่างโหนดแสดงลำดับเบสที่เป็นส่วนทับซ้อนกันของสองโหนด โดยมีโปรแกรมที่ถูกพัฒนาขึ้นเพื่อแก้ปัญหาการประกอบร่างจีโนมนี้ โดยสามารถศึกษาได้จาก [17, 18] เป็นต้น

[1] ขนาดของจีโนมคือความยาวของลำดับเบสของทุกโครโมโซมรวมกัน

ปัญหาการเทียบรีดกับจีโนมอ้างอิง

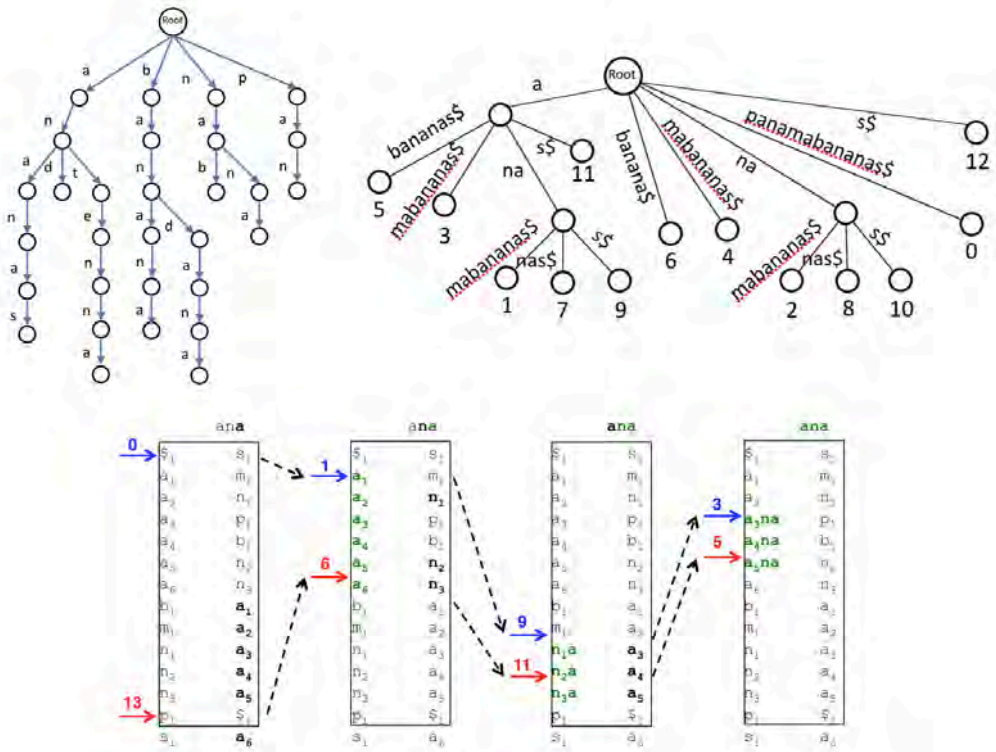
ปัญหาการเทียบรีดกับจีโนมอ้างอิง (read mapping) เชื่อมโยงมาจากปัญหาการประกอบร่างจีโนมข้างต้นโดยในกรณีที่มีจีโนมอ้างอิงที่เกิดจากการประกอบร่างจีโนมแล้ว ข้อมูลดีเอ็นเอสายสั้นจำนวนมากมายของมนุษย์คนหนึ่งที่ได้จากการถอดรหัสพันธุกรรม มักถูกนำมาเทียบกับจีโนมอ้างอิงแทนการนำมาประกอบร่างเพื่อหาจีโนมของคนนั้นๆ ทั้งนี้เพื่อเป็นการลดเวลาในการประกอบร่างจีโนมขึ้นมาใหม่ทั้งหมด และเพื่อเพิ่มความถูกต้อง โดย



รูปที่ 1.8 แนวคิดในการแก้ปัญหาการประกอบร่างจีโนม

(ที่มา: รูปที่ 3 ของ [19])

อาจยังนำรีดทั้งหมดในแต่ละบริเวณที่แมพได้ไปทำการต่อกันให้ยาวขึ้นเฉพาะในบริเวณเหล่านั้น นอกจากนี้กลุ่มก็รีดที่แมพได้บริเวณต่างๆก็สามารถนำไปวิเคราะห์เพิ่มเติมเกี่ยวกับการเกิดการแปรผันในระดับพันธุกรรมได้โดยตรง คำถามเชิงอัลกอริทึมและการคำนวณคือเราจะทำออกแบบอัลกอริทึมให้สามารถนำรีดที่มีความยาวประมาณ 100-150 เบสจำนวนเกือบพันล้านเส้นไปหาบริเวณที่เหมือนที่สุดในจีโนมโดยสามารถทำได้ถูกต้องคือแมพกับบริเวณที่เหมือนที่สุดจริงและทำได้อย่างมีประสิทธิภาพคือทำงานได้เร็วและใช้ทรัพยาการการคำนวณได้อย่างมีประสิทธิภาพ รูปที่ 1.9 แสดงตัวอย่างแนวคิดในการแก้ปัญหาการเทียบรีดกับจีโนมอ้างอิง โดยซ้ายและขวาบนแสดงแนวคิดในการใช้ซัฟฟิกซ์ไทร์และซัฟฟิกซ์ทรีตามลำดับ ส่วนด้านล่างแสดงการใช้ Burrow-Wheeler Transform (BWT) เป็นต้น ตัวอย่างโปรแกรมที่ถูกพัฒนาขึ้นที่ทำงานในลักษณะนี้สามารถศึกษาเพิ่มเติมได้จาก [20] เป็นต้น



รูปที่ 1.9 แนวคิดในการแก้ปัญหาการเทียบรีดสายสั้นกับจีโนมอ้างอิง ซ้ายบนคือซัฟฟิกซ์ไทร์ ขวาบนคือซัฟฟิกซ์ทรีและล่างคือ Burrow-Wheeler Transform (BWT)

(ที่มา: รูปที่ 9.1, 9.6 และ 9.14 ของ [21] ตามลำดับ)

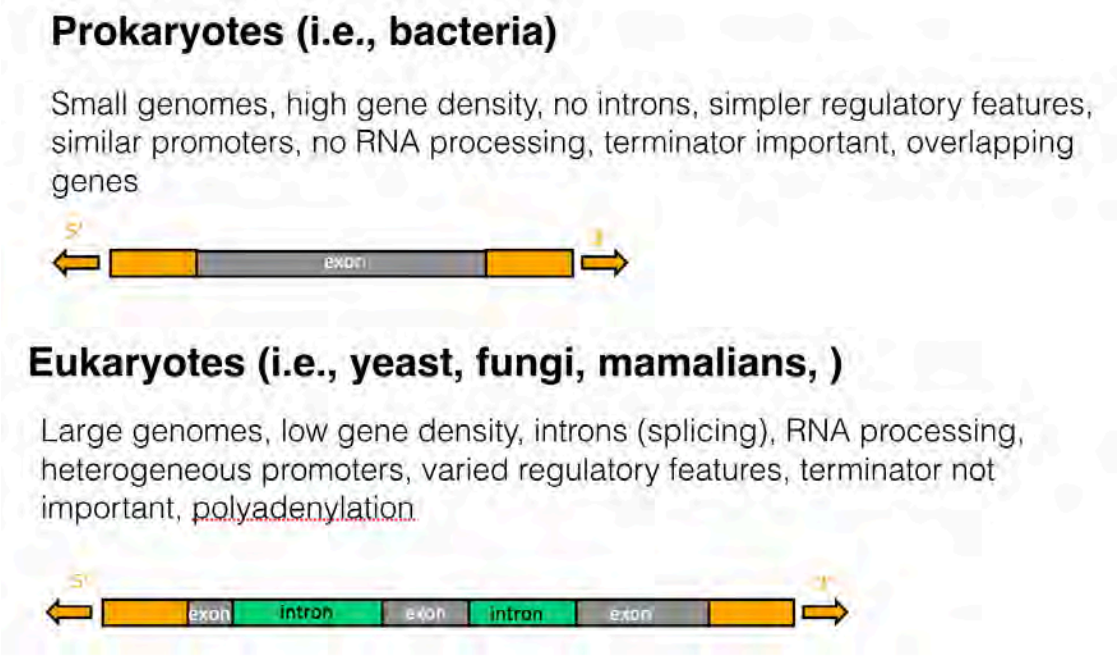
ปัญหาการตรวจหาบริเวณที่เป็นยีนในจีโนม

ปัญหาการตรวจหาบริเวณที่เป็นยีน เกิดขึ้นหลังจากเราได้ลำดับเบสของจีโนมมาเรียบร้อยแล้ว ดังในรูปที่ 1.10 ซึ่งแสดงลำดับเบสส่วนสั้นๆของจีโนม ซึ่งคำถามคือบริเวณใดในลำดับเบส A, T, C, G นี้เป็นบริเวณที่เป็นยีนที่สามารถแปลรหัสต่อไปเป็นโปรตีนได้ หรือบริเวณไหนบ้างควรถูกระบายสีแดงเพื่อบอกว่าเป็นยีนในลำดับเบสทั้งจีโนม ปัญหาการตรวจหาบริเวณที่เป็นยีนนี้เป็นตัวอย่างปัญหาที่องค์ความรู้ทางชีววิทยาจะช่วยทำให้สามารถออกแบบวิธีการการแก้ปัญหาในเชิงคำนวณได้ดีและมีความถูกต้องมากขึ้น รูปที่ 1.11 แสดงลักษณะพื้นฐานที่แตกต่างกันระหว่างจีโนมและยีนในสิ่งมีชีวิตกลุ่มยูแคริโอตและโพรแคริโอต โดยในสิ่งมีชีวิตกลุ่มโพรแคริโอต จีโนมจะมีขนาดเล็ก มีความหนาแน่นของยีน คือยีนจะอยู่ใกล้ๆกันมาก โครงสร้างของยีนไม่ซับซ้อนเพราะมีเพียงเฉพาะส่วนที่เป็นเอ็กซอน (exon) ไม่มีอินทรอน (intron) มีระบบการควบคุมการแสดงออกของยีนที่ไม่ซับซ้อน ไม่มีการประมวลผลอาร์เอ็นเอเพิ่มเติมเพราะไม่มีการเลือกส่วนของเอ็กซอนไปแปลรหัสเป็นโปรตีน และยีนมีโอกาสที่จะทับซ้อนกัน เป็นต้น สำหรับสิ่งมีชีวิตกลุ่มยูแคริโอต จีโนมจะมีขนาดใหญ่กว่ามาก (แล้วแต่ประเภทของสิ่งมีชีวิต) มีความหนาแน่นของยีนน้อย โครงสร้างของยีนซับซ้อนกว่าเพราะมีส่วนที่เป็นอินทรอนแทรกอยู่ระหว่างเอ็กซอนซึ่ง

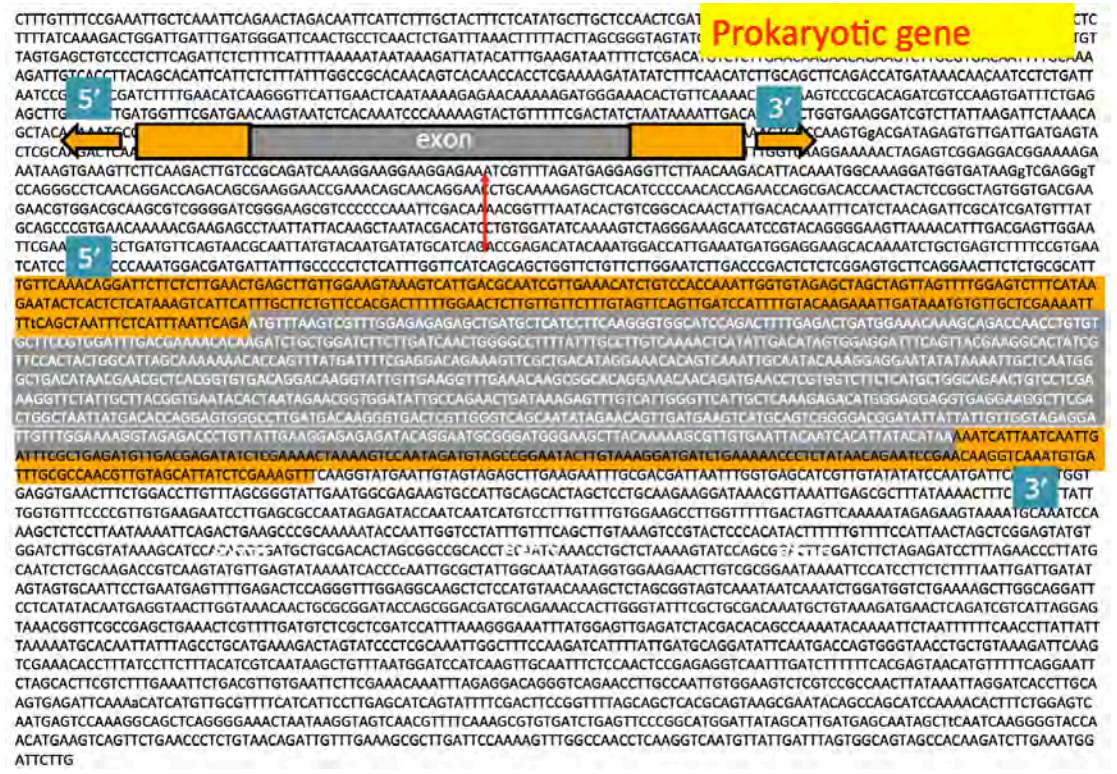
ต้องมีกระบวน การเพิ่มเติมในการประมวลผลอาร์เอ็นเอ เช่นการทำ RNA splicing คือการเลือกเอาเฉพาะบางเอ็กซอนมาต่อกันเพื่อแปลรหัสต่อไปเป็นโปรตีน เป็นต้น และมีลักษณะการควบคุมการแสดงออกของยีนที่หลากหลาย

รูปที่ 1.10 ตัวอย่างลำดับเบสในส่วนสั้นๆของจีโนม

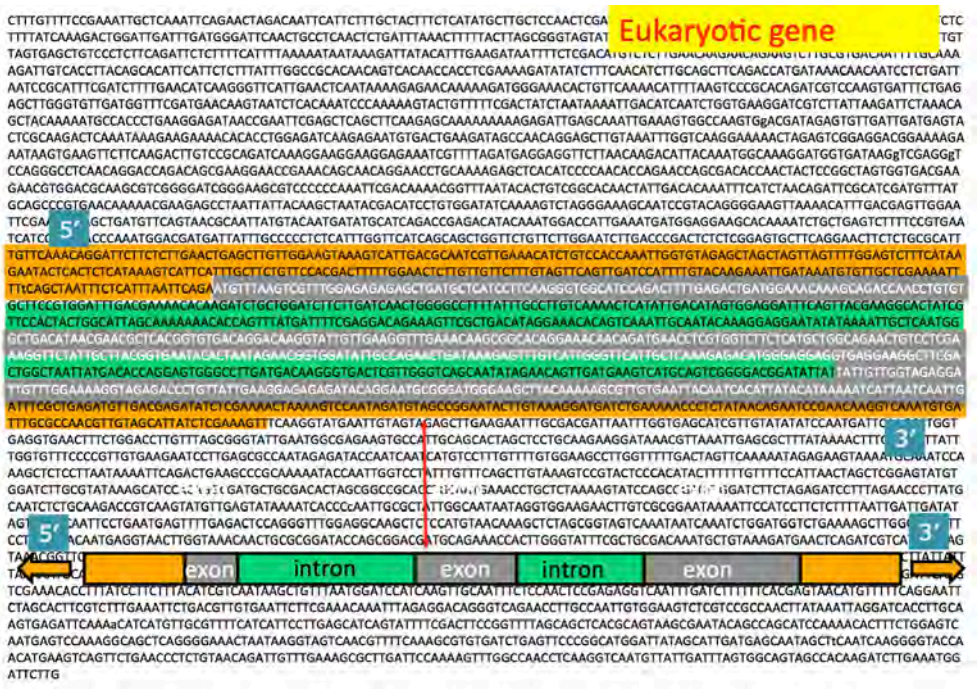
จากความรู้ทางชีววิทยานี้แสดงให้เห็นว่าถ้าเทียบลักษณะโครงสร้างยีนของสิ่งมีชีวิตกลุ่มโพรแคริโอตและยูแคริโอตกับส่วนของจีโนมข้างต้น (ซึ่งไม่ทราบว่าเป็นของสิ่งชีวิตใด) ก็จะได้ตามรูปที่ 1.12 และ 1.13 ตามลำดับ สำหรับแนวคิดเชิงคำนวณที่ใช้ในการตรวจหาบริเวณที่เป็นยีนนั้นมีความหลากหลาย โดยแนวทางหลักๆ ประกอบด้วย Homology method กับ *Ab initio* method โดยแนวทางแรกใช้การเทียบเคียงจีโนมกับยีนต่างๆที่มีการรายงานมาก่อนในสิ่งมีชีวิตอื่นๆ ส่วนแนวทางหลังนั้นสามารถแยกย่อย เช่นการหายีนโดยใช้ชุดของกฎที่สร้างมาจากลักษณะเฉพาะทางชีววิทยา การดูองค์ประกอบของลำดับเบสในยีนที่ทราบมาก่อนหน้าเพื่อหาคุณลักษณะ (feature) เฉพาะที่บ่งบอกความเป็นนยีนกับส่วนที่ไม่ใช่ยีน การมองหารูปแบบจำเพาะในบางบริเวณของยีนเช่น ส่วนหัวของยีน ส่วนท้ายของยีน ลักษณะจำเพาะของบริเวณที่เชื่อมต่อระหว่างเอ็กซอนกับอินทรอน เป็นต้น โดยอาจนำไปเข้ากระบวนการเรียนรู้ด้วยเครื่อง เช่น การทำ Decision tree, Neural Networks, และ Hidden Markov Models ต่างๆ เป็นต้น



รูปที่ 1.11 ความแตกต่างทางชีววิทยาของจีโนมและยีนในกลุ่มโพรแคริโอตและยูแคริโอต

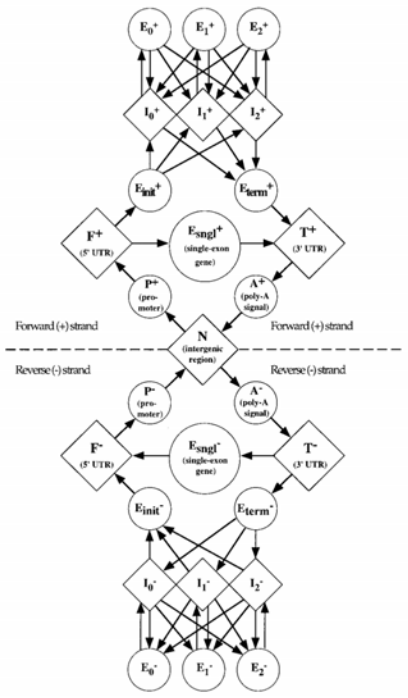


รูปที่ 1.12 โครงสร้างยีนในกลุ่มโพรแคริโอตเทียบกับส่วนของดีเอ็นเอในจีโนม



รูปที่ 1.13 โครงสร้างยีนในกลุ่มยูแคริโอตเทียบกับส่วนของดีเอ็นเอในจีโนม

รูปที่ 1.14 แสดงแบบจำลอง Hidden Markov Model ของ GENSCAN [22] ในการตรวจจับบริเวณที่เป็นยีน ตีพิมพ์ในปีค.ศ.1997

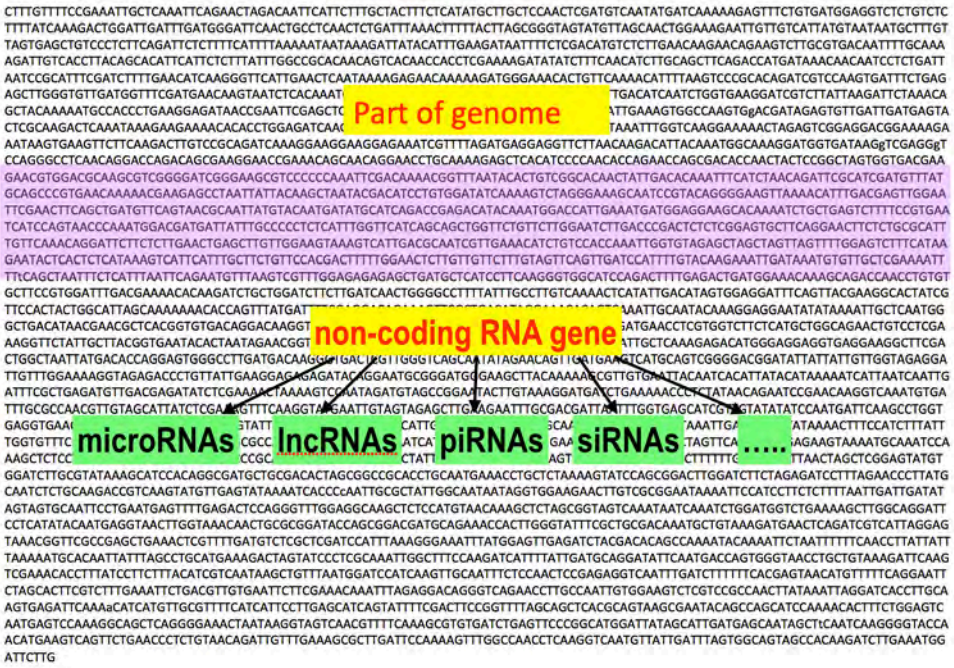


รูปที่ 1.14 แบบจำลอง Hidden Markov Model ของ GENSCAN [22] ในการตรวจจับบริเวณที่เป็นยีน

(ที่มา: รูปที่ 3 ของ [22])

ปัญหาการตรวจหาบริเวณที่เป็นนอนโคดดิงอาร์เอ็นเอในจีโนม

จีโนมมนุษย์มีขนาดประมาณ 3 พันล้านเบสและในปัจจุบันเราทราบว่ามีเพียง 2-3% ของจีโนมที่เป็นบริเวณของยีนที่สามารถแปลรหัสต่อไปเป็นโปรตีน (protein-coding gene) ส่วนอื่นๆนั้นในอดีตถือว่าเป็นบริเวณที่เป็นขยะ อย่างไรก็ตามในปัจจุบันบริเวณต่างๆใน 97% ที่เหลือได้ถูกรายงานว่ามีผลต่อการควบคุมกระบวนการต่างๆ มีผลต่อการเกิดโรค และหลายๆบริเวณเป็นยีนที่ถอดรหัสเป็นอาร์เอ็นเอแต่ไม่แปลรหัสต่อไปเป็นโปรตีน (non-coding gene) คำถามคือเราจะสามารถตรวจหาบริเวณนอนโคดดิงยีนเหล่านี้ (รูปที่ 1.15) ได้อย่างไร และจะสามารถประยุกต์ใช้แนวทางแบบเดียวกับการตรวจหาบริเวณที่เป็นยีนได้หรือไม่ อย่างไร มีองค์ความรู้ทางชีววิทยาอะไรบ้างที่จำเป็นต้องทราบก่อนการออกแบบวิธีการทางคอมพิวเตอร์ในการตรวจหา

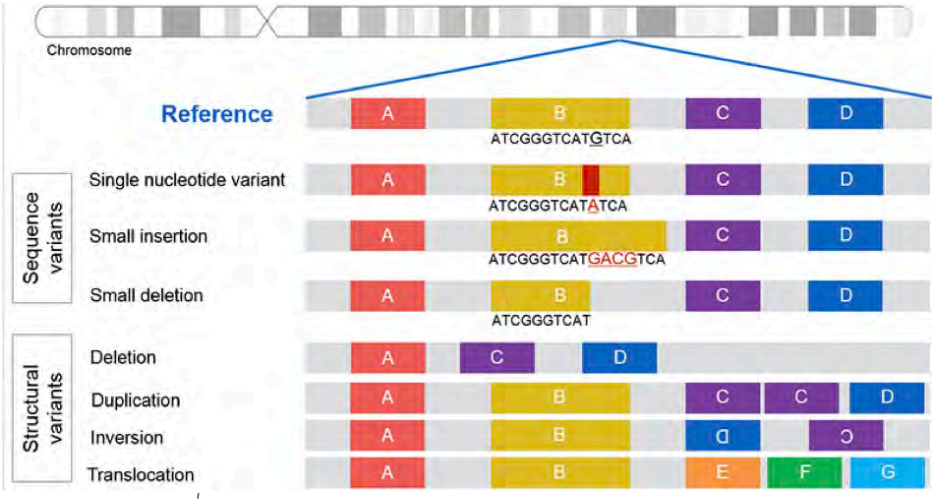


รูปที่ 1.15 เราจะตรวจหาหานอนโคดดิงอาร์เอ็นเอในส่วนของดีเอ็นเอในจีโนมได้อย่างไร

ปัญหาการตรวจหาการแปรผันของรหัสพันธุกรรมในจีโนม

การแปรผันของรหัสพันธุกรรมมีหลายประเภท ดังแสดงในรูปที่ 1.16 โดยสามารถแบ่งออกได้เป็น 2 กลุ่มหลักๆคือ กลุ่มที่การแปรผันมีผลในลำดับเบสแต่ไม่มีผลเชิงโครงสร้าง ประกอบด้วย (1) เอสเอ็นวี (Single Nucleotide Variant: SNV) ในรูปที่ 1.19 จีโนมอ้างอิง (reference genome) เป็นเบส G ในขณะที่ข้อมูลรหัสพันธุกรรมที่นำมาเทียบเป็นเบส A (2) การเกิดการแทรกลำดับเบสขนาดสั้น (small insertion) เมื่อกับจีโนมอ้างอิง โดยในรูปมีการเพิ่มลำดับเบส GACG เข้ามา และ (3) การหายไปของลำดับเบสขนาดสั้น (small deletion) โดย GTCA หายไป การหายไปหรือการเกิดการแทรกเพิ่มของลำดับเบสขนาดสั้นนี้เรียกรวมๆว่า อินเดล (indel) สำหรับกลุ่มที่การแปรผันที่มีผลในเชิงโครงสร้างประกอบด้วย deletion, insertion, duplication, inversion และ translocation โดย (1) deletion จะเกิดการหายไปของลำดับเบส โดยตัวอย่างในรูปที่ 1.19 ส่วนของบริเวณ B หายไปทั้งหมด (2) ในกรณี

ของ duplication จะเกิดการซ้ำของชุดลำดับเบส เช่นมีบริเวณ C เพิ่มเข้ามาอีกชุด (3) ในกรณีของ inversion ไม่มีการเพิ่มหรือลดของลำดับเบสแต่จะมีการกลับด้านแทนโดยในรูปที่ 1.19 ทั้ง C และ D มีลำดับเบสที่กลับด้านเมื่อเทียบกับจีโนมอ้างอิง (4) ส่วน translocation นั้นจะมีการเคลื่อนย้ายของลำดับเบสจากโครโมโซมหนึ่งไปยังอีกโครโมโซมหนึ่ง ในรูปที่ 1.19 บริเวณ E, F, และ G ย้ายมาจากโครโมโซมอื่น เป็นต้น การแปรผันเหล่านี้มีผลต่อการเกิดโรคต่างๆ ตามที่ยกตัวอย่างมาก่อนหน้า สำหรับการแก้ปัญหานี้เชิงอัลกอริทึม ในขั้นต้นจำเป็นต้องทำความเข้าใจลักษณะของการแปรผันในแต่ละประเภท รวมทั้งลักษณะของข้อมูลเข้าเพื่อการวิเคราะห์และตรวจหา เพื่อนำไปสู่การออกแบบและพัฒนาอัลกอริทึมที่มีประสิทธิภาพ ตัวอย่างอัลกอริทึมที่ใช้ในการตรวจหาการแปรผันสามารถศึกษาเพิ่มเติมได้จาก [23, 24] เป็นต้น



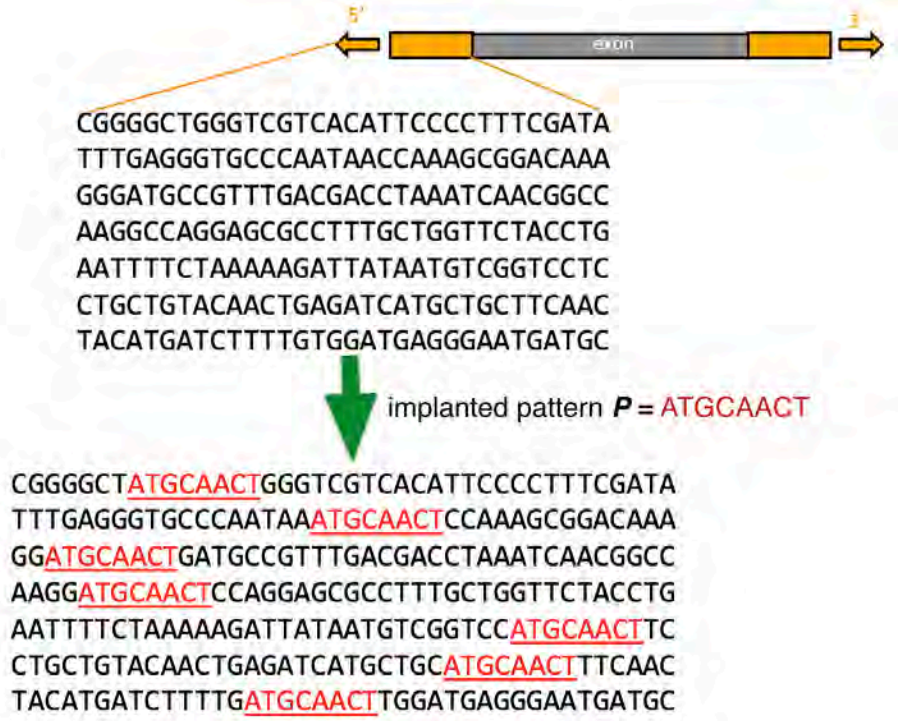
รูปที่ 1.16 ประเภทการแปรผันของรหัสพันธุกรรมในจีโนม

(ที่มารูป: http://www.labspaces.net/pictures/blog/54f1dd55a905e1425136981_blog.jpg)

ปัญหาการตรวจหาโมทิฟ

โมทิฟเป็นบริเวณบริเวณของดีเอ็นเอหรือโปรตีนที่มีรูปแบบจำเพาะ โดยมักเป็นบริเวณที่โมเลกุลจำเพาะหนึ่งๆ เข้ามาจับเพื่อควบคุมลักษณะการทำงาน เช่นดีเอ็นเอโมทิฟในส่วนหัวของยีน (ส่วน upstream region) จะเป็นบริเวณที่ทรานสคริปชันแฟคเตอร์เข้ามาจับเพื่อควบคุมการแสดงออกของยีน ในรูปที่ 1.17 ลำดับเบสที่เป็นสีแดงแสดงส่วนของโมทิฟ ในส่วนหัวของยีน 7 ยีนที่มีการแสดงออกร่วมกัน โดยสมมติฐานที่มีมาก่อนคือยีนที่มีลักษณะการแสดงออกไปในแนวทางเดียวกันน่าจะมีดีเอ็นเอโมทิฟร่วมกัน ซึ่งอาจจะเหมือนกัน 100% หรือมีความแตกต่างกันอยู่บ้างก็ได้ ในรูปที่ 1.17 นี้เนื่องจากมีการทำสีให้แตกต่างระหว่างส่วนที่เป็นโมทิฟกับส่วนอื่นๆ ก็เห็นได้ชัดว่ารูปแบบที่เกิดร่วมกันในสายดีเอ็นเอ 7 เส้นมีหน้าตาอย่างไร อย่างไรก็ตามถ้าพิจารณาในรูปที่ 1.18 จะสามารถหาได้หรือไม่ว่าโมทิฟที่แสดงไว้ในรูปที่ 1.17 นั้นอยู่ตรงไหน นอกจากนี้ถ้าอนุญาตให้โมทิฟร่วมที่มองหานั้นสามารถมีความแตกต่างกันได้บ้างอย่างเช่นสามารถต่างกันได้ไม่เกิน 2 เบส (รูปที่ 1.19) จะออกแบบอัลกอริทึมที่มี

ประสิทธิภาพเพื่อหาดีเอ็นเอโมทิฟเหล่านี้ได้อย่างไร



รูปที่ 1.17 ส่วนหัวของ 7 ยีนที่มีการแสดงออกร่วมกัน มักมีดีเอ็นเอโมทิฟ (ส่วนของลำดับเบสสีแดง) ร่วมกัน

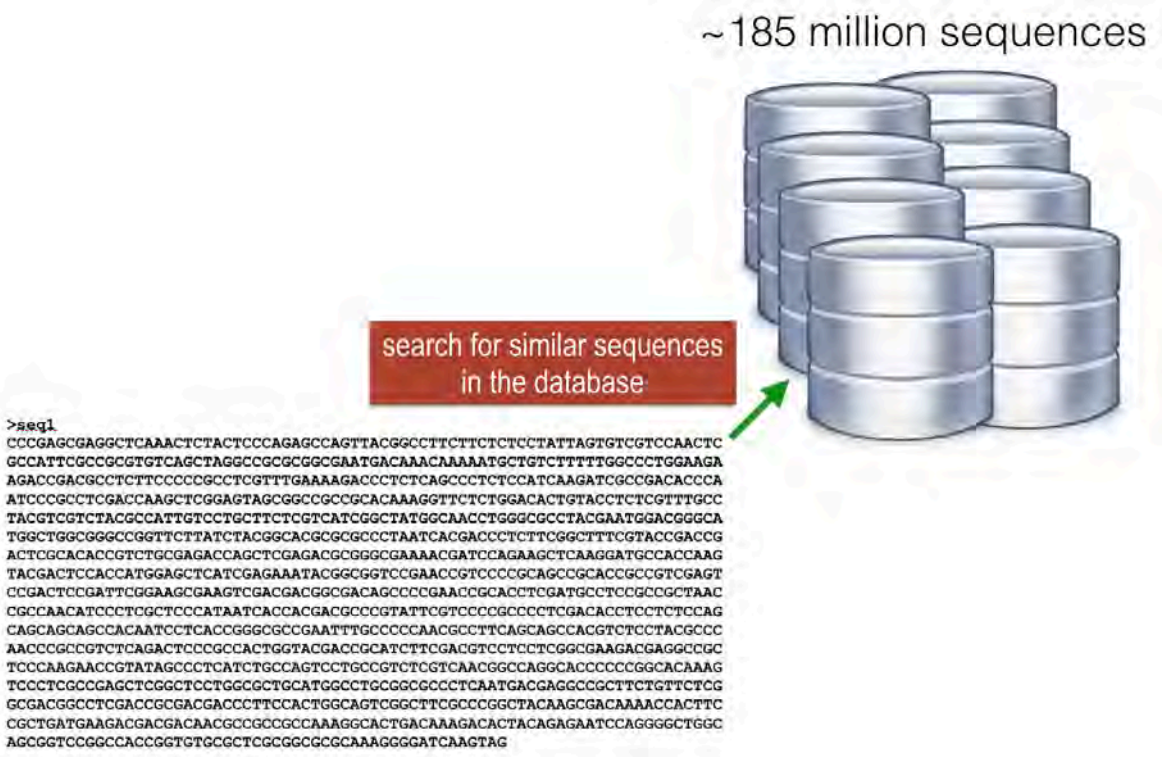
รูปที่ 1.18 ส่วนหัวของ 7 ยีนจากรูปที่แล้ว ที่มีการแสดงออกร่วมกัน แต่ลบสีส่วนที่แสดงดีเอ็นเอโมทิฟออก

CGGGGCTATCCAGCTGGGTCGTCACATTCCCCTTTCGATA
TTTGAGGGTGCCCAATAAGGGCAACTCCAAAGCGGACAAA
GGATGGATCTGATGCCGTTTGACGACCTAAATCAACGGCC
AAGGAAGCAACCCCAGGAGCGCCTTTGCTGGTTCTACCTG
AATTTTCTAAAAAGATTATAATGTCGGTCCTTGGAACTTC
CTGCTGTACAACTGAGATCATGCTGCATGCCATTTTCAAC
TACATGATCTTTTGATGGCACTTGGATGAGGGAATGATGC

รูปที่ 1.19 ตัวอย่างโมทิฟร่วมที่สามารถมีจำนวนเบสที่แตกต่างกันได้

ปัญหาการเทียบความคล้ายคลึงกันของลำดับเบสข้อมูลเข้ากับลำดับเบสในฐานข้อมูล

หลังจากได้บริเวณที่เป็นยีนในจีโนมแล้ว ข้อมูลถัดไปที่นักชีววิทยามักต้องการทราบคือยีนที่ทำนายได้เหล่านี้มีฟังก์ชันการทำงานอะไรบ้าง วิธีการหนึ่งเพื่อให้ได้มาซึ่งฟังก์ชันคือการนำลำดับเบสของยีนที่ทำนายได้ไปเทียบกับข้อมูลลำดับเบสที่อยู่ในฐานข้อมูลที่มีการระบุฟังก์ชันการทำงานมาก่อนหน้า (รูปที่ 1.20) คำถามในเชิงวิธีการคำนวณคือเราจะออก แบบอัลกอริทึมเพื่อเทียบความคล้ายคลึงกันของสายข้อมูลเข้าซึ่งอาจเป็นสายดีเอ็นเอหรือโปรตีนกับสายข้อมูลดีเอ็นเอหรือโปรตีนจำนวนมาก (มากกว่า 200 ล้านเส้น) ที่อยู่ในฐานข้อมูลได้ถูกต้องและรวดเร็วได้อย่างไร โปรแกรม BLAST [1] เป็นโปรแกรมที่ถูกใช้งานในการสืบค้นลำดับเบสที่คล้ายคลึงกันในฐานข้อมูลที่มีการใช้งานอย่างแพร่ หลายและมีอ้างอิงสูงสุดในระดับต้นๆ

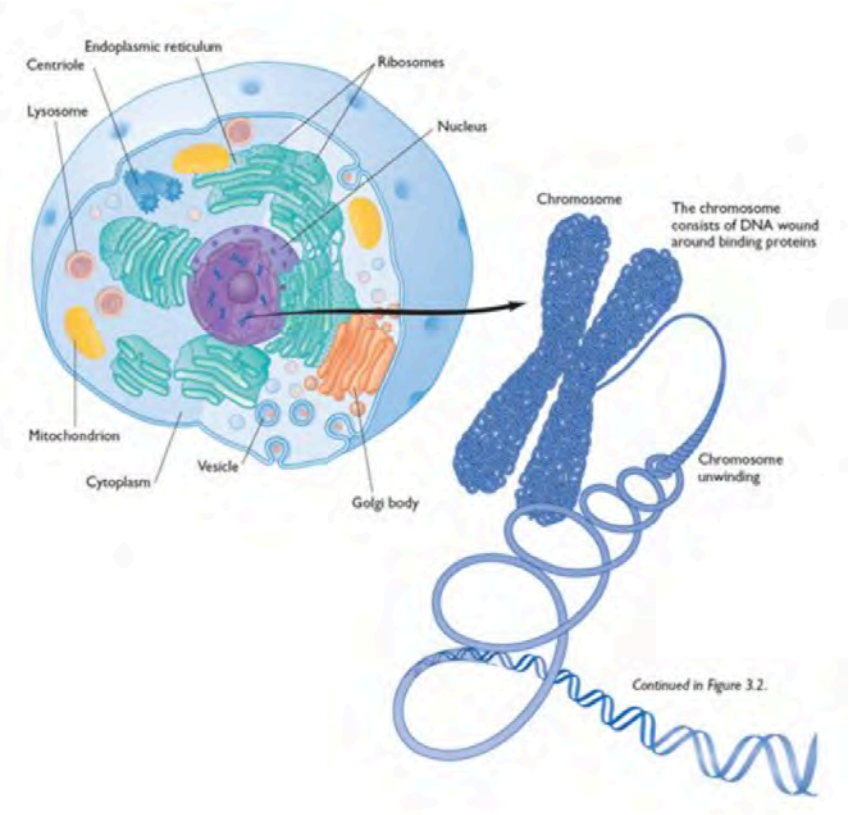


รูปที่ 1.20 การเทียบความคล้ายคลึงกันของลำดับเบสข้อมูลเข้ากับลำดับเบสในฐานข้อมูล

ความรู้พื้นฐานทางอณูชีววิทยา

เซลล์

เซลล์ (cell) (รูปที่ 1.21) เป็นองค์ประกอบพื้นฐานของสิ่งมีชีวิตทุกชนิด โดยในร่างกายมนุษย์ประกอบด้วยเซลล์จำนวนล้านล้านเซลล์ เซลล์เป็นส่วนประกอบของโครงสร้างร่างกาย รับและนำส่งสารอาหารและแปลงเป็นพลังงาน องค์ประกอบหลักของแต่ละเซลล์ประกอบด้วย ไซโตพลาสซึม (cytoplasm) ซึ่งประกอบด้วยของเหลวคล้ายวุ้นเรียกว่าไซโตซอล (cytosol) ออร์แกเนลล์ (organelle) หรือโครงสร้างย่อยอื่นๆประกอบด้วย ไลโซโซม (Lysosomes) มีหน้าที่หลักเป็นตัวนำส่งเอนไซม์ในการย่อยสลายสารต่างๆ เพอรอกซิโซม (Peroxisomes) มีโครงสร้างคล้ายไลโซโซม มีหน้าที่หลักในการย่อยสลายสารต่างๆรวมทั้งช่วยกำจัดสารที่เป็นพิษ เอนโดพลาสมิกเรติคูลัม (Endoplasmic reticulum: ER) มีหน้าที่หลากหลายโดยหน้าที่หลักๆ ประกอบด้วย ลำเลียงสารไปยังส่วนต่างๆของเซลล์ สนับสนุนการสร้างโปรตีนโดยจะเป็นที่เกาะของไรโบโซม เป็นต้น ไมโทคอนเดรีย (Mitochondrion) เป็นแหล่งพลังงานของเซลล์ กอลจิ (Golgi) มีหน้าที่หลักคือเป็นแหล่งเก็บสารที่เซลล์สร้างขึ้นก่อนส่งออกนอกเซลล์ นิวเคลียสเป็นศูนย์ควบคุมกลางของเซลล์รวมทั้งเป็นที่อยู่ดีเอ็นเอซึ่งเก็บรหัสพันธุกรรมในรูปแบบของโครโมโซม และไรโบโซม (Ribosomes) ที่มีหน้าที่สร้างโปรตีนจากรหัสพันธุกรรม เป็นต้น



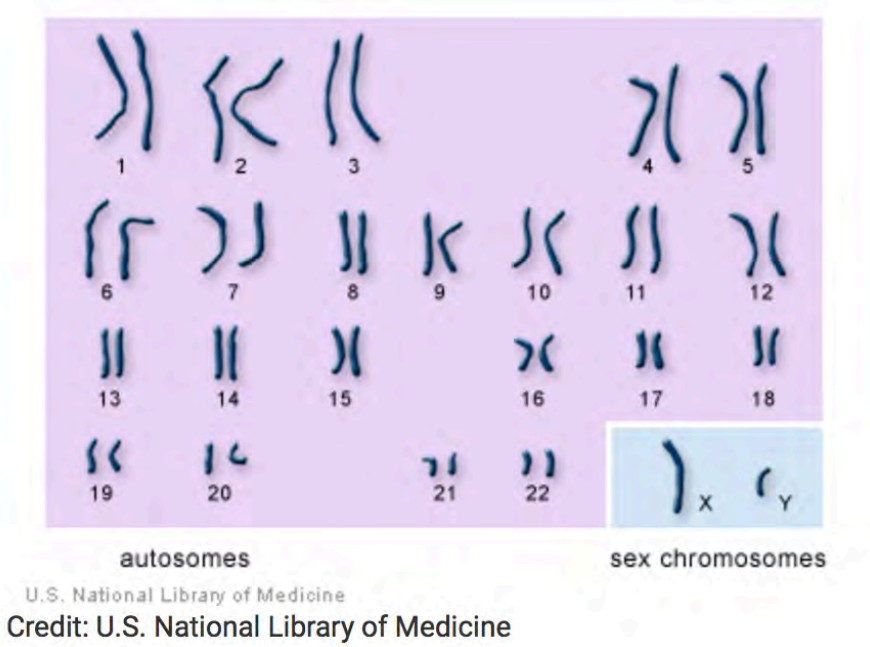
รูปที่ 1.21 องค์ประกอบพื้นฐานของเซลล์สิ่งมีชีวิตกลุ่มยูแคริโอต

(ที่มา: http://images.slideplayer.com/28/9397260/slides/slide_2.jpg) (© 2010 The McGraw-Hill Companies, Inc. All Rights Reserved)

โครโมโซม

โครโมโซม (chromosome) เป็นโมเลกุลเดี่ยวของดีเอ็นเอสายยาวโดยเป็นที่เก็บข้อมูลรหัสพันธุกรรมของสิ่งมีชีวิต รหัสพันธุกรรมในโครโมโซมมีโครงสร้างที่เป็นระบบและมีบริเวณจำเพาะที่เป็นยีนไม่มาก บริเวณที่เป็นยีนเหล่านี้สามารถถูกแปลรหัสไปเป็นอาร์เอ็นเอและเป็นโปรตีนเพื่อทำงานในกระบวนการต่างๆ โครโมโซมของสิ่งมีชีวิตบางชนิดเช่น มนุษย์ มีลักษณะเป็นสายหรือเชิงเส้น (linear) ในขณะที่โครโมโซมของสิ่งมีชีวิต เช่น แบคทีเรียมีลักษณะเป็นวงแหวน (circular) ในสิ่งมีชีวิตกลุ่มโพแคริโอต (prokaryote) วงแหวนโครโมโซมจะอยู่ในไซโทพลาสซึม (cytoplasm) ในบริเวณที่เรียกว่านิวคลีออยด์ (nucleoid) ในขณะที่ในสิ่งมีชีวิตกลุ่มยูแคริโอต (eukaryote) โครโมโซมทั้งหมดจะอยู่ในโครงสร้างที่เรียกว่านิวเคลียส (nucleus) โดยแต่ละโครโมโซมจะประกอบด้วยดีเอ็นเอที่พันตัวอย่างหนาแน่นรอบนิวเคลียร์โปรตีนที่เรียกว่าฮิสโตน (histone) ในมนุษย์โครโมโซมมี 2 ชุดโดยแต่ละชุดรับมาจากพ่อและแม่ตามลำดับ (รูปที่ 1.22) โดยจะมีทั้งหมด 46 โครโมโซม แบ่งเป็น 22 คู่ของ ออโตโซม (autosome) และมีโครโมโซมเพศอีก 2 โครโมโซม ซึ่งถ้าเป็นเพศหญิงจะมีโครโมโซมเอ็กซ์ (X) 2 โครโมโซมในขณะที่เพศชายจะมีโครโมโซมเอ็กซ์ (X) และวาย (Y) อย่างละ 1 โครโมโซม ออโตโซมคือโครโมโซมหรือชุดของโครโมโซมที่ควบคุมลักษณะทางพันธุกรรมและลักษณะที่แสดงออกต่างๆ เซลล์ที่มีโครโมโซม 2 ชุด เช่นเซลล์ทั่วไปของมนุษย์เรียกว่าเป็นดิพลอยด์ (diploid) ส่วนเซลล์สืบพันธุ์ที่จะพัฒนาต่อไปเป็นเซลล์ไข่ (egg) หรือเซลล์สเปิร์ม (sperm) เรียกว่าเป็นแฮพลอยด์ (haploid) ซึ่งจะมีโครโมโซมเพียง 1 ชุดหรือครึ่งหนึ่งของเซลล์ดิพลอยด์

ที่มา https://www.nature.com/scitable/definition/chromosome-6

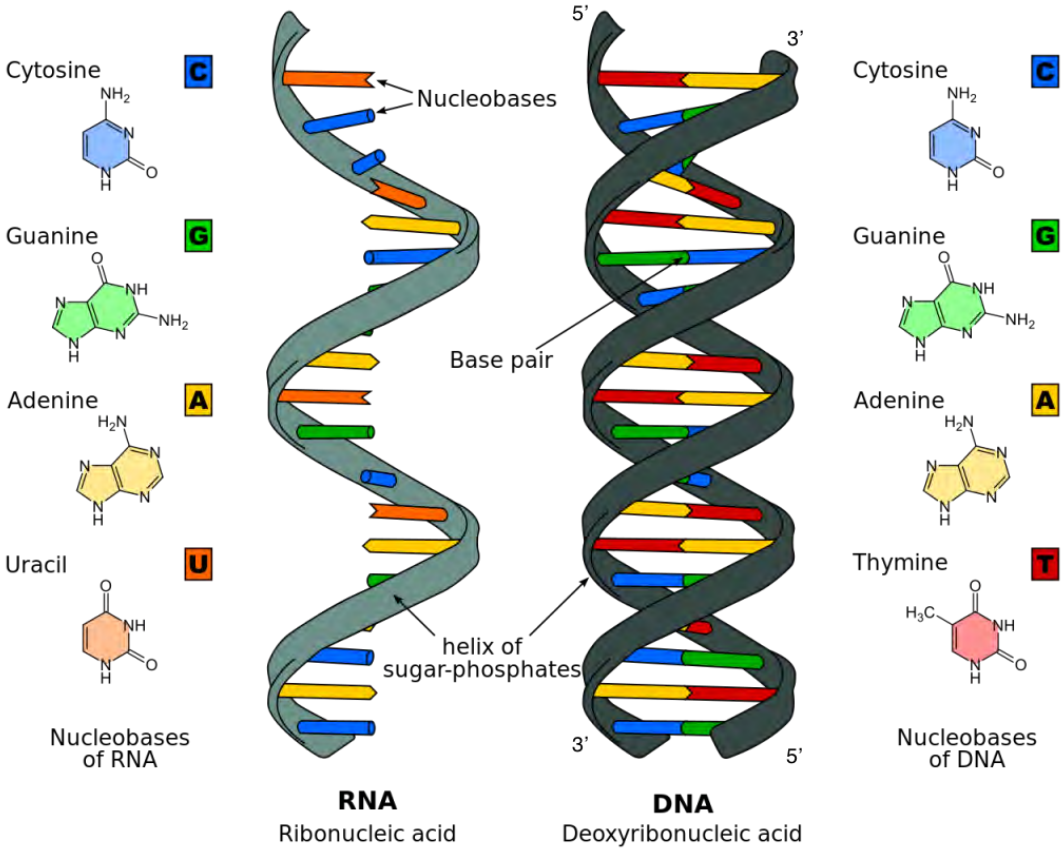


รูปที่ 1.22 โครโมโซมมนุษย์

(ที่มา: https://ghr.nlm.nih.gov/primer/basics/howmanychromosomes)

ดีเอ็นเอ

ดีเอ็นเอ (DNA) [25] หรือกรดดีออกซีไรโบนิวคลีอิค (Deoxy ribonucleic Acid) (รูปที่ 1.23) เป็นสารรหัสพันธุกรรมที่มีพบในสิ่งมีชีวิตทั้งในกลุ่มที่เป็นยูแคริโอต (Eukaryote) เช่น มนุษย์ ลิง หมู ช้าง ม้า วัว ยีสต์ และเชื้อรา เป็นต้น และกลุ่มที่เป็นโพรแคริโอต (Prokaryote) เช่นแบคทีเรีย ต่างๆ เป็นต้น รูปร่างของดีเอ็นเอมีลักษณะเป็นเกลียวคู่ (double helix) เส้นแรกจะมีการเรียงลำดับในทิศทาง 5’ (อ่านว่า five-prime) ไป 3’ (อ่านว่า three-prime) และถูกกำหนดว่าเป็นฟอร์เวิร์ดสแตรนด์ (forward strand) หรือ สแตรนด์สายบวก (plus strand) และอีกสายมีทิศทางจาก 3’ ไป 5’ ถูกกำหนดว่าเป็นรีเวิร์สสแตรนด์ (reverse strand) หรือ สแตรนด์สายลบ (minus strand) โดยดีเอ็นเอสองสายที่จับกันเป็นเกลียวนี้จะมีการเรียงลำดับของเบส (base) ในทางตรงกันข้ามกัน โดยแต่ละสายประกอบไปด้วยนิวคลีโอไทด์ (nucleotide) ที่เรียงต่อกัน ซึ่งแต่ละ นิวคลีโอไทด์ มีโครงสร้างหลักๆ ที่เป็นไปได้สี่แบบคือ ไซโตซีน (Cytocine) ไทมีน (Thymine) อะดีนีน (Adenine) และ กวานีน (Guanine) ในสายนิวคลีโอไทด์หนึ่งๆ ลำดับเบสเหล่านี้จะถูกแทนค่าด้วยตัวอักษร “C”, “T”, “A”, และ “G” ตามลำดับ ดีเอ็นเอสองสายที่จับกันเป็นเกลียวคู่นั้น คู่ของเบสที่จับกันมีได้สองแบบ คือไซโตซีน (C) จับกับ กวานีน (G) และ ไทมีน (T) จับกับ อะดีนีน (A)

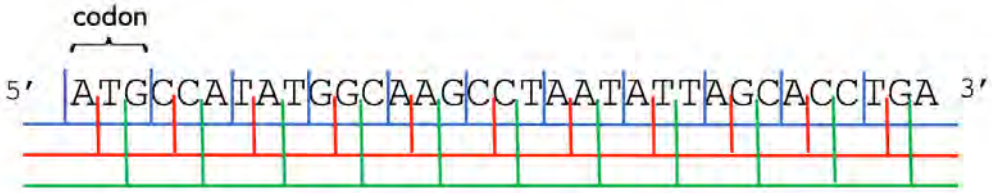


รูปที่ 1.23 ดีเอ็นเอ (DNA)

(ที่มา: http://www.basicknowledge101.com/categories/images/DNA_RNA.png)

การอ่านเฟรมในลำดับเบสนิวคลีโอไทด์

การอ่านเฟรม (reading frame) ของลำดับเบสนิวคลีโอไทด์ทั้งที่เป็นสายดีเอ็นเอและอาร์เอ็นเอ [26] สามารถอ่านได้ 3 แบบตามรูปที่ 1.24 โดยจะอ่านทีละ 3 เบสที่อยู่ติดกันเรียกว่า 1 โคดอน (codon) โดยเฟรมแบบที่หนึ่งแสดงชุดของโคดอนที่ระบุด้วยสีน้ำเงินโดยโคดอนแรกคือ “ATG” และโคดอนที่สองคือ “CCA” ตามลำดับ โดยจะไม่มีการอ่านทับซ้อนลำดับเบสระหว่างสองโคดอนภายในเฟรม สำหรับเฟรมแบบที่สองแสดงชุดของโคดอนที่มีการเลื่อนไปหนึ่งเบส (เกิด frame shift) แสดงโดยสีแดง ในกรณีนี้โคดอนแรกคือ “TGC” และโคดอนที่สองคือ “CAT” ตามลำดับ สำหรับเฟรมแบบที่สามจะมีการเลื่อนไปสองเบส (เกิด frame shift) แสดงโดยสีเขียว ในกรณีนี้โคดอนแรกคือ “GCC”และโคดอนที่สองคือ “ATA” ตามลำดับ ดังนั้นในสายอาร์เอ็นเอหนึ่งเส้นสามารถถูกอ่านออกมาได้ 3 แบบในทิศทางการอ่านจาก 5’ ไปยังทิศทาง 3’ ในฟอร์เวิร์ดสแตรนด์นอกจากนี้ถ้าพิจารณาสายที่เป็นคู่เกลียวในระดับดีเอ็นเอด้วยก็จะสามารถอ่านได้อีก 3 แบบโดยมีทิศทางจาก 3’ ไปยัง 5’ โดยอ่านจากรีเวิร์สสแตรนด์ รวมมีการอ่านที่เป็นไปได้ทั้งหมด 6 แบบ ทั้งนี้โคดอนแต่ละแบบจะสามารถแปลรหัสต่อไปเป็นกรดอะมิโนใดสามารถดูได้จากตารางโคดอนในรูปที่ 1.25



รูปที่ 1.24 การอ่านเฟรมของลำดับเบสนิวคลีโอไทด์

Open Reading Frame (ORF)

ในอณูวิทยา open reading frame (ORF) คือส่วนของเฟรมที่ถูกอ่านออกมาตามรูปแบบที่อธิบายไว้ในหัวข้อ *การอ่านเฟรมของลำดับเบสนิวคลีโอไทด์* โดย ORF เป็นส่วนของเฟรมที่มีโอกาสจะสามารถถูกแปลรหัสต่อไปเป็นโปรตีนได้โดยจะประกอบด้วยสายของโคดอนที่โคดอนแรกจะต้องเป็นรหัสพันธุกรรมเริ่มต้น (initiation codon หรือ start codon) หรือโคดอน “AUG” (แปลงมาจาก “ATG” ในระดับดีเอ็นเอ โดยการเปลี่ยนจากไทมีน (T) เป็นยูราซิล (U)) ในรูปที่ 1.25 และโคดอนหยุด (termination codon หรือ stop codon) ซึ่งสามารถเป็นโคดอน “UAA”, “UAG” หรือ “UGA” (แปลงมาจาก “TAA”, “TAG” หรือ “TGA” ในระดับดีเอ็นเอ ตามลำดับ) ก็ได้ จากรูปที่ 1.26 แสดงตัวอย่างของ ORF ที่เป็นไปได้ 2 แบบจากที่เกิดจากการอ่านเฟรมลำดับเบสนิวคลีโอไทด์ที่แตกต่างกัน โดยในเฟรมแบบที่ 3 ไม่มี ORF โดยทั่วไปในผลของการเฟรมนิวคลีโอไทด์ 3 แบบมักมีเพียงรูปแบบเดียวที่มี ORF ที่สามารถแปลงต่อไปเป็นโปรตีนที่มีลำดับกรดอะมิโนที่ถูกต้องครบถ้วนได้ สำหรับเฟรมที่พบเฉพาะโคดอนปิดจะไม่สามารถถูกแปลรหัสไปเป็นโปรตีนต่อได้ เช่นกัน

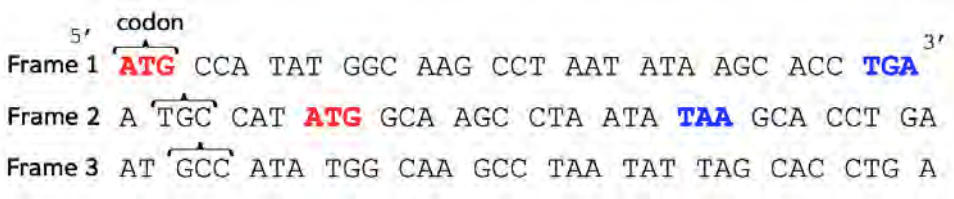
ยีน

ยีน (gene) [27] เป็นส่วนของดีเอ็นเอที่สามารถถอดรหัสต่อไปเป็นเมสเซนเจอร์อาร์เอ็นเอหรือเอ็มอาร์เอ็นเอ (messenger RNA: mRNA) หรืออาร์เอ็นเอสื่อสาร และแปลรหัสต่อไปเป็นโปรตีน (protein) ตามหลักการเซ็นทรัลดอกมา (central dogma) ยีนในสิ่งมีชีวิตกลุ่มยูแคริโอต เช่นยีนในมนุษย์ มีโครงสร้างที่ซับซ้อนมากกว่า ยีนในสิ่งมีชีวิตกลุ่มโพรแคริโอต เช่นแบคทีเรีย โดยจะมีทั้งส่วนที่เป็นเอ็กซอน (exon) และส่วนที่เป็นอินทรอน (intron) (รูปที่ 1.27) ในขณะที่กลุ่มโพรแคริโอตจะไม่มีส่วนที่เป็นอินทรอน (รูปที่ 1.28) เป็นต้น โดยยีนของทั้งสองกลุ่มยังประกอบด้วยส่วนที่ไม่สามารถแปลรหัสไปเป็นโปรตีนได้เรียกว่า untranslated region (UTR) ซึ่งอยู่ที่ส่วนหัวของยีนจากทิศทาง 5’ เรียกว่า 5’ UTR และส่วนที่เป็นส่วนท้ายของยีนจากทิศทาง 3’ เรียกว่า 3’UTR นอกจากนี้ยังมีส่วนของโปรโมเตอร์ซึ่งมักอยู่ติดกับ 5’ UTR และส่วนของเอ็นแฮนเซอร์ (enhancer) และไซเลนเซอร์ (silencer)

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

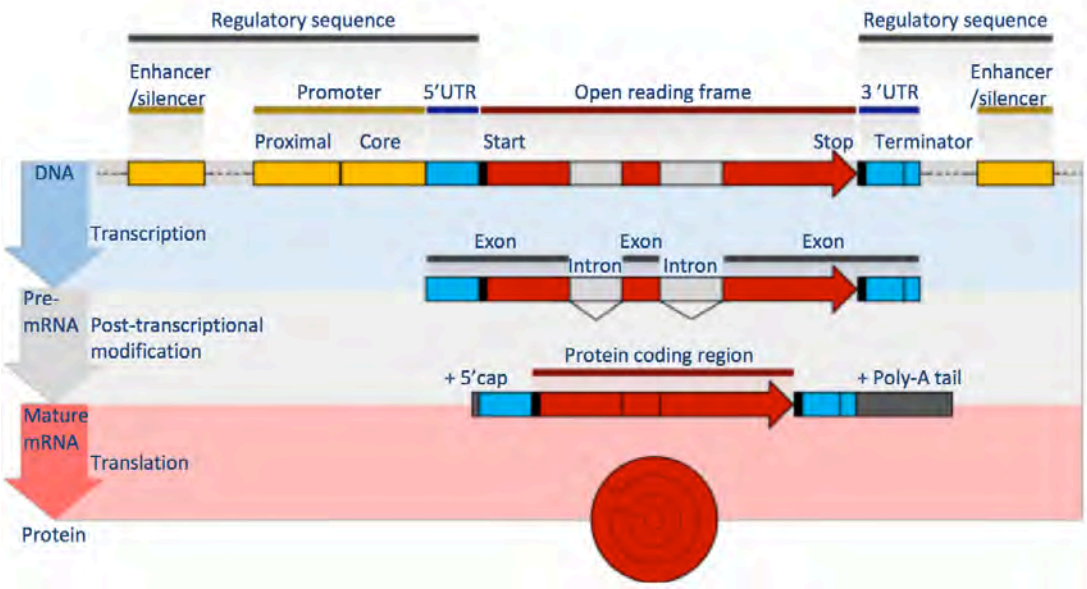
รูปที่ 1.25 ตารางการแปลงโคดอนไปเป็นกรดอะมิโน

(ที่มา: รูปที่ 2-3 หน้า 38 [25])



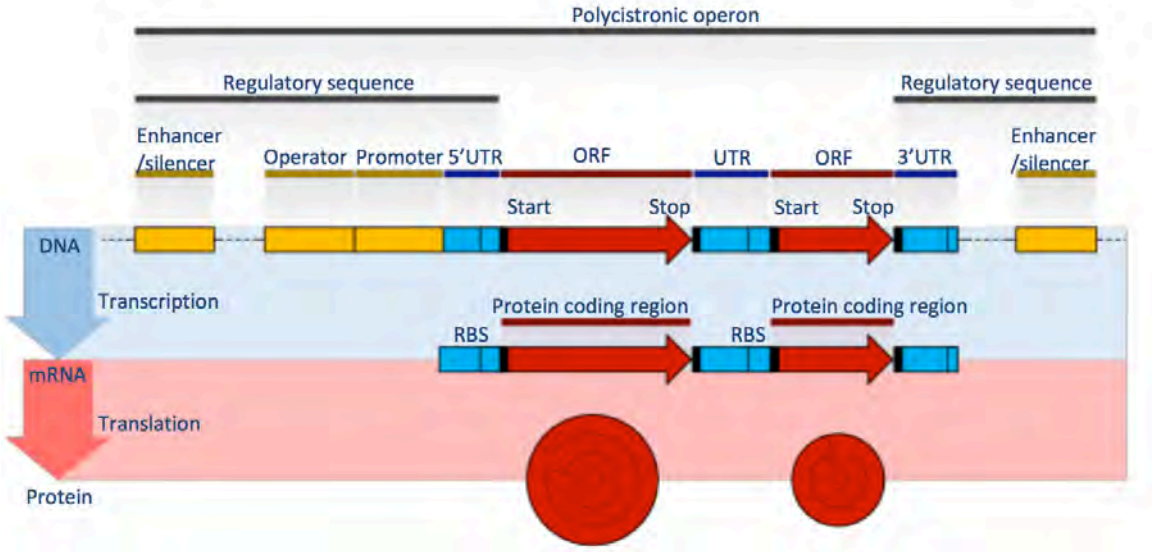
รูปที่ 1.26 ตัวอย่างของ ORFs

ที่อยู่ห่างออกไปทั้งในส่วนของ 5’ และ 3’ UTR ส่วนของโปรโมเตอร์และไซเลนเซอร์นี้เกี่ยวข้องกับการควบคุมการถอดรหัสยีนไปเป็น pre-mRNA ซึ่งจะมีการแก้ไข pre-mRNA เพิ่มเติมโดยนำเฉพาะส่วนที่เป็นเอ็กซอนมาต่อกันกลายเป็น coding sequencing และมีการเติม 5’ cap ที่ส่วนหัวของยีนให้สามารถโน้มนำให้ไรโบโซมเข้ามาจับและเติมสายของลำดับเบสอะดีนีน poly-A tail ที่ส่วนท้ายเพื่อเพิ่มความเสถียรของเมสเซ็นเจอร์อาร์เอ็นเอ ส่วน 5’ UTR และ 3’ UTR ของเมสเซ็นเจอร์อาร์เอ็นเอควบคุมการแปลรหัสไปเป็นโปรตีน



รูปที่ 1.27 ตัวอย่างโครงสร้างยีนของสิ่งมีชีวิตกลุ่มยูแคริโอต

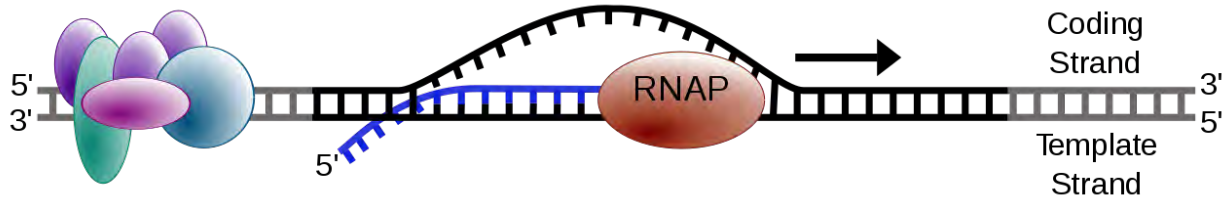
(ที่มา: รูปที่ 1 ของ [27])



รูปที่ 1.28 ตัวอย่างโครงสร้างยีนของสิ่งมีชีวิตกลุ่มโพรแคริโอต

(ที่มา: รูปที่ 2 ของ [27])

เนื่องจากดีเอ็นเอเป็นสายเกลียวคู่ ยีนหนึ่งๆอาจอยู่บนดีเอ็นเอที่เป็นสายบวกมีทิศทางจากซ้ายไปขวา (ฟอร์เวิร์ดสแตรนด์) หรือสายลบมีทิศทางจากขวามาซ้าย (รีเวิร์สสแตรนด์) ก็ได้ โดยสายที่มียีนหนึ่งๆ อยู่เรียกว่าเป็นโคดดิงสแตรนด์ (coding strand) หรือ เซ้นส์สแตรนด์ (sense strand) ของยีนนั้นๆ และดีเอ็นเออีกสายของเกลียวคู่ก็จะเป็นเทมเพลตสแตรนด์ (template strand) หรือ แอนไทเซ้นส์สแตรนด์ (antisense strand) โดยในการถอดรหัสของยีนนั้นๆ ไปเป็นเมสเซนเจอร์อาร์เอ็นเอรวมไปถึงโปรตีนของยีนนั้นๆ จะมีทิศทางเดียวกับยีนที่อยู่บนโคดดิงสแตรนด์ทั้งนี้เนื่องจากในระหว่างการอ่านนิวคลีโอไทด์บนสายดีเอ็นเอเกลียวคู่อาร์เอ็นเอโพลีเมอเรส (RNAP ในรูปที่ 1.29) จะเคลื่อนที่ในทิศทางจาก 5’ ไป 3’ โดยการอ่านและสร้างสายนิวคลีโอไทด์ของเมสเซนเจอร์อาร์เอ็นเอ (เส้นสีน้ำเงิน) จากสายที่เป็นเทมเพลตสแตรนด์ซึ่งมีทิศทาง 3’ ไป 5’ จึงได้ลำดับเบสของเมสเซนเจอร์อาร์เอ็นเออยู่ในทิศทาง 5’ ไป 3’ ตามโคดดิงสแตรนด์



รูปที่ 1.29 การอ่านดีเอ็นเอเพื่อถอดรหัสไปเป็นเมสเซ็นเจอร์อาร์เอ็นเอ

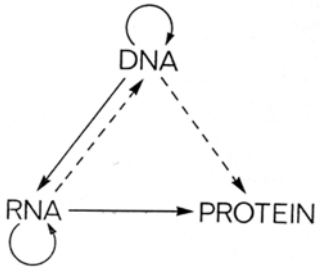
(ที่มา: https://en.wikipedia.org/wiki/File:Simple_transcription_elongation1.svg)

หลักการเซ็นทรัลดอกมา

หลักการเซ็นทรัลดอกมา (central dogma) เป็นการอธิบายของการไหลของข้อมูลรหัสพันธุกรรมโดยมีการอธิบายไว้ครั้งแรกโดย ฟรานซิส คริก (Francis Crick) ในปี 1958 [28] และมีการนำเสนอในปี 1970 [29] โดยคริกอีกครั้งโดยระบุว่าในกระบวนการถ่ายโอนข้อมูลของรหัสพันธุกรรมนั้นสามารถถ่ายโอนได้ระหว่างกรดนิวคลีอิค หรือจากกรดนิวคลีอิคไปเป็นโปรตีน แต่จะไม่มีการถ่ายโอนระหว่างโปรตีน หรือการถ่ายโอนย้อนกลับจากโปรตีนมายังกรดนิวคลีอิค (รูปที่ 1.30)

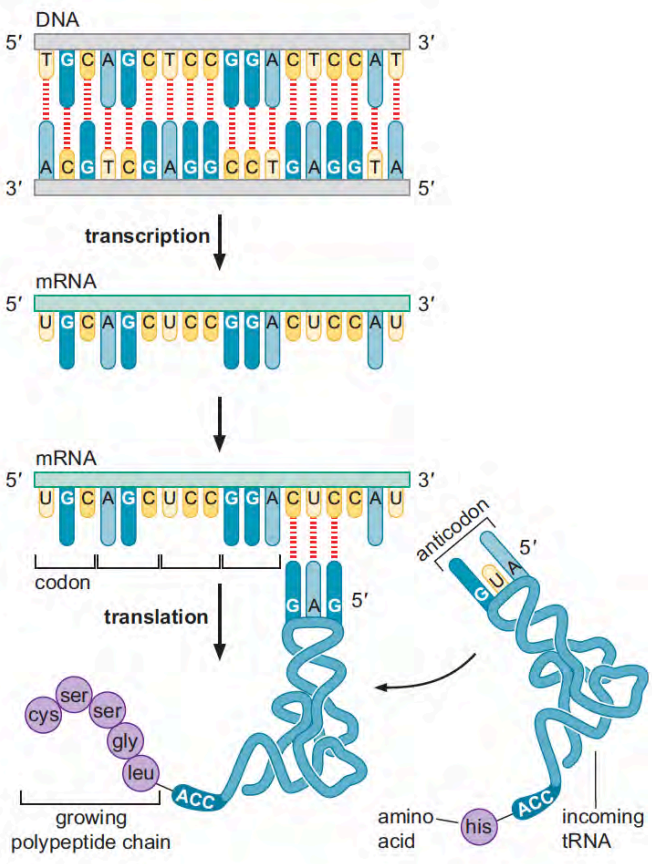
หลักการเซ็นทรัลดอกมา [25] ถือเป็นกระบวนการพื้นฐานที่สำคัญในอณูชีววิทยาของสิ่งมีชีวิต หลักการทำงานในส่วนของการสร้างโปรตีนจากดีเอ็นเอ (รูปที่ 1.31) ประกอบด้วย 1) การถอดรหัสจากสายดีเอ็นเอ (DNA) ไปยังเมสเซ็นเจอร์อาร์เอ็นเอโดยอาร์เอ็นเอโพลีเมอเรส (RNA polymerase) เรียกว่ากระบวนการทรานสคริปชัน (transcription) โดยเบสที่เป็นไทมีน (T) เดิมจะถูกเปลี่ยนเป็นนิวคลีโอไทด์ใหม่เรียกว่ายูราซิล (Uracil) ซึ่งจะถูกแทนด้วยตัวอักษร “U” 2) การแปลรหัสจากเมสเซ็นเจอร์อาร์เอ็นเอไปยังโปรตีน (protein) ซึ่งเรียกว่ากระบวนการทรานสเลชัน (translation) ในกระบวนการทรานสเลชันนี้ไรโบโซม (ribosome) ทำหน้าที่เป็นตัวจับและอ่านข้อมูลจากเมสเซนเจอร์อาร์เอ็นเอในทิศทางจาก 5’ ไปยัง 3’ เพื่อกำหนดกรดอะมิโนที่ต้องการ โดยมีทรานสเฟอร์อาร์เอ็นเอหรือทีอาร์เอ็นเอ (transfer RNA: tRNA) ทำหน้าที่ช่วยขนย้ายกรดอะมิโนที่จำเพาะนั้นมา

ยังไรโบโซมและเชื่อมต่อกรดอะมิโนที่นำมานั้นเข้ากับสายโพลีเปปไทด์ (โปรตีน) ทั้งนี้แต่ละสามเบสที่อยู่ติดกันในเมสเซ็นเจอร์อาร์เอ็นเอจะนับเป็นหนึ่งหน่วยโคดอน (codon) ซึ่งแต่ละโคดอนจะถูกแปลรหัสไปเป็นหนึ่งกรดอะมิโนตามรูปที่ 1.25



รูปที่ 1.30 แสดงเส้นทางการถ่ายโอนข้อมูลรหัสพันธุกรรมที่เป็นไปได้โดยฟรานซิส คริก ในปีค.ศ. 1970 โดยลูกศรเส้นทึบแสดงทิศทางการถ่ายโอนข้อมูลรหัสพันธุกรรมที่เกิดโดยทั่วไป (probable) ลูกศรเส้นประแสดงการถ่ายโอนข้อมูลรหัสพันธุกรรมที่เป็นไปได้ (possible) และเส้นที่หายไปคือไม่สามารถเกิดขึ้นได้

(ที่มา: ภาพที่ 2 ของ [29])

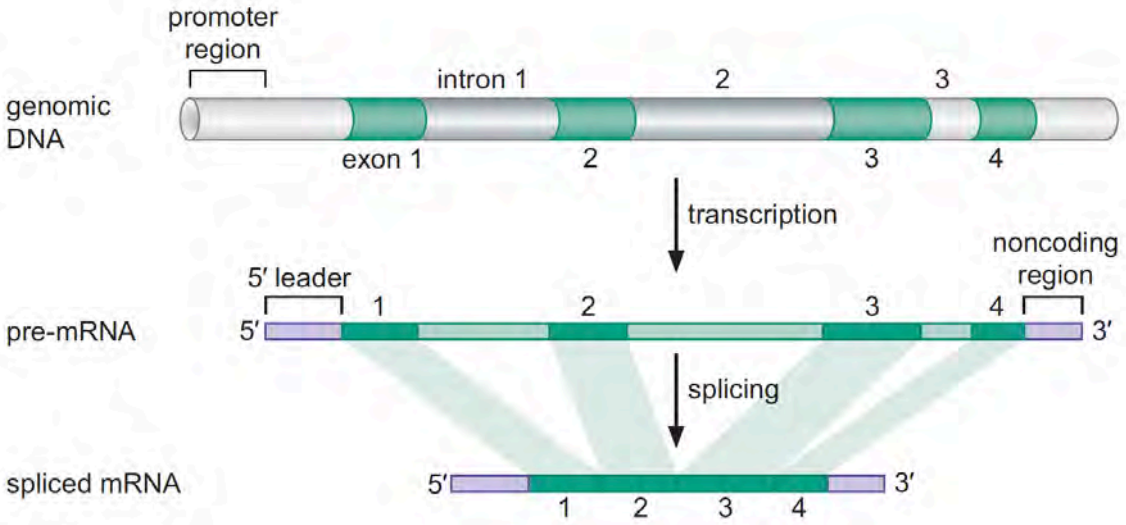


รูปที่ 1.31 กระบวนการทรานสคริปชันและทรานสเลชัน

(ที่มา: รูปที่ 2-15 หน้า 36 [25])

RNA Splicing

ในสิ่งมีชีวิตกลุ่มยูแคริโอตเมสกระบวนการแปลรหัสหรือทรานสคริปชันมีรายละเอียดแยกย่อยดังแสดงในรูปที่ 1.32 โดยในผลขั้นแรกของการแปลรหัสจะได้สาย primary transcript หรือ pre-mRNA ซึ่งยังมีส่วนที่เป็นอินทรอนประกอบอยู่ หลังจากนั้นจะมีกระบวนการที่สองคือการทำ RNA splicing หรือการต่ออาร์เอ็นเอโดยบริเวณที่เป็นเอ็กซอนใน pre-mRNA จะถูกเลือกมาต่อกันให้เป็นเมสเซ็นเจอร์อาร์เอ็นเอเรียกว่า mature mRNA ซึ่งประกอบเป็น coding sequence (CDS) เริ่มด้วย start codon (“AUG”) และปิดท้ายด้วย stop codon (“UGA”, “UAG”, หรือ “UAA”) เป็นหลักที่พร้อมจะถูกนำไปแปลรหัสเป็นโปรตีนต่อไป

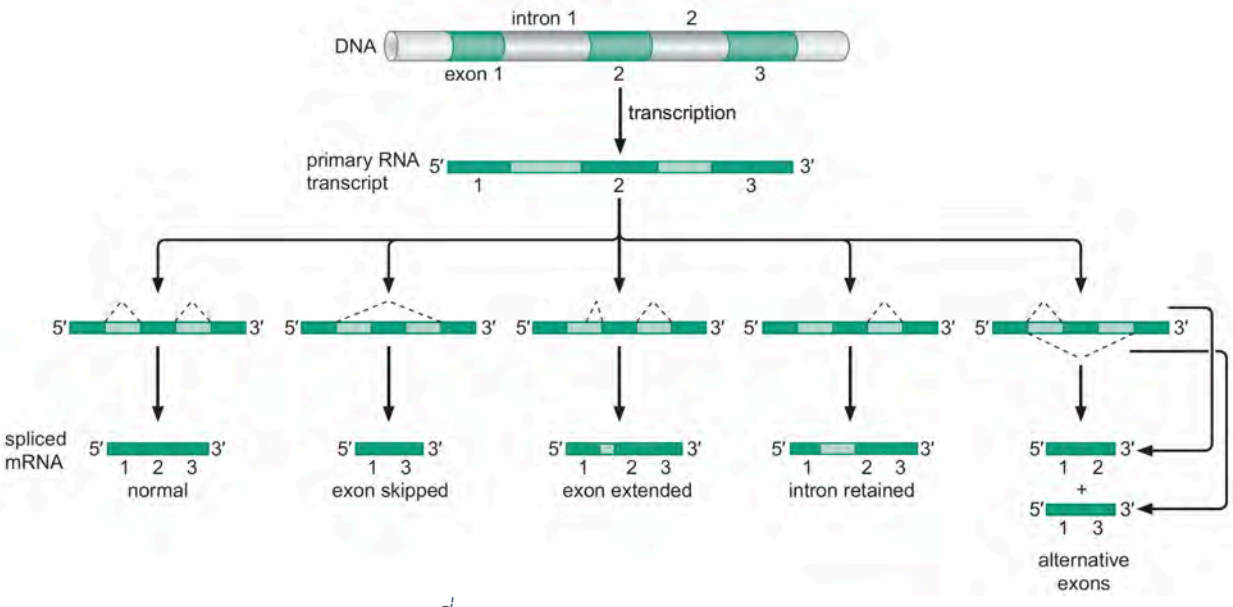


รูปที่ 1.32 โครงสร้างพื้นฐานของยีนในกลุ่มยูแคริโอต

(ที่มา: รูปที่ 14-1 หน้า 468 [25])

ทั้งนี้กระบวนการทำ RNA splicing นี้ เอ็กซอนที่อยู่ใน pre-mRNA อาจถูกเลือกมาต่อไม่ครบทำให้สามารถมีเมสเซ็นเจอร์อาร์เอ็นเอที่พร้อมนำไปแปลรหัสเป็นโปรตีนได้มากกว่า 1 แบบ (มากกว่า 1 ไอโซฟอร์ม) จากยีนเดียวกัน นอกจากนี้ในบางกรณีส่วนที่เป็นอินทรอนอาจยังถูกนำมาต่อเป็นส่วนของเมสเซ็นเจอร์อาร์เอ็นเอที่จะถูกนำไปแปลรหัสด้วย ดังแสดงในรูปที่ 1.33 โดยในรูปแบบที่ 3 และ 4 ยังมีส่วนของอินทรอนประกอบอยู่

รูปที่ 1.34 แสดงตัวอย่างลำดับเบสในส่วนของโคดดิงซีเควนซ์ (มีเฉพาะส่งเอ็กซอนที่ถูกนำมาต่อกัน) ของยีน BRCA1 ที่เกี่ยวข้องกับมะเร็งเต้านม โดยอยู่ในรูปแบบฟาสต้า (FASTA format) ซึ่งประกอบด้วย 2 บรรทัด บรรทัดแรกขึ้นต้นด้วยอักขระ “>” เสมอ จากนั้นตามด้วยคำอธิบายลำดับเบส อาจเป็นชื่อยีนและรายละเอียดหรืออาจเป็นรหัสตัวเลขก็ได้ ไม่มีข้อจำกัดในส่วนนี้ บรรทัดที่สองเป็นลำดับเบสของโคดดิงซีเควนซ์ และรูปที่ 1.35 แสดงตัวอย่างลำดับกรดอะมิโนที่ถูกแปลรหัสมาจากโคดดิงซีเควนซ์ของยีน BRCA1 โดยอยู่ในรูปแบบฟาสต้า (FASTA format) เช่นกัน



รูปที่ 1.33 Alternative splicing

(ที่มา: รูปที่ 14-15 หน้า 484 [25])

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

รูปที่ 1.34 ตัวอย่างโคดิงซีเควนซ์ของยีน BRCA1 ในรูปแบบฟาสต้า

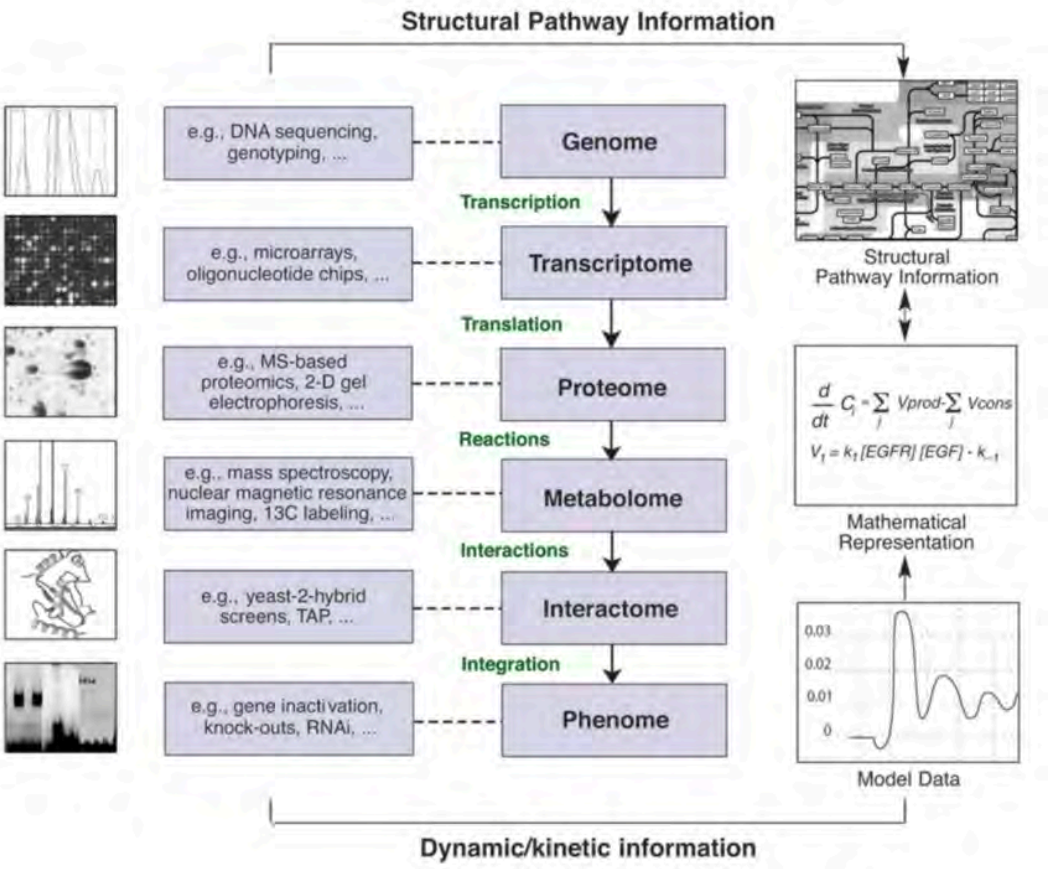
(ที่มา: http://www.ebi.ac.uk/ena/data/view/U64805)

```
>sp|P38398|BRCA1_HUMAN Breast cancer type 1 susceptibility protein OS=Homo sapiens GN=BRCA1 PE=1 SV=2
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 ตัวอย่างลำดับกรดอะมิโนที่ถูกแปลรหัสมาจากโคดดิงซีเควนซ์ของยีน BRCA1 โดยอยู่ในรูปแบบฟาสต้า

เทคโนโลยีโอมิกส์

เทคโนโลยีโอมิกส์ (Omics technology) [30] (รูปที่ 1.36) เป็นการศึกษารายละเอียดในองค์รวมเกี่ยวกับโมเลกุลต่างๆ ที่ประกอบขึ้นเป็นเซลล์ เนื้อเยื่อ อวัยวะ และสิ่งมีชีวิต โดยประกอบไปด้วยเทคโนโลยีจีโนมิกส์ (genomics) ที่ใช้ในการศึกษารหัสพันธุกรรมเพื่อตรวจจับยีนทั้งหมดในจีโนม (genome) เทคโนโลยีทรานสคริปโตมิกส์ (transcriptomics) ที่ใช้ในการตรวจวัดเมสเซ็นเจอร์อาร์เอ็นเอ (messenger RNAs: mRNAs) หรือยีนที่แสดงออกโดยยีนทั้งหมดที่แสดงออกเรียกว่าทรานสคริปโตม เทคโนโลยีโปรตีโอมิกส์ (proteomics) ที่ใช้ในการตรวจวัดโปรตีน โดยโปรตีนที่แสดงออกทั้งหมดเรียกว่าโปรตีโอม และเทคโนโลยีเมทาโบโลมิกส์ (metabolomics) ที่ใช้ในการตรวจวัดเมทาโบไลท์ (metabolites) ที่อยู่ในเซลล์ตัวอย่างหนึ่งๆ โดยเมทาโบไลท์ทั้งหมดที่แสดงออกเรียกว่าเมทาโบโลม ทั้งนี้จำนวน (เช่นมีเมสเซ็นเจอร์อาร์เอ็นของยีน 1000 จาก 30000 ยีนที่แสดงออก) และปริมาณเมสเซ็นเจอร์อาร์เอ็นเอ โปรตีน และเมทาโบไลท์ที่วัดได้ในเซลล์หนึ่งๆนั้น จะมีลักษณะพลวัตสามารถเปลี่ยนแปลงตามเวลาที่เปลี่ยนไป มีความจำเพาะกับเงื่อนไขทดสอบในการตรวจวัด รวมทั้งมีความจำเพาะกับประเภทของเซลล์และรวมทั้งเนื้อเยื่อที่ใช้ทดสอบ

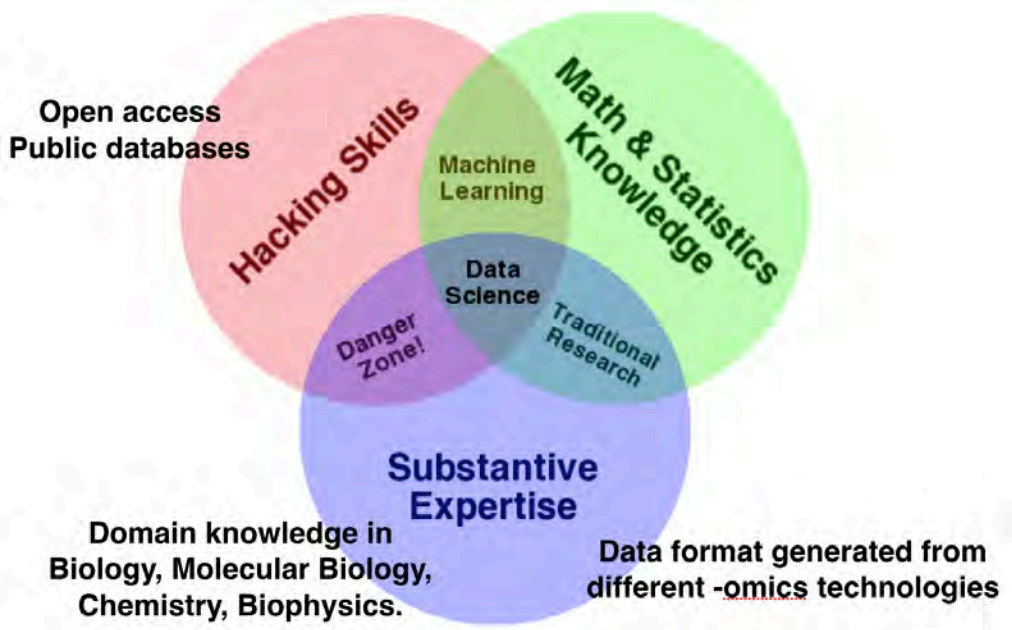


รูปที่ 1.36 เทคโนโลยีโอมิกส์

(ที่มา: http://pubs.niaaa.nih.gov/publications/arh311/49-59.htm)

วิทยาศาสตร์ข้อมูลทางชีววิทยา

รูปที่ 1.37 แสดงแผนภาพเวนน์ที่นำเสนอโดย Drew Conway เกี่ยวกับองค์ความรู้และทักษะที่จำเป็นในการศึกษาและทำงานทางด้านวิทยาศาสตร์ข้อมูล ซึ่งประกอบด้วย 3 ส่วนหลักคือ (1) องค์ความรู้ทางคณิตศาสตร์และสถิติ (2) ความสามารถในการได้มาซึ่งข้อมูล และการจัดการข้อมูลให้อยู่ในรูปแบบที่เหมาะสม และ (3) ความเข้าใจในข้อมูลที่จำเพาะกับบริบทเช่นข้อมูลทางการตลาด ข้อมูลความเชื่อมโยงของเครือข่ายออนไลน์ เป็นต้น และสามารถอธิบายความหมายได้ ทั้งนี้งานทางด้านวิทยาศาสตร์ข้อมูลทางชีววิทยาสามารถอธิบายโดยใช้แผนภาพเวนน์ของ Drew Conway เช่นกัน โดยต้องการ (1) องค์ความรู้ทางคณิตศาสตร์และสถิติ (2) ความสามารถในการได้มาซึ่งข้อมูล และการจัดการข้อมูลให้อยู่ในรูปแบบที่เหมาะสม เช่นรู้ว่าจะได้มาซึ่งข้อมูลต่างๆ ได้อย่างไร รู้ว่าข้อมูลที่ต้องการนั้นสามารถหาได้จากฐานข้อมูลไหน ความน่าเชื่อถือของแต่ละฐานข้อมูล ลักษณะการเข้าถึงข้อมูล สามารถดาวน์โหลดข้อมูลได้ทั้งชุดหรือต้องเรียกดาวน์โหลดเป็นรายการผ่าน API เป็นต้น และ (3) ความเข้าใจกระบวนการและข้อมูลทางชีววิทยาและอณูวิทยา ลักษณะของข้อมูลที่เกิดจากเทคโนโลยีโอมิกส์ต่างๆ ความหมายของข้อมูล เป็นต้น



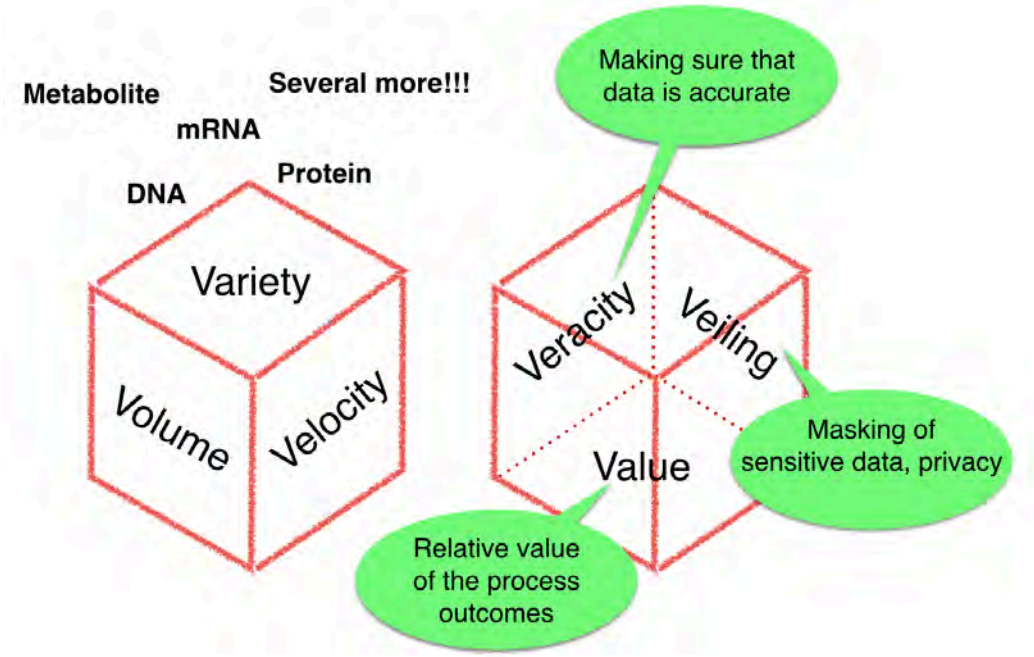
รูปที่ 1.37 แผนภาพเวนน์ของ Drew Convey เพื่ออธิบายวิทยาศาสตร์ข้อมูลทางชีววิทยา

(ที่มา: http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram โดยปรับเพิ่มเติมตัวอย่างที่เกี่ยวข้องกับข้อมูลทางชีววิทยา)

บิกดาต้ากับชีวสารสนเทศ

ข้อมูลใดๆ ที่ถูกขนานนามว่าเป็นข้อมูลขนาดใหญ่หรือบิกดาต้านั้นเกิดขึ้นด้วยองค์ประกอบหลัก 3 ประการ (รูปที่ 1.38) คือ ขนาดของข้อมูล (Volume) ความหลากหลายของข้อมูล (Variety) และอัตราการเกิดข้อมูลใหม่ (Velocity) ตัวอย่างข้อมูลขนาดใหญ่เช่นข้อมูลโพสของเฟสบุ๊ค (Facebook) โดยข้อมูลมีลักษณะที่หลากหลายไม่ได้จำกัดเฉพาะตัวอักษรและประโยคต่างๆ แต่ยังรวมถึงวีดีโอ เสียง เป็นต้น และข้อมูลเหล่านี้มีจำนวนเพิ่มขึ้นอย่างมากในทุกๆวันจากทั่วโลก ในกรณีของข้อมูลโอมิกส์ขนาดข้อมูลโดยเฉพาะข้อมูลจีโนม (genomes) และเอ็กโซม (exomes) มีจำนวนเพิ่มขึ้นอย่างมาก (รูปที่ 1.39) ด้วยราคาของเทคโนโลยีการถอดรหัสพันธุกรรมที่ถูกลงอย่างมากเมื่อเทียบกับเมื่อหลายปีก่อน (รูปที่ 1.40) ในเชิงของความหลากหลายของข้อมูลนอกจากข้อมูลรหัสพันธุกรรมแล้วยังมีข้อมูลจากเทคโนโลยีโอมิกส์อื่นๆ ไม่ว่าจะเป็นข้อมูลการแสดงออกของยีนผ่านเทคโนโลยีอาร์เอ็นเอซีค (RNA-Seq) ที่มีลักษณะเดียวกับข้อมูลลำดับเบสดีเอ็นเอ หรือข้อมูลไมโครอะเรย์ที่มีลักษณะเป็นตารางสองมิติโดยแต่ละบรรทัดแสดงข้อมูลของยีนหนึ่งโดยแต่ละคอลัมน์แสดงปริมาณการแสดงออกของยีนนั้นๆตามเงื่อนไขการทดลองจำเพาะหนึ่งๆ ข้อมูลการออกแสดงของโปรตีนจากเทคโนโลยีแมสสเปคโตรเมทรี (mass spectrometry) ที่อยู่ในรูปแบบของพีค (peak) จำนวนมาก ข้อมูลการแสดงออกของเมทาโบไลท์ เป็นต้น นอกจากองค์ประกอบหลัก 3 ส่วนข้างต้นแล้วยังมีการเพิ่มเติมการพิจารณาข้อมูลในอีก 3 มิติคือ ความถูกต้องของข้อมูล (Veracity) คุณค่าของข้อมูล (Value) และการรักษาสิทธิส่วนบุคคลของข้อมูล ข้อมูลรหัสพันธุกรรมเป็น

ตัวอย่างสำคัญที่ต้องจำเป็นต้องพิจารณามิติเหล่านี้ด้วย



รูปที่ 1.38 องค์ประกอบ 3 วี (Vs) ของข้อมูลขนาดใหญ่ในบริบทของข้อมูลทางชีวสารสนเทศ

รูปที่ 1.39 แสดงจำนวนเบสและลำดับเบสที่เพิ่มขึ้นในฐานข้อมูล GenBank (https://www.ncbi.nlm.nih.gov/genbank/) อย่างก้าวกระโดดในช่วง 25 ปีที่ผ่านมา โดยเฉพาะหลังการตีพิมพ์จีโนมมนุษย์เป็นครั้งแรกในปีค.ศ. 2001 รูปที่ 1.40 แสดงราคาต่อเบสและต่อจีโนมที่ลดลงอย่างมากในช่วง 10 ปีที่ผ่านมา ซึ่งอัตราที่ลดลงนี้ลดลงมากกว่ากฎของมัวร์ (Moore's law)

จีโนมิกส์บนคลาวด์

ด้วยข้อมูลจีโนมที่มีการเพิ่มขึ้นอย่างมากด้วยราคาที่ถูกลง บริษัทยักษ์ใหญ่ที่ให้บริการคลาวด์อย่างกูเกิ้ลและอเมซอนได้มีบริการกูเกิ้ลจีโนมิกส์ (Google Genomics) (https://cloud.google.com/genomics/) และ Genomics in the Cloud (https://aws.amazon.com/health/genomics/) ตามลำดับ โดยทั้งสองบริษัท มีไปปไลน์พื้นฐานพร้อมใช้สำหรับวิเคราะห์ข้อมูลจีโนม มีสำเนาฐานข้อมูลสาธารณะตัวอย่างเช่นข้อมูลจากโครงการ 1000 จีโนม ข้อมูลจีโนมอ้างอิง ข้อมูลจาก The Cancer Genome Atlas (TCGA) เป็นต้น ที่เกี่ยวข้องกับการวิเคราะห์ข้อมูลจีโนมที่สามารถเข้าถึงได้โดยสะดวก ในกรณีของกูเกิ้ลจีโนมิกส์ผู้ใช้สามารถวิเคราะห์การเกิดการแปรผันในรหัสพันธุกรรมโดยใช้ BigQuery เป็นต้น นอกจากกูเกิ้ลและอเมซอนซึ่งนักวิจัยสามารถใช้โครงสร้างพื้นฐานเพื่อการประมวลผลข้อมูลจีโนมมิกส์แล้ว แล้วก็ยังมีบริษัทอื่นๆอื่นอีก เช่น บริษัทอิลูมินาที่มีอิลูมินาเบสสเปซ (Illumina BaseSapce: https://basespace.illumina.com/home/index) ซึ่งเป็นคลาวด์ของบริษัทอิลูมินาโดยลูกค้าของบริษัทสามารถเข้าถึงรวมทั้งวิเคราะห์ข้อมูลที่ส่งไปทำการถอดรหัสที่บริษัทได้ นอกจากนี้ยังมีบริการจากบริษัท

SevenBridges ที่ได้รับเงินสนับสนุนจากสถาบันมะเร็งแห่งชาติ (National Cancer Institute: NCI) ภายใต้สถาบันสุขภาพแห่งชาติ (National Institutes of Health) หรือเอ็นไอเอช (NIH) ในการจัดเตรียมโครงสร้างพื้นฐานบนคลาวด์ชื่อ CGC หรือ Cancer Genomics Cloud [31] (http://www.cancergenomicscloud.org) เพื่อวิเคราะห์ข้อมูลเกี่ยวกับจีโนมิกส์ของโรคมะเร็งโดยมีข้อมูลหลักจาก TCGA (The Cancer Genome Atlas: https://cancergenome.nih.gov) ที่เอ็นไอเอช โดยทาง CGC มีการเตรียมไปป์ไลน์พื้นฐานเพื่อการวิเคราะห์ข้อมูลโดยผู้ใช้สามารถอัพโหลดข้อมูลเพิ่มเติมเพื่อการวิเคราะห์ร่วมกับข้อมูลจาก TCGA ได้ นอกจากนี้ยังมี Canadian Genomics Cloud (https://genomicscloud.ca) ที่เป็นแพลตฟอร์มสาธารณะเพื่อการวิเคราะห์ข้อมูลทั้งข้อมูลจีโนมิกส์และข้อมูลเชิงคลีนิกสำหรับนักวิทยาศาสตร์และนักวิจัยของประเทศแคนาดา เป็นต้น

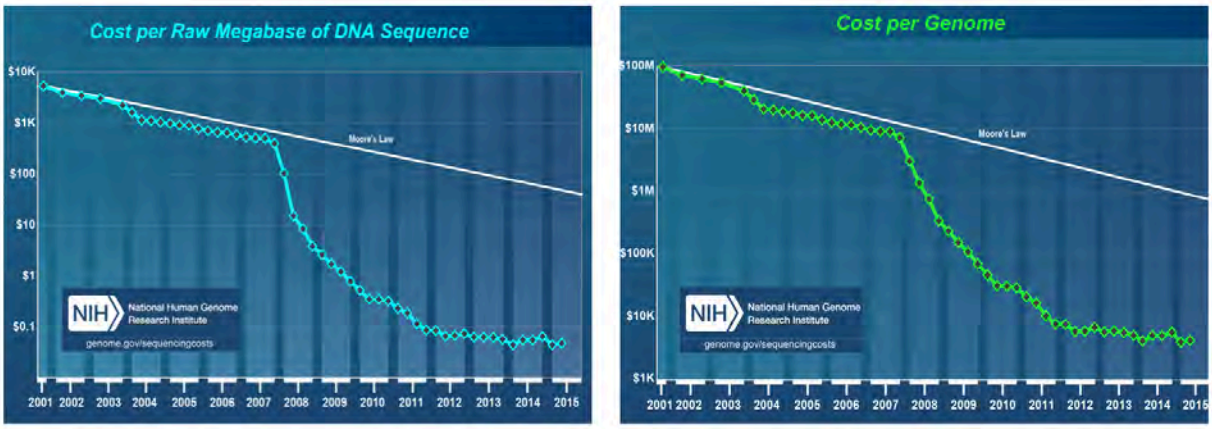
Growth of GenBank and WGS

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
211	Dec 2015	203939111071	189232925	1297865618365	317122157
212	Feb 2016	207018196067	190250235	1399865495608	333012760
213	Apr 2016	211423912047	193739511	1452207704949	338922537
214	Jun 2016	213200907819	194463572	1556175944648	350278081
215	Aug 2016	217971437647	196120831	1637224970324	359796497
216	Oct 2016	220731315250	197390691	1676238489250	363213315
217	Dec 2016	224973060433	198565475	1817189565845	395301176
221	Aug 2017	240343378258	203180606	2242294609510	499965722
222	Oct 2017	244914705468	203953682	2318156361999	508825331
223	Dec 2017	249722163594	206293625	2466098053327	551063065

รูปที่ 1.39 แสดงจำนวนเบสและลำดับเบสที่เพิ่มขึ้นในฐานข้อมูล GenBank

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



รูปที่ 1.40 แสดงค่าใช้จ่ายต่อเบสและต่อจีโนมที่ลดลงเป็นอย่างมากตั้งแต่ปีค.ศ. 2007

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

ตัวอย่างฐานข้อมูลสาธารณะ

NCBI

เอ็นซีบีไอ (NCBI: National Center for Biotechnology Information) (https://www.ncbi.nlm.nih.gov) ถูกตั้งขึ้นเมื่อวันที่ 4 พฤศจิกายน ค.ศ. 1988 (พ.ศ. 2531) โดยเป็นส่วนหนึ่งของหอสมุดแพทย์แห่งชาติอเมริกัน (National Library of Medicine) หรือเอ็นแอลเอ็ม (NLM) ภายใต้เอ็นไอเอช (NIH) ทั้งนี้เพื่อเป็นแหล่งข้อมูลทางสารสนเทศเพื่อสนับสนุนงานวิจัยและพัฒนาทางด้านการแพทย์และเทคโนโลยีชีวภาพ เอ็นซีบีไอเป็นฐานข้อมูลสาธารณะขนาดใหญ่ ที่ถูกใช้อ้างอิงเป็นระดับต้นๆของโลก ประกอบด้วยฐานข้อมูลจำเพาะที่สำคัญหลายฐาน ตัวอย่างเช่น ฐานข้อมูลนิวคลีโอไทด์ (Nucleotide) ที่เก็บลำดับเบสของสายดีเอ็นเอ ฐานข้อมูลโปรตีน (Protein) เก็บลำดับกรดอะมิโนในส่วนของโปรตีน ฐานข้อมูลจีโนม (Genome) ที่เก็บลำดับเบสดีเอ็นเอของจีโนมสิ่งมีชีวิตต่างๆ ฐานข้อมูลการแสดงออกของยีน (GEO DataSets, GEO profiles) ฐานข้อมูลความแปรผันในลำดับเบสเดี่ยวๆ (dbSNP) ฐานข้อมูลความแปรผันในกลุ่มเบส (dbVar) เช่น การเกิดส่วนของดีเอ็นเอซุดซ้ำ มีการเพิ่มกลุ่มของเบสหรือกลุ่มของเบสของสายดีเอ็นเอหายไปบางส่วน หรือเกิดการกลับด้านของสายดีเอ็นเอ เป็นต้น โดยข้อมูลในฐานข้อมูลเหล่านี้มาจากข้อมูล ผลงานวิจัยที่ได้รับการตีพิมพ์ในวารสารต่างๆ ซึ่งถูกจัดเก็บในฐานข้อมูลผับเมด (PubMed) ซึ่งมีมากกว่า 27 ล้านรายการ (เข้าถึงออนไลน์เมื่อวันที่ 30 กันยายน พ.ศ. 2560) โดยผับเมดก็เป็นฐานข้อมูลหลักที่สำคัญภายใต้เอ็นซีบีไอ เช่นกัน

UniProt

ยูนิพรอต (UniProt) (https://www.uniprot.org) เป็นฐานข้อมูลที่เน้นการเก็บข้อมูลโปรตีนที่มีคุณภาพทั้งลำดับกรดอะมิโนและฟังก์ชันของสายโปรตีนของสิ่งมีชีวิตต่างๆ โดยเป็นฐานข้อมูลอ้างอิงหลักเกี่ยวกับข้อมูลโปรตีนอีก

แหล่งนอกเหนือจากฐานข้อมูลโปรตีนที่เอ็นซีบีไอ ภายในฐานข้อมูลได้มีการแบ่งข้อมูลออกเป็นสองส่วนหลักๆ 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 มีข้อมูลโปรตีน 90,050,711 รายการ ซึ่งมีปริมาณข้อมูลมากกว่า Swiss-Prot มากแต่ข้อมูลส่วนใหญ่ยังไม่ผ่านการตรวจสอบด้วยมือและการทวนสอบ

สารานุกรมขององค์ประกอบดีเอ็นเอ

สารานุกรมขององค์ประกอบดีเอ็นเอ (ENCODE: Encyclopedia of DNA Elements) ถูกสร้างขึ้นโดยความร่วมมือของกลุ่มวิจัยต่างๆในรูปแบบของสมาคม (consortium) โดยได้รับเงินทุนสนับสนุนจาก National Human Genome Research Institute (NHGRI) โดยเป้าหมายหลักของโครงการคือการสร้างองค์ความรู้เชิงลึกเกี่ยวกับฟังก์ชันขององค์ประกอบต่างๆ ในจีโนมของมนุษย์ และรวมไปถึงองค์ประกอบที่มีการทำงานในระดับอาร์เอ็นเอและโปรตีน องค์ประกอบที่เกี่ยวกับการควบคุมการแสดงออกของยีน เป็นต้น โดยข้อมูลจากการทดลองต่างๆได้ถูกรวบรวมไว้และได้เปิดให้เข้าถึงโดยสาธารณะที่
https://www.encodeproject.org

ตัวอย่างฐานข้อมูลเปิดอื่นๆ

นอกจากตัวอย่างของฐานข้อมูลหลักข้างต้น ยังมีฐานข้อมูลเปิดอีกมากมายซึ่งมีลักษณะข้อมูลแตกต่างกันไป เช่น ฐานข้อมูลที่จำเพาะกับสิ่งมีชีวิตหนึ่ง เช่น ฐานข้อมูลจีโนมมนุษย์ที่ University of California, Santa Cruz (UCSC) (https://genome.ucsc.edu) ฐานข้อมูล GENCODE (https://www.gencodegenes.org) ที่เก็บข้อมูลลำดับเบสลำดับกรดอะมิโน รวมทั้งคำอธิบายประกอบ (annotation) ฟังก์ชันของยีน โปรตีนที่มีข้อมูลของมนุษย์และหนูเม้าส์ ฐานข้อมูล TAIR (https://www.arabidopsis.org) ซึ่งเก็บข้อมูลต่างๆ ทั้งรหัสพันธุกรรมและฟังก์ชันของยีนของพืชใบเลี้ยงคู่ *Arabidopsis thaliana* ที่ถูกเลือกมาเป็นต้นแบบในการถอดรหัสพันธุกรรม ฐานข้อมูลโปรตีนดาต้าแบงค์ (RCDB Protein Data Bank: RCSB PDB) (https://www.rcsb.org) ที่เก็บโครงสร้าง 3 มิติของโปรตีนที่มีการทดลองจากห้องปฏิบัติการ ฐานข้อมูลโปรตีนแฟมิลีหรือพีแฟม (Pfam: http://pfam.xfam.org) ที่มีการจัดกลุ่มของโปรตีนตามการปรากฏของโปรตีนโดเมน (protein domain) หรือชุดของโปรตีนโดเมนที่ปรากฏร่วมกัน โดยแต่ละโปรตีนโดเมนจะเป็นส่วนของสายโปรตีนที่มักมีฟังก์ชันการทำงานจำเพาะ ฐานข้อมูลอาร์เอ็นเอแฟมิลีหรืออาร์แฟม (http://rfam.xfam.org) ที่มีการจุดกลุ่มของอาร์เอ็นเอตามการปรากฏร่วมกันของโครงสร้างหรือส่วนของโครงสร้าง 2 มิติ (secondary structure) ที่มักมีผลต่อฟังก์ชันการทำงานเฉพาะ ฐานข้อมูล KEGG Pathway (https://www.genome.jp/kegg/pathway.html) เป็นแหล่งรวบรวมข้อมูลพาธ์เวย์หรือชีววิถีของสิ่งมีชีวิตต่างๆ เป็นต้น

แบบฝึกหัดบทที่ 1

1. ให้เขียนโปรแกรมเพื่อแก้ปัญหาโจทย์ที่โรซาลินด์ (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) Protein Translation (http://rosalind.info/problems/ptra/)
   7) Introduction to Protein Databases (http://rosalind.info/problems/dbpr/)
   8) Complementing a Strand of DNA (http://rosalind.info/problems/rvco/)
2. จากข้อ 1.6 ของโรซาลินด์ข้างต้น จงเขียนฟังก์ชัน translate() เองโดยใช้ตารางโคดอนในรูปที่ 1.5
3. ไฟล์ในรูปแบบฟาสต้า (FASTA) มีลักษณะอย่างไร จงอธิบาย
4. ไฟล์ในรูปแบบฟาสคิว (FASTQ) มีลักษณะอย่างไร จงอธิบาย
5. จงยกตัวอย่างฐานข้อมูลสาธารณะ พร้อมตัวอย่างข้อมูลที่อยู่ในฐานข้อมูลเหล่านั้น

ภาคผนวกบทที่ 1

รูปแบบไฟล์ที่เก็บรหัสพันธุกรรมจากเครื่องถอดรหัสพันธุกรรม

ข้อมูลทางชีววิทยาที่เกิดจากเทคโนโยลีโอมิกส์ต่างๆ รวมทั้งผลของการวิเคราะห์ข้อมูลเบื้องต้นมักอยู่ในรูปแบบของเท็กซ์ไฟล์ (Text file) ที่มีโครงสร้างจำเพาะ สำหรับข้อมูลลำดับเบสของนิวคลีโอไทด์ที่ได้อ่านได้จากเครื่องถอดรหัสพันธุกรรมจะอยู่ในรูปแบบ FASTQ และเมื่อทำการประกอบร่างจีโนมได้ผลลัพธ์เป็นดีเอ็นสายยาวแล้วมักจะถูกบันทึกในรูปแบบ FASTA

FASTQ

เป็นไฟล์เก็บลำดับเบสของสายดีเอ็นเอที่ได้จากการถอดรหัสจีโนมโดยความยาวของสายดีเอ็นเอแต่ละเส้นขึ้นอยู่กับเทคโนโยลีที่ใช้ในการถอดรหัส ตัวอย่างเช่น ถ้าใช้ Illumina HiSeq ก็จะมีความยาวของสายดีเอ็นเอตั้งแต่ 50 ถึง 250 เบส ขึ้นอยู่กับชุดคิท (Kit) ในห้องปฏิบัติการที่ใช้ในการเตรียมข้อมูลก่อนการถอดรหัส รูปที่ 1.41 แสดงตัวอย่างโครงสร้างข้อมูลในไฟล์ FASTQ โดยดีเอ็นเอแต่ละเส้นจะใช้ 4 บรรทัดในการแสดงข้อมูล โดยบรรทัดที่ 1 ที่แสดงรหัสสายดีเอ็นเอจะขึ้นต้นด้วยเครื่องหมายแอด (@) เสมอ และบรรทัดที่สามที่ออกแบบไว้ให้เพิ่มเติมข้อมูลได้ในอนาคตจะขึ้นต้นด้วยเครื่องหมายบวก (+) เสมอ

บรรทัดที่ 1 @รหัสสายดีเอ็นเอ และข้อมูลอื่นๆ เกี่ยวกับดีเอ็นเอนั้น
บรรทัดที่ 2 ลำดับเบสของสายดีเอ็นเอที่ถูกถอดรหัสออกมา
บรรทัดที่ 3 + อาจมีข้อมูลเพิ่มเติมหรือเป็นปล่อยว่าง เตรียมไว้สำหรับอนาคต
บรรทัดที่ 4 อักขระแสดงคุณภาพของแต่ละเบสในสายดีเอ็นเอในบรรทัดที่ 2

รูปที่ 1.41 โครงสร้างข้อมูลในไฟล์ FASTQ

ตัวอย่างข้อมูลในรูปที่ 1.42 ชื่อของสายดีเอ็นเอคือ `HWI-ST797:281:D198UACXX:5:1101:1945:2049 1:N:0:GGCTAC`ซึ่งมีลำดับเบสที่ถอดรหัสได้เป็น `NTTATCCTCCACACAATTCCTTTCACTTTAGACAAAGAGATTTGTATTGCTCAGAAGCAGAGAATCTAGGTTTCTGTGGAATCTATTGGAGTTAGAAGGTA` โดยแต่ละตำแหน่งมีอักขระที่เป็นไปได้ 4 ตัว คือ A, T, C, และ G แสดงลำดับนิวคลีโอไทด์ อะดีนีน (Adenine) ไทมีน (Thymine) ไซโตซีน (Cytocine) และ กวานีน (Guanine) ตามลำดับ สำหรับอักษร N แสดงถึงความไม่แน่ใจของเครื่องถอดรหัสว่าเป็นนิวคลีโอไทด์ใด และคุณภาพของแต่ละลำดับเบสที่ถอดรหัสออก มาได้นี้คือ `#1:DDDDDHHHFHJJGJIEHHEHHGGHJGHIGHHIIJJJFHIJJHGIAFFGIJIIII@BFBFG@CCHGJIDCGIIDCAEHHDEFFBEEE>;ACCC@ACB;;` เรียงตามลำดับนิวคลีโอไทด์เบสต่อเบส หมายเหตุบรรทัดที่ 2 และ 4 ในตัวอย่างนี้ได้ทำเป็น*ตัวอักษรเอียง*เพื่อให้เห็นความแตกต่างระหว่างบรรทัดชัดเจนขึ้น

```
@HWI-ST797:281:D198UACXX:5:1101:1945:2049 1:N:0:GGCTAC
NTTATCCTCCACACAATTCCTTTCACTTTAGACAAAGAGATTTGTATTGCTCAGAAGCAGAGAATCTAGG
TTTCTGTGGAATCTATTGGAGTTAGAAGGTA
+
#1:DDDDDHHHFHJJGJIEHHEHHGGHJGHIGHHIIJJJFHIJJHGIAFFGIJIIII@BFBFG@CCHGJI
DCGIIDCAEHHDEFFBEEE>;ACCC@ACB;;
```

รูปที่ 1.42 ตัวอย่างข้อมูลในไฟล์ FASTQ

สำหรับอักขระแสดงคุณภาพในบรรทัดที่ 4 ตัวอักขระ “!” แสดงคุณภาพน้อยสุด และอักขระ “~” แสดงคุณภาพสูงสุดของเบสหนึ่งๆที่ถูกถอดรหัสออกมา ภาพต่อไปนี้เรียงลำดับของอักขระแสดงคุณภาพจากน้อยสุดไปมากสุด โดยอักขระเหล่านี้สามารถเชื่อมโยงไปยัง Phred quality score หรือ q score (คำศัพท์ถัดไป) ตามรูปที่ 1.43

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

ที่มา: https://en.wikipedia.org/wiki/FASTQ_format#Quality

Phred quality score

อีวิงก์ (Ewing) แลกรีน [32] ได้พัฒนาอัลกอริทึมที่ใช้ในการอ่านลำดับเบสโดยอัตโนมัติจากเครื่องถอดรหัสพันธุกรรมโดยมีการให้คะแนนกับแต่ละเบสที่อ่านออกตามค่า q ในสมการลอการิทึมต่อไปนี้

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

โดยที่ p คือค่าประมาณของความน่าจะเป็นที่อาจจะเกิดความผิดพลาดในการอ่านเบสนั้นๆ ตัวอย่างเช่นถ้าเบสนั้นมีค่า 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 นี้ยังถูกใช้เป็นมาตรวัดอย่างแพร่หลายในการประเมินความถูกต้องของแพลตฟอร์มที่ใช้ในการถอดรหัสพันธุกรรม เช่นอาจเทียบว่าสองแพลตฟอร์มมี % ของเบสที่มีค่า q score >= 30 ของรีด 1 และ รีด 2 (ในกรณีของ paired-end reads) เท่าไหร่และมีค่าเฉลี่ยของทั้งสองรีดรวมกันเท่าไหร่ 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 ค่าคุณภาพของแต่ละเบสโดยเทียบกับค่า Phred quality score

(ที่มา: https://www.drive5.com/usearch/manual/quality_score.html)

FASTA

ฟาสต้า (FASTA) เป็นไฟล์ที่ใช้เก็บลำดับเบสของสายดีเอ็นเอ อาร์เอ็นเอ หรือลำดับของกรดอะมิโนในสายของโปรตีน โดยแต่ละสายของดีเอ็นเอ อาร์เอ็นเอ หรือโปรตีน จะใช้ 2 บรรทัดในการแสดงข้อมูลดังรูปที่ 1.44 โดยบรรทัดที่แรกจะขึ้นต้นด้วยเครื่องหมายมากกว่า (>) เสมอ และสิ่งที่ตามมาคือรหัสของลำดับสายดีเอ็นเอ อาร์เอ็นเอ หรือโปรตีน ส่วนบรรทัดที่สองจะเป็นลำดับเบสในดีเอ็นเอ อาร์เอ็นเอ หรือลำดับกรดอะมิโนของสายโปรตีน

บรรทัดที่ 1 >รหัสสายดีเอ็นเอ
บรรทัดที่ 2 ลำดับเบสของสายดีเอ็นเอ อาร์เอ็นเอ หรือลำดับกรดอะมิโนในสายโปรตีน

รูปที่ 1.44 โครงสร้างข้อมูลในไฟล์ฟาสต้า

รูปที่ 1.45 แสดงตัวอย่างข้อมูลในไฟล์ฟาสต้าโดย `ENA|U64805|U64805.1 Homo sapiens Brca1-delta11b (Brca1) mRNA, complete cds.` แสดงชื่อของสายอาร์เอ็นเอโดยมีรายละเอียด

ว่าเป็นของมนุษย์ (Homo sapiens) และเป็นลำดับอาร์เอ็นเอประเภทเมสเซ็นเจอร์อาร์เอ็นเอ (messenger RNA: mRNA) ที่สมบูรณ์สามารถถูกแปลรหัสต่อไปเป็นโปรตีน (รูปที่ 1.46) สำหรับลำดับเบส **ATGGATTTATCT GCTCT…** ในบรรทัดถัดๆมาถือเป็นบรรทัดที่สองที่เก็บลำดับเบสอาร์เอ็นเอตามรหัสชื่อที่ระบุในบรรทัดแรก

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

รูปที่ 1.45 ตัวอย่างข้อมูลในไฟล์ฟาสต้า โดยเป็นลำดับเบสของสายอาร์เอ็นเอ

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

รูปที่ 1.46 ตัวอย่างข้อมูลในไฟล์ฟาสต้า โดยเป็นลำดับกรดอะมิโนของสายโปรตีน

บทที่ 2 การประกอบร่างจีโนมแบบไม่มีจีโนมอ้างอิง (*De novo* genome assembly)

วัตถุประสงค์

- เพื่อให้นิสิตเข้าใจกระบวนการและเทคโนโลยีที่เกี่ยวข้องกับการถอดรหัสจีโนม
- เพื่อให้นิสิตคุ้นเคยกับตัวอย่างข้อมูลตั้งต้นที่ได้จากการถอดรหัสจีโนมและเข้าใจการทำงานของอัลกอริทึมพื้นฐานที่ใช้ในการประกอบร่างจีโนม
- เพื่อให้นิสิตได้เห็นตัวอย่างงานวิจัยและผลงานวิจัย รวมทั้งตัวอย่างโปรแกรมที่ใช้ในการประกอบร่างจีโนม
- เพื่อให้นิสิตได้เห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทาย รวมทั้งงานวิจัยอื่นๆ ที่เกี่ยวข้อง

ผลลัพธ์ที่คาดหวัง

- นิสิตสามารถอธิบายความแตกต่างรวมทั้งข้อดีข้อเสียระหว่างแพลตฟอร์มที่ใช้ในการถอดรหัสจีโนมได้
- นิสิตเข้าใจคุณลักษณะของข้อมูลตั้งต้นที่ได้จากการถอดรหัสจีโนม
- นิสิตสามารถอธิบายการทำงานของอัลกอริทึมหลักๆ ที่ถูกใช้ในการประกอบร่างจีโนมได้
- นิสิตสามารถเขียนโปรแกรมที่ใช้ในการประกอบร่างจีโนมอย่างง่ายได้
- นิสิตสามารถยกตัวอย่างโปรแกรมประกอบร่างจีโนมที่มีการใช้งานกันอย่างแพร่หลายได้
- นิสิตสามารถยกตัวอย่างความท้าทายที่ยังมีอยู่ในการประกอบร่างจีโนมและสามารถยกตัวอย่างความท้าทายที่ยังมีอยู่และสามารถนำเสนอแนวทางในการพัฒนาวิธีการแก้ปัญหาเหล่านี้ได้ รวมทั้งสามารถประยุกต์องค์ความรู้จากบทเรียนเพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้

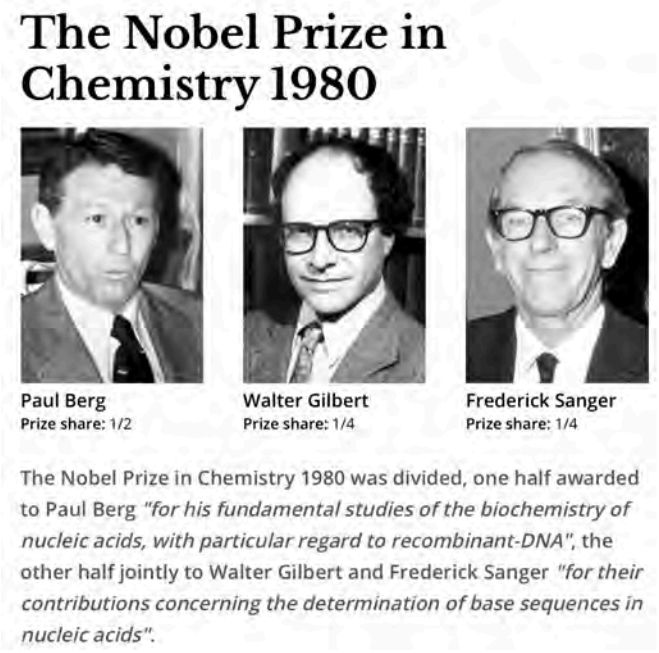
เนื้อหาโดยสรุป

เทคโนโลยีการถอดรหัสพันธุกรรมโดยใช้เทคโนโลยี Next Generation Sequencing (NGS) ในปัจจุบัน ลักษณะของข้อมูลและปริมาณข้อมูลที่ได้จากการถอดรหัสจีโนม โจทย์ทางชีวสารสนเทศ การประกอบร่างจีโนมแบบไม่มีจีโนมอ้างอิง (*de novo* genome assembly) โดยการพยายามนำดีเอ็นเอสั้นยาวประมาณ 100-150 เบสจำนวนมากมายที่ได้มาจากเทคโนโลยีการถอดรหัสมาเชื่อมต่อกันให้เป็นสายดีเอ็นเอที่ยาวขึ้นจนเป็นสายโครโมโซมที่ถูกต้อง ตัวอย่างอัลกอริทึมและโครงสร้างข้อมูลที่เกี่ยวข้องกับการประกอบร่างจีโนม เช่น ปัญหาการสร้างสาย

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

บทที่ 2 การประกอบร่างจีโนมแบบไม่มีจีโนมอ้างอิง (*De novo* genome assembly)

การได้มาซึ่งรหัสพันธุกรรมในระดับจีโนมเป็นจุดเริ่มต้นในการทำความเข้าใจกระบวนการต่างๆ ในทางชีววิทยาและอณูวิทยาของสิ่งมีชีวิต ในปีค.ศ. 1977 วอลเตอร์ กิลเบิร์ต (Walter Gilbert) และ เฟดเดริค แซงเกอร์ (Frederick Sanger) ต่างพัฒนาวิธีในการถอดรหัสพันธุกรรมดีเอ็นเอ และในปีค.ศ. 1980 ทั้งสองคนได้รับรางวัลโนเบลในสาขาเคมีร่วมกัน (รูปที่ 2.1)



รูปที่ 2.1 วอลเตอร์ กิลเบิร์ต (Walter Gilbert) และ เฟดเดริค แซงเกอร์ (Frederick Sanger) ได้รับรางวัลโนเบลในปี ค.ศ. 1980 ในสาขาเคมี ในเรื่องการหาลำดับเบสในสายของกรดนิวคลีอิค

อย่างไรก็ตามนักวิทยาศาสตร์ยังใช้เวลาอีกร่วม 20 ปี กับงบประมาณอีกกว่า 3 พันล้านยูเอสดอลล่าร์ ในการถอดรหัสจีโนมมนุษย์จนได้โครงร่างแรกในปีค.ศ. 2001 [2] โดยระหว่างทาง ในปีค.ศ. 1990 ฟรานซิส คอลลิน (Francis Collins) เป็นผู้นำโครงการถอดรหัสจีโนมมนุษย์ (Human Genome Project) ซึ่งเป็นโครงการเปิดต่อสาธารณะโดยมีเป้าหมายว่าจะสามารถถอดรหัสจีโนมมนุษย์ได้สำเร็จในปีค.ศ. 2005 และในปีค.ศ. 1997 เครก เวนเตอร์ (Craig Venter) ได้ก่อตั้งบริษัทเอกชนภายใต้ชื่อ Celera Genomics โดยมีเป้าหมายเดียวกัน

หลังประสบความสำเร็จในการถอดรหัสจีโนมมนุษย์ในปีค.ศ. 2001 จีโนมของสิ่งมีชีวิตอื่นๆ ในกลุ่มยูแคริโอตได้ถูกถอดรหัสออกมาอย่างต่อเนื่อง ทั้งจีโนมหนูเม้าส์ตัวเล็ก *Mus musculus* [33] และหนูแรทตัวโต *Rattus norvegicus* [34] สุนัข (*Canis lupus familiaris*) [35] ชิมแปนซี (*Pan troglodytes*) [36] ลิงวอก (*Macaca*

mulatta) [37] ม้า (*Equus caballus*) [38] โอพอสซัม (*Didelphimorphia*) [39] วัว (*Bos taurus*) [40] โดยในสมัยแรกนั้นการถอดรหัสจีโนมใช้เทคโนโลยีแซงเกอร์ซึ่งพบว่ามีข้อจำกัดจึงได้มีการพัฒนาวิธีการถอดรหัสพันธุกรรมที่ใช้เวลาน้อยกว่า มีความถูกต้อง และมีราคาถูกกว่ามาก จึงได้มีจีโนมของสิ่งมีชีวิตอื่นๆออกมาอีกมากมายเช่น จีโนมหมีแพนด้า (*Ailuropoda melanoleura*) [41] จีโนมงูหลามพม่า (*Python molurus bivittatus*) [42] นอกจากการถอดรหัสจีโนมมนุษย์และสัตว์หลากหลายชนิดแล้ว ยังมีการถอดรหัสจีโนมพืชมากมายเช่น ข้าวสายพันธุ์จาโปนิกา (*Oryza sativa L. ssp. japonica*) [43] และสายพันธุ์อินดิกา (*Oryza sativa L. ssp. indica*) ยางพารา (*Hevea brasiliensis*) [44, 45] ปาล์มน้ำมัน (*Elaeis guineensis*) [46] และทุเรียน (*Durio zibethinus*) [47] เป็นต้น องค์ความรู้จากการวิเคราะห์จีโนมของสิ่งมีชีวิตเหล่านี้สามารถนำไปประยุกต์ใช้ในการวินิจฉัยและวิจัยเพิ่มเติมทางการแพทย์ การปรับปรุงพันธุ์พืช และเทคโนโลยีชีวภาพ เป็นต้น

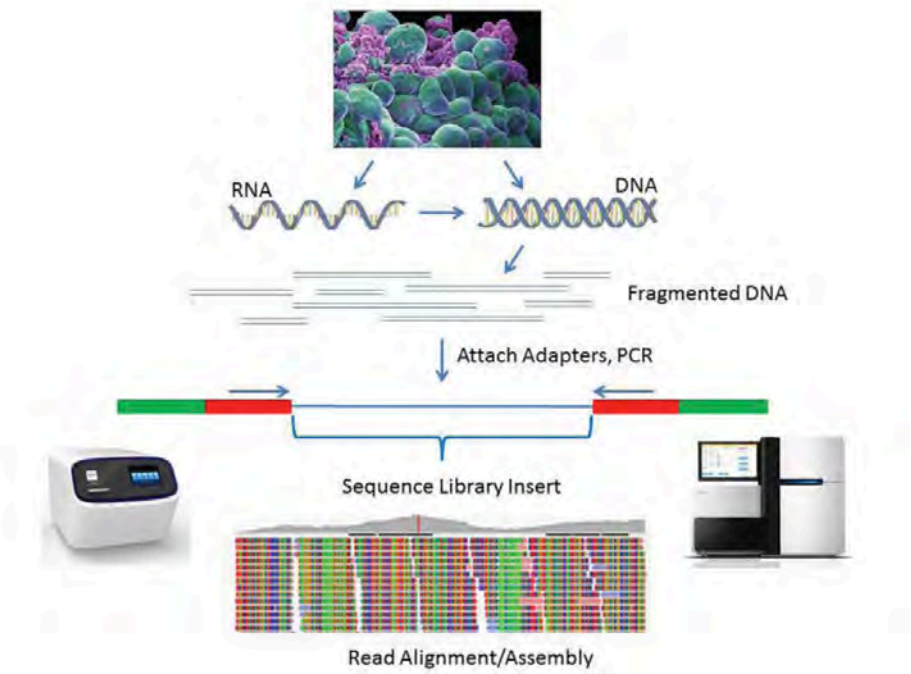
ความก้าวหน้าของเทคโนโลยีการถอดรหัสพันธุกรรม

ช่วงปลายทศวรรษของคริสต์ศตวรรษ ที่ 2000 ตลาดของเทคโนโลยีและเครื่องมือถอดรหัสพันธุกรรมมีการขยายตัวอย่างมาก บริษัทอิลูมินา (Illumina) สามารถลดราคาของจีโนมมนุษย์ 1 คนจาก 3 พันล้านยูเอสดอลล่าร์เหลือประมาณ 1 หมื่นดอลล่าร์ โดยใช้เทคโนโลยี Next Generation Sequencing หรือ NGS ในขณะที่บริษัทคอมพลีต จีโนมิกส์ (Complete Genomics) ก่อตั้งโรงงานจีโนมิกส์ในซิลิคอล วัลลีย์ (Silicon Valley) โดยให้บริการถอดรหัสจีโนมหลายร้อยจีโนมต่อเดือน สถาบันจีโนมของปักกิ่ง (Beijing Genome Institute: BGI) ได้สั่งเครื่องถอดรหัสพันธุกรรมมาใช้ในสถาบันหลายร้อยเครื่องและกลายมาเป็นศูนย์ถอดรหัสพันธุกรรมที่ใหญ่ที่สุดในโลก ปี ค.ศ. 2010 มีโครงการถอดรหัสจีโนมของสัตว์มีกระดูกสันหลัง 10,000 ชนิด ปีค.ศ. 2015 ประเทศอังกฤษได้เริ่มโครงการถอดรหัสจีโนมของชาวอังกฤษ 100,000 คนผ่านโครงการจีโนมิกส์อิงแลนด์ (Genomics England) [48] และประเทศกาต้าร์เริ่มโครงการถอดรหัสจีโนมของชาวกาต้าร์ 10,000 คนผ่านโครงการ (Qatar Genome Program) (https://qatargenome.org.qa) เป็นต้น ในขณะที่บริษัทอิลูมินามีเป้าหมายจะพยายามลดราคาของการถอดรหัสจีโนมลงให้เหลือ 100 ยูเอสดอลล่าร์ต่อคน ทั้งนี้ข้อมูลรหัสพันธุกรรมที่ได้จากเทคโนโลยี NGS นั้นจะเป็นข้อมูลสายสั้นๆ คำถามคือข้อมูลเหล่านี้จะนำมาต่อกันให้เป็นลำดับเบสของโครโมโซมต่างๆ ที่ประกอบกันเป็นจีโนมได้อย่างไร

การเตรียมดีเอ็นเอเพื่อการถอดรหัสพันธุกรรม

ขั้นตอนพื้นฐานในการถอดรหัสพันธกรรม [49] (รูปที่ 2.2) ประกอบด้วยการสกัดดีเอ็นเอจากเนื้อเยื่อหรือเซลล์ตัวอย่าง (ในกรณีการถอดรหัสพันธุกรรมอาร์เอ็นเอ อาร์เอ็นเอจะถูกแปลงให้เป็น cDNA (complementary DNA) หรือดีเอ็นเอต้นแบบที่ถูกสังเคราะห์ย้อนกลับจากเมสเซ็นเจอร์อาร์เอ็นโดยเอ็นไซม์ชื่อรีเวิร์สทรานสคริปเตส (reverse transcriptase) จากนั้นดีเอ็นเอจะถูกทำให้เป็นเส้นสั้นๆ และทำให้เป็นไลบรารีโดยการนำดีเอ็นเอสาย

สั้นเหล่านี้ไปเชื่อมต่อกับอะแดปเตอร์โดยวิธีการไลเกชัน (ligation) ซึ่งอะแดปเตอร์เหล่านี้ถูกออกมาให้มีรูปแบบลำดับเบสที่มีความจำเพาะกับแพลตฟอร์มที่ใช้ในการถอดรหัสเช่น เป็นอะแดปเตอร์ที่สามารถเชื่อมต่อได้กับโฟเซลล์ (flow-cell) ของแพลตฟอร์มอิลูมินา (Illumina) หรือบีดส์ (beads) ของแพลตฟอร์ม Ion Torrent เป็นต้น หลังจากต่อกับอะแดปเตอร์แล้วก็จะเป็นการเพิ่มจำนวนดีเอ็นเอในไลบรารีซึ่งมีวิธีการแตกต่างกันไปตามแพลตฟอร์ม เช่นกัน และนำไปทำการอ่านรหัสตามวิธีการจำเพาะของแต่ละแพลตฟอร์มต่อไป



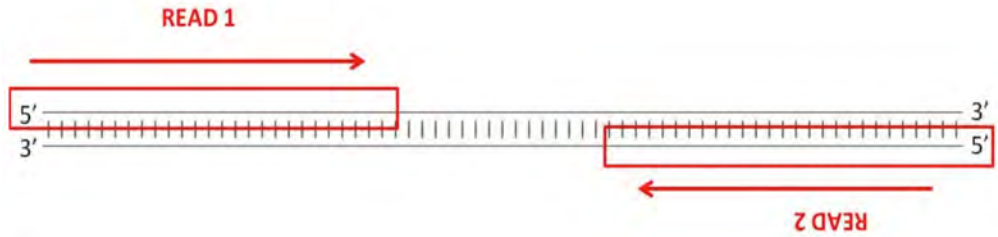
รูปที่ 2.2 ขั้นตอนพื้นฐานการเตรียมไลบรารีของดีเอ็นเอสายย่อยเพื่อถอดรหัสพันธุกรรมในเทคโนโลยีเอ็นจีเอส (NGS: Next Generation Sequencing)

(ที่มา: รูปที่ 1 ของ [49])

เทคโนโลยีในการถอดรหัสพันธุกรรม

หลังความสำเร็จในการถอดรหัสพันธุกรรมมนุษย์เป็นครั้งแรกในปีค.ศ. 2001 [2] เทคโนโลยีที่เรียกกันโดยรวมๆ ว่าเทคโนโลยีเอ็นจีเอส Next-Generation Sequencing (NGS) ที่ใช้ในการถอดรหัสพันธุกรรมในระดับจีโนมได้มีการพัฒนาอย่างก้าวกระโดด บางเทคโนโลยีพยายามเพิ่มจำนวนของเบสที่สามารถอ่านได้ต่อหนึ่งหน่วยเวลาในขณะที่บางเทคโนโลยีเน้นความยาวของสายลำดับเบสที่สามารถถอดรหัสได้ บางเทคโนโลยีถอดรหัสพันธุกรรมเป็นสายเดี่ยว (single-end sequencing) บางเทคโนโลยีถอดรหัสพันธุกรรมเป็นสายคู่ (paired-end sequencing) โดยการถอดรหัสพันธุกรรมเป็นสายเดี่ยวดีเอ็นเอเกลียวคู่จะถูกถอดรหัสออกมาเป็นสายรหัสพันธุกรรมจำนวนมากในทิศทางเดียวคือทิศทางจาก 5’ ไป 3’ ในขณะที่การถอดรหัสพันธุกรรมเป็นสายคู่ ดีเอ็นเอจะถูกถอดรหัสจากทั้งสองทิศทางคือจากทั้งฟอร์เวิร์ดสแตรนด์และรีเวิร์สสแตรนด์ (รูปที่ 2.3)

ในกรณีที่ใช้แพลตฟอร์มอิลูมินาข้อมูลรหัสพันธุกรรมของดีเอ็นแบบสายคู่ (paired-end) จะอยู่ในรูปแบบไฟล์ฟาสคิว (FASTQ) โดยแยกเป็น 2 ไฟล์ คือ [ชื่อไฟล์]_1.fastq และ [ชื่อไฟล์]_2.fastq โดยไฟล์ที่ลงท้ายด้วย _1.fastq เก็บข้อมูลสายรหัสพันธุกรรมดีเอ็นเอหรือรีดที่อ่านได้จากฟอร์เวิร์ดสแตรนด์ (READ 1 ในรูปที่ 2.3) และไฟล์ที่ลงท้ายด้วย _2.fastq เก็บข้อมูลสายรหัสพันธุกรรมดีเอ็นเอหรือรีดที่อ่านได้จากรีเวิร์สสแตรนด์ (READ 2 ในรูปที่ 2.3) ดังตัวอย่างลำดับเบสจริงในรูปที่ 2.4



รูปที่ 2.3 การถอดรหัสดีเอ็นเอในกรณีของ paired-end sequencing

(ที่มา: http://www.cureffi.org/2012/12/19/forward-and-reverse-reads-in-paired-end-sequencing/)

sample_1.fastq

```
@NB501835:10:HHJN7BGX3:1:11101:8426:1042 1:N:0:9
TTTCCNTTAGGAAGTAGAACTCCTCATTACCCTAATTANATCAGAAAAAGGAAGCCTGGGTTTTACAGTAACCAA
+
AAAAA#EEEEEEEEEEEEEEEEEEEEEEEEEEEEE#E6EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
@NB501835:10:HHJN7BGX3:1:11101:3305:1042 1:N:0:9
TTGTGNGTTAGATACTATTATCTTCATCTTCCAGATGGNGAAACAGAGGCTCAGTGAAGTTAAATAATCTGCCTC
+
AAAAA#/EEEEEEEEAEAEEEEAEEEEEE/EEEEE/AA#EEEEEEEEE/EEEEEAEEEE/<EEEEEEEEEEEEEE
@NB501835:10:HHJN7BGX3:1:11101:8130:1043 1:N:0:9
AGGCGNTGGCTCACGCCTGTAATCCCAACACTTTGGGAGGCCAGACGGGCGGATCACGAGGTCAGGGGATCAAGAC
+
A/A/A#AAEAEEAEEEAA<EEEEEEEEEAE/AEEEEEEEEEEEEEEEEEEEAEEEEE<EAEEEEAEEEAAEEAEAA
```

sample_2.fastq

```
@NB501835:10:HHJN7BGX3:1:11101:8426:1042 2:N:0:9
CTCACCTTTATGAGCCGGTCCCCAGGTTTTNGCNTTCCATNNCTNTTGGCTGNANNCNNTATNACATNNNGANC
+
AAAAAEEEEEEEEEEEEEEEEEEEEEEEEE#EE#EEEEEE##EE#EEEEEEE#E##E##EEE#EEEE###EE#A
@NB501835:10:HHJN7BGX3:1:11101:3305:1042 2:N:0:9
CTTCATTTGAGGCAGATTATTTAACTTCACNGANCCTCTGNNTCNCCATCTGNANNANNAAGNTAATNNNATNT
+
AA/A66/EEEEEE6EE/EAEEA/EE/EE/E#EE#EEEEEE##EE#EEAAEEE#A##E##EEE#EEE/###/E#E
@NB501835:10:HHJN7BGX3:1:11101:8130:1043 2:N:0:9
GCTAATTTTTTGTATTTTTAGTAGAGACGGNGTNTCACCANGTTNGCCAGGANGNTCNNGATNCCCTNNNCTNGTN
+
AAAAAEEEEEEEE/E/EEAEEEEEEEEEAE#EE#EEEAEE#AEE#AEEEAEE#/#EE##E/E#//<E###<E#EE#
```

รูปที่ 2.4 ตัวอย่างลำดับเบสจริงในไฟล์ FASTQ ของรีดในไฟล์ sample_1.fastq และ ในไฟล์ sample_2.fastq

Goodwin, S et al. [50] ได้แบ่งเทคโนโลยีที่ใช้ในการถอดรหัสพันธุกรรมออกเป็นกลุ่มใหญ่ๆ ตามตารางในรูปที่ 2.5 ประกอบด้วย

1. การถอดรหัสพันธุกรรมแบบสายสั้น (Short-read NGS) ซึ่งสามารถแบ่งออกเป็นกลุ่มกว้างๆ ได้เป็น 3 กลุ่มย่อยคือ

1.1 Sequencing by ligation (SBL) ตัวอย่างแพลตฟอร์มเช่น SOLiD และ Complete Genomics เป็นต้น ความยาวของสายรหัสพันธุกรรมที่ผลิตได้จากแต่ละแพลตฟอร์มที่อยู่ในกลุ่มนี้อยู่ ระหว่าง 50-100 base pair (bp) และมีทั้งแบบที่ให้ผลเป็นรหัสพันธุกรรมสายเดี่ยว (single-end reads) และแบบสายคู่ (paired-end reads)

1.2 Sequencing by synthesis (SBS): CRT (Cyclic Reversible Termination) ตัวอย่างแพลตฟอร์มเช่น Illumina และ Qiagen เป็นต้น ความของสายรหัสพันธุกรรมที่ผลิตได้จากแต่ละแพลตฟอร์มที่อยู่ในกลุ่มนี้อยู่ระหว่าง 36-300 bp และมีทั้งแบบที่ให้ผลเป็นรหัสพันธุกรรมสายเดี่ยวและแบบสายคู่ เช่นกัน

1.3 Sequencing by synthesis (SBS): SNA (Single-Nucleotide Addition) ตัวอย่างแพลตฟอร์มเช่น 454 และ Ion Torrent เป็นต้น ความยาวของสายรหัสพันธุกรรมที่ผลิตได้จาก 454 แต่ละแพลตฟอร์มอยู่ระหว่าง 400-1,000 bp และมีทั้งแบบที่ให้ผลเป็นรหัสพันธุกรรมสายเดี่ยวและแบบสายคู่ สำหรับแพลตฟอร์ม Ion Torrent ให้ความยาวของสายรหัสพันธุกรรมเป็น 200 หรือ 400 bp และเป็นแบบสายเดี่ยวเท่านั้น

เปรียบเทียบระหว่างแพลตฟอร์มเหล่านี้ เทคนิค Sequence by Ligation ที่ถูกใช้โดยแพลตฟอร์ม SOLiD และ Complete Genomics ให้ความถูกต้องสูงถึงประมาณ 99.99% แต่ อย่างไรก็ตามก็มีข้อดีข้อเสียระหว่างค่าความไว (sensitivity) และความจำเพาะ (specificity) โดยมีการรายงานว่าความแปรผันในระดับดีเอ็นเอที่เกิดจริงหายไป (ไม่ถูกถอดรหัสออกมา) ในขณะที่ความแปรผันที่ไม่จริงบางตำแหน่งกลับปรากฏในลำดับเบสที่ถูกถอดรหัสออกมา นอกจากนี้ยังมีหลักฐานว่าแพลตฟอร์มในกลุ่มนี้ถอดรหัสได้ไม่แม่นยำในบริเวณของสายดีเอ็นเอที่เป็น AT-rich หรือมีนิวคลีโอไทด์อะดีนีนติดกับไทมีนซ้ำๆ รวมทั้งบริเวณที่เป็น GC-rich ที่มีนิวคลีโอไทด์กวานีนและไซโตซีนอยู่ติดกันหลายๆเบส นอกจากนี้ข้อจำกัดหลักของแพลตฟอร์มกลุ่มนี้คือความยาวของสายรหัสพันธุกรรมที่สามารถถอดออกมาได้ โดยมีความยาวมากที่สุดที่ 75 bp ในแพลตฟอร์ม SOLiD และ 28-100 bp ในกรณีของ Complete Genomics ทำให้ไม่สามารถนำข้อมูลที่ได้ไปวิเคราะห์ลักษณะความผันแปรในเชิงโครงสร้าง (structural variation) ของสายดีเอ็นได้ ปัจจุบันแพลตฟอร์มอิลูมินา ที่เป็นแบบสายคู่มีการใช้งานอย่างกว้างขวางที่สุด เนื่องจากมีการรายงานเรื่องไบแอส (bias) ของการถอดรหัสพันธุกรรมแบบสายเดี่ยวและแพลต ฟอร์มอิลูมินามีรุ่นทางการตลาดที่มีความหลากหลายตามจำนวนรหัสพันธุกรรมที่สามารถถอดรหัสได้ต่อหนึ่งหน่วยเวลาที่ความถูกต้อง >= 99.5% สำหรับแพลตฟอร์ม 454 และ Ion Torrent จะให้สายรหัสพันธุกรรมที่ยาวกว่าสองกลุ่มข้างต้นที่ความยาวเฉลี่ย 700 และ 400 bp ตามลำดับ ซึ่งจะมีประโยชน์จำเพาะกับการวิเคราะห์ข้อมูลดีเอ็นเอในบริเวณที่มีความซับซ้อนหรือมีจำนวนซ้ำของนิวคลีโอไทด์จำนวนมาก อย่างไรก็ตามทั้งสองเทคโนโลยีนี้ใช้หลักการ SNA (Single-Nucleotide Addition) ทำให้มีโอกาสเกิดข้อผิดพลาดมากกว่าแพลตฟอร์มในสองกลุ่มแรกในการ

อ่านรหัสในบริเวณที่เกิดอินเดล (การมีชุดของลำดับเบสเพิ่มหรือหายไป) จากบริเวณสายของดีเอ็นเอที่ถูกอ่านเพื่อถอดรหัสอยู่ บริษัท Roche ได้หยุดการผลิตแพลตฟอร์ม 454 ไปในปีค.ศ. 2516

2. การถอดรหัสพันธุกรรมแบบสายยาว (Long-read NGS)

ด้วยความรู้ที่เพิ่มมากขึ้นจากการถอดรหัสจีโนมของสิ่งมีชีวิตต่างๆ โดยเฉพาะจีโนมในกลุ่มยูแคริโอตพบว่าจีโนมของหลายๆสิ่งมีชีวิตรวมทั้งของมนุษย์มีความซับซ้อน เนื่องจากประกอบด้วยบริเวณที่มีลำดับเบสหรือคู่ของเบสที่เกิดความซ้ำจำนวนมาก การเกิดความซ้ำ การหายไป การกลับด้านของลำดับเบส และการย้ายบริเวณของลำดับเบส จากบริเวณหนึ่งไปยังอีกบริเวณหนึ่งในโครโมโซมเดียวกันหรือต่างโครโมโซมกัน หรือการเกิดเหตุการณ์ข้างต้นมากกว่าหนึ่งเหตุการณ์ร่วมกัน ซึ่งข้อมูลรหัสพันธุกรรมที่ได้จากการใช้เทคโนโลยีการถอดรหัสพันธุกรรมแบบสายสั้น (short reads) จำนวนมาก จะไม่เพียงต่อการนำมาวิเคราะห์ลักษณะของจีโนมข้างต้น จำเป็นต้นมีเทคโนโลยีที่สามารถถอดรหัสพันธุกรรมได้สายยาวมากขึ้น (Long-read sequencing) เพื่อให้สามารถวิเคราะห์ความแปรผันเชิงโครงสร้าง รวมทั้งการเกิดความซ้ำในลักษณะต่างๆ ได้ถูกต้องมากขึ้น โดยเทคโนโลยีที่สามารถถอดรหัสพันธุกรรมแบบสายยาวที่มีอยู่ในปัจจุบัน สามารถแบ่งออกได้เป็น 2 ประเภทใหญ่ๆ ดังต่อไปนี้

2.1 Single-molecule real-time sequencing ตัวอย่างแพลตฟอร์มเช่น Pacific Biosciences และ Oxford Nanopore Technologies (ONT) เป็นต้น ความยาวของสายรหัสพันธุกรรมที่ผลิตได้จากแต่ละแพลตฟอร์มที่อยู่ในกลุ่มนี้อยู่ระหว่าง 8-12 Kb (Kilo bases) ในกรณีที่ใช้แพลตฟอร์ม PacBio Sequel และ ~ 20 Kb ในกรณีของ PacBio RS II และ ยาวถึง 200 Kb ในกรณีของ Oxford Nanopore MK 1 MinION

2.2 Synthetic long-read sequencing ตัวอย่างแพลตฟอร์มเช่น Illumina และ 10X Genomics เป็นต้น โดยแพลตฟอร์มกลุ่มนี้จะอาศัยฐานเทคโนโลยีจากการถอดรหัสพันธุกรรมสายสั้นที่มีอยู่แต่มีการเพิ่มบาร์โคดเพื่อให้สามารถแยกได้ว่ารหัสพันธุกรรมที่อ่านออกมาได้นั้นอยู่ในสายดีเอ็นเอตั้งต้นเดียวกันโดยจะสร้างรหัสพันธุกรรมสายยาวโดยใช้คอมพิวเตอร์ โดยความยาวของสายรหัสพันธุกรรมที่สามารถสร้างได้จากทั้งสองแพลตฟอร์มอยู่ที่ประมาณ 100 Kb

ในปัจจุบันแพลตฟอร์มที่ใช้ในการถอดรหัสพันธุกรรมสายยาวที่มีการใช้มากที่สุดคือ PacBio RS II โดยมีความยาวเฉลี่ยอยู่ที่ 10-15 Kb ซึ่งจะช่วยให้การประกอบร่างจีโนม (genome assembly) สำหรับสิ่งมีชีวิตที่ยังไม่เคยมีการถอดรหัสพันธุกรรมทำได้ง่ายขึ้นมาก อย่างไรก็ตามข้อจำกัดของทั้งสองแพลตฟอร์มนี้คือยังมีความผิดพลาดของการอ่านค่าข้อมูลสูงมากประมาณ 15% ซึ่งความผิดพลาดหลักอยู่ที่บริเวณที่เป็น indel อย่างไรก็ตามความผิดพลาดจากการอ่านนี้เกิดขึ้นกระจายแบบสุ่มดังนั้น ถ้ามีถอดรหัสพันธุกรรมโดยใช้ดีเอ็นเอที่มีสำเนาจำนวนมาก (high coverage) จำนวนสายที่อ่านได้จากเครื่องในบริเวณหนึ่งๆจะมีหลายเส้นซึ่งสามารถใช้ในการทวนสอบการอ่านตำแหน่งหนึ่งๆให้ถูกต้องขึ้นได้

รูปที่ 2.5 ตารางเปรียบเทียบเทคโนโลยีถอดรหัสจีโนม

(ที่มา: ตารางที่ 1 ของ [50])

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)

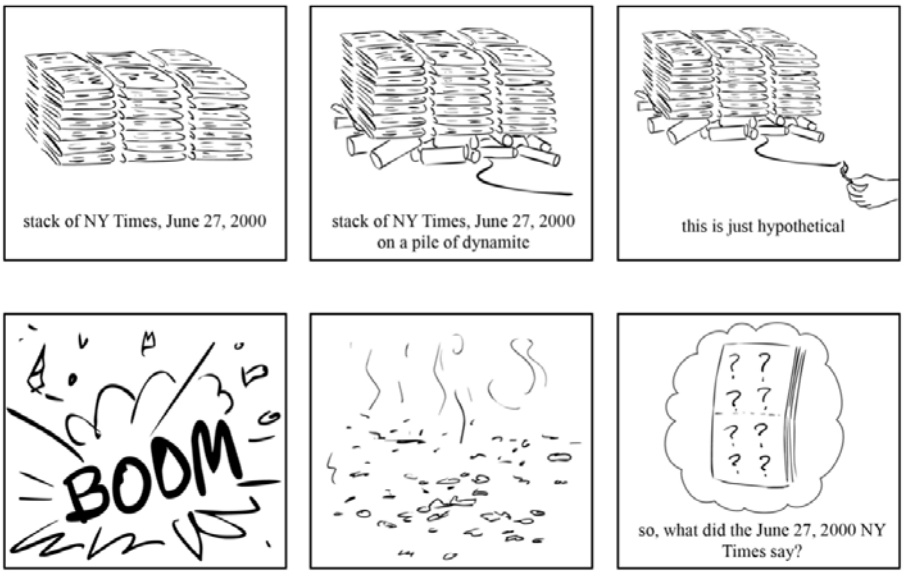
รูปที่ 2.6 ตารางเปรียบเทียบเทคโนโลยีถอดรหัสจีโนม (ต่อ)

(ที่มา: ตารางที่ 1 (ต่อ) ของ [50])

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]

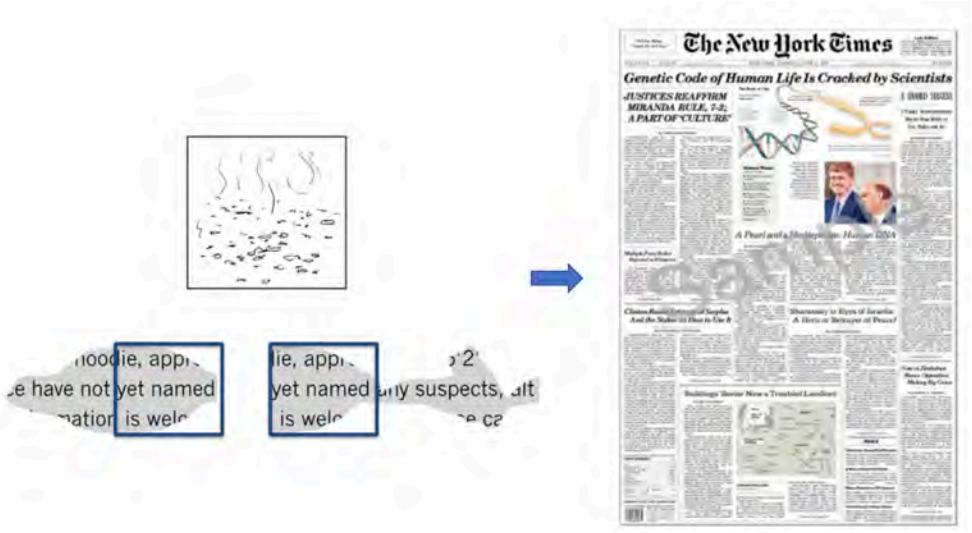
ปัญหากองหนังสือพิมพ์ระเบิด

สมมติว่าหนังสือพิมพ์ฉบับเดียวกันกองอยู่หนึ่งตั้งพร้อมจะไปส่ง ในขณะเองก็เกิดระเบิดขึ้นโดยหนังสือพิมพ์ที่กองอยู่ฉีกขาดกระเด็นเป็นชิ้นเล็กชิ้นน้อยตามรูปที่ 2.7 คำถามคือเราจะต่อชิ้นส่วนหนังสือพิมพ์ที่ฉีกขาดจากแรงระเบิดนี้กลับมาเป็นหนังสือพิมพ์ต้นฉบับได้อย่างไร ปัญหาการต่อชิ้นส่วนหนังสือพิมพ์นี้เป็นปัญหาของการหาชิ้นส่วนที่ทับซ้อน (overlap) กันแล้วเอามาต่อเข้าด้วยกันจนในที่สุดต่อได้ครบถ้วนเท่าต้นฉบับเดิมดังแสดงในรูปที่ 2.8



รูปที่ 2.7 ปัญหากองหนังสือพิมพ์ระเบิด

(ที่มา: รูปที่ 3.1 ของ [21])



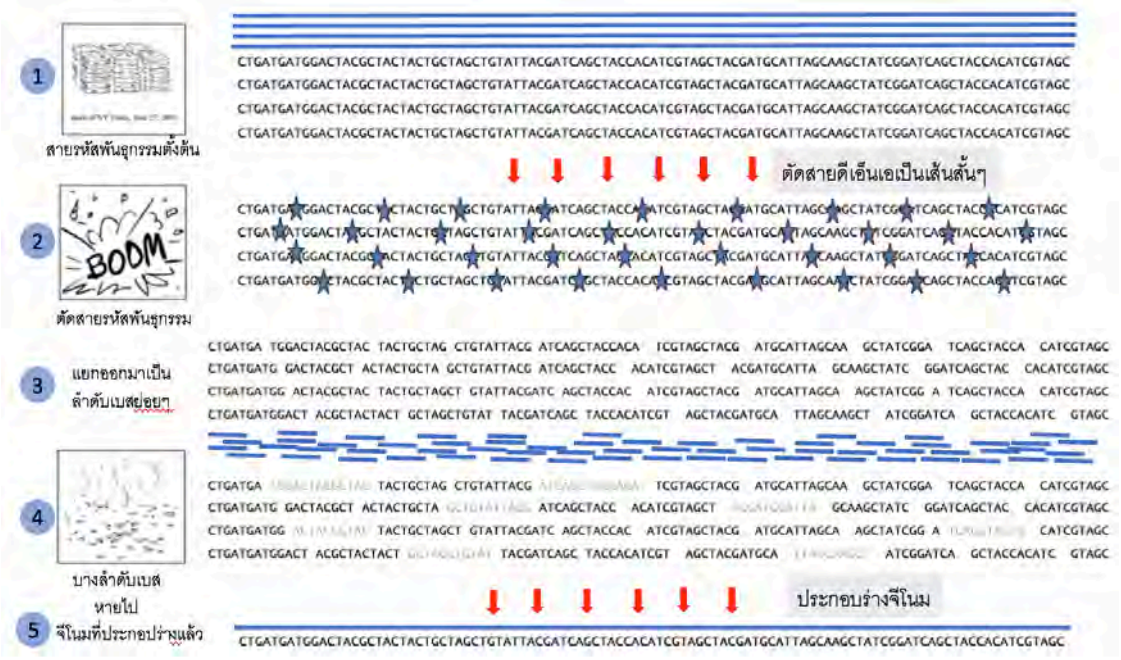
รูปที่ 2.8 การประกอบร่างชิ้นส่วนหนังสือพิมพ์

(ที่มา: แก้ไขจากรูปที่ 3.2 ของ [21] โดยเพิ่มตัวอย่างต้นฉบับทางขวา)

ปัญหาการต่อชิ้นส่วนหนังสือพิมพ์นี้สามารถเปรียบเทียบได้ใกล้เคียงกับปัญหาการประกอบร่างจีโนมตัวอย่างในรูปที่ 2.9 โดยสายดีเอ็นเอเปรียบเสมือนหนังสือพิมพ์ 1 ฉบับและการทำสำเนาดีเอ็นเอหลายๆ สำเนาเปรียบได้กับมีตั้งของหนังสือพิมพ์ฉบับเดียวกัน โดยสายดีเอ็นเอเหล่านี้จะถูกนำมาตัดออกเป็นสายสั้นๆ (fragmentation) เพื่อให้เครื่องถอดรหัสพันธุกรรมแบบสายสั้นสามารถอ่านรหัสพันธุกรรมออกมาได้ การตัดรหัสพันธุกรรมดีเอ็นเอออกเป็นสายสั้นอาจทำได้หลายวิธีตัวอย่าง เช่น การใช้เอ็นไซม์ การใช้คลื่นอัลตราโซนิค การตัดดีเอ็นเอโดยใช้ไนโตรเจนที่ถูกบีบอัดหรือการใช้แรงดันอากาศตัดสายดีเอ็นเอที่ถูกบังคับให้เคลื่อนผ่านรูที่มีขนาดเล็กมาก เป็นต้น โดยประสิทธิภาพของดีเอ็นเอสายสั้นที่ได้จากการตัดด้วยวิธีการต่างๆ เหล่านี้ได้มีการประเมินไว้ใน [51] การตัดดีเอ็นเอเป็นส่วนๆนี้เทียบได้การระเบิดกองหนังสือพิมพ์ ซึ่งรหัสพันธุกรรมสายสั้นบางส่วนก็จะหายไปเหมือนกับชิ้นส่วนที่เกิดไฟไหม้ไปนั่นเอง ปัญหาการประกอบร่างจีโนมมีข้อมูลเข้าหลักเป็นดีเอ็นเอสายสั้นเหล่านี้จำนวนมากมายและผลลัพธ์ที่คาดหวังคือสายรหัสพันธุกรรมที่เป็นผลจากการต่อดีเอ็นเอสายสั้นเหล่านี้ซึ่งคาดหวังว่าจะเป็นตัวแทนดีเอ็นเอต้นฉบับที่มีความถูกต้อง รูปที่ 2.9 (ขั้นตอนที่ 5)

คำถาม	ทำไมการถอดรหัสจีโนมถึงต้องเตรียมสายดีเอ็นเอหลายๆสำเนา

ความท้าทายของปัญหาการประกอบร่างจีโนมเกิดจากเทคโนโลยีการถอดรหัสจีโนมในปัจจุบันที่ไม่สามารถอ่านลำดับรหัสพันธุกรรมได้ครบทั้งโครโมโซม ในกรณีของแพลตฟอร์มอิลูมินาที่มีการใช้งานกันอย่างแพร่หลายใน



รูปที่ 2.9 เทียบเคียงปัญหาการประกอบชิ้นส่วนหนังสือพิมพ์กับการการประกอบร่างจีโนม

(ที่มา: แก้ไขเพิ่มเติมข้อมูลจากรูปที่ 3.1 ของ [21])

ปัจจุบันความยาวของลำดับเบสอยู่ที่ 100-150 นิวคลีโอไทด์เป็นหลัก และปัญหาการประกอบร่างจีโนมไม่ใช่ปัญหาการต่อจิ๊กซอว์ แต่เป็นปัญหาหาส่วนที่ทับซ้อน (overlap) ระหว่างดีเอ็นเอสายสั้นเหล่านี้

ปัญหาการต่อสายสตริง

k-mer (อ่านว่า เค-เมอร์) แสดงลำดับเบสที่เป็นส่วนย่อย (substring/fragment) ของสตริงต้นแบบ โดย k คือค่าจำนวนเต็มที่บอกจำนวนของเบสในลำดับเบสย่อยหนึ่งๆ ตัวอย่างเช่น ถ้ามีลำดับเบสต้นฉบับเป็น
GATTCAACGCTTAGCTT
และมี k=3 องค์ประกอบย่อย 3-mer ของลำดับเบสต้นฉบับนี้ประกอบด้วย GAT, ATT, TTC, TCA, ..., CTT (โดยมีการขยับลำดับเบสไปทีละ 1 เบสเพื่อสร้าง 3-mer ส่วนถัดไป ปัญหาการต่อสตริง (String Reconstruction Problem) ถูกนิยามดังแสดงไว้ในนิยามปัญหาที่ 2.1 ต่อไปนี้

นิยามปัญหาที่ 2.1 ปัญหาการต่อสายสตริง

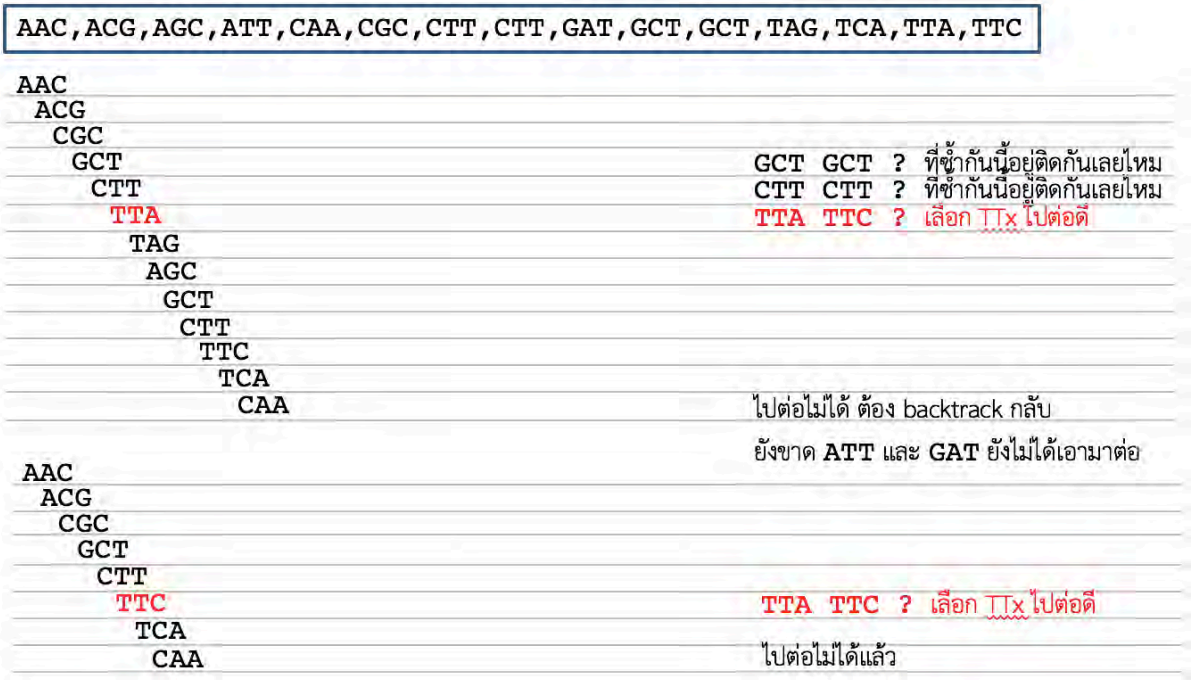
ปัญหาการต่อสายสตริง (String Reconstruction Problem) สร้างสายสตริงต้นฉบับจากลำดับเบสที่เป็นส่วนย่อยของสตริงขนาด k-mers จำนวนมาก	
ข้อมูลเข้า	ชุดของลำดับเบสย่อยขนาด k
ผลลัพธ์	สายสตริงต้นฉบับที่ประกอบด้วยชุดของลำดับเบสย่อยขนาด k ทั้งหมด

วิธีการแก้ปัญหาการต่อสายสตริงแบบง่ายๆ (Naïve Approach)

ขั้นตอน	สิ่งที่ทำ
1.	เรียงชุดของ k-mer ที่ได้มาตามลำดับตัวอักษรตามพจนานุกรม 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 แล้วดูว่ามี k-mer อื่นๆใดบ้างที่ขึ้นต้นด้วย AC จะได้มา k-mers ลำดับที่ 2 ที่เข้าเงื่อนไข
4.	นำ k-mer ที่สองมาต่อกับลำดับเบสตั้งต้นของต้นฉบับ ได้เป็น AACG และทำซ้ำข้อ 3. จนกว่าจะ k-mers ทั้งหมดจะถูกใช้

ฝึกหัด	จากตัวอย่าง k-mers และคำอธิบายเกี่ยวกับขั้นตอนการต่อสายสตริงแบบง่ายๆ จงเขียนแผนภาพแสดงการเชื่อมต่อระหว่าง k-mers

แก้ปัญหาการต่อสายสตริงแบบง่ายๆ ที่อธิบายข้างต้นมีข้อจำกัดคือในกรณีอาจพบเงื่อนไขที่ไม่สามารถไปต่อได้ ดังตัวอย่างชุดของ k-mers ข้างต้น เพราะระหว่างการต่ออยู่ถ้าเลือก k-mer TTA มาต่อ k-mer ถัดไปที่จะต้องนำมาต่อคือ TAG ซึ่งหลังจากนี้ก็จะต้องเป็น k-mer ที่ขึ้นต้นด้วย AGC ซึ่งต่อไปได้จนถึง CAA ก็ต้องมีการย้อนการทำงานกลับ (backtrack) โดยไปเลือก TTC แทน TTA ซึ่งก็จะสามารถต่อ k-mer บางส่วน อย่างไรก็ตามเมื่อสิ้นสุดการทำงาน (ไม่สามารถไปต่อไปได้แล้ว) ก็ยังไม่ได้สายสตริงต้นฉบับที่ถูกต้องครบถ้วนดังแสดงในรูปที่ 2.10



รูปที่ 2.10 แสดงตัวอย่างข้อจำกัดของการต่อสายสตริงโดยวิธีการแบบง่ายๆ

ความซับซ้อนเพิ่มเติมจากการมี k-mer เดียวกันหลายซ้ำภายในชุด

การประกอบสายสตริงต้นฉบับจากชุดของ k-mers ข้างต้นมีความซับซ้อนมากขึ้นเมื่อ k-mer อย่าง **ATG** มีสำเนาเป็น 3 ซ้ำทำให้เรามีเส้นทางแยก 3 เส้นทางคือ **TGG**, **TGC** และ **TGT** ที่สามารถนำมาต่อกับ **ATG** ได้ รวมทั้งมีความเป็นไปได้ที่ k-mer หลายซ้ำนี้อยู่ในตำแหน่งที่หลากหลายตัวอย่างเช่นอยู่ติดกัน อยู่ไม่ติดกัน โดยมี k-mer อื่นแทรกอยู่ 1 k-mer หรือมี k-mer อื่นแทรกอยู่ 2 k-mers เป็นต้น พิจารณาตัวอย่างต่อไปนี้ จากชุดของไบนารีสตริง(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) ในการปรากฏสตริงย่อย “01” อย่างน้อย 1 ครั้งในแต่ละไบนารีสตริง 4-mer เป็นเท่าไหร่ ซึ่งคำตอบคือ 11/16 โดยพบสตริงย่อย “01” ในสตริง 4-mer ที่ถูกขีดเส้นใต้กำกับไว้ดังต่อไปนี้

0000 0001 0010 0011 0100 0101 0110 0111

1000 1001 1010 1011 1100 1101 1110 1111

ถ้ามีคำถามเดียวกันแต่เปลี่ยนสตริงย่อยที่สนใจเป็น “11” จะได้ค่าความน่าจะเป็นเท่ากับ 8/16 หรือ 1/2 ในสตริง 4-mer ที่ถูกขีดเส้นใต้กำกับไว้ต่อไปนี้

0000 0001 0010 0011 0100 0101 0110 0111

1000 1001 1010 1011 1100 1101 1110 1111

คำถามคือทำไมค่าความน่าจะเป็นของการพบสตริงย่อย “11” ดูน้อยกว่าการพบสตริงย่อย “01” ในสตริง 4-mer ทั้งหมดที่เกิดจากการสร้างขึ้นมาแบบสุ่ม และถ้าเปลี่ยนเงื่อนไขในการหาค่าความน่าจะเป็นใหม่ โดยเปลี่ยนเป็นหาค่าความน่าจะเป็นในการปรากฏสตริงย่อย “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

และถ้าเปลี่ยนสตริงย่อยที่สนใจเป็น “11” โดยใช้เงื่อนไขเดียวกันจะได้ค่าความน่าจะเป็นเท่ากับ 3/16 ดังที่แสดงไว้ต่อไปนี้

0000 0001 0010 0011 0100 0101 0110 0111

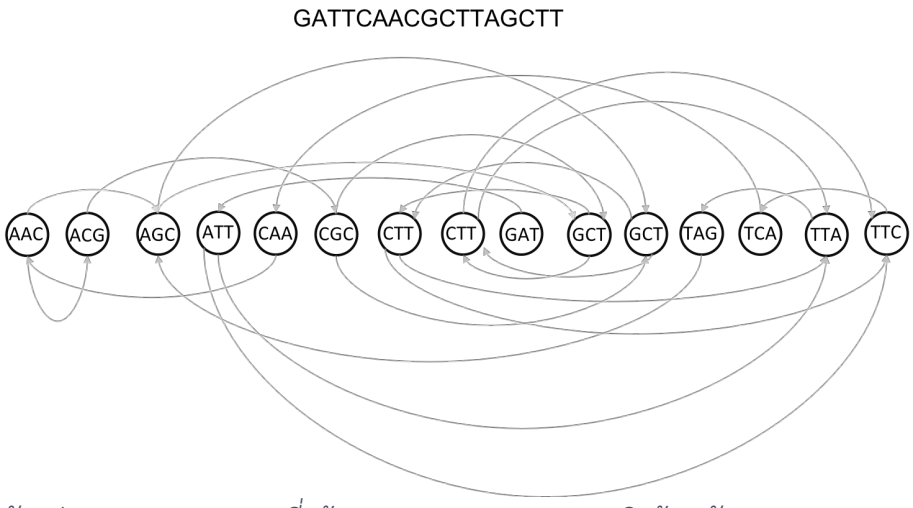
1000 1001 1010 1011 1100 1101 1110 1111

จากตัวอย่างข้างต้นจะเห็นว่าแต่ละ 4-mer ข้างต้นมีโอกาสในการพบสตริงย่อยที่มีรูปแบบจำเพาะหนึ่งๆ ไม่เท่ากัน โดยทั่วไปปรากฏการณ์นี้เรียกว่า overlapping word paradox โดยหมายถึงการพิจารณาสตริงย่อยโดยอนุญาตให้เกิดการคาบเกี่ยว (overlap) กันได้ ตามตัวอย่างข้างต้นปรากฏการณ์นี้มีผลต่อจำนวนที่ปรากฏของสตริงย่อยบางรูปแบบเช่น รูปแบบซ้ำเดิม “11” ในขณะที่รูปแบบอื่นๆจะไม่มีผลกระทบ ดังต่ออย่างสุดท้ายข้างต้นที่ 0111 และ 1110 ถูกนับด้วยว่าพบสตริงย่อย “11” อย่างน้อยสองครั้ง เป็นต้น ซึ่งปรากฏการณ์นี้ทำให้การคำนวณค่าความน่าจะเป็นในการปรากฏรูปแบบจำเพาะหนึ่งๆมีความซับซ้อนมากขึ้นเนื่องจากขึ้นอยู่กับทั้งรูปแบบที่มีความจำเพาะของสตริงย่อยหนึ่งๆ และจำนวนที่คาดว่าจะพบสตริงย่อยนั้นๆ ใน k-mer หนึ่งๆ

วิธีการแก้ปัญหาการต่อสายสตริงโดยใช้เส้นทางฮามิลโทเนียน (Hamiltonian Path)

แนวทางที่สองในการสร้างสายสตริงต้นฉบับจากชุดของลำดับเบสที่เป็นส่วนย่อยของสตริงขนาด k-mers คือการสร้างกราฟแสดงความคาบเกี่ยว (overlap graph) (ปัญหาที่ 2.2) แล้วหาเส้นทางฮามิลโทเนียน (Hamiltonian path) (ปัญหาที่ 2.3) โดยกราฟแสดงความคาบเกี่ยวแสดงการเชื่อมต่อกันระหว่าง k-mers โดยแต่ละโหนดใน

กราฟคือแต่ละ k-mer และเส้นเชื่อมที่มีทิศทาง (directed edge) จากโหนด A ไป B (A→ B) แสดงความสัมพันธ์ว่า k-mer ของโหนด A และ B นั้นมีรูปแบบที่มีความคาบเกี่ยวกัน สามารถเอามาเชื่อมต่อกันได้ โดยอธิบายผ่านฟังก์ชัน OVERLAP(Patterns) ได้ว่าจะมีเส้นเชื่อมระหว่าง A และ B ก็ต่อเมื่อ SUFFIX(Pattern) = PREFIX(Pattern') โดย Pattern คือรูปแบบจำเพาะของโหนด A และ Pattern' เป็นรูปแบบจำเพาะของ B ตามลำดับ ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 ตัวอย่าง overlap graph ที่สร้างจาก 3-mer ของสายสตริงต้นฉบับ GATTCAACGCTTAGCTT
(ที่มา: ดัดแปลงจากรูปที่ 3.9 ของ [21])

นิยามปัญหาที่ 2.2 ปัญหาสร้างกราฟแสดงความคาบเกี่ยวระหว่างโหนด

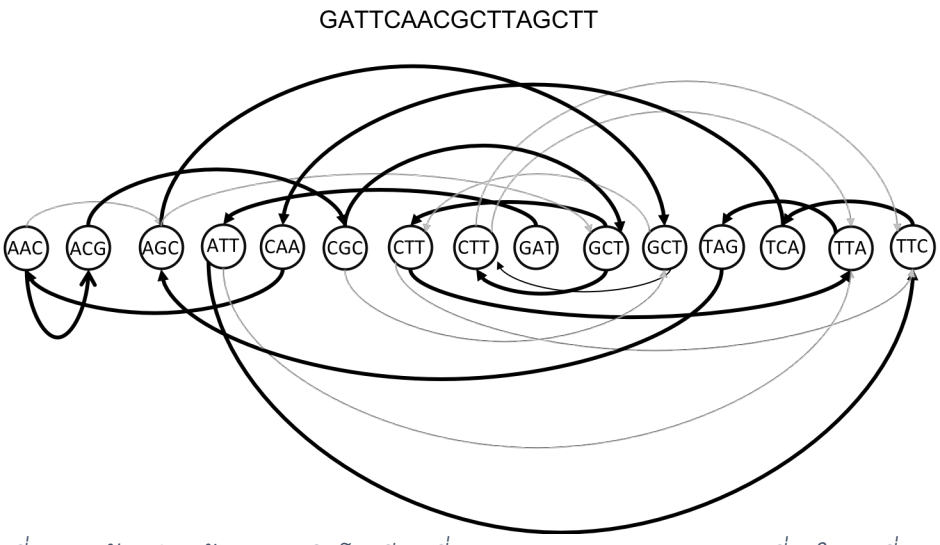
ปัญหาสร้างกราฟแสดงความคาบเกี่ยวระหว่างโหนด (Overlap Graph Problem)	
ข้อมูลเข้า	ชุดของสตริงย่อยที่มีขนาด k-mer
ผลลัพธ์	กราฟแสดงความคาบเกี่ยวระหว่างโหนด โดยผ่านฟังก์ชัน OVERLAP (Patterns)

กราฟแสดงความคาบเกี่ยวระหว่างโหนดนี้จะเป็นข้อมูลเข้าของปัญหาที่ 2.3 และผลลัพธ์ที่คาดหวังคือเส้นทางฮามิลโทเนียน ดังตัวอย่างเส้นสีดำในรูปที่ 2.12 ซึ่งแสดงเส้นทางการต่อกันของ k-mer ทุกโหนดเพื่อให้ได้เป็นสายสตริงต้นฉบับนั่นเอง

นิยามปัญหาที่ 2.3 ปัญหาการหาเส้นทางฮามิลโทเนียน

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

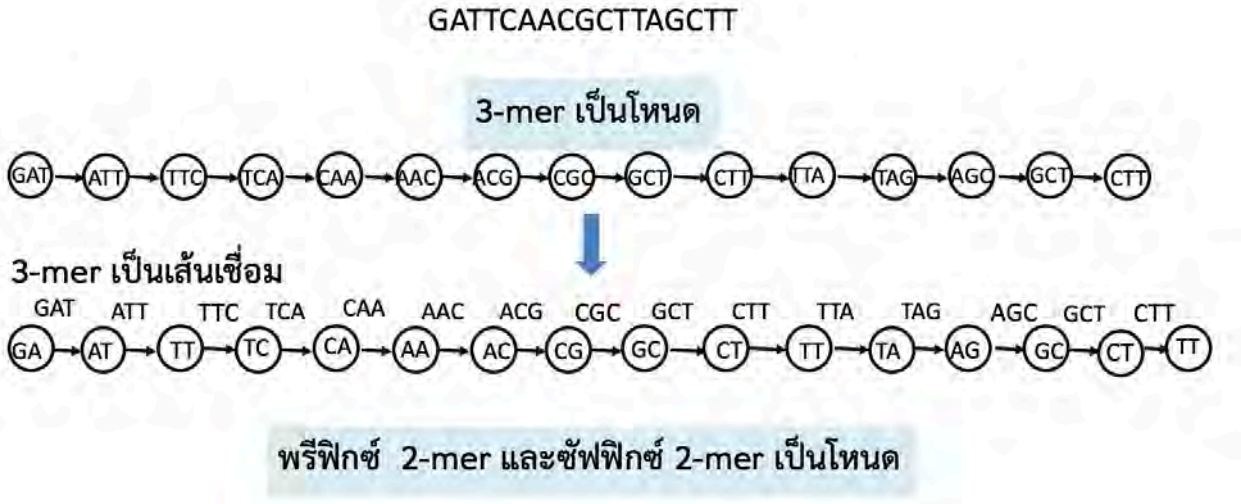
ความซับซ้อนของการหาเส้นทางฮามิลโนเนียนในกราฟเป็น NP-complete



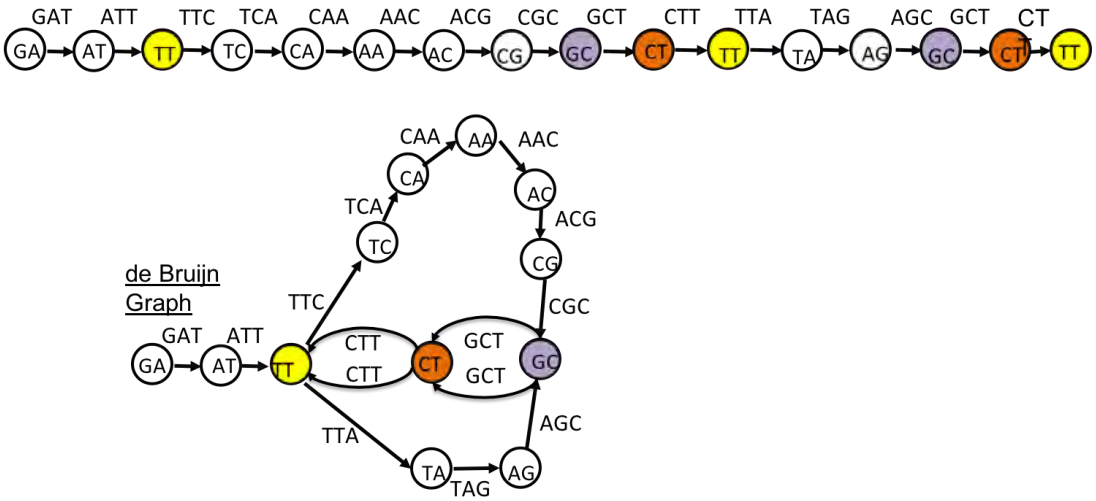
รูปที่ 2.12 ตัวอย่างเส้นทางฮามิลโทเนียนที่หาจากกราฟแสดงความคาบเกี่ยวในรูปที่ 2.11
(ที่มา: ดัดแปลงจากรูปที่ 3.9 บน ของ [21])

วิธีการแก้ปัญหาการต่อสายสตริงโดยใช้เส้นทางออยเลอร์ (Euler Path)

แนวทางที่สามในการสร้างสายสตริงต้นฉบับจากชุดของลำดับเบสที่เป็นส่วนย่อยของสตริงขนาด k-mers คือการสร้างกราฟ de Bruijn (ปัญหาที่ 2.4) แล้วหาเส้นทางออยเลอร์ (ปัญหาที่ 2.5) โดยกราฟ de Bruijn แสดงการเชื่อมต่อกันระหว่าง k-mers โดยแต่ละโหนดในกราฟคือ *ส่วนของ k-mer* โดยเป็น PREFIX(k-mer) และ SUFFIX(k-mer) ตามลำดับ และเส้นเชื่อมระหว่าง PREFIX และ SUFFIX โหนดของ k-mer เดียวกันจะแสดงลำดับเบสของ k-mer นั้นๆ ตัวอย่างโหนดและเส้นเชื่อมของกราฟ de Bruijn แสดงในรูปที่ 2.13 และจะมีการหาและรวมโหนดที่มีลำดับเบส k-n mers แบบเดียวกันทีละกลุ่มๆ จนได้กราฟ de Bruijn สุดท้ายในรูปที่ 2.14



รูปที่ 2.13 ตัวอย่างโหนดและเส้นเชื่อมในกราฟ de Bruijn



รูปที่ 2.14 ตัวอย่างกราฟ de Bruijn ที่สร้างจาก 3-mer ของสายสตริงต้นฉบับ GATTCAACGCTTAGCTT

นิยามปัญหาที่ 2.4 ปัญหาสร้างกราฟ de Bruijn

ปัญหาสร้างกราฟ de Bruijn	
ข้อมูลเข้า	ชุดของสตริงย่อยที่มีขนาด 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) ซึ่งแสดงเส้นทางที่ผ่านทุกเส้นเชื่อมเพื่อให้ได้เป็นสายสตริงต้นฉบับนั่นเอง

ทฤษฎีบทของออยเลอร์ (Euler's Theorem) ประยุกต์ใช้กับกราฟมีทิศทาง (directed graph) โดยกราฟหนึ่งๆ จะมีคุณสมบัติ Eulerian ถ้ากราฟนั้น สมดุล (balanced) *และ* มีคุณสมบัติ strongly connected คือทุกโหนดในกราฟจะต้องสามารถเชื่อมถึงกันได้โดยเส้นทางใดเส้นทางหนึ่ง โดยกราฟจะสมดุลถ้าทุกโหนดในกราฟมีความสมดุลซึ่งหมายถึง 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 เมื่อเดินถึงจุดตั้งต้นที่เลือกนี้ให้เพิ่มเส้นทางนี้ใน 1. และ กลับไป 2.

กราฟ de Bruijn และ กราฟแสดงความคาบเกี่ยว

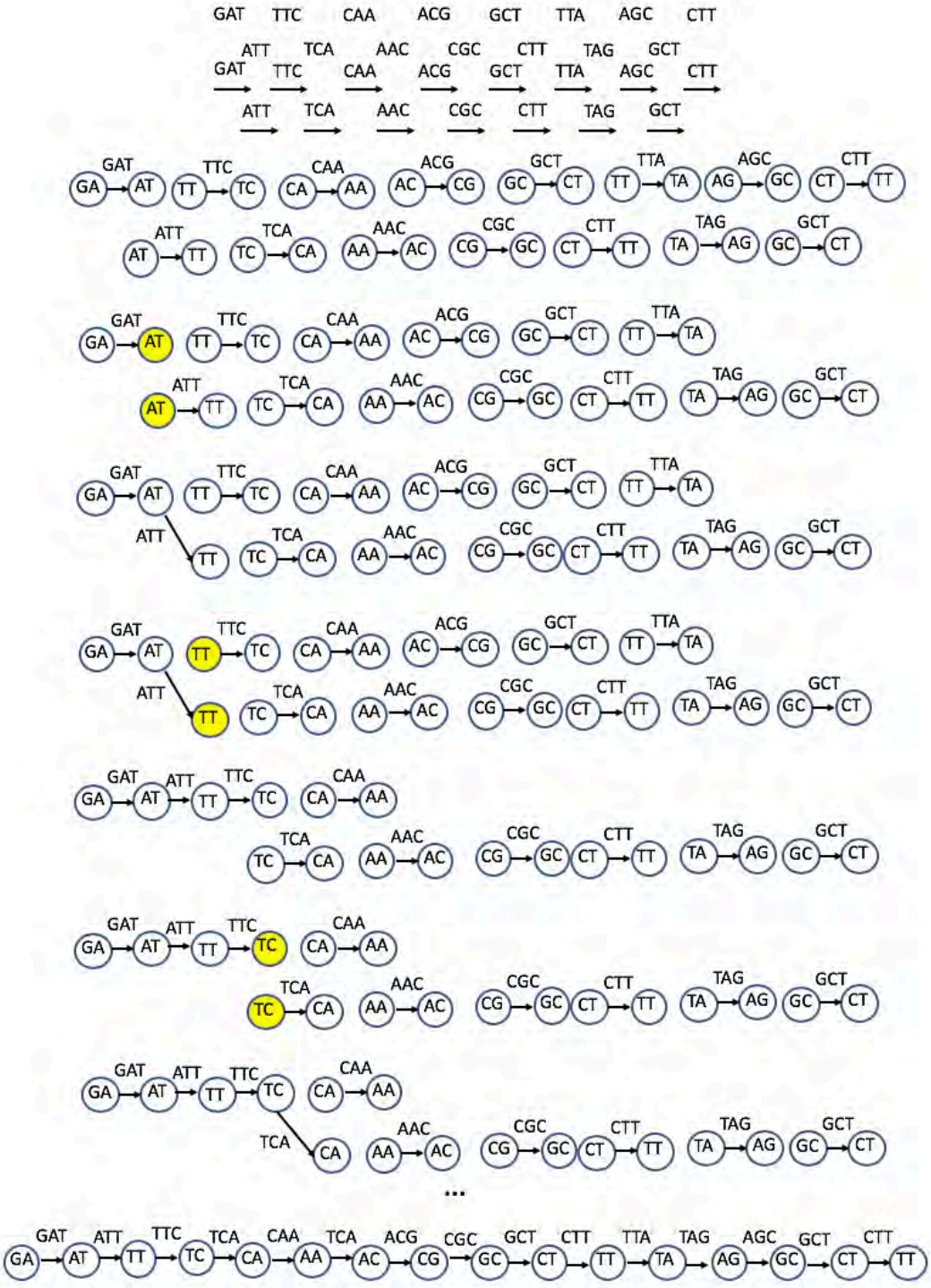
ถึงจุดนี้เรามีแนวทางแก้ปัญหาการประกอบร่างจีโนมจากชุดของ k-mer โดยการหาเส้นทางฮามิลโทเนียนจากกราฟแสดงความคาบเกี่ยวโดยต้องเดินผ่านทุกโหนดในกราฟเพียงครั้งเดียว และการหาเส้นทางออยเลอร์จากกราฟ de Bruijn โดยต้องเดินผ่านทุกเส้นเชื่อมในกราฟเพียงครั้งเดียว คำถามคือเราควรเลือกวิธีไหน

อัลกอริทึมที่ใช้ในการหาเส้นทางออยเลอร์จากกราฟ de Bruijn ใช้เวลาในการทำงานเป็นระดับโพลีโนเมียล (polynomial) แต่ยังไม่มีอัลกอริทึมใดที่สามารถหาเส้นทางฮามิลโทเนียลอยู่ภายในกรอบเวลาที่เป็นโพลีโนเมียล ถ้าย้อนประวัติไปในการประกอบร่างรหัสพันธุกรรมดีเอ็นเอใน 20 ปีย้อนหลังจะพบว่ามีการพยายามใช้กราฟแสดงความคาบเกี่ยวในการประกอบร่างจีโนมมนุษย์ ซึ่งในเวอร์ชันแรกๆ ของจีโนมมนุษย์ก็ได้มาจากวิธีการประกอบร่างจีโนมจากกราฟแสดงความคาบเกี่ยว จนเมื่อมีการนำกราฟ de Bruijn เข้ามาจำลองการเชื่อมต่อของดีเอ็นเอสายสั้นโดยมีข้อมูลบนเส้นเชื่อมต่อแสดงความทับซ้อนระหว่างดีเอ็นสายสั้นสองสายแทนกราฟแสดงความคาบเกี่ยวเดิม อัลกอริทึมที่ถูกออกแบบและพัฒนาในสมัยหลังจากนั้นจึงมีการใช้กราฟ de Bruijn เป็นหลักในการประกอบร่างจีโนม

การสร้างกราฟ de Bruijn จากชุดของดีเอ็นเอสายสั้น

ตัวอย่างข้างต้นอธิบายลักษณะของกราฟ de Bruijn โดยสร้างกราฟจากจีโนมที่ทราบลำดับเบสอยู่แล้ว อย่างไรก็ตามในการทำงานจริงเราไม่ทราบลำดับเบสของจีโนม คำถามคือเราจะสร้างลำดับเบสของจีโนมจากชุดของดีเอ็น

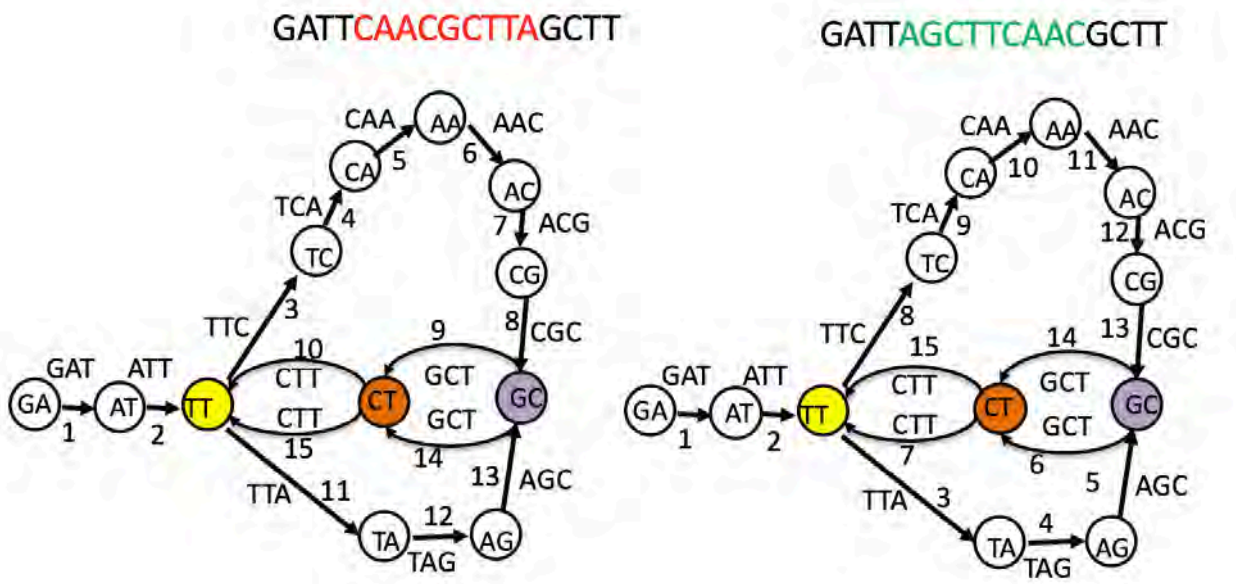
เอสายสั้นได้อย่างไร รูปที่ 2.15 แสดงการสร้างชุดของโหนดและเส้นเชื่อมในรูปแบบกราฟ de Bruijn โดยหลังจากสามารถเชื่อมโยงทุกโหนดให้เป็นกราฟเดียวกันแล้ว กราฟนี้จะถูกนำไปสร้างเป็น de Bruijn สุดท้ายในรูปที่ 2.14



รูปที่ 2.15 ขั้นตอนการสร้างกราฟ de Bruijn จากชุดของดีเอ็นเอสายสั้น

การประกอบร่างจีโนมโดยใช้ดีเอ็นสายคู่

ในกราฟ de Bruijn หนึ่งๆ สามารถมีเส้นทางออยเลอร์ได้มากกว่า 1 เส้นทาง ดังตัวอย่างในรูปที่ 2.16 แต่ในจีโนมต้นฉบับนั้นมีลำดับเบสเป็น **GATTCAACGCTTAGCTT** ซึ่งก็คือเส้นทางซ้ายในรูป เพื่อเป็นการลดความคลุมเครือของการประกอบร่างจีโนมนี้ วิธีแรกที่เป็นไปได้คือการเพิ่มความยาวของ k-mer (อย่างไรก็ตามการเพิ่มความยาวของ k-mer เท่ากับมีสมมติฐานว่าเครื่องถอดรหัสพันธุกรรมสามารถอ่านลำดับเบสสายยาวได้ ซึ่งในปัจจุบัน กลางปีพ.ศ. 2561 (ค.ศ. 2518) นี้การถอดรหัสสายยาวยังมีราคาสูงมากและมีความผิดพลาดในการอ่านมากเมื่อเทียบกับการถอดรหัสสายสั้นโดยเทคโนโลยี NGS โดยแพลตฟอร์มอิลูมินา [50]



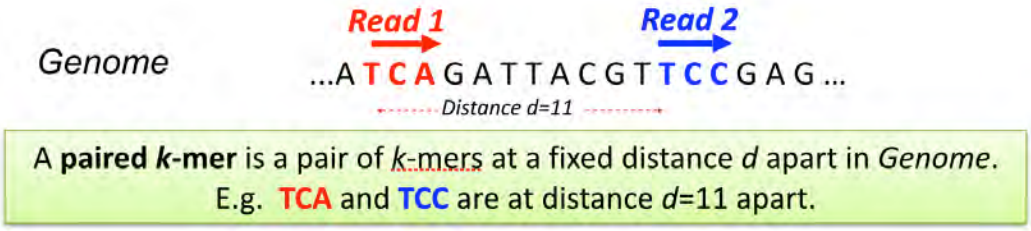
รูปที่ 2.16 ตัวอย่างเส้นทางออยเลอร์สองเส้นทางในกราฟ de Bruijn ที่สร้างจากสายสตริงต้นฉบับ GATTCAACGCTTAGCTT

อีกแนวทางที่เป็นไปได้คือการออกแบบการทดลองให้เครื่องถอดรหัสพันธุกรรมสามารถถอดรหัสพันธุกรรมที่เป็นสายคู่ได้ ซึ่งปัจจุบันเทคโนโลยีการถอดรหัสพันธุกรรมแบบสายสั้นอย่างอิลูมินานั้น เน้นการถอดรหัสพันธุกรรมออกมาเป็นแบบสายคู่เป็นหลัก 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 เราจะกำหนดพรีฟิกซ์ (prefix) และ ซัฟฟิกซ์ (suffix) ของ k-mer สายคู่นี้ (สมมติว่า $d = 1$) ดังต่อไปนี้

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

SUFFIX(($a_1 \ldots a_k | b_1, \ldots b_k$)) = ($a_2 \ldots a_k | b_2, \ldots b_k$)

ตัวอย่างเช่น PREFIX((**GAC | TCA**)) = (**GA | TC**) และ SUFFIX((**GAC | TCA**)) = (**AC | CA**) เป็นต้น ซึ่งซัฟฟิกซ์ของ k-mer แรกก็จะเป็นพรีฟิกซ์ของ k-mer ถัดไปนั่นเอง



รูปที่ 2.17 การนับระยะห่างระหว่างคู่ของสายดีเอ็นเอ

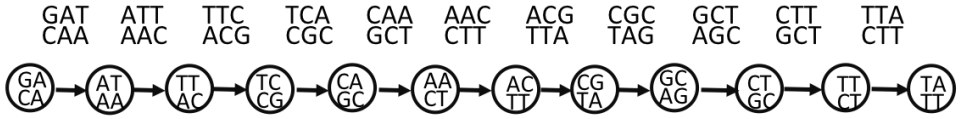
รูปที่ 2.18 แสดงตัวอย่างของสายคู่เทียบกับจีโนมต้นฉบับโดยในตัวอย่างนี้แต่ละสายของคู่ยาว 3-mer และแต่ละคู่ห่างกัน 1 เบส และรูปที่ 2.19 แสดงเส้นทางกราฟที่เชื่อมต่อ 3-mer สายคู่ที่สร้างจากสายสตริงต้นฉบับ GATTCAACGCTTAGCTT โดยโหนดและเส้นเชื่อมอยู่ในรูปแบบของกราฟ de Bruijn

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

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 ตัวอย่างการสร้างคู่ของ 3-mer ที่ห่างกัน 1 เบสของสายสตริงต้นฉบับ GATTCAACGCTTAGCTT



รูปที่ 2.19 ตัวอย่างเส้นทางกราฟ (path graph) ที่เชื่อมต่อ 3-mer สายคู่ที่สร้างจากสายสตริงต้นฉบับ GATTCAACGCTTAGCTT

(ที่มา: ดัดแปลงจากรูปที่ 3.33 ของ [21])

เราสามารถสร้างกราฟ de Bruijn ของ k-mer สายคู่จากเส้นทางกราฟในรูปที่ 2.19 โดยการรวมโหนดที่มีลำดับเบสภายในโหนดเป็นแบบเดียวกันซึ่งในกรณีนี้ไม่มี ดังนั้นกราฟ de Bruijn สุดท้ายก็จะเป็นกราฟเดียวกับรูปที่ 2.19 และมีเส้นทางออยเลอร์เดียวที่เป็นไปได้คือ GATTCAACGCTTAGCTT นั่นเอง จะเห็นกราฟ de Bruijn ที่

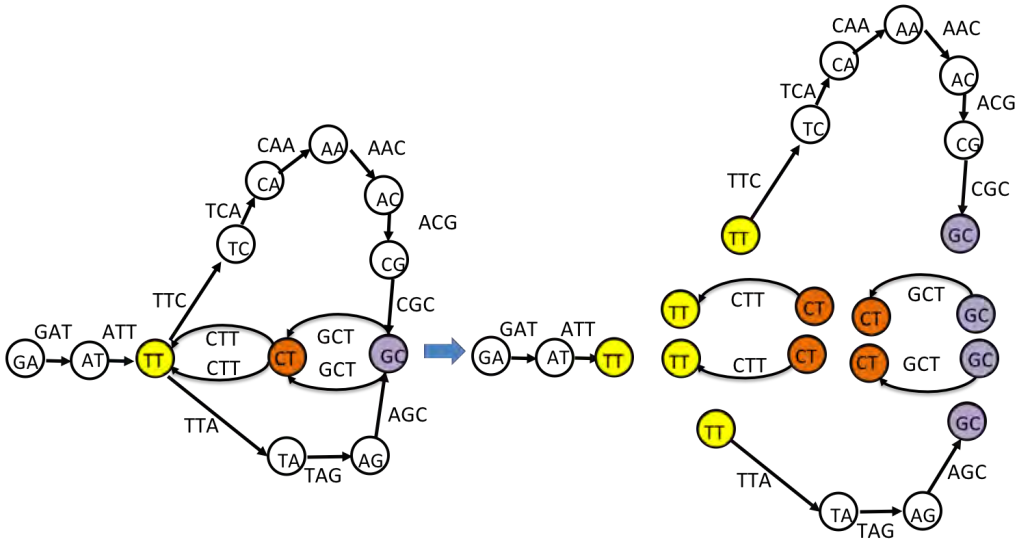
สร้างจาก k-mer สายคู่สามารถช่วยลดความคลุมเครือในการเลือกเส้นทางเดินได้เมื่อเทียบกับกราฟที่สร้างจาก k-mer สายเดี่ยวในรูปที่ 2.16

บทส่งท้าย

ในขณะที่บทเรียนข้างต้นสามารถนำมาประยุกต์ใช้ในการแก้ปัญหาการประกอบร่างจีโนม (genome assembly) โดยพื้นฐานในการออกแบบอัลกอริทึมเพื่อให้การประกอบร่างจีโนมจากข้อมูลจริงที่ได้จากเครื่องถอดรหัสที่สามารถเกิดข้อผิดพลาดในการอ่านข้อมูลได้ตลอดนั้น มีความซับซ้อนมากขึ้นในมิติต่างๆ เช่น

(1) *มีการแบ่งข้อมูลลำดับเบสที่อ่านได้ซึ่งเรียกว่ารีด (read)* ออกเป็นลำดับเบสย่อยขนาด k-mer เพื่อให้แต่ละ k-mer มีโอกาสถูกต้องทุกลำดับเบสมากขึ้น (แต่ก็เพิ่มโอกาสในการเกิดทางเลือกมากขึ้น ทำให้การหาเส้นทางออยเลอร์ ซับซ้อนมากขึ้น เช่นกัน) เพื่อเป็นการเพิ่มความครอบคลุม (coverage) ของลำดับเบสที่ถูกถอดรหัสออกมาได้จากจีโนมในบริเวณหนึ่งๆ ให้มีความถูกต้องน่าเชื่อถือมากขึ้น เปรียบได้กับการมีชิ้นส่วนของหนังสือพิมพ์ที่ใกล้เคียงกันหลายๆชิ้นอาจจะชิ้นเล็กๆ แต่ถูกต้องเพื่อตรวจสอบซึ่งกันและกัน

(2) *มีการแตกจีโนมออกเป็นส่วนๆ เรียกว่าคอนทิก (contig)* เพื่อแก้ปัญหาที่เครื่องถอดรหัสไม่สามารถอ่านรหัสพันธุกรรมในบางบริเวณของจีโนมได้หรืออ่านได้ไม่ดี หรือบริเวณนั้นมีลำดับเบสซ้ำ (repeats) จำนวนมาก ทำให้เกิดช่องว่าง (gap) ของข้อมูลบริเวณเหล่านั้น (ข้อมูลบริเวณนั้นหายไป) ซึ่งนำไปทำให้เส้นเชื่อมระหว่างโหนดในกราฟ de Bruijn บางส่วนหายไป ทำให้ไม่สามารถหาเส้นทางออยเลอร์ที่สมบูรณ์ได้ โดยแต่ละคอนทิกจะถูกแสดงโดย maximal non-branching path ดังแสดงในรูปที่ 2.20

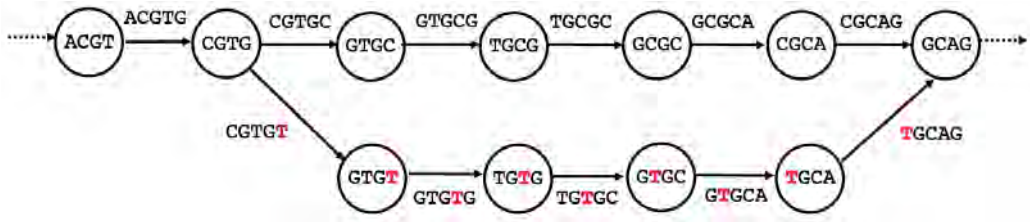


รูปที่ 2.20 กราฟ de Bruijn ที่ถูกแตกออกเป็น 7 maximal non-branching paths ซึ่งถูกแสดงโดย GATT, CTT, CTT, GCT, GCT, TTCAACGC และ TTAGC

(ที่มา: ดัดแปลงจากรูปที่ 3.38 ของ [21])

ตัวอย่างผลงานวิจัยของผู้เขียนตำรานี้ในครงการถอดรหัสจีโนมของเชื้อราชื่อ *Ophiocordyceps polyrha chis-furcata* [52] โดยแพลตฟอร์ม 454 (Roche 454 GS FLX) ซึ่งเป็นเครื่องถอดรหัสพันธุกรรมสายสั้นแต่ยาวกว่าแพลตฟอร์มอิลูมินา ผลของการประกอบร่างจีโนมแบบ *de novo* assembly (คือประกอบร่างขึ้นมาใหม่โดยไม่มีการเทียบกับจีโนมอ้างอิง เนื่องจากเป็นเชื้อที่ไม่เคยมีการถอดรหัสพันธุกรรมมาก่อน) โดยใช้โปรแกรม Newbler v2.8 ได้จำนวนคอนทิก (contig) ทั้งสิ้น 4,419 คอนทิก และเพื่อให้สามารถเชื่อมต่อคอนทิกเหล่านี้ให้ยาวขึ้น ได้มีการถอดรหัสพันธุกรรมเพิ่มเติมโดยใช้แพลตฟอร์มอิลูมินาที่ให้ข้อมูลเป็น ไลบรารี mate pairs โดยได้ผลข้อมูลเป็นรหัสพันธุกรรมสายคู่ที่มีช่องว่างระหว่างคู่ของสายขนาด 3, 6 และ 8 Kb (กิโลเบส) และนำรหัสพันธุกรรมสายคู่เหล่านี้มาประกอบร่างเข้ากับคอนทิกที่มีอยู่ก่อนหน้าโดยใช้โปรแกรมสร้าง scaffolds ชื่อ SSPACE 2.0 [53] ได้ผลเป็น 418 scaffolds (แต่ละ scaffold ประกอบด้วยชุดของคอนทิกที่คั่นด้วยช่องว่าง หรือ gap และมีคู่ของรีด (คู่ของ mate pair) ที่สายที่หนึ่งที่อยู่ในคอนทิกแรกและคู่ของมันตกอยู่ในคอนทิกที่สอง) โดยมี 59 scaffolds ที่มีความยาวมากกว่า 1 Kb และมีค่า N50 เท่ากับ 3.3 ล้านเบส ทั้งนี้ N50 คือค่ามัธยฐาน (median) ของความยาวของคอนทิก หรือ scaffolds ในกรณีนี้ โดยแสดงถึงคุณภาพของจีโนมที่ประกอบร่างขึ้นมาโดยถ้า N50 มีค่ามากแสดงว่าจีโนมที่ประกอบร่างขึ้นมานี้มีคุณภาพดีถึงดีมาก ทั้งนี้จากการเปรียบเทียบค่า N50 ของเชื้อรานี้กับราอื่นๆ ที่มีการถอดรหัสจีโนมมาก่อนถือว่าจีโนมที่ได้มีคุณภาพดีมาก เชื้อรา *Ophiocordyceps polyrhachis-furcata* มีขนาดของจีโนมโดยประมาณ 43 Mb (43 ล้านเบส) ในขณะที่จีโนมมนุษย์มีขนาดประมาณ 3 พันล้านเบส (3Gb)

(3) *มีการจัดการเรื่องบับเบิ้ล (bubble)* ในกราฟที่เป็นผลจากการเกิดความผิดพลาดในการอ่านรีด (สายของลำดับเบสที่อ่านได้จากเครื่องถอดรหัสพันธุกรรม) ตัวอย่างเช่น ลำดับเบส **ACGTGTGCAG** ตรงตำแหน่งนิวคลีโอไทด์ **T** ถูกอ่านมาไม่ถูกต้องโดยเปลี่ยนจาก **C** มาเป็น **T** รูปที่ 2.21 แสดงชุดของ 5-mers ที่ไม่ถูกต้องเนื่องจากมีเบส **T** นี้เป็นส่วนประกอบ **GTGTG**, **TGTGC**, **GTGCA** และ **TGCAG**



รูปที่ 2.21 การเกิด bubble ในกราฟ de Bruijn จากการเกิดลำดับเบสที่ผิดโดยตำแหน่งที่เป็น C ถูกอ่านเป็น T (ที่มา: ดัดแปลงจากรูปที่ 3.39 (บน) ของ [21])

ซึ่ง 5-mers ชุดนี้ทำให้เกิดเส้นทางที่ไม่ถูกต้องจากโหนด **CGTG** ไปยังโหนด **GCAG** ซึ่งหมายความว่าถ้ามีชุดของ 5-mers ที่สามารถสร้างเส้นทางที่ถูกต้องด้วยก็จะมีสองเส้นทางที่เชื่อมระหว่างโหนด **CGTG** ไปยังโหนด **GCAG** ในกราฟ de Bruijn ซึ่งโครงสร้างนี้เรียกว่าบับเบิ้ลซึ่งหมายถึงมีสองเส้นทางที่แยกออกจากกันโดยทั้งสอง

เส้นทางมีโหนดตั้งต้นก่อนแยกและโหนดปลายที่กลับมาร่วมเส้นทางกันเป็นคู่โหนดเดียวกัน (โดยเส้นทางเหล่านี้จะต้องสั้นกว่าค่า threshold ที่กำหนด) การจัดการบับเบิ้ลนี้โปรแกรมประกอบร่างจีโนม (genome assembler) ส่วนใหญ่จะทำการกำจัดบับเบิ้ลเหล่านี้ออกไปซึ่งมีโอกาสที่เส้นทางที่ถูกต้องจะถูกกำจัดออกไปด้วยทำให้การประกอบร่างจีโนมมีข้อผิดพลาด นอกจากนี้ข้อมูลลำดับเบสซ้ำแต่ไม่ซ้ำทั้งหมด เช่น เกิดความแปรผันในบางลำดับเบส ก็ทำให้เกิดบับเบิ้ลซ้อนๆกันได้ ซึ่งการกำจัดบับเบิ้ลโดยเลือกเส้นทางใดเส้นทางหนึ่งก็จะทำให้เกิดข้อผิดพลาดในการประกอบร่างจีโนม เช่นกัน ดังนั้นโปรแกรมประกอบร่างจีโนมสมัยหลังๆ จึงมีความพยายามในการแยกความแตกต่างระหว่างการเปลี่ยนแปลงของลำดับเบสที่เกิดจากการอ่านข้อมูลผิดพลาดของเครื่องถอดรหัสพันธุกรรมกับการเปลี่ยนแปลงของลำดับเบสที่เกิดจากความแปรผันของจีโนมที่เกิดขึ้นจริง

นอกจากนี้สมมติฐานอื่นๆ ในออกแบบอัลกอริทึมเพื่อประกอบร่างจีโนมยังไม่เป็นจริง ในชุดของข้อมูลที่ได้จากเครื่องถอดรหัส เช่น ระยะห่างระหว่างลำดับเบสสายคู่ (paired-end reads) มีค่าคงที่ รีดที่อ่านได้จากเครื่องถอดรหัสครอบคลุมทุกบริเวณในจีโนม เราทราบรูปแบบที่เป็นไปได้ทั้งหมดของ k-mer ในการถอดรหัสจีโนมของสิ่งมีชีวิตหนึ่งๆ เป็นต้น

ตัวอย่างโปรแกรมประกอบร่างจีโนมที่มีการใช้งานกันอย่างแพร่หลาย

ตัวอย่างของโปรแกรมที่มีการใช้งานอย่างแพร่หลายในการประกอบร่างจีโนมของสิ่งมีชีวิตกลุ่มยูแคริโอต เช่น SOAPdenovo2 [54, 55] ALLPATHS [56] Velvet [57] ABySS [58] เป็นต้น ซึ่งทั้งสี่โปรแกรมใช้กราฟ de Bruijn เป็นฐาน รีวิวของ Nagarajan N. และ Pop M. [59] ได้ทำการเปรียบเทียบข้อดีข้อเสียของโปรแกรมที่ใช้ประกอบร่างจีโนม รวมทั้งการประยุกต์ใช้โปรแกรมเหล่านี้กับข้อมูลรหัสพันธุกรรมในระดับอาร์เอ็นเอ และข้อมูลรหัสพันธุกรรมที่ถอดรหัสจากกลุ่มเชื้อ (meta-genomics) เช่น กลุ่มของเชื้อที่เก็บจากสิ่งแวดล้อม กลุ่มของเชื้อที่เก็บจากลำไส้ เป็นต้น

แบบฝึกหัดบทที่ 2

ให้เขียนโปรแกรมเพื่อแก้ปัญหาที่เกี่ยวข้องกับการประกอบร่างจีโนม (genome assembly) โดยใช้โจทย์ที่โรซาลินด์ (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

WGS (Whole Genome Sequencing) คือการถอดรหัสพันธุกรรมในระดับจีโนมซึ่งครอบคลุมทั้งส่วนที่เป็นยีนที่สามารถแปลรหัสไปเป็นโปรตีนและส่วนอื่นๆ ทั้งหมด ในขณะที่ WES (Whole Exome Sequencing) เน้นการถอดรหัสพันธุกรรมของเอ็กซอนทั้งหมดที่อยู่ในจีโนม ซึ่งจะครอบคลุมเฉพาะส่วนที่เป็นยีนที่สามารถแปลรหัสต่อไปเป็นโปรตีนซึ่งครอบคลุมรหัสพันธุกรรมเพียงประมาณ 1-2% ของจีโนมทั้งหมดโดยแพลตฟอร์มหลักๆที่ใช้ในการถอดรหัสเอ็กโซมคือ NimbleGen, Agilgent และ Illumina [60] โดยข้อมูลรหัสพันธุกรรมเอ็กโซมที่ได้จะถูกนำไปเทียบกับจีโนมอ้างอิงเพื่อวิเคราะห์ความแปรผันต่างๆ เช่นการแปรผันของลำดับเบสเดี่ยวๆ ที่เรียกว่าเอสเอ็นวี (Single Nucleotide Variant: SNV) และ การเพิ่มหรือการหายไปของลำดับเบสสายสั้นๆ ที่เรียกรวมๆว่าอินเดล (indels) เป็นต้น

บทที่ 3 การเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิง (Short read mapping to reference genome)

วัตถุประสงค์

- เพื่อให้นิสิตได้เห็นขั้นตอนหลัก (ขั้นตอนแรก) ในการวิเคราะห์ข้อมูลรหัสพันธุกรรมของจีโนมหนึ่งๆ เทียบกับจีโนมอ้างอิง
- เพื่อให้นิสิตคุ้นเคยกับข้อมูลที่เกี่ยวข้องและเข้าใจการพัฒนาวิธีการหาสตริงย่อยในสตริงหลักแบบเหมือนทั้งเส้นโดยเน้นความเร็วและการใช้หน่วยความจำที่มีประสิทธิภาพ
- เพื่อให้นิสิตได้เห็นตัวอย่างงานวิจัยและผลงานวิจัย รวมทั้งตัวอย่างโปรแกรมที่ใช้ในหาสตริงย่อยในสตริงหลัก
- เพื่อให้นิสิตได้เห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทาย รวมทั้งงานวิจัยอื่นๆ ที่เกี่ยวข้อง

ผลลัพธ์ที่คาดหวัง

- นิสิตสามารถอธิบายความแตกต่างรวมทั้งข้อดีข้อเสียระหว่างแพลตฟอร์มที่ใช้ในการถอดรหัสจีโนมได้
- นิสิตเข้าใจคุณลักษณะของข้อมูลตั้งต้นที่ได้จากการถอดรหัสจีโนม
- นิสิตสามารถอธิบายการทำงานของอัลกอริทึมหลักๆ ที่ใช้ในการหาสตริงย่อยในสตริงหลักได้
- นิสิตสามารถเขียนโปรแกรมที่ใช้ในการสตริงย่อยในสตริงหลักอย่างง่ายได้
- นิสิตสามารถยกตัวอย่างโปรแกรมที่ใช้ในการเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิงที่มีการใช้งานกันอย่างแพร่หลายได้
- นิสิตสามารถยกตัวอย่างความท้าทายที่ยังมีอยู่และสามารถนำเสนอแนวทางในการพัฒนาวิธีการแก้ปัญหาเหล่านี้ได้ รวมทั้งสามารถประยุกต์องค์ความรู้จากบทเรียนเพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้

เนื้อหาโดยสรุป

ในบทที่ 2 เราพูดถึงการถอดรหัสจีโนมและการประกอบร่างจีโนมจากดีเอ็นเอสายสั้นมากมาย โดยการประกอบร่างจีโนมนี้เน้นการทำแบบ *de novo* คือสร้างจีโนมขึ้นมาจากการประกอบดีเอ็นเอสายสั้นๆเหล่านั้นเข้าด้วยกันโดยไม่มีจีโนมอ้างอิง ในบทนี้จะแตกต่างจากบทที่ 2 โดยมีสมมติฐานว่ามีจีโนมอ้างอิงอยู่แล้วตัวอย่างเช่น จีโนมมนุษย์ เป็น

ต้น โจทย์คือถ้ามีดีเอ็นเอสายสั้นมากมายจากการถอดรหัสพันธุกรรมจะเอาดีเอ็นเอสายสั้นเหล่านี้ไปเทียบกับจีโนมอ้างอิงว่าตรงกับบริเวณไหนที่สุด โดยจะทำให้การเทียบนี้เร็วและมีประสิทธิภาพสูงสุดได้อย่างไร ผลการเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิง (short read mapping/alignment) นี้นอกจากจะเป็นทางเลือกในการประกอบร่างจีโนมโดยมีสมมติฐานว่ามีจีโนมอ้างอิงแล้ว ยังสามารถใช้เป็นข้อมูลพื้นฐานในการวิเคราะห์ความแปรผันของดีเอ็นในลักษณะต่างๆ เช่น เอสเอ็นวี (Single Nucleotide Variant: SNV) ที่เกิดความแปรผันในลำดับเบสเดี่ยวๆ การแปรผันในจำนวนซ้ำของบางบริเวณในจีโนม (Copy Number of Variation: CNV) ความแปรผันในเชิงโครงสร้างของจีโนม (structural variation) ในลักษณะอื่นๆ เช่นเกิด การกลับด้านของลำดับเบส (inversion) ในรหัสพันธุกรรมของผู้ป่วยเทียบกับจีโนมอ้างอิง ความแปรผันเมื่อเทียบระหว่างจีโนมของบุคคลภายในครอบครัวเช่นพ่อแม่ ลูก (Trio) ความแปรผันในระดับกลุ่มประชากร ความแปรผันระหว่างจีโนมของเซลล์ปกติและเซลล์มะเร็ง เป็นต้น ตัวอย่างอัลกอริทึมที่เกี่ยวข้องกับการเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิง เช่น ไทร์ (trie) ซัฟฟิกซ์ทรี (suffix trees) ซัฟฟิกซ์อะเรย์ (suffix arrays) แนวคิดในเรื่องการบีบอัดสตริง (string compression) และ Burrows-Wheeler Transform (BWT) ตัวอย่างโปรแกรมที่มีการใช้งานกันอย่างแพร่หลาย และงานวิจัยที่เกี่ยวข้อง

บทที่ 3 การเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิง (Short read mapping to reference genome)

ประมาณ 1% ของทารกแรกแรกเกิดจะมีปัญหาความบกพร่องด้านสติปัญญา (mental retardation) ซึ่งความบกพร่องนี้สามารถเกิดความผิดปกติของพันธุกรรมที่หลากหลายจึงยังไม่ทราบสาเหตุที่ชัดเจน ตัวอย่างโรคเช่น Ohdo syndrome ที่ผู้ป่วยจะไม่สามารถแสดงออกทางสีหน้าได้ (expressionless) หรือมีหน้าที่เหมือนใส่หน้ากาก (mask-like) ในปีค.ศ. 2011 นักชีววิทยาได้ทราบสาเหตุทางพันธุกรรมของโรคจากข้อมูลการกลายพันธุ์ (mutations) ในตำแหน่งต่างๆ ที่เกิดร่วมกันในผู้ป่วยซึ่งนักวิจัยได้ค้นพบว่าสาเหตุหลักของโรค Ohdo syndrome ซึ่งเกิดจากการกลายพันธุ์ในระดับดีเอ็นเอในบริเวณหนึ่งที่มีผลต่อการแปลรหัสไปเป็นโปรตีนที่ไม่สมบูรณ์ [61] โดยการแปลรหัสเพื่อสร้างสายโปรตีนจะหยุดการแปลเร็วกว่าปกติ อีกตัวอย่างคือกรณีเด็กชายอายุ 6 ปีชื่อนิโคลาส วอลเกอร์ (Nicolas Volker) ที่เกิดอาการอักเสบลำไส้อย่างรุนแรง โดยแพทย์ไม่สามารถที่มาของอาการและโรค แพทย์ทำการรักษาตามอาการโดยการผ่าตัดลำไส้หลายครั้งจนกระทั่งโรงเรียนแพทย์ที่วิสคอนซินทำการถอดรหัสดีเอ็นเอของนิโคลาสและทราบสาเหตุของอาการลำไส้อักเสบรุนแรงซึ่งเกิดจากการกลายพันธุ์ของยีนชื่อ XIAP (X-linked inhibitor of apoptosis) และนำไปสู่ความผิดปกติของระบบภูมิคุ้มกัน หลังจากทราบสาเหตุที่แท้จริงแล้ว แพทย์ได้ทำการรักษาโดยวิธีการรักษาด้วยภูมิคุ้มกันบำบัด (immunotherapy) ซึ่งทำให้สามารถรักษาชีวิตของนิโคลาสไว้ได้ การหาบริเวณที่เกิดความแปรผัน (variation) ในจีโนมของบุคคลหนึ่งๆ เทียบกับจีโนมอ้างอิง และความแปรผันที่อาจเป็นการกลายพันธุ์ (mutation) คือมีผลต่อการเกิดโรคหรือความรุนแรงของโรค สามารถนำไปสู่การวิฉัยโรคและวิธีการรักษาที่ตรงจุด ด้วยเทคโนโลยีการถอดรหัสพันธุกรรมที่มีอยู่ การหาความแปรผันในจีโนมทำได้โดยการนำข้อมูลรหัสพันธุกรรมในรูปแบบดีเอ็นเอสายสั้น (รีด) จำนวนมากของผู้ป่วยไปเทียบกับจีโนมอ้างอิงและดูว่ามีบริเวณใดบ้างที่มีความแตกต่างกัน ซึ่งความแปรผันมีได้หลายรูปแบบไม่ว่าจะเป็นการแปรผันเพียงหนึ่งเบสหรือเบสเดี่ยวๆในบริเวณหนึ่งๆของจีโนม เรียกว่าเอสเอ็นวี (Single Nucleotide Variant: SNV) การแปรผันโดยมีการชุดของลำดับเบสสั้นๆเพิ่มเติมหรือหายไปเมื่อเทียบกับจีโนมอ้างอิง (insertions/deletions: indels อ่านว่าอินเดล) การแปรผันโดยมีชุดของลำดับเบสกลับด้านกับจีโนมอ้างอิง (inversion) การแปรผันโดยมีชุดของลำดับเบสย้ายจากโครโมโซมหนึ่งไปยังอีกโครโมโซมหนึ่ง (translocation) การแปรผันของจำนวนชุดดีเอ็นเอที่แตกต่างจากจีโนมอ้างอิง (Copy Number of Variations: CNVs อ่านว่าซีเอ็นวี) เป็นต้น ซึ่งในบทเรียนนี้เราจะเน้นการหาความแปรผันในลำดับเบสเดี่ยวๆ โดยเริ่มจากการแนะนำอัลกอริทึมต่างๆ ที่ถูกออกแบบและพัฒนามาใช้ในการค้นหาสตริงสายย่อยซึ่งเป็นตัวแทนของดีเอ็นสายสั้นหรือที่เรียกว่ารีด ในสตริงสายหลักซึ่งเป็นตัวแทนของจีโนมอ้างอิง โดยเน้นการหาแบบที่สตริงย่อยทั้งสายเหมือนกับบริเวณใดๆ ในสตริงสายหลัก (exact match) และในตอนท้ายจะแสดงตัวอย่างวิธีการหาเอสเอ็นวีโดยวิธีการหาสตริงย่อยในสตริงหลักแบบโดยประมาณ

ปัญหาการเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิง

การเทียบดีเอ็นเอสายสั้นจำนวนมากกับจีโนมอ้างอิงนี้เป็นตัวอย่างปัญหาการหาชุดของสตริงย่อยในสายสตริงหลัก ซึ่งถูกนิยามดังแสดงไว้ในนิยามปัญหาที่ 3.1 ต่อไปนี้

นิยามปัญหาที่ 3.1 ปัญหาการหาชุดของสตริงย่อยในสายสตริงหลัก

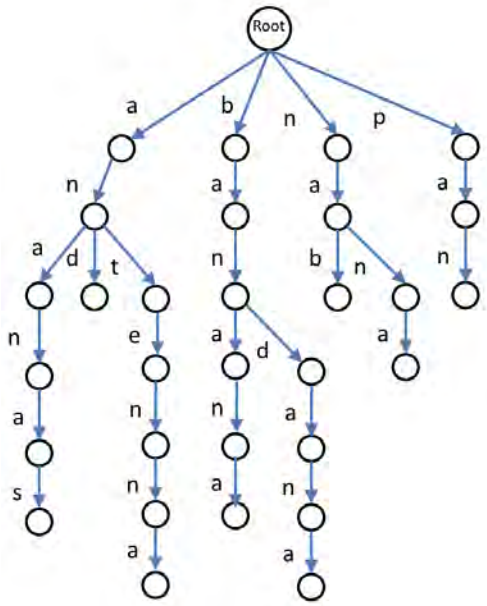
ปัญหาการหาชุดของสตริงย่อยในสายสตริงหลัก (Multiple Pattern Matching Problem) หาตำแหน่งที่พบทั้งหมดของแต่ละสตริงย่อยในสายสตริงหลัก	
ข้อมูลเข้า	สายสตริงหลักและชุดของสตริงย่อย
ผลลัพธ์	ตำแหน่งเริ่มต้นทั้งหมดในสตริงหลักที่พบสตริงย่อยแต่ละรูปแบบ

วิธีการแก้ปัญหาการหาชุดของสตริงย่อยในสายสตริงหลักแบบ Brute force

หลักการพื้นฐานของแนวทาง Brute force สำหรับการหาชุดของสตริงย่อยในสายสตริงหลักจะเป็นลักษณะของการนำแต่ละสตริงย่อยมาเทียบหาบริเวณที่เหมือนที่สุดในสตริงหลัก โดยแต่ละสตริงย่อยเป็นอิสระต่อกัน ถ้าความยาวของสายสตริงหลักคือ |Text| และความยาวของแต่ละสตริงย่อยเท่ากับ |Pattern| เวลาที่ใช้ในการหาสตริงย่อยนั้นๆ ในสตริงหลักจะเท่ากับ O(|Text| * |Pattern|) และถ้าหาชุดของสายสตริงโดย |Patterns| คือความยาวของแต่ละสตริงย่อยทั้งหมดรวมกัน จะได้เวลาที่ใช้เป็น O(|Text| * |Patterns|) ถ้านำวิธีการ Brute force นี้ไปใช้ในการหาว่าแต่ละรีดที่อ่านมาได้จากเครื่องถอดรหัสพันธุกรรมอยู่ตรงไหนในจีโนมอ้างอิงจะใช้เวลานานมาก โดยความยาวรวมของทุกรีด (เทียบได้กับ |Patterns|) จะประมาณ 1 TB (Terabytes) ในขณะที่ความยาวของจีโนมอ้างอิงจะประมาณ 3 GB (Gigabytes)

วิธีการแก้ปัญหาการหาชุดของสตริงย่อยในสายสตริงหลักโดยใช้ไทร์ (Trie)

แนวทางที่สองในการหาชุดของสตริงย่อยในสายสตริงหลัก อัลกอริทึมจะถูกออกแบบให้อ่านผ่านสายสตริงหลักเพียงครั้งเดียว โดยถ้าวิธีการ Brute force เปรียบได้กับการเอาแต่ละสายสตริงย่อยใส่รถยนต์ส่วนตัวแล้ววิ่งไปตามถนนสายสตริงหลักโดยแต่ละสตริงย่อยมีรถยนต์เป็นของตัวเอง ในขณะที่วิธีการที่สองนี้จะเหมือนกับการบรรทุกชุดของของสายสตริงย่อยทั้งหมดลงในรถโดยสารคันเดียวแล้วใช้รถโดยสารนี้วิ่งไปตามถนนสายสตริงหลักเพียงครั้งเดียว โดยโครงสร้างข้อมูลที่ถูกนำมาบรรจุชุดของสตริงย่อยทั้งหมดเข้าด้วยกันนี้เรียกว่าไทร์ (Trie) ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 ตัวอย่างไทร์ (Trie) ที่มีชุดของสตริงย่อยประกอบด้วย “ananas”, “and”, “antenna”, “banana”, “bandana”, “nab”, “nana” และ “pan”

(ที่มา: รูปที่ 9.1 ของ [21])

ปัญหาที่ 3.2 ปัญหาการสร้างไทร์จากชุดของสตริงย่อย

ปัญหาการสร้างไทร์จากชุดของสตริงย่อย	
ข้อมูลเข้า	ชุดของสตริงย่อย
ผลลัพธ์	ไทร์ของชุดของสตริงย่อย

วิธีการสร้างไทร์จากชุดของสตริงย่อย

สูโดโคดที่ 3.1 TrieConstruction()

```
1  TrieConstruction(Patterns)
2      Trie <- กราฟที่มีโหนดเดียวคือรูทโหนด
3      for แต่ละสตริงย่อย pattern ใน Patterns
4          currentNode <- root ของ Trie
5          for แต่ละตัวอักษร c ใน pattern
6              if มีเส้นเชื่อมที่มีค่าเป็น c จาก currentNode ไปโหนด n
7                  เปลี่ยน currentNode ไปที่โหนด n
8              else
9                  เพิ่มโหนดใหม่ใน Trie
10                 เพิ่มเส้นเชื่อมชี้จาก currentNode มาที่โหนดใหม่นี้
11                     และใส่ค่าเส้นเชื่อมนี้เป็น c
12                 เปลี่ยน currentNode มาที่โหนดใหม่นี้
13     ส่งกลับ Trie
```

หยุดคิด	ไทร์ที่สร้างขึ้นนี้จะถูกนำไปค้นหาตำแหน่งของสตริงย่อยเหล่านี้ในสตริงหลักอย่างไร

ไทร์ที่สร้างขึ้นสามารถนำมาหาสตริงย่อยในสายของสตริงหลักได้ ดังแสดงในสูโดโคดที่ 3.2 `Prefix TrieMatching()`โดยมีหลักการทำงานคือเริ่มอ่านจากอักขณะแรกของสายสตริงหลักเทียบกับค่าของแต่ละเส้นเชื่อมจากโหนดราก (โหนดตั้งต้นในไทร์) ถ้าพบก็อ่านอักขระถัดไปจากสายสตริงหลัก พร้อมทั้งเปลี่ยนค่าโหนดตั้งต้นในไทร์เป็นโหนดที่ถูกชี้โดยเส้นเชื่อมที่มีค่าตรงกับอักขระที่อ่านมาได้ก่อนหน้า และวนกลับไปทดสอบว่ามีเส้นเชื่อมจากโหนดตั้งต้นใหม่นี้ที่มีค่าตรงกับอักขระล่าสุดที่อ่านมาได้จากสตริงหลักหรือไม่ วนซ้ำการทำงานนี้ และส่งกลับค่าเส้นทางจากโหนดรากถึงโหนดใบ (สตริงย่อยที่พบ) ถ้าสามารถหาเส้นทางนั้นๆได้

วิธีการหาชุดของสตริงย่อยในสายสตรงหลักโดยวิธีการ prefix trie matching

สูโดโคดที่ 3.2 PrefixTrieMatching()

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

เนื่องจาก `PrefixTrieMatching()` จะตรวจสอบสายสตริงหลักโดยเริ่มอ่านจากอักขระแรกเสมอ ดังนั้นเพื่อให้สามารถนำ `PrefixTrieMatching()`มาใช้ในการหารูปแบบของสตริงย่อยทั้งหมดที่พบในสตริงหลักก็สามารถทำได้โดยนำ `PrefixTrieMatching()`มาสแกนสายสตริงหลักโดยที่ในแต่ละรอบของการทำงาน สายสตริงหลักที่เป็นข้อมูลเข้าของ `PrefixTrieMatching()` จะมีขนาดสั้นลงโดยอักขระทางซ้ายมือสุดจะถูกตัดออกไป ดังสูโคด 3.3 ต่อไปนี้

สูโดโคดที่ 3.3 `TrieMatching()`

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

ถ้าวิเคราะห์เวลาที่ใช้ในการหาชุดของสตริงย่อยในสตริงหลักโดยใช้ไทร์แล้วจะพบว่าจะใช้เวลา 2 ส่วน ส่วนแรกเป็นเวลาในการสร้างไทร์ซึ่งเท่ากับ O(|Patterns|) ตามความยาวรวมของสตริงย่อยทั้งหมด ส่วนที่สองเป็นส่วนของการใช้ไทร์ในการค้นหาสตริงย่อยทั้งหมดในสตริงหลัก `TrieMatching()` ซึ่งใช้เวลาO(|Text| * |Longest pattern|) โดย |Text| คือความยาวของสตริงหลักและ |Longest Pattern| คือความยาวของสตริงย่อยที่ยาวที่สุด ในขณะที่เวลาที่ใช้ในกรณีของ Brute force เท่ากับ O(|Text| * |Patterns|)

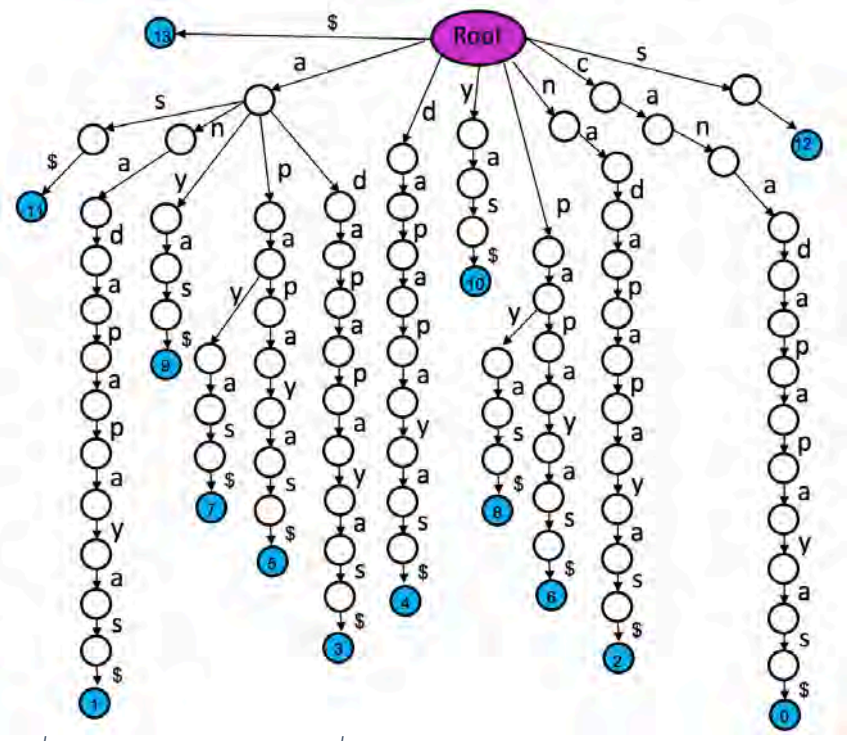
หยุดคิด	ดูเหมือนว่าเราสามารถใช้ไทร์เป็นตัวช่วยในการหาชุดของสตริงย่อยในสตริงหลักได้เร็วกว่าวิธีการ Brute force มาก คำถามคือการใช้ไทร์มีข้อจำกัดอะไรบ้างหรือไม่

ถึงแม้ว่าการใช้ไทร์จะทำให้การหาสตริงย่อยทำได้เร็วขึ้นมากแต่ก็ต้องใช้หน่วยความจำเป็นจำนวนมากในการเก็บโครงสร้างข้อมูลไทร์ ถ้าใช้ไทร์ในการเก็บข้อมูลรีดทั้งหมดที่ได้จากการถอดรหัสจีโนมมนุษย์อาจต้องใช้หน่วยความจำถึง 1 TB

วิธีการหาชุดของสตริงย่อยในสายสตริงหลักโดยใช้ซัฟฟิกซ์ไทร์

เนื่องจากการสร้างไทร์จากชุดของสตริงย่อยหรือรีดทั้งหมดที่อ่านได้จากเครื่องถอดรหัสพันธุกรรมจะได้ไทร์ที่มีขนาดใหญ่มากซึ่งจะต้องใช้หน่วยความจำจำนวนมาก จึงได้มีแนวความคิดในการสร้างไทร์จากชุดของซัฟฟิกซ์ทั้งหมดที่เป็นไปได้ของข้อมูลสตริงสายหลักแทน และเมื่อต้องการหาชุดของสตริงย่อยในสตริงสายหลักก็นำสตริงสายย่อยเหล่านี้มาเทียบกับซัฟฟิกซ์ไทร์ (suffix trie) ของสตริงสายหลัก รูปที่ 3.2 แสดงตัวอย่างของซัฟฟิกซ์ไทร์ ที่สร้างจากสตริงสายหลัก “canadapapayas$” ซึ่งจะมีการใส่โหนดใบปิดท้ายของแต่ละซัฟฟิกซ์ในไทร์เป็นดัชนีของตัวอักขระแรกของซัฟฟิกซ์นั้นๆ ซึ่งในกรณีนี้มีทั้งหมด 13 ซัฟฟิกซ์ ตัวอย่างเช่น ซัฟฟิกซ์ “canadapapayas$” ตัวอักขระแรกคือตัวอักษร c มีดัชนีที่ 0 ซัฟฟิกซ์ “anadapapayas$” ตัวอักขระแรกคือตัว a มีดัชนีที่ 1 เป็นต้น การหาสตริงย่อยอย่าง “yas” จะสามารถเทียบลำดับอักขระจนพบโหนดใบ มีค่าเป็น 10 ซึ่งหมายถึงพบคำว่า “yas” ในสายสตริงหลักโดยตำแหน่งคำนับจากตัวอักษรแรก (y) อยู่ในตำแหน่งที่ 10 (ใช้ดัชนีเริ่มที่ 0) ถ้าค้นหาคำว่า “apa” จะพบทางแยกเป็นสองเส้นทาง ซึ่งหมายถึงพบ “apa” 2 ตำแหน่งในสายสตริงหลักคือตำแหน่งที่ 5 และ 7 ตามลำดับ

การสร้างสตริงหลักเป็นซัฟฟิกซ์ไทร์แทนการสร้างไทร์จากสตริงสายย่อยทั้งหมด จะได้จำนวนซัฟฟิกซ์ทั้งหมดเท่ากับความยาวของสตริงหลัก (ในตัวอย่างสตริงหลัก canadapapayas$ มีทั้งหมด 14 ซัฟฟิกซ์โดยรวมสตริงว่างด้วย) และถ้าสายสตริงหลักมีขนาด |Text| จำนวนโหนดที่แทนซัฟฟิกซ์ทั้งหมดจะเท่ากับ |Text| * (|Text|-1)/2 = $O(|Text|^2)$

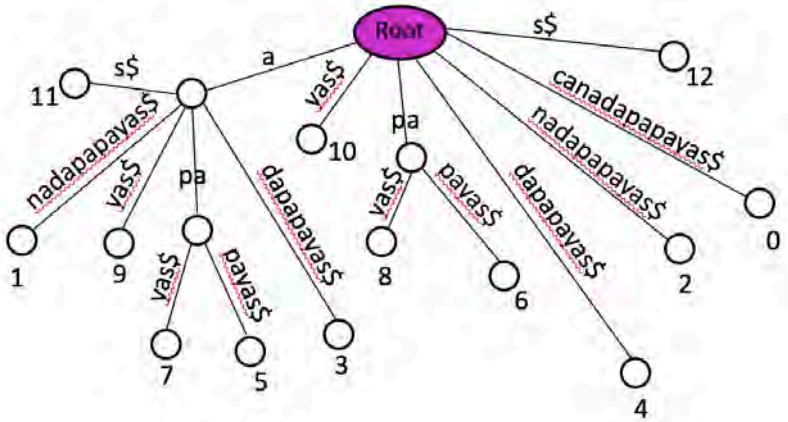


รูปที่ 3.2 ตัวอย่างซัฟฟิกซ์ไทร์ที่สร้างจากสตริงสายหลัก "canadapapayas$"

(ที่มา: ดัดแปลงจากรูปที่ 9.3 ของ [21])

วิธีการหาชุดของสตริงย่อยในสายสตริงหลักโดยใช้ซัฟฟิกซ์ทรี

ซัฟฟิกซ์ทรี (suffix tree) เป็นโครงสร้างข้อมูลที่ลดจำนวนโหนดในซัฟฟิกซ์ไทร์ลงโดยเส้นเชื่อมทั้งหมดที่ไม่มีทางแยกจะถูกรวมเข้าด้วยกันกลายเป็นเส้นเชื่อมเดียวและมีค่าของเส้นเป็นชุดของอักขระที่ต่อกันแทน ในการหาสตริงสายย่อยในสตริงสายหลักก็จะมีวิธีการเดียวกันกับการใช้ซัฟฟิกซ์ไทร์ รูปที่ 3.3 แสดงตัวอย่างของซัฟฟิกซ์ทรีที่สร้างจากซัฟฟิกซ์ไทร์ในรูปก่อนหน้า



รูปที่ 3.3 ตัวอย่างซัฟฟิกซ์ทรีที่สร้างจากสตริงสายหลัก "canadapapayas$"

(ที่มา: ดัดแปลงจากรูปที่ 9.6 ของ [21])

ในขณะที่ซัฟฟิกซ์ไทร์อาจมีจำนวนโหนดในไทร์เป็นตัวเลขสมการกำลังสอง (quadratic) ของความยาวของสตริงสายหลัก ในกรณีของซัฟฟิกซ์ทรีจำนวนของโหนดในทรีมีอย่างมากที่สุด 2 * |Text| (มาจากจำนวนโหนดใบเท่ากับ |Text| และโหนดภายในอีกอย่างมาก |Text| โหนด) ดังนั้นจะใช้หน่วยความจำอย่างมาก O(|Text|) อย่างไรก็ตาม การเก็บซัฟฟิกซ์ทรีตามตัวอย่างข้างต้น อาจจะใช้หน่วยความจำไม่ต่างจากการเก็บซัฟฟิกซ์ไทร์ เพราะยังต้องเก็บค่าของเส้นเชื่อมตามจำนวนไบต์ของอักขระที่นำมาต่อกัน อย่างไรก็ตามในทางปฏิบัติเราไม่จำเป็นต้องเก็บสายของสตริงย่อยที่เป็นค่าของแต่ละเส้นเชื่อมไว้แต่สามารถเก็บใช้ตัวชี้ (pointer) ของตำแหน่งเริ่มต้นและความยาวของสตริงย่อยนั้นๆแทน นอกจากนี้เรายังสามารถสร้างซัฟฟิกซ์ทรีได้โดยตรงใช้เวลาเป็นสมการเส้นตรงโดยไม่ต้องสร้างจากซัฟฟิกซ์ไทร์อีกที

อย่างไรก็ตามถึงแม้ว่าซัฟฟิกซ์ทรีต้องการหน่วยความจำลดลงมากจาก $O(|Text|^2)$ เป็น O(|Text|) โดยเฉลี่ยเมื่อเทียบกับซัฟฟิกซ์ไทร์ ในเชิงปฏิบัติซัฟฟิกซ์ทรียังต้องการหน่วยความจำประมาณ 20 เท่าของ |Text| ดังนั้นถ้าต้องการสร้างซัฟฟิกซ์ทรีของจีโนมมนุษย์ (สตริงสายหลักที่เป็นจีโนมอ้างอิง) ขนาด 3 GB จะต้องใช้หน่วยความจำประมาณ 60 GB ซึ่งก็ดีขึ้นมากเมื่อเทียบกับการสร้างไทร์จากรีดทั้งหมดที่ได้จากการถอดรหัสจีโนมซึ่งใช้หน่วยความจำประมาณ 1 TB ก่อนที่จะศึกษาแนวทางในการลดหน่วยความจำลงไปอีก ขอให้ลองแก้ปัญหาต่อไปนี้โดยใช้ซัฟฟิกซ์ทรี

ฝึกหัด

ปัญหาการหาสตริงย่อยซ้ำที่ยาวที่สุด (Longest Repeat Problem)

หาสตริงย่อยที่เหมือนกันที่ยาวที่สุดและพบในสายสตริงหลักมากกว่า 1 ตำแหน่ง

ข้อมูลเข้า	สายสตริงหลัก
ผลลัพธ์	สตริงย่อยที่เหมือนกันที่ยาวที่สุดและพบในสายสตริงหลักมากกว่า 1 ตำแหน่ง

ฝึกหัด

ปัญหาการหาสตริงย่อยที่เหมือนกันที่ยาวที่สุดและพบในสตริงหลักทั้งสองเส้น

(Longest Shared Substring Problem)

หาสตริงย่อยที่เหมือนกันที่ยาวที่สุดและพบในสายสตริงหลักทั้งสองเส้น

ข้อมูลเข้า	สายสตริงหลัก 2 เส้น
ผลลัพธ์	สตริงย่อยที่เหมือนกันที่ยาวที่สุดและพบในสายสตริงหลักทั้งสองเส้น

ฝึกหัด	
ปัญหาการหาสตริงย่อยที่สั้นที่สุดที่พบในสตริงหลักเส้นเดียว (Shortest Non-Shared Substring Problem) หาสตริงย่อยที่สั้นที่สุดและพบในสายสตริงหลักเส้นที่ 1 เท่านั้น	
ข้อมูลเข้า	สายสตริงหลัก 2 เส้น
ผลลัพธ์	สตริงย่อยที่สั้นที่สุดที่พบในสายสตริงหลักเส้นที่ 1 เท่านั้น

วิธีการหาชุดของสตริงย่อยในสายสตริงหลักโดยใช้ซัฟฟิกซ์อะเรย์

วิธีการสร้างซัฟฟิกซ์อะเรย์นั้นจะต้องทำการเรียงลำดับซัฟฟิกซ์ทั้งหมดที่มีอยู่ (รูปที่ 3.2) ตามลำดับตัวอักษรในพจนานุกรมดังในรูปที่ 3.4 โดยถือว่าอักขระ $ เป็นอักขระแรกของอักขระที่เป็นตัวอักษร โดยซัฟฟิกซ์อะเรย์จะเก็บลิสต์ของดัชนีตัวอักษรแรกของแต่ละซัฟฟิกซ์ที่มีการเรียงลำดับแล้ว ดังแสดงในบรรทัดต่อไปนี้

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

ทั้งนี้ซัฟฟิกซ์อะเรย์สามารถสร้างโดยง่ายหลังจากได้ชุดของซัฟฟิกซ์ที่มีเรียงลำดับแล้ว โดยอัลกอริทึมเรียงข้อมูลที่เร็วที่สุดยังต้องมีการเปรียบเทียบค่าระหว่างข้อมูล O(n log n) ครั้ง ซึ่งก็คือต้องมีการเปรียบเทียบ O(|Text| log |Text|) ครั้ง อย่างไรก็ตามก็มีอัลกอริทึมที่สามารถสร้างซัฟฟิกซ์อะเรย์โดยใช้เวลาในสมการเชิงเส้นและต้องการหน่วยความจำประมาณ 1/5 ของหน่วยความจำที่ใช้ในการเก็บซัฟฟิกซ์ทรีซึ่งก็จะใช้หน่วยความจำประมาณ 12 GB สำหรับเก็บซัฟฟิกซ์อะเรย์ของจีโนมมนุษย์

The Burrows-Wheeler Transform

การใช้ซัฟฟิกซ์อะเรย์ลดความต้องการใช้หน่วยความจำลงไปเป็นอย่างมากและเป็น state of the art ในการทำ pattern matching จนกระทั่งต้นคริสตศักราช 2000 ก็มีคำถามว่ามีโครงสร้างข้อมูลอะไรไหมที่จะสามารถเข้ารหัสสตริงสายหลัก (Text) โดยใช้หน่วยความจำที่มีขนาดประมาณขนาดของสตริงสายหลักได้

ก่อนที่จะตอบปัญหานี้ลองพิจารณาการบีบอัดสายสตริง (text compression) โดยใช้การเข้ารหัสแบบ run-length encoding ดูก่อน การเข้ารหัสในวิธีการนี้จะแทนที่ลำดับของสายอักขระที่เป็นอักขระเดียวกันเรียกว่ารัน (run) ด้วยจำนวนซ้ำที่พบกับอักขระตัวนั้นๆ ตัวอย่างเช่น **TTTTTGGGAAAACCCCCCA** จะถูกแทนที่ด้วย **5T3G4A6C1A** เป็นต้น การเข้ารหัสแบบ run-length นี้จะมีประสิทธิภาพถ้าในสายสตริงนั้นมีชุดของอักขระซ้ำยาวๆ จำนวนมาก อย่างไรก็ตามในกรณีของจีโนมมนุษย์ไม่ได้มีชุดอักขระซ้ำจำนวนมาก แต่จะมีความซ้ำของชุดอักขระเป็นชุดๆเรียกว่า รีพีท (repeat) แทรกในสตริงหลักหรืออยู่ต่อเนื่องกันไป 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 รายการของซัฟฟิกซ์ทั้งหมดของสตริงหลัก "canadapapayas$" ที่มีการเรียงลำดับตามตัวอักษรแล้ว (โดยถือว่าอักขระ $ มาเป็นลำดับแรก) และตำแหน่งเริ่มต้นของซัฟฟิกซ์นั้นๆ ที่พบในสตริงหลัก

(ที่มา: ดัดแปลงจากรูปที่ 9.7 ของ [16])

ลองพิจารณาการสร้างรันโดยวิธีนำอักขระในสายสตริงหลัก อย่างเช่น **TACGTAACGATACGAT** มาเรียงลำดับตามตัวอักษรได้เป็น **AAAAACCCGGGTTTT** ซึ่งเข้ารหัสเป็น **5A3C3G4T** ซึ่งวิธีการนี้จะแสดงจีโนมมนุษย์ที่มีขนาด 3 GB โดยใช้เพียงตัวเลข 4 ตัว

หยุดคิด	วิธีการเข้ารหัสจีโนมข้างต้นมีข้อผิดพลาดอย่างไร

การนำอักขระทั้งหมดมาเรียงลำดับตามตัวอักษรแล้วจากนั้นนับจำนวนซ้ำ ใช้ไม่ได้กับการบีบอัดข้อมูล เพราะว่าสตริงที่แตกต่างกัน เช่น **GCATCATGCAT** และ **ACTGACTACTG** ที่มีชุดและจำนวนของอักขระเท่ากันแต่มีลำดับของตัวอักษรแตกต่างกันจะถูกเรียงลำดับเป็น **AAACCCGGTTT** เหมือนกันและบีบอัดออกมาเป็นสตริงเดียวกันคือ **3A3C2G3T** เป็นต้น ซึ่งไม่สามารถแตก (decompress) สตริงที่ถูกบีบอัดนี้ออกมาเป็นสตริงต้นฉบับที่ถูกต้องได้ (เพราะไม่รู้ว่าตกลงมาจากต้นฉบับไหน)

การสร้าง Burrows-Wheeler Transform

ลองพิจารณาอีกวิธีการในการแปลงรีพีทต่างๆ ให้เป็นรัน ซึ่งถูกนำเสนอโดย ไมเคิล เบอร์โรวส์ (Michael Burrows) และเดวิด วีลเลอร์ (David Wheeler) ในปีค.ศ. 1994 โดยวิธีการนี้เริ่มจากขั้นแรกทำการสร้างลำดับของอักขระในสายของสตริงหลักโดยการทำ cyclic rotation โดยตัดอักขระตัวขวาสุดไปเพิ่มไว้ที่ตำแหน่งซ้ายสุดและเลื่อนสาย

สายอักขระที่เหลือไปทางขวา 1 ตำแหน่ง ทำอย่างนี้ไปเรื่อยๆ จนกว่าจะการหมุนสตริงนี้จะได้ผลกลับมาเป็นสตริงหลักตั้งต้น ขั้นที่สองทำการเรียงลำดับสายอักขระที่เกิดจากการหมุนรอบละ 1 ตัวอักขระนี้ตามลำดับพจนานุกรม ได้เป็นเมทริกซ์ของเบอร์โรวส์-วีลเลอร์ โดยแสดงด้วย `M(Text)` ดังแสดงในรูป 3.5 โดยตัวอย่าง Text ในรูปนี้คือ “canadapapayas$” และได้ Burrows-Wheeler Transform หรือ `BTW(Text)` เป็น “sncdpyp$aaaaaa” (คอลัมน์ขวาสุดของเมทริกซ์)

ผลการหมุนสายสตริงหลักทีละ 1 อักขระ	ผลการเรียงสายสตริงที่ได้ทางซ้ายตามตัวอักษร 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

รูปที่ 3.5 (ซ้าย) แสดงผลการหมุนสายสตริงหลัก “canadapapayas$” (ขวา) เมทริกซ์เบอร์โรวส์-วีลเลอร์ที่เป็นผลของการเรียงสายสตริงทั้งหมดที่เกิดจากการหมุน โดยคอลัมน์ขวาสุด คือ Burrow-Wheeler Transform

นิยามปัญหาที่ 3.3 ปัญหาการสร้าง Burrows-Wheeler Transform

ปัญหาการสร้าง Burrows-Wheeler Transform สร้าง Burrows-Wheeler Transform จากสายสตริง	
ข้อมูลเข้า	สายสตริง
ผลลัพธ์	BWT(สายสตริง)

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 ใน **BWT("canadapapayas$") = "**sncdpyp$aaaaaa**"** จะสังเกตเห็นว่ามีรัน เช่น "aaaaaa" เกิดขึ้น คำถามคือทำไม Burrows-Wheeler Transform ถึงมีรันนี้เกิดขึ้น ลองจินตนาการว่าถ้าเราเอาคำทั้งหมดในผลงานตีพิมพ์ในปีค.ศ. 1958 ของวัตสันและคริกเกี่ยวกับดีเอ็นเอสายคู่มาสร้างเมทริกซ์ของเบอร์โรวส์-วีลเลอร์ จะพบว่าคำว่า "and" บ่อยๆ ซึ่งถ้าพิจารณาการผลของการหมุนสตริงทั้งหมดในเมทริกซ์ของเบอร์โรวส์-วีลเลอร์ที่มีการเรียงลำดับสตริงแล้วจะพบว่าทุกบรรทัดของสตริงที่ขึ้นต้นด้วย "nd" คอลัมน์สุดท้ายของบรรทัดเดียวกันมักเป็นตัวอักษร "a" และบรรทัดเหล่านี้จะอยู่กันเป็นกลุ่ม ดังแสดงในรูปที่ 3.6 สตริงย่อยอย่าง "apa" ในสายสตริงหลัก "canadapapayas$" ก็เทียบได้กับคำว่า "and" ในตัวอย่างนี้ ซึ่งก็เป็นคำอธิบายของตัวอักษร "aaaaaa" 2 ใน 6 ตัว ในรีพีท "aaaaaa" ของ **BWT("canadapapayas$") = "**sncdpyp$**aaaaaa"** การประยุกต์ใช้ Burrows-Wheeler Transform กับจีโนมทำให้สามารถแปลงรีพีทต่างๆ ให้อยู่ในรูปแบบของรันได้ ซึ่งก็สามารถใช้วิธีการเข้ารหัสอย่าง 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) ที่ถูกเลือกออกมา โดย Text คือคำทั้งหมดที่ได้จากผลงานตีพิมพ์ของวัตสันและคริกเกี่ยวกับดีเอ็นเอสายคู่ในปีค.ศ. 1958 โดยบรรทัดที่ขึ้นต้นด้วย "nd" มักมีคอลัมน์สุดท้ายเป็น "a" ซึ่งเป็นที่มาของรันที่ปรากฏใน BWT(Text)

(ที่มา: รูปที่ 9.9 ของ [21])

การแปลง Burrows-Wheeler Transform กลับเป็นสายสตริงตั้งต้น

การบีบอัดข้อมูลจะไม่มีประโยชน์ถ้าไม่สามารถแปลงจากข้อมูลที่ถูกบีบอัดเป็นข้อมูลต้นฉบับที่ถูกต้องได้ อย่างไรก็ตามสายสตริงต้นฉบับที่ถูกต้องสามารถแปลงกลับมาจาก BTW ได้ พิจารณาตัวอย่างของ BWT(Text) = "ard$rcaaaabb" ในเมทริกซ์ของเบอร์โรวส์-วีลเลอร์ที่ทราบเฉพาะคอลัมน์ซ้ายสุดและขวาสุดต่อไปนี้

```
$??????????a
a??????????r
a??????????d
a??????????$
a??????????r
a??????????c
b??????????a
b??????????a
c??????????a
d??????????a
r??????????b
r??????????b
```

อักขระคอลัมน์ขวาสุดเรียงจากบนลงล่างคือ BWT ส่วนคอลัมน์ซ้ายสุดคืออักขระตัวแรกแต่ละสายสตริงที่ผ่านการหมุน 1 ตัวอักษรและมีการเรียงลำดับแล้ว คำถามคืออักขระแรกสุดของสตริงต้นฉบับคือตัวอะไร ซึ่งถ้าพิจารณาจากเมทริกซ์นี้ก็จะพบคือตัวอักษร “a” ซึ่งมาจากตัวอักษรซ้ายสุดของบรรทัดที่ 4 ที่มี “$” เป็นตัวอักษรทางขวาสุด

```
$a????????a
a??????????r
a??????????d
a??????????$
a??????????r
a??????????c
b??????????a
b??????????a
c??????????a
d??????????a
r??????????b
r??????????b
```

หยุดคิด	อักขระตัวถัดไปของสตริงตั้งต้นข้างต้นคือตัวอะไร

ใช้วิธีการคิดตามหลักการข้างต้นจะได้ว่าตัวอักขระถัดไปของสตริงต้นฉบับอาจเป็น “b”, “c”, หรือ “d” ก็ได้ ซึ่งเป็นตัวอักขระซ้ายสุดในบรรทัดที่ 7, 9, และ 10 ตามลำดับ เพราะทั้งสามบรรทัดมี “a” เป็นตัวอักขระขวาสุด ถ้า “a” โดยจะเป็น “b”, “c” หรือ “d” ขึ้นอยู่กับว่า “a” ตัวซ้ายสุดในบรรทัดที่ 4 นั้นเป็น a ตัวไหนใน BTW “ard$rcaaaabb” เช่นถ้า “a” นั้นเป็น “a” ตัวที่ 7 อักขระตัวที่สองในสตริงต้นฉบับก็จะเป็น “b” ถ้า “a” นั้นเป็นตัวที่ 9 อักขระตัวที่สองในสตริงต้นฉบับก็จะเป็น “c” และถ้า “a” เป็นตัวที่ 10 อักขระตัวที่สองในสตริงต้นฉบับก็จะเป็น “d” เป็นต้น คำถามคือเราจะทราบได้อย่างไรว่า “a” ที่เป็นตัวซ้ายสุดจากบรรทัดที่ 4 นี้ เป็น “a” ตัวไหนใน BTW

คุณสมบัติ First-Last

เพื่อให้สามารถตัดสินใจได้ว่าตัวอักขระที่เกิดซ้ำตัวหนึ่งๆนั้นเป็นตัวไหน ได้เพิ่มการบอกตำแหน่งที่ปรากฏของอักขระนั้นๆในคอลัมน์ซ้ายสุดจากบนลงล่าง ตามตัวอย่างต่อไปนี้ (แสดงเฉพาะของตัวอักขระ “a”) พิจารณา a_1 ที่เป็นอักขระซ้ายสุดในบรรทัดที่ 2 จากเฉลยของสตริงต้นฉบับที่มีอยู่ “$a_1$dapapayas$can” ถ้ามีการหมุนจนได้สตริงต้นฉบับจะได้ “cana$_1$dapapayas$” ซึ่ง a_1 อยู่ในลำดับที่ 2 ของ “a” ทั้งหกตัวที่ปรากฏในสตริงต้นฉบับ

$canadapapayas
$a_1$dapapayas$can
$a_2$nadapapayas$c
$a_3$papayas$canad
$a_4$payas$canadap
a_5s$canadapapay
$a_6$yas$canadapap
canadapapayas$
dapapayas$cana
nadapapayas$ca
papayas$canada
payas$canadapa
s$canadapapaya
yas$canadapapa

พิจารณาต่อว่า a_1 ที่อยู่คอลัมน์ซ้ายสุดในบรรทัดที่ 2 นี้เป็น “a” ไหนในคอลัมน์ขวาสุด ถ้ามีการหมุน a_1 ในบรรทัดนี้ไปทางขวา 1 ตำแหน่ง จะได้สตริงใหม่คือ “dapapayas$cana$_1$” ซึ่งตรงกับบรรทัดที่ 9 ดังในตัวอย่างต่อไปนี้

$canadapapayas
$a_1$dapapayas$can
$a_2$nadapapayas$c
$a_3$papayas$canad
$a_4$payas$canadap
a_5s$canadapapay
$a_6$yas$canadapap
canadapapayas$
dapapayas$cana$a_1$
nadapapayas$ca
papayas$canada
payas$canadapa
s$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” เหล่านี้อยู่ในลำดับเดียวกันกับลำดับที่อยู่ในคอลัมน์ซ้ายสุดดังแสดงในรูปข้างล้างนี้ โดยสามารถสรุปคุณสมบัติ First-Last ได้ว่าอักขระใดๆ ในคอลัมน์ซ้ายสุดที่อยู่ในลำดับ k ของสายสตริงต้นฉบับ อักขระตัวเดียวกันนั้นที่อยู่ในคอลัมน์ขวาสุดจะอยู่ในลำดับที่ k เดียวกันในสายสตริงต้นฉบับ

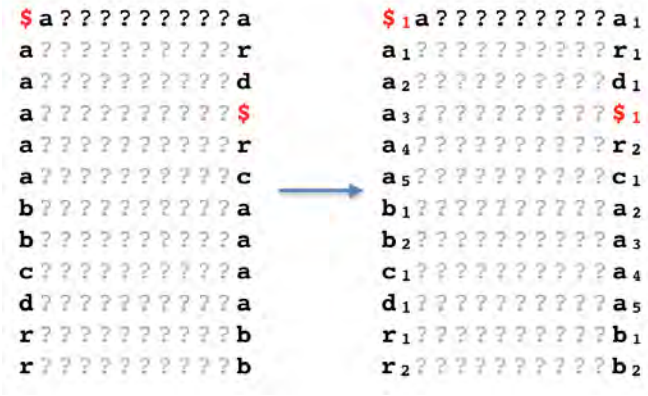
$canadapapayas
a1dapapayas$can
a2nadapapayas$c
a3papayas$canad
a4payas$canadap
a5s$canadapapay
a6yas$canadapap
canadapapayas$
dapapayas$cana1
nadapapayas$ca2
papayas$canada3
payas$canadapa4
s$canadapapaya5
yas$canadapapa6

เพื่อเป็นการอธิบายว่าทำไมคุณสมบัติ First-Last ข้างต้นถึงเป็นจริงพิจารณาเมทริกซ์ย่อยต่อไปนี้โดยเน้นเฉพาะบรรทัดที่มี a_i อยู่ในคอลัมน์ซ้ายสุดในเมทริกซ์ย่อยทางซ้ายมือ ถ้าเราลองหมุน a_i ในแต่ละบรรทัดไปต่อท้ายสตริงในบรรทัดเดียวกันตามที่แสดงในเมทริกซ์ย่อยทางขวามือ จะพบว่าไม่มีผลกระทบกับลำดับของสตริงในแต่ละบรรทัดซึ่งลักษณะนี้เป็นจริงสำหรับทุกอักขระและกับสตริงใดๆ

การประยุกต์ใช้คุณสมบัติ First-Last ในการแปลง Burrows-Wheeler Transform กลับเป็นสายสตริงตั้งต้น

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

ตำแหน่งแรกคือ a_3 และอักขระถัดไปหาได้จากอักขระที่อยู่ในคอลัมน์ซ้ายสุดของบรรทัดที่มี a_3 เป็นอักขระที่อยู่ในคอลัมน์ขวาสุดนั่นเอง จากนั้นอักขระถัดๆไปก็จะเป็น “r” (r_2), “a” (a_4), c (c_1), “a” (a_5) ไปเรื่อยๆ ตามลำดับ



ฝึกหัด	ลองสร้างสตริงต้นฉบับจาก BWT “enwvpeoseu$llt”

วิธีการหาชุดของสตริงย่อยในสายสตริงหลักโดยใช้ Burrows-Wheeler Transform

อาศัย BTW ที่มีข้อมูลเฉพาะคอลัมน์แรกและคอลัมน์สุดท้ายและมีดัชนีของแต่ละอักขระแล้ว เราสามารถหาสตริงย่อยในสายสตริงหลักโดยการตรวจดูแต่ละอักขระในสตริงย่อยเรียงลำดับจากขวามาซ้าย ดังแสดงในตัวอย่างต่อไปนี้ (รูปที่ 3.8 โดยให้เลขที่บรรทัดเริ่มจาก 0) สมมติว่าต้องการหาว่ามีสตริงย่อย “apa” ในสตริงต้นฉบับ อักขระ “a” ทางขวาสุดใน “apa” จะถูกนำมาดูในคอลัมน์ซ้ายสุดว่ามีบรรทัดใดบ้างที่เป็น “a” และดูว่าอักขระตัวก่อนหน้ามันคือ “p” หรือไม่โดยดูจากคอลัมน์ขวาสุดในบรรทัดเดียวกัน พบว่ามี 2 บรรทัดคือบรรทัดที่ 4 และ 6 ที่เข้าเงื่อนไข ซึ่งคือ p_1, และ p_2 ตามลำดับ นำ p_1 และ p_2 นี้มาว่าอยู่บรรทัดใดบ้างในคอลัมน์ซ้ายสุด และดูว่าคอลัมน์ทางขวาสุดคือ “a” หรือไม่ ซึ่งพบว่าทั้ง 2 บรรทัดคือบรรทัดที่ 10 และ 11 เข้าเงื่อนไข นำ a_3 และ a_4 นี้มาดูว่าบรรทัดใดบ้างในคอลัมน์ซ้ายสุด และเนื่องจากไม่มีอักขระในสตริงย่อยแล้ว หมายความว่าพบสตริงย่อย “apa” 2 ตำแหน่งในสตริงต้นฉบับ

<u>หมายเหตุ</u>

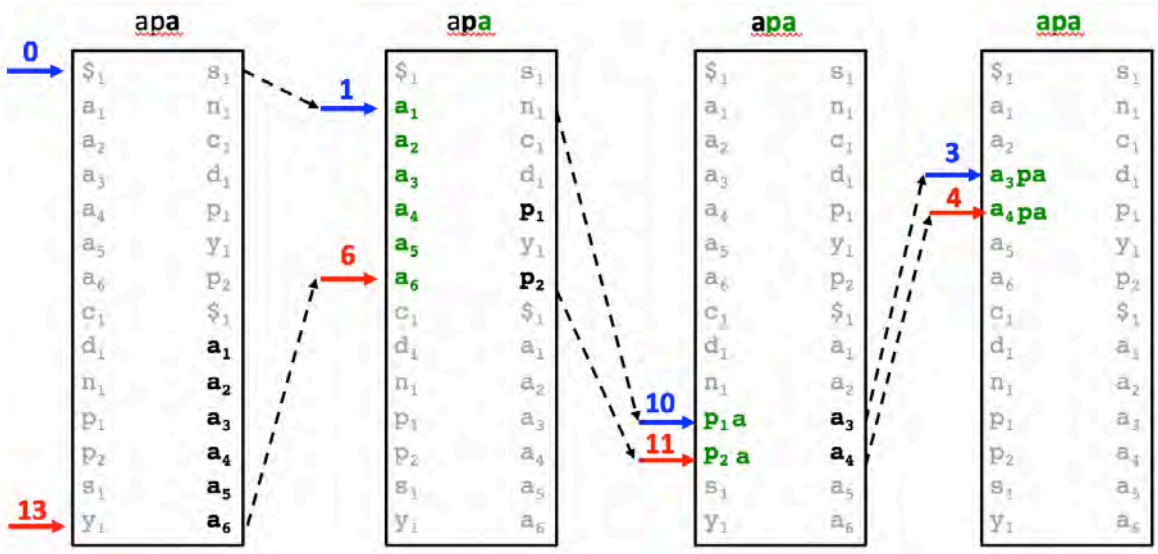
เริ่มไล่อักขระจากคอลัมน์ซ้ายสุด อักขระในคอลัมน์ขวาสุดในบรรทัดเดียวกันคือดูตัวที่มาก่อนหน้า

เริ่มไล่อักขระจากคอลัมน์ขวาสุด อักขระในคอลัมน์ซ้ายสุดในบรรทัดเดียวกันคือดูตัวที่ตามมา

การหาบรรทัดในคอลัมน์ซ้ายสุดของอักขระทางขวาสุด

จากคำอธิบายข้างต้นในการหาสตริงย่อยโดยไล่ดูทีละอักขระในสตริงย่อยจากขวามาซ้าย ในขั้นตอนเดินย้อนกลับแต่ละรอบนี้จะต้องมีการเปลี่ยนชุดของบรรทัดทางซ้ายที่ต้องพิจารณาในรอบนั้นๆ ดังแสดงในรูปที่ 3.8 ต่อไปนี้ โดยในรอบแรกชุดของบรรทัดทางซ้ายที่ต้องพิจารณาคือบรรทัด 1-6 ในรอบที่สองจะเป็นบรรทัด 10-11 และใน

รอบที่สามจะเป็นบรรทัด 3-4 ซึ่งจะเห็นว่าสามารถใช้ตัวชี้เพียง 2 ตัวคือ *top* กับ *bottom* เช่น ในรอบแรก *top* = 1 และ *bottom* = 6 ในรอบที่สอง *top* = 10 และ *bottom* = 11 เพื่อให้การเปลี่ยนชุดของบรรทัดที่ต้องพิจารณาทำได้รวดเร็วในแต่ละรอบ ได้มีการเพิ่มฟังก์ชัน `LASTTOFIRST(i)` ที่ทำหน้าที่หาเลขที่บรรทัดทางซ้ายของตัวอักขระที่ถูกระบุโดยเลขที่บรรทัดทางขวา จากตัวอย่างในรูปที่ 3.8 `LASTTOFIRST(4)` จะได้ค่าส่งกลับเป็น 10 และ `LASTTOFIRST(6)` จะได้ค่าส่งกลับเป็น 11 เป็นต้น สูโดโคดที่ 3.4 แสดงโคดที่ใช้ในการหาจำนวนตำแหน่งในสตริงต้นฉบับที่อยู่ในรูปของ BWT ที่พบสายสตริงย่อยที่ต้องการหา



รูปที่ 3.7 การเปลี่ยนค่าของตัวชี้ top และ bottom ของบรรทัดที่ต้องพิจารณาในแต่ละรอบ

(ที่มา: ดัดแปลงจากรูปที่ 9.14 ของ [21])

สูโดโคดที่ 3.4 BWMatching()

```
1  BWMatching(FirstColumn, LastColumn, Pattern, LASTTOFIRST)
2      top <- 0
3      bottom <- จำนวนบรรทัดทั้งหมด
4      while top <= bottom
5          if ยังมีอักขระในสตริงย่อย Pattern
6              symbol <- อักขระทางขวาสุดของสตริงย่อย
7              นำอักขระขวาสุดนี้ออกจากสตริงย่อย
8              if มีตำแหน่งอักขระใน LastColumn ในช่วงระหว่าง top ถึง bottom ที่เป็นตัวเดียวกับ symbol
9                  topIndex <- ตำแหน่งแรกในช่วง top ถึง bottom ที่เป็นตัวเดียวกับ symbol
10                 bottomIndex <- ตำแหน่งสุดท้ายในช่วง top ถึง bottom ที่เป็นตัวเดียวกับ symbol
11                 top <- LASTTOFIRST(topIndex)
12                 bottom <- LASTTOFIRST(bottomIndex)
13             else
14                 ส่งกลับ ค่า 0
15         else
16             ส่งกลับ ค่า bottom-top+1 ซึ่งคือค่าจำนวนตำแหน่งในสตริงหลักที่พบสายสตริงย่อย
```

วิธีการหาชุดของสตริงย่อยในสายสตริงหลักโดยไม่ต้องเหมือนกันทั้งสาย

วิธีการที่อธิบายมาก่อนหน้านี้ทั้งหมดเน้นการหาสตริงย่อยในสายสตริงหลักที่มีประสิทธิภาพและใช้หน่วยความจำน้อยที่สุด โดยสตริงย่อยทั้งสายจะต้องเหมือนกับบางส่วนของสายสตริงหลัก ในหัวข้อนี้เราจะนำวิธีการข้างต้นมาประยุกต์ใช้ในการหาสตริงย่อยในสายสตริงหลักโดยอาจมีบางอักขระในสายสตริงย่อยที่ไม่เหมือนกับสตริงหลัก

นิยามปัญหาที่ 3.4 ปัญหาการหาชุดของสตริงย่อยในสายสตริงหลักแบบโดยประมาณ

ปัญหาการหาชุดของสตริงย่อยในสายสตริงหลักแบบโดยประมาณ (Multiple Approximate Pattern Matching Problem) หาตำแหน่งทั้งหมดที่พบสตริงย่อยโดยบางส่วนของสตริงย่อยไม่ต้องเหมือนสายสตริงหลัก	
ข้อมูลเข้า	สายสตริงหลัก ชุดของสตริงย่อย และค่าจำนวนเต็ม d
ผลลัพธ์	ตำแหน่งเริ่มต้นทั้งหมดในสตริงหลักที่พบสตริงย่อยแต่ละรูปแบบโดยสตริงย่อยเหล่านั้นสามารถต่างกับสตริงหลักได้ d อักขระ

ลองพิจารณาสตริงสองสายต่อไปนี้ จะเห็นว่าเราสามารถหาว่าสายสตริงย่อยต่างจากสายสตริงหลัก 1 อักขระโดยการแบ่งสตริงย่อยออกเป็นส่วนแล้วเอาสองส่วนนั้นมาหาในสายสตริงหลักโดยหาแบบต้องเหมือนทั้งเส้น (exact match) ถ้าทั้งสองส่วนเหมือนกับสายสตริงหลักทั้งเส้น ก็ตรวจสอบเพิ่มเติมว่าสตริงย่อยทั้งสายมี 1 อักขระที่ต่างจากสายสตริงหลักหรือไม่ วิธีการนี้สามารถนำไปประยุกต์ใช้กับจำนวนความต่างมากกว่า 1 ได้ด้วย (d > 1) โดยถ้าสตริงย่อยต่างจากสตริงหลักอย่างมาก d อักขระ หมายความว่า ทั้งสตริงย่อยและสตริงหลักจะมีส่วนของสตริงที่

สตริงย่อย **acttggct**

สตริงหลัก **....ggcacactaggctcc....**

เหมือนกันยาวที่สุด k อักขระ (k-mer) 1 สาย ตัวอย่างเช่น ถ้าเรามีสายสตริงย่อยยาว 20 อักขระและค่า d = 3 สตริงย่อยนี้จะถูกแบ่งออกเป็น d+1 ส่วน โดยมีความยาวของแต่ละส่วนเป็น 20/(3+1) = 5 ซึ่งเราสามารถนำทั้งสี่ส่วนไปหาบนสายสตริงหลักแบบเหมือนทั้งเส้น เช่น

สตริงย่อย acttaggctcgggataatcc

สตริงหลัก ....actaagtctcgggataagcc....

สิ่งที่สังเกตได้นี้มีประโยชน์เพราะสามารถนำแต่ละส่วนย่อยนั้นไปเทียบหากับสตริงหลักแบบเหมือนทั้งเส้นโดยใช้วิธีการอย่างซัฟฟิกซ์ทรีหรือซัฟฟิกซ์อะเรย์ได้

หยุดคิด	ถ้าสตริงย่อยมีความยาว 23 อักขระและต่างจากสายสตริงหลัก 3 อักขระ เราจะสามารถสรุปได้ไหมว่าสตริงย่อยจะมีส่วนของสตริงที่เหมือนกับสตริงหลัก 6 อักขระ หรือ 5 อักขระ

ในคำอธิบายข้างต้นเรายังไม่ได้พิจารณาว่าค่า k ที่มีการพูดถึงนั้นจะมีค่าเท่าไหร่ โดยค่า k นี้มีทฤษฎีอธิบายดังต่อไปนี้

ทฤษฎี ถ้าในส่วนที่เหมือนกัน n อักขระระหว่างสตริงสองสาย และมีอย่างมาก d อักขระที่ต่างกัน สตริงสองสายนั้นจะต้องมีส่วนของสายยาว k อักขระ (k-mer) ร่วมกัน โดย k = $\lfloor n/(d+1) \rfloor$

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

acttaggctcgggataatccgga

ถ้าเรากระจายตำแหน่งที่ไม่เหมือน 3 ตำแหน่งลงไปในสตริงส่วนต่างๆข้างต้น จะพบว่าตำแหน่งที่ไม่เหมือนเหล่านี้จะมีผลกระทบกับสตริง 3 ส่วน ซึ่งจะเหลืออีกส่วนที่ยาวอย่างน้อย k ที่ไม่ถูกกระทบ ดังนั้น ส่วนย่อยที่ยาว k นี้จะเหมือนกันระหว่างสตริงสองสาย

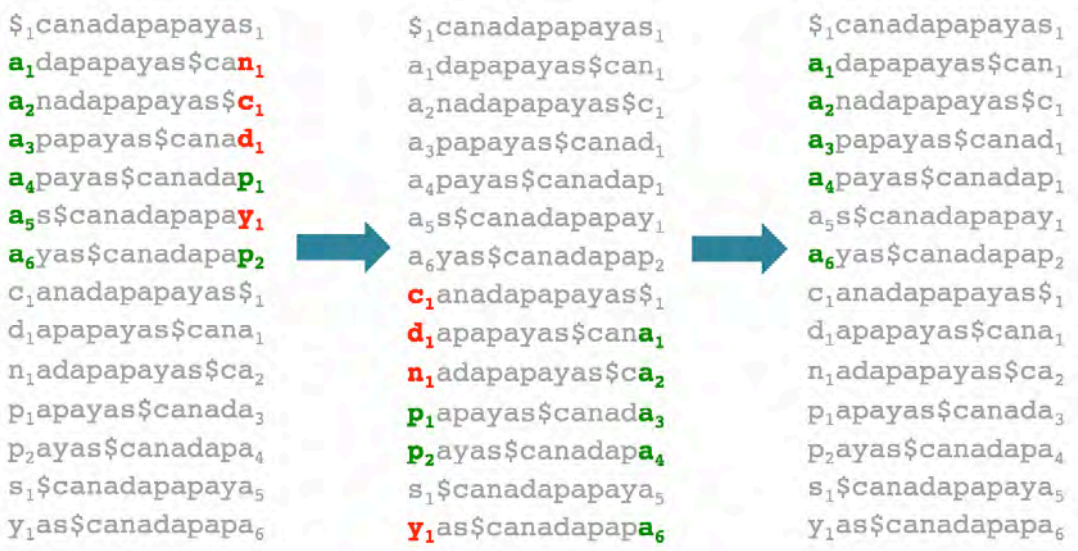
ถึงจุดนี้เราจะได้ขั้นตอนหลักๆของอัลกอริทึมที่ใช้ในการหาสตริงย่อยที่มีความยาว n อักขระ ในสายสตริงหลัก โดยสามารถมีอักขระที่ต่างกันได้ไม่เกิน d อักขระ โดยขั้นแรกแบ่งสตริงย่อยยาว n ออกเป็น d+1 ส่วน โดย d ส่วนแรกยาว k = $\lfloor n/(d+1) \rfloor$ เรียกว่าสีด (seeds) หรือชิ้นส่วนเริ่มต้น นำสีดเหล่านี้ไปหาในสตริงหลักและตรวจสอบดูว่าสีดไหนบ้างลำดับอักขระทั้งเส้นเหมือนกับส่วนของสตริงหลัก เมื่อได้สีดเหล่านี้แล้วขั้นถัดไปจะทำการขยายสีดทั้งด้านซ้ายและขวาเพื่อดูว่าสตริงย่อยทั้งเส้นที่เกิดจากการขยายสีดหนึ่งๆนี้ ยังมีอักขระที่ต่างกับสตริงสายหลักไม่เกิน d อักขระหรือไม่

วิธีการหาชุดของสตริงย่อยในสายสตริงหลักแบบโดยประมาณโดยใช้ Burrows-Wheeler Transform

การประยุกต์ใช้แนวทางของเบอร์โรวส์-วีลเลอร์ กับการหาชุดของสตริงย่อยในสายสตริงหลักแบบโดยประมาณนั้นตอนที่เทียบทีละอักขระของสตริงย่อยกับคอลัมน์ซ้ายสุดถึงไม่ตรงก็จะไม่หยุดการทำงานถ้าจำนวนอักขระที่ไม่ตรงนั้นยังไม่เกิน d อักขระ จากรูปที่ 3.8 BTW ทางซ้ายมือแสดงการพบอักขระ “a” ตัวขวาสุดของสตริงย่อยในคอลัมน์ซ้ายสุด 6 บรรทัด ซึ่งตัวอักขระก่อนหน้า “a” ใน 6 บรรทัดนี้มี 4 บรรทัดเป็น “n”, “c”, “d”, “y” ซึ่งไม่ตรงกับอักขระ “p” อย่างไรก็ตามถ้า d = 1 หมายความว่าจะยังเดินย้อนไปต่อได้ โดยจะเก็บไว้ทั้ง 6 บรรทัด โดยอักขระในคอลัมน์ขวาสุด 6 บรรทัดนี้จะถูกนำมาพิจารณาตรวจดูว่าอยู่บรรทัดไหนในคอลัมน์ซ้ายสุดของ BWT กลาง และตามต่อไปที่คอลัมน์ขวาสุดของ BTW กลางเพื่อกลายเป็นคอลัมน์ซ้ายสุดของ BWT ทางขวา ซึ่งพบว่ามีทั้งสิ้น 5 ตำแหน่ง คือ “ada”, “ana”, “aya” และอีก 2 “apa” ในสายสตริงหลักที่เหมือนกับสายสตริงย่อยโดยประมาณคือมีอักขระตัวกลางที่แตกต่างกัน 1 อักขระ

ในทางปฏิบัติเราไม่ต้องการที่จะอนุญาตให้เกิดอักขระที่ไม่ตรงกับสตริงหลักตั้งแต่ต้นๆของการอ่านจากขวามาซ้าย เนื่องจากจะเพิ่มจำนวนบรรทัดที่ต้องตรวจสอบเพิ่มเติมมาก แนวทางหนึ่งที่เป็นไปได้คือมีการกำหนดเงื่อนไขเพิ่มเติมว่าซัฟฟิกซ์จำนวน x อักขระจะต้องเหมือนกับสตริงสายหลักก่อนก่อนที่จะเดินย้อนต่อและอนุญาต

ตำแหน่งไม่เหมือนหลังจากนั้น นอกจากนี้ขนาดของ d มีผลต่อเวลาที่ใช้ในการหาเนื่องจากเพิ่มจำนวนการเดินย้อนมากขึ้นเพราะมีรูปแบบที่เป็นไปได้มากขึ้นมาก ในทางปฏิบัติค่า d มักจะไม่เกิน 3



รูปที่ 3.8 แสดงการใช้ Burrows-Wheeler Transform กับการหาสตริงย่อย “apa” ในสายสตริงหลักแบบประมาณ

(ที่มา: ดัดแปลงจาก รวบย่อส่วนจากรูปที่ 9.19 ของ[21])

บทส่งท้าย

วิธีการและอัลกอริทึมที่ใช้ในการเทียบ (align หรือ map) หรือค้นหาชุดของสายสตริงย่อย เช่น รีดหรือดีเอ็นเอสายสั้นจำนวนมากในสายสตริงหลัก เช่น จีโนมอ้างอิงของมนุษย์ เป็นจุดเริ่มต้นของการหาความแปรผันของดีเอ็นเอของบุคคลหนึ่งเทียบกับจีโนมอ้างอิง เทียบกันระหว่างจีโนมของบุคคลในครอบครัว หรือเทียบกันระหว่างจีโนมของกลุ่มประชากร ซึ่งความแปรผันมีได้หลากหลายรูปแบบเช่น การแปรผันในลำดับเบสเดียวที่บริเวณหนึ่งๆ ที่เรียกว่าเอสเอ็นวี (SNV) หรือการแปรผันเชิงโครงสร้าง (structural variation) เช่น มีชุดของรีพีทที่แตกต่างกันระหว่างจีโนม เป็นต้น โดยวิธีการการค้นหาชุดของสายสตริงย่อยในสายสตริงหลักแบบโดยประมาณเป็นตัวอย่างของวิธีการหาเอสเอ็นวี นอกจากการเทียบดีเอ็นเอสายสั้นจำนวนมากกับจีโนมอ้างอิงตามที่อธิบายในบทเรียนนี้แล้ว การค้นหาสตริงย่อยในสตริงหลักยังสามารถนำไปประยุกต์ใช้ในการตอบโจทย์ทางชีวการแพทย์อื่นๆ เช่น

- วิเคราะห์การแสดงออกของยีนในระดับทรานสคริปโตมิกส์ (transcriptomics) โดยการเทียบสายอาร์เอ็นเอในรูปแบบซีดีเอ็นเอจำนวนมากกับจีโนมอ้างอิงด้วยเทคโนโลยีเอ็นจีเอส (NGS) เรียกว่า RNA-Seq (อ่านว่าอาร์เอ็นเอซีค) เพื่อใช้เป็นข้อมูลในการวิเคราะห์ปริมาณการแสดงออกของยีน (gene expression) ทั้งจีโนม ในเงื่อนไขจำเพาะต่างๆ เช่นการใช้ยาที่แตกต่าง การแสดงออกของยีนตามเวลาที่เพิ่มขึ้น เป็นต้น
- วิเคราะห์ความแปรผันในสายดีเอ็นเอเทียบกับจีโนมอ้างอิงแต่ชุดดีเอ็นเอสายสั้นที่ได้เป็นดีเอ็นเอเฉพาะ

บริเวณที่เป็นเอ็กซอน (exons) ของทั้งจีโนมซึ่งเป็นผลของการทำ Whole Exome Sequencing (WES) แทนที่จะเป็น Whole Genome Sequencing (WGS) ทั้งนี้เพื่อมุ่งเป้าการวินิจฉัยการแปรผันของดีเอ็นเอเฉพาะบริเวณที่เป็นยีนที่สามารถแปลรหัสต่อไปเป็นโปรตีน ซึ่งส่วนใหญ่รู้ฟังก์ชันและหรือความสัมพันธ์กับการเกิดโรค เหมาะกับการวินิจฉัยทางคลินิก รวมทั้งจำนวนรีดที่ได้ทั้งหมดมีจำนวนน้อยกว่ารีดที่ได้จากการถอดรหัสทั้งจีโนมมาก

- การประยุกต์ใช้ซัฟฟิกซ์ทรีในการเทียบ short tandem repeats (STR) กับจีโนมอ้างอิงในงานนิติเวชศาสตร์ ที่มีการใช้เทคโนโลยี NGS ในการถอดรหัสพันธุกรรมในบริเวณที่เป็น STR โดยทั้งนี้จะได้ข้อมูลในรายละเอียดเพิ่มขึ้นในการพิสูจน์อัตลักษณ์และเปรียบเทียบหาความสัมพันธ์ระหว่างบุคคล

นอกจากการประยุกต์ใช้อัลกอริทึมต่างๆ ตามที่ได้มีการอธิบายในบทเรียนนี้แล้ว ยังมีการวิจัยและพัฒนาเพิ่มเติมในเชิงของการหาสตริงย่อยในสายสตริงหลักที่มีความจำเพาะ เช่นการวิเคราะห์ข้อมูลเกี่ยวกับอีพิจีโนมิกส์ Epigenomics) และเมตาจีโนมิกส์ [62] โดยในกรณีอีพิจีโนมิกส์ต้องมีการพิจารณาเพิ่มเติมการเกิดเมธิลเลชัน (Methylation) โดยมีการเติมหมู่เมธิลให้กับนิวคลีโอไทด์ซึ่งหมายถึงอาจมีการพิจารณาเพิ่มอักขระจาก “A”, “T”, “C”, และ “G” ที่เป็นตัวแทนของกรดนิวคลีโอไทด์พื้นฐานให้มีอักขระที่แสดงถึงนิวคลีโอไทด์ที่มีการเติมหมู่เมธิลหรือการจัดการโอกาสที่จะเกิดลำดับเบสที่แตกต่างมากขึ้น เป็นต้น ในขณะที่งานวิจัยในสาขาเมตาจีโนมิกส์ (Meta genomics) หรือบางครั้งถูกเรียกว่า Environmental genomics หรือ Community genomics [63, 64] ซึ่งจะทำการถอดรหัสพันธุกรรมจากตัวอย่างที่เก็บมาจากสิ่งแวดล้อม เช่น จากธารน้ำร้อน จากบ่อย่อยไขมันของโรงงานอุตสาหกรรม จากลำไส้กุ้ง จากลำไส้ ผิวหนังหรือช่องปากมนุษย์ เป็นต้น โดยดีเอ็นเอที่ถอดรหัสพันธุกรรมออกมาได้นั้นจะประกอบด้วยรหัสพันธุกรรมของเชื้อต่างๆ หลายชนิดอยู่รวมกัน (ซึ่งมักไม่สามารถทำการเพาะเลี้ยงเชื้อเหล่านี้ในห้องทดลองได้) โดยความคาดหวังหลักจากการวิจัยคือสามารถระบุได้ว่าตัวอย่างที่เก็บมานั้นประกอบด้วยเชื้อชนิดใดบ้าง ปริมาณเท่าไหร่และหรืออยู่ในสายวิวัฒนาการ (phylogenetic group) กลุ่มใด ดังนั้นการออกแบบและพัฒนาวิธีการเทียบดีเอ็นเอกับจีโนมอ้างอิงในงานเชิงเมตาจีโนมิกส์นี้ต้องคำนึงถึงเรื่องชุดของจีโนมอ้างอิงที่ปัจจุบันมีจีโนมของเชื้อต่างๆอยู่มากมาย ซึ่งจะนำไปสู่ความคลุมเครือในการตัดสินว่ารหัสพันธุกรรมเหล่านั้นเป็นของเชื้อใดบ้าง โดยเฉพาะถ้ามีบริเวณที่มีลำดับเบสที่คล้ายคลึงกันหรือเหมือนกันระหว่างจีโนมของเชื้อมากกว่าหนึ่งชนิด เป็นต้น นอกจากนี้เรนเนิร์ทและคณะ [62] ได้กล่าวถึงการนำเสนอวิธีการทำ *de novo* assembly หรือการประกอบร่างจีโนมแบบวิธีการผสมที่จะใช้การเทียบดีเอ็นสายสั้นกับจีโนมอ้างอิงก่อน และดีเอ็นสายสั้นที่ตกอยู่ในบริเวณเดียวกันทั้งหมดที่คาบเกี่ยวกันจะถูกนำไปทำกราฟแสดงความคาบเกี่ยว (overlap graph) เพื่อทำการประกอบร่างเฉพาะบริเวณๆไป ให้มีความถูกต้องมากขึ้นต่อไป ในเชิงของปัญหาที่มีอยู่เรนเนิร์ทและคณะ [62] ได้อธิบายถึงความซับซ้อนที่เกิดจากรีพีทในการเทียบสายดีเอ็นเอกับจีโนมอ้างอิง โดยในจีโนมมนุษย์มีรีพีทเป็นส่วน ประกอบราว 50% [2] ในขณะที่จีโนมข้าวโพดมีรีพีทเป็นส่วนประกอบมากกว่า 80% [65] ซึ่งเป็นการยากในการระบุที่มาของสายดีเอ็นเอที่อ่านมาได้ว่าเป็นของรีพีทไหนในกลุ่มของรีพีทที่เป็นประเภทเดียวกัน โดยทั่วไปถ้า

ผลของการเทียบดีเอ็นเอสายสั้นแล้วพบว่ามีความเหมือนกับหลายๆบริเวณพร้อมๆกัน โปรแกรมที่ทำหน้าที่เทียบนี้จะรายงานผลเป็นความมั่นใจต่ำ ในทางตรงกันข้ามถ้าบริเวณในจีโนมอ้างอิงที่เหมือนกับดีเอ็นเอสายสั้นมีบริเวณเดียวโปรแกรมตัวเทียบก็จะให้ค่าความมั่นใจสูง เป็นต้น นอกจากรีพีทแล้วอีกประเด็นที่ถูกกล่าวไว้คือโดยพื้นฐานจีโนมของมนุษย์แต่ละคนเหมือนกันประมาณ 99.8% ซึ่งหมายความว่าจะมีเบสที่แตกต่างกันโดยปกติอยู่แล้วและลำดับเบสที่แตกต่างกันนี้มีการกระจายตัวที่ไม่สม่ำเสมอในลักษณะการแปรผันโดยรวม [66] และการแปรผันแบบ indels โดยเฉพาะ [67] ซึ่งอาจมีผลต่อการแปลผลการเทียบสายดีเอ็นเอกับจีโนมอ้างอิง เนื่องจากมีบางบริเวณในจีโนมอ้างอิงที่มีจำนวนรีดมาตกน้อยหรือไม่มีเลย ซึ่งอาจเป็นผลของการเกิด deletion ในบริเวณของจีโนมที่ถูกถอดรหัสออกมาเป็นชุดของรีดที่นำมาเทียบ หรือในบริเวณนั้นของจีโนมมนุษย์แต่ละคนมีความแปรผันมากจนไม่มีดีเอ็นสายสั้นจากจีโนมมนุษย์คนไหนที่เหมือนกับจีโนมอ้างอิงมากพอ โดยแนวทางในการแก้ปัญหานี้ประกอบด้วยการสร้างจีโนมอ้างอิงให้กับกลุ่มประชากรของตัวเอง โดยในระยะหลังมีแนวทางนี้เกิดขึ้นมากเนื่องจากราคาในการถอดรหัสพันธุกรรมระดับจีโนมถูกลงมากในขณะที่คุณภาพของรหัสพันธุกรรมที่ได้ดีขึ้นมาก เช่น สร้างจีโนมอ้างอิงของคนเกาหลีที่ตีพิมพ์ในนิตยสารเนเจอร์ปีค.ศ. 2016 [68] หรือการสร้างจีโนมอ้างอิงจากบุคคลในภายในครอบครัว [69] เป็นต้น อีกแนวทางหนึ่งคือการเปลี่ยนลำดับเบสในจีโนมอ้างอิงในตำแหน่งต่างๆที่พบว่าความแปรผันสูง เช่นตำแหน่งที่อาจพบเบส “T” หรือ “C” ก็สามารถเข้ารหัสเป็นตัวอักษรใหม่เช่น “Y” ตามมาตรฐาน International Union of Pure and Applied Chemistry (IUPAC) ซึ่งใช้ระบุว่าอาจเป็นเบส “T” หรือ “C” ก็ได้ ซึ่งในกรณีนี้อัลกอริทึมหรือโปรแกรมจะต้องการไฟล์ที่มีข้อมูลระบุความหมายของรหัสเพิ่มเติมเหล่านี้ นอกจากนี้อีกแนวทางหนึ่งที่น่าสนใจคือการนำกราฟมาแสดงองค์ประกอบของลำดับเบสในจีโนมแทนการใช้สตริงอย่างที่มีการใช้กันอยู่อย่างกว้างขวาง เนื่องจากกราฟเป็นโครงสร้างข้อมูลที่สามารถแสดงการเกิดความแปรผันได้อย่างเป็นธรรมชาติ เช่น แต่ละโหนดสามารถมีเส้นเชื่อมในลักษณะทางแยกและสามารถกลับมารวมกันได้ในลำดับเบสถัดๆไปเกิดเป็นบับเบิ้ลในสายทางเดินภายในกราฟ เป็นต้น โดยงานวิจัยมีทั้งแนวทางเน้นการสร้างกราฟที่ใช้หน่วยความจำอย่างมีประสิทธิภาพและสามารถสืบค้นข้อมูลลำดับเบสในตำแหน่งต่างๆ ได้อย่างรวดเร็ว [70, 71] และมีการพัฒนาออกมาในรูปแบบของเครื่องมือทางชีวสารสนเทศที่ช่วยในการสร้างกราฟจากลำดับเบสจีโนม[72] และเครื่องมือที่สามารถสร้างกราฟและนำรีดมาเทียบกับกราฟได้ [73] นอกจากนี้ก็มีโครงการ vg (https://github.com/vgteam/vg) ที่เปิดเป็นสาธารณะบนกิต (git) เพื่อการพัฒนาเครื่องมือในการสร้าง variant graph จากไฟล์ VCF (Variant Call Format) และสามารถนำสตริงของรีดมาเทียบกับกราฟที่สร้างขึ้นได้ เป็นต้น บริษัทเอกชนอย่าง SevenBridges (https://www.sevenbridges.com/graph/) ก็นำกราฟมาเป็นโครงสร้างข้อมูลพื้นฐานในการแสดงจีโนมและหาความแปรผันประเภทต่างๆ ดิลเดยธ์และคณะ [74] นำเสนอจีโนมอ้างอิงในรูปแบบกราฟที่สร้างจากกลุ่มประชากรโดยนำมาประยุกต์ใช้กับบริเวณ MHC (major histocompatibility complex) ในโครโมโซมที่ 6 รวมทั้งนำข้อมูลอื่นๆเข้ามาร่วม เช่น ข้อมูลลำดับเบสที่ทราบแน่ชัดของอัลลีลต่างๆ ของ HLA (Human Leukocyte Antigen) และข้อมูล 87,640 สนิปส์จากโครงการจีโนมมนุษย์ 1000 จีโนม เป็นต้น โดยได้แสดงให้เห็นว่าการใช้จีโนมกราฟช่วยเพิ่มความถูกต้องในการระบุบริเวณจำเพาะบนจีโนมได้

เรนเนิร์ทและคณะ [62] ได้คาดการณ์ว่าการใช้ฮาร์ดแวร์ เช่น หน่วยประมวลผลกราฟิกส์ (Graphics Processing Unit: GPU) จะเป็นแนวทางสำคัญในการเพิ่มประสิทธิภาพในการเทียบชุดของสตริงย่อยกับสตริงหลักแบบโดยประมาณ สำหรับแนวทางในเชิงข้อมูลคือเรื่องของการเพิ่มความถูกต้องของผลการวิเคราะห์ในบริเวณที่เป็นรีพีท เป็นต้น

ตัวอย่างโปรแกรมที่มีการใช้งานกันอย่างแพร่หลาย

ตัวอย่างโปรแกรมที่มีการใช้งานกันอย่างแพร่หลายโดยมีการใช้ Burrows-Wheeler Transform (BWT) และซัฟฟิกซ์อะเรย์เป็นฐาน เช่น โปรแกรม BWA [75, 76], Bowtie [77], Bowtie2 [78] เป็นต้น ทั้งนี้ CUSHAW [79] ใช้แนวทางเดียวกันแต่พัฒนาให้เป็นการทำงานแบบขนานโดยใช้คูดา (CUDA: compute unified device architecture) ในขณะที่ SOAP2 [80] และ SOAP3 [81] ปรับแต่ง BWT ให้เป็นสองทิศทางทั้งสายบวกและลบซึ่งทำให้สามารถค้นหาดีเอ็นเอในจีโนมอ้างอิงในทั้งสองทิศทางได้พร้อมกัน รายละเอียดเพิ่มเติมเกี่ยวกับการเทียบดีเอ็นสายสั้นกับจีโนมอ้างอิงและหรือแอปพลิเคชั่นที่เกี่ยวข้องสามารถศึกษาได้จากรีวิว [20, 62, 82, 83]

แบบฝึกหัดบทที่ 3

ให้เขียนโปรแกรมเพื่อแก้ปัญหาที่เกี่ยวข้องกับการเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิงโดยใช้โจทย์ที่โรซาลินด์ต่อไปนี้

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

อัลลีล (Allele)

อัลลีลเป็นรูปแบบที่เป็นไปได้ที่แตกต่างกันของยีนเดียวกัน บางยีนอาจมีหลายอัลลีลซึ่งจะอยู่ในตำแหน่งของยีนและโครโมโซมเดียวกัน มนุษย์เป็นตัวอย่างของสิ่งมีชีวิตที่เรียกว่าดิพลอยด์ (diploid) ซึ่งแต่ละยีนจะมีสองอัลลีลรับมาจากพ่อและแม่อย่างละหนึ่งอัลลีล แต่ละคู่ของอัลลีลนี้เป็นตัวแทนจีโนไทป์ (genotype) ของยีนนั้นๆ โดยจีโนไทป์ จะเป็นแบบโฮโมไซกัส (homozygous) ถ้าทั้งสองอัลลีลเหมือนกัน และเป็นเฮเทอโรไซกัส (heterozygous) ถ้าสองอัลลีลต่างกัน อัลลีลเหล่านี้ส่งผลต่อลักษณะที่ปรากฏ (phenotype) นอกจากนี้บางอัลลีลอาจเป็นอัลลีลเด่น

(dominant allele) หรืออัลลีลด้อย (recessive) ถ้าที่โลคัสหนึ่งๆ เป็นเฮเทอโรไซกัส โดยมีอัลลีลเด่นและอัลลีลด้อยอย่างละหนึ่งอัลลีล ลักษณะที่แสดงออกจะเป็นของอัลลีลเด่น
ที่มา https://www.nature.com/scitable/definition/allele-48

สนิปส์ (SNP)

Single Nucleotide Polymorphism หรือ SNP อ่านว่าสนิปส์ เป็นลักษณะการแปรผันของลำดับเบสเดี่ยวๆ ระหว่างจีโนมของแต่ละคน จากพื้นฐานที่สายดีเอ็นเอประกอบไปลำดับเบสนิวคลีโอไทด์ “A”, “T”, “C” และ “G” ถ้ามีมากกว่า 1% ของจำนวนประชากรที่มีตำแหน่งหนึ่งๆในสายดีเอ็นเอเป็นนิวคลีโอไทด์ที่แตกต่างกัน ความแปรผันในตำแหน่งนั้นๆสามารถเรียกได้ว่าเป็นสนิปส์ ถ้าสนิปส์เกิดในยีน ยีนนั้นจะมีมากกว่าหนึ่งอัลลีล ซึ่งในกรณีนี้อาจมีผลต่อการความแปรผันในการแปลรหัสยีนนั้นๆไปเป็นโปรตีน ทั้งนี้สนิปส์พบทั้งในบริเวณที่เป็นยีนและไม่ใช่ยีน สนิปส์บางตำแหน่งอาจส่งผลหรือเกี่ยวข้องกับการเกิดโรค
ที่มา https://www.nature.com/scitable/definition/single-nucleotide-polymorphism-snp-295

รูปแบบไฟล์ที่เกี่ยวข้อง

SAM/BAM

รูปแบบไฟล์แซม (SAM: Sequence Alignment/Map) [84] ถูกใช้ในการแสดงผลการเทียบหรือแมพรีดที่ถอดรหัสได้กับจีโนมอ้างอิง เป็นไฟล์แบบเท็กซ์ (text) ที่มีรูปแบบจำเพาะโดยแต่ละคอลัมน์คั่นด้วยแท็บ (tab) และประกอบด้วย 2 ส่วนคือส่วนหัว (ซึ่งอาจจะหรือไม่มีก็ได้ ถ้ามี ทุกบรรทัดของส่วนหัวจะขึ้นต้นด้วยอักขระ “@” และต้องมาก่อนส่วนที่เป็นผล) และส่วนที่เป็นผลของการเทียบดีเอ็นเอสายสั้นกับจีโนมอ้างอิง โดยส่วนที่เป็นผลนี้จะประกอบด้วย 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 หมายความว่าไม่มีค่าคุณภาพของการแมพ CIGAR string เป็นสายของอักขระโดยที่อักขระแต่ละตัวมีความหมายจำเพาะแตกต่างกันไป ตัวอย่างเช่น “M” หมายถึงรีดแมพได้กับจีโนมอ้างอิง “I” หมายถึงเกิดการแทรก (insertion) ส่วนนี้อยู่ในจีโนมอ้างอิง “D” หมายถึงมีการตัดออก (deletion) จากจีโนมอ้างอิง “S” หมายถึงเกิด soft clipping “=” หมายถึงรีดเส้นนั้นตรงกับจีโนมอ้างอิง เป็นต้น QUAL เป็นรหัส ASCII ที่แสดงค่าคะแนน Phred (ภาคผนวกบทที่ 1) + 33 เป็นต้น

ไฟล์รูปแบบแบม (BAM) เป็นไฟล์การแสดงผลของการเทียบหรือแมพรีด เช่นเดียวกับไฟล์แซมแต่อยู่ในรูปแบบไบนารีและถูกบีบอัดในรูปแบบ BGZF ซึ่งสามารถบีบอัดได้ดีและยังอนุญาตให้สามารถเข้าถึงไฟล์แบมในแต่ละส่วนได้โดยตรงผ่านชุดของดัชนีการสืบค้นที่สร้างมาพร้อมกัน สำหรับรายละเอียดเพิ่มเติมของรูปแบบไฟล์

ทั้ง SAM และ BAM สามารถศึกษาได้จาก https://samtools.github.io/hts-specs/SAMv1.pdf ทั้งนี้การจัดการไฟล์ในรูปแบบแซมและแบมสามารถทำได้โดยใช้ชุดของโปรแกรมชื่อ SAMtools (http://samtools.sourceforge.net) ซึ่งตีพิมพ์เป็นส่วนหนึ่งของรูปแบบแซมข้างต้น

บทที่ 4 การหาบริเวณที่ควบคุมการแสดงออกของยีน (Regulatory motif finding)

วัตถุประสงค์

- เพื่อให้นิสิตเห็นความเชื่อมโยงของการวัดการแสดงออกของยีนกับการหาบริเวณที่ควบคุมการแสดงออกของยีน
- เพื่อให้นิสิตคุ้นเคยกับตัวอย่างข้อมูลตั้งต้นที่เกี่ยวข้องและเข้าใจการทำงานของอัลกอริทึมพื้นฐานที่ใช้ในการหาบริเวณที่ควบคุมการแสดงออกของยีน
- เพื่อให้นิสิตได้เห็นตัวอย่างงานวิจัยและผลงานวิจัยรวมทั้งตัวอย่างโปรแกรมที่ใช้ในการหาบริเวณที่ควบคุมการแสดงออกของยีน
- เพื่อให้นิสิตได้เห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทาย รวมทั้งงานวิจัยอื่นๆ ที่เกี่ยวข้อง รวมทั้งเห็นแนวทางอื่นๆที่สามารถใช้ในการการหาบริเวณที่ควบคุมการแสดงออกของยีนได้เช่นกัน

ผลลัพธ์ที่คาดหวัง

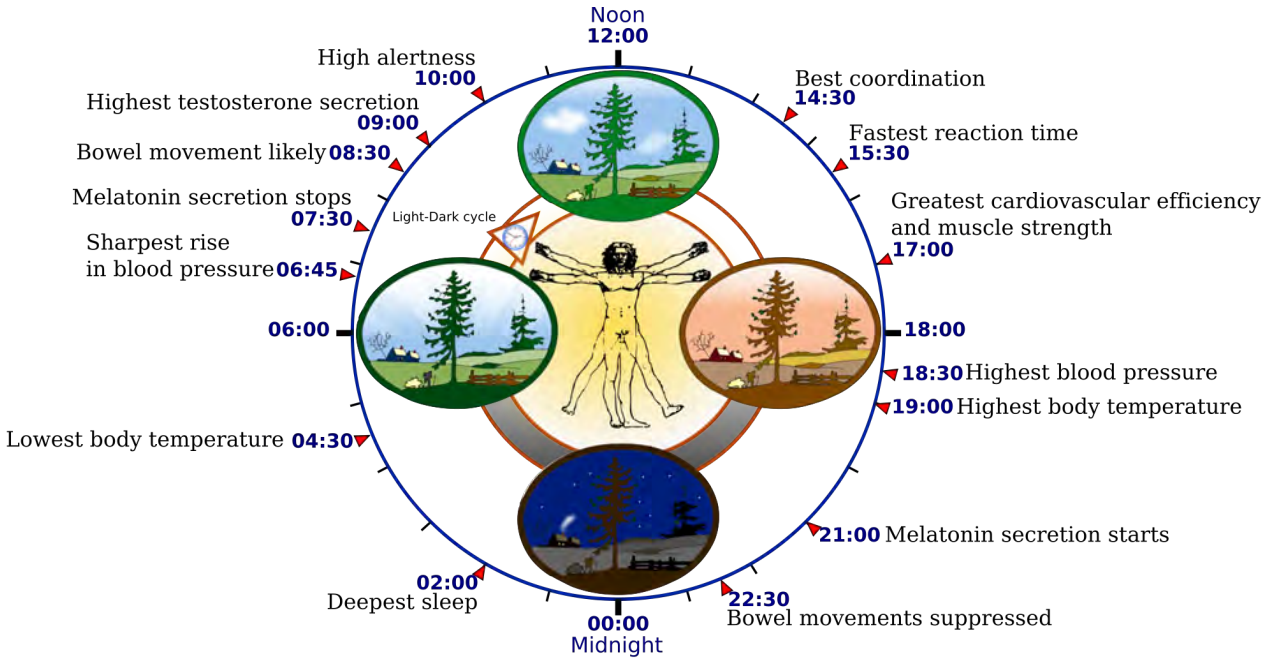
- นิสิตสามารถอธิบายความเชื่อมโยงของการวัดการแสดงออกของยีนกับการหาบริเวณที่ควบคุมการแสดงออกของยีน
- นิสิตเข้าใจคุณลักษณะของข้อมูลตั้งต้นที่ใช้ในการหาบริเวณที่ควบคุมการแสดงออกของยีน
- นิสิตสามารถอธิบายการทำงานของอัลกอริทึมหลักๆ ที่ใช้หาบริเวณที่ควบคุมการแสดงออกของยีน
- นิสิตสามารถเขียนโปรแกรมที่ใช้ในการหาบริเวณที่ควบคุมการแสดงออกของยีนได้
- นิสิตสามารถยกตัวอย่างการหาบริเวณที่ควบคุมการแสดงออกของยีนที่มีการใช้งานกันอย่างแพร่หลายได้
- นิสิตสามารถยกตัวอย่างความท้าทายที่ยังมีอยู่ในการหาบริเวณที่ควบคุมการแสดงออกของยีนและสามารถนำเสนอแนวทางในการพัฒนาวิธีการแก้ปัญหาเหล่านี้ได้ รวมทั้งสามารถประยุกต์องค์ความรู้จากบท เรียนเพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้
- นิสิตสามารถยกตัวอย่างแนวทางอื่นๆ ในการแก้ปัญหาการหาบริเวณที่ควบคุมการแสดงออกของยีนได้

เนื้อหาโดยสรุป

การวัดการแสดงออกของยีนและความเชื่อมโยงของการวัดการแสดงออกกับการหาบริเวณที่ควบคุมการแสดงออก โดยมีสมมติฐานว่าชุดของยีนที่มีรูปแบบของการแสดงออกแบบเดียวกันจะมีบริเวณในโปรโมเตอร์ที่มีรูปแบบของลำดับเบสร่วมกันเรียกว่าเรกูลาทอรีโมทิฟ (regulatory motif) หรือเรียกสั้นๆว่าโมทิฟ ซึ่งจะเป็นบริเวณที่ตัวควบคุมการแสดงออก เช่น ทรานสคริปชันแฟคเตอร์ (transcription factor: TF) มาจับ รูปแบบของข้อมูลเข้า โจทย์ทางชีวสารสนเทศ อัลกอริทึมพื้นฐานในการหาโมทิฟ เช่น โปรไฟล์เมทริกซ์ (profile matrix) คอนเซ็นซัสสตริง (consensus string) ปัญหามีเดียนสตริง (median string problem) การค้นหาโมทิฟแบบโลภ (greedy motif search) การค้นหาโมทิฟแบบสุ่ม (randomized motif search) และการสุ่มเลือกแบบกิปส์ (Gibbs sampling) การแสดงผลของโมทิฟผ่าน sequence logo และโปรแกรมที่มีการใช้งานกันอย่างแพร่หลาย โจทย์อื่นๆที่เกี่ยวข้อง รวมทั้งแนวทางหาโมทิฟโดยวิธีการเรียนรู้ของเครื่อง

บทที่ 4 การหาบริเวณที่ควบคุมการแสดงออกของยีน (Regulatory motif finding)

การดำเนินชีวิตในแต่ละวันของสิ่งมีชีวิตทั้งสัตว์และพืชถูกควบคุมโดยนาฬิกาเวลาภายในหรือนาฬิกาชีวิตซึ่งเรียกว่านาฬิกาเซอร์คาเดียน (circadian clock) การเกิดอาการเจ็ทแลก (jet lag) ตอนเดินทางข้ามทวีปเป็นตัวอย่างที่แสดงว่านาฬิกาเซอร์คาเดียนไม่เคยหยุดทำงาน รูปที่ 4.1 แสดงนาฬิกาเซอร์คาเดียนของมนุษย์ อีกตัวอย่างที่ยืนยันการทำงานของนาฬิกาเซอร์คาเดียนคือการทดลองโดยกักบริเวณหนูแรทและมนุษย์ที่เป็นอาสาสมัครไว้ในที่หลบภัยที่มีแต่ความมืดตลอดเวลาพบว่ากระบวนการทำงานพื้นฐานยังเป็นรอบปกติประมาณ 24 ชั่วโมงของการดำเนินชีวิตในแต่ละวัน อย่างไรก็ตามนาฬิกาเซอร์คาเดียนก็สามารถทำงานผิดพลาดได้ซึ่งในกรณีนี้จะทำให้เกิดโรคที่เรียกว่า delayed sleep-phase syndrome (DSPS)



รูปที่ 4.1 นาฬิกาเซอร์คาเดียนของมนุษย์

(ที่มา: https://commons.wikimedia.org/wiki/File:Biological_clock_human.svg)

จากผลการทดลองและอาการเจ็ทแลกที่เกิดขึ้นข้างต้น มีคำถามว่านาฬิกาเซอร์คาเดียนมีผลต่อกระบวนการของสิ่งมีชีวิตในระดับอณูชีววิทยาหรือชีวโมเลกุลอย่างไร ยีนนาฬิกา (clock gene) มีการทำงานและเกี่ยวข้องกับการแสดงออกของยีนและโปรตีนต่างๆในแต่ละช่วงของวันอย่างไร หรือเราจะสามารถอธิบายได้ไหมว่าทำไมการเกิดหัวใจล้มเหลวมักเกิดขึ้นในช่วงเช้าบ่อยกว่าช่วงเวลาอื่นๆ หรือการเกิดอาการภูมิแพ้มักเกิดตอนกลางคืน

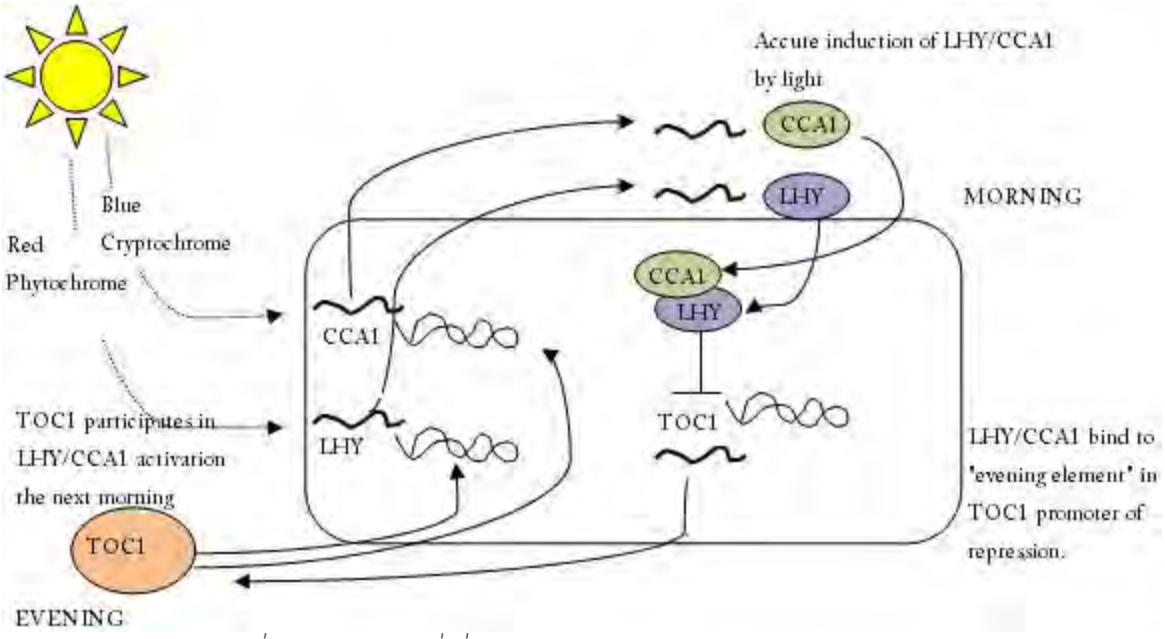
เป็นต้น และเราจะสามารถอธิบายได้ไหมว่ามียีนอะไรบ้างที่เกี่ยวข้องกับการทำให้นาฬิกาชีวิตทำงานผิดพลาดและเกิดโรค DSPS ขึ้น

ในช่วงต้นค.ศ. 1970 รอน โคนอปกา (Ron Konopka) และ ซีมัวร์ เบนเซอร์ (Seymour Benzer) ได้ทำการศึกษาแมลงวันที่มีการกลายพันธุ์โดยมีรูปแบบของนาฬิกาเซอร์คาเดียนที่ผิดปกติและสามารถค้นพบยีนเดี่ยวที่ให้ทำเกิดความผิดปกตินั้น [85] หลังจากนั้นอีก 20 ปี นักชีววิทยาได้ค้นพบยีนในลักษณะเดียวกันที่อยู่ในสัตว์เลี้ยงลูกด้วยนม (mammalians) ซึ่งเป็นเพียงข้อมูลแรกของจิกซอว์ภาพใหญ่ ปัจจุบันได้ค้นพบยีนที่เกี่ยวข้องกับนาฬิกาเซอร์คาเดียนอีกหลายยีนตัวอย่างเช่น ยีนชื่อ *timeless*, *clock*, *cycle* ที่ทำหน้าที่ควบคุมและประสานการทำงานของยีนอื่นๆอีกหลายร้อยยีนรวมทั้งมีความอนุรักษ์ของยีนเหล่านี้ในเชิงวิวัฒนาการ (Evolutionary conservation) ระหว่างสปีชีส์

ในกรณีของพืชนาฬิกาเซอร์คาเดียนมีความสำคัญมากเพราะหมายถึงการอยู่รอด มีการพิจารณาว่ามียีนอะไรบ้างในพืชที่มีการแสดงออกหรือการทำงานในช่วงเช้า เช่น ตอนพระอาทิตย์ขึ้นกับช่วงหลังพระอาทิตย์ตกดินไปแล้ว นักชีววิทยาได้มีการประมาณว่ามีมากกว่า 1000 ยีนที่มีความเกี่ยวข้องกับนาฬิกาชีวิตในพืช โดยยีนเหล่านี้อยู่ในกระบวนการสังเคราะห์แสง (photosynthesis) และการออกดอก (flowering) เป็นต้น คำถามคือยีนเหล่านี้ทราบได้อย่างไรว่าเป็นช่วงไหนของวันและควรเป็นเวลาที่ยีนนั้นๆ ควรแสดงออกและทำงาน ในการศึกษาทดลองพบว่าเซลล์ของพืชทุกๆ เซลล์เป็นอิสระต่อกันในการตรวจสอบช่วงเวลาของวันและมียีนหลัก 3 ยีนคือ LHY (LATE ELOGATED HYPOCOTYL), CCA1 (CIRCADIAN CLOCK ASSOCIATED 1) และ TOC1 (TIMING OF CAB EXPRESSION 1) โดย TOC1 จะควบคุมการแสดงออกของ LHY และ CCA1 ในขณะที่ LHY และ CCA1 ก็สามารถยับยั้งการแสดงออกของยีน TOC1 ได้ เกิดเป็นลูปเรียกว่า negative feedback loop โดยในตอนเช้าแสงแดดจะกระตุ้นการแสดงออกของยีน LHY และ CCA1 ซึ่งก็จะไปยับยั้งการแสดงออกของยีน TOC1 เมื่อตกเย็นแสงแดดไม่มีแล้วการแสดงออกของยีน LHY และ CCA1 ก็จะลดลงทำให้การแสดงออกของยีน TOC1 มีมากขึ้นและมีมากที่สุดในช่วงกลางคืนและมีหน้าที่เพิ่มการแสดงออกของยีน LHY และ CCA1 ซึ่งมีเส้นทางย้อนกลับในการควบคุมการแสดงออกของยีน TOC1 อีกที (รูปที่ 4.2) โปรตีนของทั้งสามยีนคือ LHY, CCA1, และ TOC1 เป็นทรานสคริปชันแฟคเตอร์ (transcription factor: TF) มีหน้าที่ควบคุมการทำงานของยีนอื่นๆอีกที โดยจะไปจับกับตำแหน่งจำเพาะของยีนเป้าหมายเรียกว่าเรียกว่าเรกูลาทอรีโมทิฟ (regulatory motif) หรือเรียกสั้นๆว่าโมทิฟ หรือทรานสคริปชันแฟคเตอร์ไบนดิงไซต์ (transcription factor binding site: TFBS) ซึ่งโดยทั่วไปจะอยู่ด้านหน้าของยีนที่เรียกว่า upstream region โดยครอบคลุมช่วงประมาณ 600-1,000 เบส รวมบริเวณที่เป็นโปรโมเตอร์ โดยยีนเป้าหมายหลายๆยีนที่มี CCA1 มาจับจะมีโมทิฟคือ **AAAAAATCT** ร่วมกัน

การหาโมทิฟข้างต้นจะมีความซับซ้อนเพิ่มขึ้นในกรณีที่ยีนเป้าหมายแต่ละยีนของทรานสคริปชันแฟคเตอร์เดียวกันอาจมีโมทิฟที่แตกต่างกันได้ในบางเบสโดยไม่จำเป็นต้องเหมือนกัน 100% ตัวอย่างเช่น CCA1 อาจจับได้กับโมทิฟ **AAGAACTCT** นอกจากนี้ถ้าเราไม่ทราบว่ามาก่อนว่าโมทิฟของทรานสคริปชันแฟคเตอร์ หนึ่งๆ มีรูปแบบอย่างไร เราจะหาโมทิฟในชุดของสาย upstream region ของยีนเป้าหมายได้อย่างไร ในบทเรียน นี้เราจะ

ศึกษาอัลกอริทึมที่ใช้ในการหาโมทิฟ (motif finding) โดยพยายามหารูปแบบจำเพาะหรือข้อความสั้นๆ (โมทิฟ) ที่ซ่อนอยู่ และพบอยู่ร่วมกันในชุดของสาย upstream region ของยีนเป้าหมาย



รูปที่ 4.2 ยีนหลักๆที่เกี่ยวข้องกับนาฬิกาเซอร์คาเดียนในพืช

(ที่มา: By Bjholm - Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=12116856)

ความซับซ้อนของการหาโมทิฟ

การหา Evening element

ในปีค.ศ. 2000 สตีฟ เคย์ (Steve Kay) ใช้ดีเอ็นเออะเรย์ในการตรวจสอบว่ามียีนใดบ้างในพืชใบเลี้ยงคู่ *Arabidopsis thaliana* ที่มีการแสดงออกในช่วงเวลาต่างๆของวัน จากนั้นเคย์ได้ทำการสกัดข้อมูลเฉพาะส่วนที่เป็น upstream regions ของประมาณ 500 ยีนที่มีการแสดงออกในลักษณะที่สัมพันธ์กับวงรอบเวลาเซอร์คาร์เดียน และทำการหารูปแบบลำดับเบสจำเพาะที่พบบ่อยเป็นพิเศษใน upstream regions ของ 500 ยีนนี้ ทั้งนี้เคย์พบว่าลำดับเบส **AAAATATCT** ถูกพบบ่อยเป็นพิเศษและโดยพบทั้งสิ้น 46 ครั้ง

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

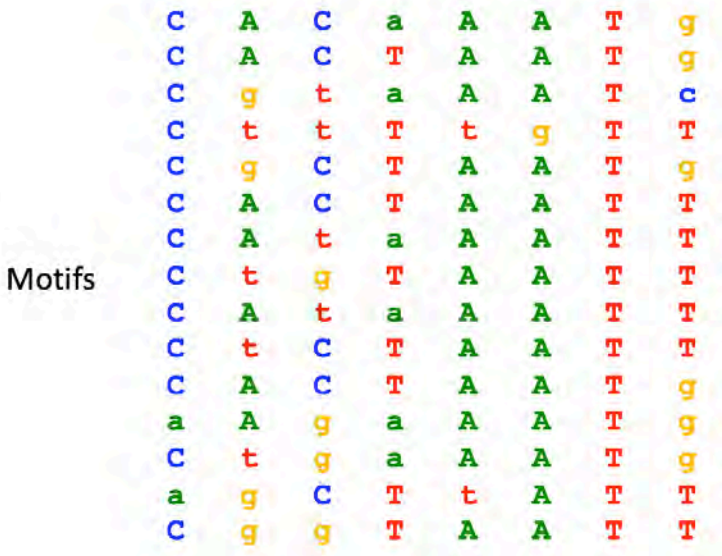
ทั้งนี้คีย์ได้เรียกโมทิฟนี้ว่า evening element และได้ทำการทดลองเพิ่มเติมเพื่อยืนยันว่าโมทิฟนี้มีผลต่อการแสดง

ออกของยีนที่เกี่ยวข้องกับนาฬิกาเซอร์คาร์เดียนใน *Arabidopsis thaliana* จริง โดยทำการเปลี่ยนลำดับเบสในบริเวณที่เป็น evening element ของหนึ่งยีนและวัดการแสดงออก ซึ่งพบว่าไม่มีการแสดงออกในลักษณะที่เกี่ยวข้องกับนาฬิกาเซอร์คาร์เดียนอีกต่อไป รูปที่ 4.3 แสดงโมทิฟของยีน CCA1 โดยมีที่มาจากฐานข้อมูล JASPAR 2018

รูปที่ 4.3 โมทิฟของยีน CCA1

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

ในขณะที่การหา evening elements ในยีนเป้าหมายของพืชอาจไม่ซับซ้อนนัก เนื่องจากรูปแบบของโมทิฟนั้นค่อนข้างมีความอนุรักษ์ (conserved) ระหว่างโมทิฟที่พบใน upstream region ของแต่ละยีน อย่างไรก็ตามถ้าพิจารณาบริเวณของยีนเป้าหมายที่ถูกจับโดยทรานสคริปชันแฟคเตอร์ HOXA5 (Homeobox protein Hox-A5) ซึ่งมีหน้าที่ในส่วนควบคุมพัฒนาการของเซลล์ที่เกี่ยวข้องกับแกนเส้นผ่าศูนย์กลางหน้าหลัง (anterior-posterior axis) โดยโมทิฟของ HOXA5 ขนาด 8-mer แสดงในรูปที่ 4.4



รูปที่ 4.4 ตัวอย่างของโมทิฟของ HOXA5 ที่พบในยีนเป้าหมายโดยตัวอักษรใหญ่ในแต่ละคอลัมน์ระบุเบสที่พบถี่สุดในคอลัมน์นั้นๆ

(สร้างจากข้อมูลของ http://jaspar.genereg.net/matrix/MA0158.1/)

เป้าหมายของเราคือการแปลงตัวอย่างโจทย์ทางชีววิทยาข้างต้นให้อยู่ในรูปแบบที่สามารถแก้ปัญหาได้โดยคอมพิวเตอร์ พิจารณาสายดีเอ็นเอ 10 เส้นที่ถูกจำลองขึ้นต่อไปนี้ โดยแต่ละเส้นมีการแทรกลำดับเบสแบบเดียวกัน aaaaaaaaggggggg ขนาด 15-mer เข้าไปโดยการสุ่มตำแหน่ง สายดีเอ็นเอที่ถูกจำลองขึ้นนี้เปรียบได้กับ upstream regions ของ 10 ยีนที่มีการแสดงออกร่วมกัน และลำดับเบส 15-mer ที่แทรกเข้าไปเทียบได้กับทรานสคริปชันแฟคเตอร์ไบนดิงไซต์หรือโมทิฟของยีนเหล่านี้นั่นเอง

```
1  atgaccgggatactgataaaaaaaaggggggggcgtacacattagataaacgtatgaagtacgttagactcgggcccgccdcg
2  acccctatttttgagcagatttagtgacctggaaaaaaaatttgagtacagaaacttttccgaataaaaaaaggggggga
3  tgagtatccctgggatgacttaaaaaaaaggggggggtgctctcccgatttttgaatatgtaggatcattcgccagggtccga
4  gctgagaattggatgaaaaaaaaggggggggtccacgcaatcgcgaaccaacgcggacccaaaggcaagaccgataaaggaga
5  tccctttgcggtaatgtgccgggaggctggttacgtagggaagccctaacggacttaataaaaaaaaggggggggcttatag
6  gtcaatcatgttcttgtgaatggatttaaaaaaaaggggggggaccgcttggcgcacccaaattcagtgtgggcgagcgcaa
7  cggttttggcccttgttagaggcccccgtaaaaaaaaggggggggcaattatgagagagctaatctatcgcgtgcgtgttcat
8  aacttgagttaaaaaaaaggggggggctggggcacatacaagaggagtcttccttatcagttaatgctgtatgacactatgta
9  ttggcccattggctaaaagcccaacttgacaaatggaagatagaatccttgcataaaaaaaaggggggggaccgaaagggaag
10 ctggtgagcaacgacagattcttacgtgcattagctcgcttccggggatctaatagcacgaagcttaaaaaaaaggggggga
```

หยุดคิด	จากดีเอ็นเอที่จำลองขึ้น 10 เส้นข้างต้น เราจะหาลำดับเบส 15-mer ที่เราแทรกเข้าไปแบบสุ่มได้อย่างไร

ปัญหานี้สามารถแก้ได้โดยง่ายโดยการนำสายดีเอ็นที่จำลองขึ้นนี้มาต่อกันแล้วหาคำที่มีความถี่มากที่สุด ซึ่งจะได้ผลลัพธ์ดังต่อไปนี้

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

เนื่องจากดีเอ็นเอที่จำลองขึ้นมานี้แต่ละเบสก็ถูกสร้างขึ้นแบบสุ่ม ดังนั้นโอกาสที่ 15-mer อื่นๆ จะปรากฏบ่อยเป็นไปได้ยาก อย่างไรก็ตามถ้าลำดับเบส 15-mer ที่แทรกเข้าไปนี้สามารถมีความแตกต่างกันได้ 4 เบสดังตัวอย่างต่อไปนี้ วิธีการหาโมทิฟโดยหาความถี่ของคำจะไม่สามารถนำมาประยุกต์ใช้ได้

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

การหาโมทิฟโดยวิธีการ Brute force

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

นิยามปัญหาที่ 4.1 ปัญหาการหาโมทิฟที่แทรกอยู่ในสายดีเอ็นเอ

ปัญหาการหาโมทิฟที่แทรกอยู่ในสายดีเอ็นเอ (Implanted Motif Problem) หาโมทิฟ (k,d) ทั้งหมดที่อยู่ในสายเอ็นเอโดยที่ k คือความยาวโมทิฟและ d คือจำนวนเบสที่สามารถแตกต่างกันได้ระหว่างโมทิฟ	
ข้อมูลเข้า	ชุดของสายดีเอ็นเอ *Dna* และจำนวนเต็ม k และ d
ผลลัพธ์	ชุดของโมทิฟ (k,d) ทั้งหมดที่พบในสายดีเอ็นเอ

การหาโมทิฟโดยวิธีการ Brute force search หรือ exhaustive search เป็นการแก้ปัญหาโดยการตรวจสอบคำตอบที่เป็นไปได้ทั้งหมดว่าคำตอบใดบ้างถูกต้อง โดยอัลกอริทึมในกลุ่มนี้ไม่มีการออกแบบมากนักและสามารถหาคำตอบที่ถูกต้องได้แต่ปัญหาหลักคือเวลาที่ต้องใช้ในการตรวจสอบคำตอบโดยเฉพาะในกรณีที่มีจำนวนคำตอบที่เป็นไปได้มากมายมหาศาล สำหรับการหาโมทิฟโดยวิธีการ Brute force นี้ เราสามารถสร้างคำตอบที่เป็นไปได้ทั้งหมดจาก k-mer ใดๆในสายดีเอ็นเอ โดยคำตอบที่เป็นไปได้ทั้งหมดเหล่านั้นต้องต่างจาก k-mer นั้นๆ อย่างมาก d เบส จากนั้นทำการตรวจสอบว่าแต่ละคำตอบที่เป็นไปได้นี้ปรากฏในทุกสายของดีเอ็นเอหรือไม่ ถ้าใช่ก็เก็บคำตอบที่เป็นไปได้คำตอบนี้เข้าเป็นสมาชิกของชุดโมทิฟ (k,d) ที่เป็นคำตอบสุดท้าย ตามตัวอย่างสูโดโคดที่ 4.1 ต่อไปนี้

สูโดโคดที่ 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
```

จากสูโดโคดข้างต้นจะเห็นว่าวิธีการนี้จะใช้เวลาในการหาโมทิฟนานมากโดยเฉพาะเมื่อ k และ d มีค่ามาก เพื่อเป็นการลดเวลาในการหาโมทิฟ ได้มีการเสนอวิธีการหาโมทิฟโดยการหา k-mer สองสายที่มีความแตกต่าง กันมากสุด d เบสจากคู่ของสายดีเอ็นเอ อย่างไรก็ตามวิธีการนี้ไม่สามารถแก้ปัญหาได้โดยพิจารณาตัวอย่าง 15-mer สองสายนี้ **AgAAgAAAGGttGGG** และ **cAAtAAAAcGGGGcG** ซึ่งแต่ละเส้นต่างจาก k-mer ที่ถูกต้อง **AAAAAAAAGGGGGGG** 4 เบส แต่ k-mer สองสายนี้เองต่างกันถึง 8 ตำแหน่งตามภาพข้างล่าง ซึ่งถ้า k-mer สองสายนี้เป็นตัวอย่างคำตอบสุดท้าย วิธีการหาโมทิฟ (k,d) โดยการพยายามหา k-mer ในดีเอ็นเอสองสายที่ต่างกันมากสุด d เบสก็จะไม่ได้คำตอบสุดท้ายข้างต้น

ฝึกหัด	จงเขียนโคดเพื่อสร้างสายดีเอ็นยาว 600 เบส จำนวน 10 เส้น โดยการสุ่มแต่ละเบสมาจาก “A”, “T”, “C” และ “G” โดยมีโอกาสที่จะเกิดแต่ละเบสเท่าๆกัน จากนั้นทำการสร้างโมทิฟที่เป็นคำตอบสุดท้ายขนาด 15-mer โดยแต่ละเบสสุ่มมาแบบเดียวกับการสร้างสายดีเอ็นเอข้างต้น และโมทิฟที่สร้างขึ้นนี้สามารถต่างจากโมทิฟ **AAAAAAAA GGGGGGG** ได้มากสุด 4 เบส จากนั้นทำสุ่มตำแหน่งในสายดีเอ็นเอเส้นที่หนึ่งและทำการแทรกโมทิฟที่สร้างขึ้นมานี้ในตำแหน่งนั้น ทำการสร้างโมทิฟที่เป็นคำตอบสุดท้ายและทำการแทรกในดีเอ็นเอสายถัดไปจนครบ 10 เส้น จากนั้น ใช้สายดีเอ็นเอที่มีการแทรกโมทิฟแล้วเป็นข้อมูลเข้า และลองทำการหาโมทิฟโดยการพยายามหา 15-mer ที่เหมือนกันที่สุดระหว่างดีเอ็นสองสายใดๆที่เป็นข้อมูลเข้า และรายงานผล

การให้คะแนนโมทิฟ

การใช้ชุดของโมทิฟเพื่อสร้างโปรไฟล์เมทริกซ์และสายสตริงเสียงข้างมาก

การอธิบายปัญหาการหาโมทิฟโดยใช้ตัวอย่างการแทรกโมทิฟเข้าไปในสายดีเอ็นเอข้างต้นมีข้อจำกัด เนื่องจากวิธีการหาโมทิฟข้างต้นมีสมมติฐานว่าสายดีเอ็นเอที่เป็นข้อมูลเข้าทั้งหมดจะต้องมีโมทิฟที่ถูกต้องแทรกอยู่ ซึ่งสมมติฐานนี้ *ไม่เป็นจริง* ในมุมมองของนักชีววิทยา ตัวอย่างเช่น การทดลองของสตีฟ เคย์ ที่ใช้ดีเอ็นเออะเรย์ในการตรวจสอบว่ามียีนใดบ้างใน *Arabidopsis thaliana* ที่มีการแสดงออกในช่วงเวลาต่างๆของวันนั้น เคย์ไม่ได้คาดหวังว่า upstream region ของทุกยีนจะต้องมี evening element ทั้งนี้ด้วยเหตุผลจากการทดลองดีเอ็นเออะเรย์ที่มักมีสัญญาณรบกวนและยีนหลายยีนที่แสดงออกในรูปแบบเดียวกับกลุ่มยีนเป้าหมายอาจไม่เกี่ยวข้องกับการทำงานกับนาฬิกาเซอร์คาร์เดียนก็ได้ ในกรณีของการหาชุดของยีนเป้าหมายของ HOXA5 ก็เช่นกัน

ด้วยเหตุผลข้างต้นวิธีการในการหาโมทิฟได้ถูกนำเสนอในแนวทางที่สอดคล้องกับลักษณะข้อมูลการทดลองมากขึ้นโดยใช้การให้คะแนนโมทิฟหนึ่งๆ ที่สามารถถูกจับได้โดยทรานสคริปชันแฟคเตอร์ เทียบกับโมทิฟที่เป็นอุดมคติคือสามารถถูกจับได้โดยทรานสคริปชันแฟคเตอร์ได้ดีที่สุด อย่างไรก็ตามในทางปฏิบัติเรามักไม่ทราบโมทิฟที่เป็นอุดมคตินั้น ดังนั้นแนวทางที่เป็นไปได้คือพยายามเลือก k-mer จากแต่ละสายดีเอ็นเอและพยายามให้คะแนนโมทิฟเหล่านี้จากความเหมือนกับโมทิฟอื่นๆที่ถูกเลือกมาเช่นกัน

ในการกำหนดวิธีการให้คะแนน ข้อมูลที่เกี่ยวข้องประกอบด้วย t คือจำนวนสายดีเอ็นเอ โดยแต่ละสายยาว n เบส และทำการเลือก k-mer จากดีเอ็นเอแต่ละสาย เพื่อให้ได้ชุดของโมทิฟ *Motifs* ซึ่งถูกแสดงด้วย โมทิฟเมทริกซ์ (motif matrix) ขนาด $t \cdot k$ ดังแสดงในรูปที่ 4.5 ซึ่งเป็นตัวอย่างของโมทิฟเมทริกซ์ของตำแหน่งจับบนสายดีเอ็นเอของ HOXA5 จากรูปที่ 4.4 โดยในรูปที่ 4.5 นี้ตัวอักษรใหญ่ในแต่ละคอลัมน์แสดงเบสที่พบบ่อยที่สุดในตำแหน่งนั้นๆ โดยในตัวอย่างนี้คอลัมน์ที่ 1, 5, 6, และ 7 มีความอนุรักษ์ของเบสมากโดยในคอลัมน์ที่ 7 นั้นเป็นเบส “T” ทั้งหมด ในขณะที่คอลัมน์ที่ 2 และ 3 มีความอนุรักษ์ของเบสจำเพาะน้อยสุด เราสามารถสร้างโมทิฟเมทริกซ์ได้มากมายโดยการเลือกชุดของ k-mers จาก t ดีเอ็นเอ อย่างไรก็ตามเป้าหมาย ของเราคือการเลือก k-mers ที่ทำให้ได้โมทิฟเมทริกซ์ที่มีความอนุรักษ์มากที่สุด หรืออีกนัยยะคือมีอักษรตัวใหญ่ มากที่สุดในทุกคอลัมน์ จากรูปที่ 4.5 ฟังก์ชัน SCORE(*Motifs*) จะแสดงค่าผลรวมของตัวอักษรเล็กทั้งหมด ที่ปรากฏอยู่ในโมทิฟเมทริกซ์ โดย SCORE(*Motifs*) นี้สามารถนำมาใช้ในการวัดความอนุรักษ์ของโมทิฟ เมทริกซ์ซึ่งยิ่งมีค่าน้อยก็หมายความว่าโมทิฟเมทริกซ์นั้นๆ มีความอนุรักษ์มากนั่นเอง

ฝึกหัด	ค่าน้อยสุดที่เป็นไปได้ของ SCORE(*Motifs*) คือ 0 ซึ่งหมายความว่าเบสในแต่ละคอลัมน์เหมือนกันทุกบรรทัด คำถามคือค่าที่มากที่สุดที่เป็นไปได้ของ SCORE(*Motifs*) เป็นเท่าไหร่ โดยแสดงค่าในรูปแบบของตัวแปร t และ k ข้างต้น

Motifs	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
SCORE(Motifs)		2	+ 8	+ 8	+ 6	+ 2	+ 1	+ 0	+ 7 = 34
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
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
CONSENSUS(Motifs)		C	A	C	T	A	A	T	T

รูปที่ 4.5 โมทิฟเมทริกซ์ ผลของ SCORE(Motifs) COUNT(Motifs) โปรไฟล์เมทริกซ์และสายสตริงเสียงข้างมาก ของ HOXA5

(ที่มา: Sequence Logo มาจาก http://jaspar.genereg.net/matrix/MA0158.1/)

เราสามารถสร้างเมทริกซ์ 4 x *k* ที่แสดงจำนวนครั้งที่แต่ละเบสปรากฏในแต่ละคอลัมน์ โดยแสดงผ่านฟังก์ชัน `COUNT(Motifs)` นอกจากนี้ถ้าเรานำจำนวน t มาหารแต่ละค่าที่ปรากฏใน เมทริกซ์ 4 x *k* ก็ได้เมทริกซ์ใหม่เรียกว่า โปรไฟล์เมทริกซ์ *P* = `PROFILE(Motifs)` ซึ่งแต่ละค่าในโปรไฟล์เมทริกซ์แสดงค่าความถี่ของการเกิดเบสจำเพาะหนึ่งในแต่ละคอลัมน์ และท้ายสุดเราสามารถสร้างสายสตริงเสียงข้างมาก (consensus string) ผ่านฟังก์ชัน `CONSENSUS(Motifs)` จากเบสที่พบบ่อยที่สุดในแต่ละคอลัมน์ ถ้าเราสามารถ เลือก k-mers ที่ถูกต้องจากสาย upstream regions ผลของ `CONSENSUS(Motifs)`ก็จะได้เรกูลาทอรีโมทิฟ (regulatory

motif) ที่เป็นอุดมคติของ upstream regions ชุดนี้ ตัวอย่างจากรูปที่ 4.5 สายสตริงเสียงข้างมากซึ่งเป็นตำแหน่งที่ HOXA5 มาจับคือ CACTAATT

การปรับปรุงการให้คะแนน

จากรูปที่ 4.5 ถ้าพิจารณาคอลัมน์ที่ 2 และคอลัมน์สุดท้ายของโมทิฟเมตริกซ์จะพบว่าทั้งสองคอลัมน์มี 4 คะแนนเท่ากันโดยคอลัมน์ที่ 2 มีเบส t และ a เป็นเสียงข้างน้อยอย่างละ 2 คะแนน ในขณะที่คอลัมน์สุดท้ายทั้ง 4 คะแนนมาจาก t สี่ตัว

หยุดคิด	4 คะแนนของสองคอลัมน์ที่เท่ากันนี้มีความหมายทางชีววิทยาเท่ากันไหม

ในทางชีววิทยาบางตำแหน่งของโมทิฟหรือไบนดิงไซต์ที่ทรานสคริปชันแฟคเตอร์หนึ่งๆมาจับอาจเป็นนิวคลีโอไทด์ ได้มากกว่า 1 แบบ ตัวอย่างเช่น โมทิฟ CSRE ในยีสต์ (*Saccharomyces cerevisiae*) ที่ยาว 16-mer ประกอบด้วย 5 ตำแหน่งที่มีความอนุรักษ์มาก ในขณะที่อีก 11 ตำแหน่งจะมีความอนุรักษ์น้อยโดยแต่ละตำแหน่งเหล่านี้มีเบสที่แตกต่างกันดังต่อไปนี้

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C	G/C	G/T	T/A	C/T	G/C	C/G	A	T	G/T	C/G	A	T	C/T	C/T	G/T

รูปที่ 4.6 โมทิฟ CSRE ในยีสต์ (Saccharomyces cerevisiae) ที่มีความอนุรักษ์มากเพียง 5 ตำแหน่ง (1, 8, 9, 12, และ 13) จาก 16 ตำแหน่ง ในขณะที่ 11 ตำแหน่งที่เหลือนั้นแต่ละตำแหน่งสามารถเป็นนิวคลีโอไทด์ได้สองประเภท

(ที่มา: รูปที่ 2.3 ของ [21])

จากตัวอย่างข้างต้น บางตำแหน่งจะเป็นนิวคลีโอไทด์ได้สองประเภท ซึ่งมีความถี่ในการปรากฏเท่ากันดังแสดงในรูปที่ 4.6 จากตัวอย่างการแสดงสายสตริงเสียงข้างมากของ CSRE ในยีสต์ สายสตริงเสียงข้างมากของ HOXA5 ข้างต้น สามารถปรับให้ละเอียดมากขึ้นดังต่อไปนี้

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

รูปที่ 4.7 โมทิฟของ HOXA5 ที่ถูกแสดงโดยสายสตริงเสียงข้างมากที่มีการแสดงนิวคลีโอไทด์ที่เป็นไปได้ในแต่ละตำแหน่ง

เอนโทรปีและโลโก้โมทิฟ

พิจารณาโปรไฟล์เมทริกซ์ในรูปที่ 4.5 แต่ละคอลัมน์แสดงค่าการกระจายตัวของค่าความน่าจะเป็นที่ตำแหน่งนั้นจะเป็นนิวคลีโอไทด์แต่ละแบบ และผลรวมของทุกแถวจะต้องเป็น 1 ตัวอย่างเช่นในคอลัมน์ที่สองหรือตำแหน่งที่

สองของโมทิฟมีความน่าจะเป็นที่จะเป็นนิวคลีโอไทด์ “A”, “C”, “G” และ “T” เท่ากับ 0.46, 0.0, 0.27 และ 0.27 ตามลำดับ

เอนโทรปี (Entropy) ใช้วัดค่าความไม่แน่นอนของการกระจายตัวของค่าความน่าจะเป็น (p_1,..., p_N) โดยแสดงด้วยสูตรต่อไปนี้

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

ตัวอย่างเช่นค่าเอนโทรปีของการกระจายตัวของค่าความน่าจะเป็น (0.46, 0.0, 0.27, 0.27) ในคอลัมน์ที่สองมีค่าเท่ากับ

$$-(0.46 \log_2 0.46 + 0.0 \log_2 0.0 + 0.27 \log_2 0.26 + 0.27 \log_2 0.27) \approx 1.54$$

ในขณะที่ค่าเอนโทรปีของคอลัมน์ที่มีความอนุรักษ์มากกว่าอย่างคอลัมน์ที่หนึ่งที่มีการกระจายตัวของค่าความน่าจะเป็น (0.13, 0.87, 0.0, 0.0) มีค่าเท่ากับ

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

และค่าเอนโทรปีของคอลัมน์ที่มีความอนุรักษ์มากอย่างคอลัมน์ที่ 6 ที่มีการกระจายตัวของค่าความน่าจะเป็น (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 $0 \log_2 0$ เรากำหนดให้มีค่า เท่ากับ 0

หยุดคิด	ค่าเอนโทรปีที่มากที่สุดและที่น้อยที่สุดเป็นเท่าไหร่ ถ้าการกระจายตัวของค่าความน่าจะเป็นในแต่ละคอลัมน์มีค่าที่เป็นไปได้ทั้งหมดสี่แบบ

สำหรับค่าเอนโทรปีของคอลัมน์ที่มีการอนุรักษ์มากที่สุด ตัวอย่างเช่นคอลัมน์ที่ 7 มีค่าเท่ากับ 0 ซึ่งเป็น ค่าเอนโทรปีน้อยที่สุดที่เป็นไปได้ ในทางกลับกันคอลัมน์ที่มีการปรากฏของนิวคลีโอไทด์ทั้งสี่ตัวเท่ากัน (นิวคลีโอ ไทด์ละ ¼) มีค่าเอนโทรปีเท่ากับ $-4 \cdot \frac{1}{4} \cdot \log_2 \frac{1}{4} = 2$ โดยทั่วไปคอลัมน์ที่มีความอนุรักษ์มากกว่าจะมีค่าเอนโทรปีน้อยกว่า ดังนั้นเราสามารถใช้ค่าเอนโทรปีในการปรับปรุงการให้คะแนนโมทิฟเมทริกซ์ โดยค่าเอนโทรปีของ โมทิฟเมทริกซ์จะเท่ากับผลบวกของค่าเอนโทรปีในแต่ละคอลัมน์ การใช้เอนโทรปีเพื่อการกำหนดคะแนนให้กับโม ทิฟเมทริกซ์ถูกใช้มากกว่าการนับ SCORE(*Motifs*) ที่อธิบายไปก่อนหน้า อย่างไรก็ตามเพื่อให้การอธิบาย ในหัวข้อถัดๆไปไม่ซับซ้อน เราจะยังคงใช้ SCORE(*Motifs*)โดยไม่ได้ใช้เอนโทรปี

ฝึกหัด	จงคำนวณค่าเอนโทรปีของโมทิฟเมทริกซ์ HOXA5 ที่แสดงในรูปที่ 4.5

sequence logo หรือ motif logo ที่แสดงในรูปที่ 4.5 ข้างต้นเป็นตัวอย่างของการประยุกต์ใช้เอนโทรปี โดย sequence logo จะแสดง (visualize) ความอนุรักษ์ของนิวคลีโอไทด์จำเพาะในแต่ละตำแหน่งในรูปแบบของตัวอักษรที่กองซ้อนกันอยู่ (stack) ทั้งนี้ขนาดของตัวอักษรแต่ละตัวในกองซ้อนแต่ละคอลัมน์บ่งชี้ความถี่ของการปรากฏของนิวคลีโอไทด์นั้นๆ ในตำแหน่งนั้นและความสูงรวมของกองซ้อนในแต่ละคอลัมน์ขึ้นอยู่กับเนื้อหาข้อมูล (information content) ในคอลัมน์นั้นๆ ซึ่งคำนวณจาก $2 - H(p_1, \dots, p_N)$ โดยยิ่งมีค่าเอนโทรปีน้อยก็จะมีค่าเนื้อหาข้อมูลมาก หรือหมายถึงคอลัมน์ที่มีความสูงมากจะมีความอนุรักษ์มาก

การหาโมทิฟโดยวิธีการหามีเดียนสตริง

หลังจากได้มีการกำหนดการให้คะแนนของชุดของ k-mers แล้ว เราสามารถนิยามปัญหาการหาโมทิฟได้ดังต่อไปนี้

นิยามปัญหาที่ 4.2 ปัญหาการหาโมทิฟ

ปัญหาการหาโมทิฟ (Motif Finding Problem) รับข้อมูลเป็นชุดของสตริง ให้หาหนึ่ง k-mer จากแต่ละสตริง เพื่อมาสร้างโมทิฟเมทริกซ์ที่มีค่าคะแนนน้อยที่สุด	
ข้อมูลเข้า	ชุดของสตริง *Dna* และค่าจำนวนเต็ม *k*
ผลลัพธ์	ชุดของโมทิฟ (*Motifs*) ที่มีขนาด k-mer โดยแต่ละ k-mer มาจากแต่ละสตริงใน *Dna* โดยให้ค่าของ SCORE(*Motifs*)น้อยที่สุด

จากปัญหาข้างต้น ถ้าใช้วิธีการ Brute force ในการหาโมทิฟ จะต้องมีพิจารณาชุดของ k-mers ทั้งหมดที่เป็นไปได้ และดูว่าชุดของ k-mers ไหนที่ให้คะแนนโมทิฟน้อยที่สุด เนื่องจากสตริงแต่ละเส้นมี k-mers ที่เป็นไปได้ทั้งหมด *n*-*k*+1 สายและเนื่องจาก *Dna* มีทั้งหมด *t* เส้น จะมีชุดของ k-mers เพื่อสร้าง *Motifs* ทั้งสิ้น $(n\text{-}k+1)^t$ ชุด และเมื่อได้แต่ละโมทิฟที่เป็นไปได้แล้วต้องนำมาคำนวณคะแนนโดยใช้ฟังก์ชัน SCORE(*Motifs*)ซึ่งต้องใช้ อีก $k \cdot t$ ขั้นการคำนวณ ดังนั้น ถ้ามีสมมติฐานว่า *k* สั้นกว่า *n* มาก เวลาที่ใช้ในการหาโมทิฟโดยวิธีการ Brute force ข้างต้นจะเท่ากับ $O(n^t \cdot k \cdot t)$ ซึ่งช้ามาก

กำหนดแนวทางการปัญหาใหม่อีกครั้ง

จากปัญหาความไม่มีประสิทธิภาพของวิธีการ Brute force ข้างต้น เราเริ่มจากการหาชุดของ k-mers เพื่อสร้างเป็นโมทิฟเมทริกซ์ที่มีคะแนนน้อยสุดและหาโมทิฟที่เป็นคำตอบผ่านการหาสายสตริงเสียงข้างมากจากโมทิฟเมทริกซ์ที่มีคะแนนน้อยสุดตามลักษณะการทำงานต่อไปนี้

Motifs -> CONSENSUS(*Motifs*)

เพื่อเป็นการลดเวลาในการหาโมทิฟโดยวิธีการ Brute force ข้างต้น ลองพิจารณาว่าเราจะสามารถสร้างโมทิฟเมทริกซ์ย้อนกลับจากสตริงเสียงข้างมากได้หรือไม่ พิจารณารูปที่ 4.8 ต่อไปนี้จะพบว่า SCORE(*Motifs*) เท่ากับ 34 ที่คำนวณมาก่อนหน้าในรูปที่ 4.5 โดยนับผลรวมของอักษรตัวเล็กในแต่ละคอลัมน์บวกกันนั้นได้ค่าเท่ากับการหาผลรวมของอักษรที่เล็กที่นับได้ในแต่ละแถว และถ้าพิจารณาให้ละเอียดมากขึ้นจะพบว่าอักษรตัวเล็กที่นับได้ในแต่ละแถวนั้นแสดงความแตกต่างหรือระยะทางแฮมมิง (Hamming distance) ของโมทิฟสายนั้นเทียบกับสตริงเสียงข้างมากนั่นเอง 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 แสดงผลการคำนวณคะแนนโมทิฟเมทริกซ์ผ่าน SCORE(Motifs) ซึ่งเป็นผลบวกของอักษรตัวเล็กตามคอลัมน์เทียบกับผลบวกของอักษรตัวเล็กนับตามแถว โดยอักษรตัวเล็กในแต่ละแถวคือนิวคลีโอไทด์ที่ต่างจากนิวคลีโอไทด์ที่อยู่ในสายสตริงเสียงข้างมาก (consensus string) ในตำแหน่งเดียวกัน

ถ้ามีข้อมูลเข้าเป็นชุดของ k-mers *Motifs* = {$Motif_1$,..., $Motif_t$} และ k-mer ที่เป็น *Pattern* หรือโมทิฟที่เป็นสายสตริงเสียงข้างมาก เรานิยาม

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

และถ้าย้อนกลับไปพิจารณา SCORE(*Motifs*)ข้างต้น จะพบว่า

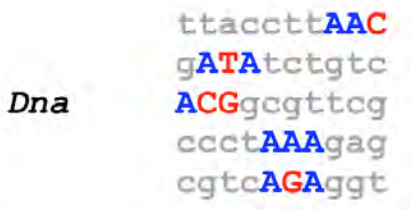
$$SCORE(Motifs) = d(CONSENSUS(Motifs), Motifs)$$

สมการนี้ได้ให้แนวคิดว่าแทนที่เราจะหาชุดของ k-mer *Motifs* ที่ทำให้ได้ SCORE(*Motifs*)น้อยสุด เราสามารถ

หาสายสตริงเสียงข้างมาก *Pattern* ที่ทำให้ค่า $d(Pattern, Motifs)$ มีค่าน้อยที่สุด จาก k-mer *Patterns* และ k-mer *Motifs* ทั้งหมดที่เป็นไปได้จากชุดของสายดีเอ็นเอ ปัญหานี้เทียบเท่ากับปัญหาการหา โมทิฟ (นิยามปัญหาที่ 4.2)

ปัญหามีเดียนสตริง

อาจมีความสงสัยว่าการแก้ปัญหาการหาโมทิฟโดยใช้สายสตริงเสียงข้างมากข้างต้นจะมีความซับซ้อนเพิ่มเติมขึ้นหรือไม่ เพราะว่านอกจากต้องพิจารณาชุดของ k-mers เพื่อสร้าง *Motifs* ทั้งหมดที่เป็นไปได้แล้วยังต้องพิจารณาสายสตริงเสียงข้างมากที่เป็นไปได้ทั้งหมดอีกด้วย อย่างไรก็ตามถ้ามีการพิจารณาการแก้ปัญหานี้ในรายละเอียดจะพบว่าเราไม่จำเป็นต้องพิจารณาทุกชุดของ k-mers ที่เป็นไปได้เพื่อหา $d(Pattern, Motifs)$ ที่มีค่าน้อยที่สุด เพื่อเป็นการอธิบายประโยคข้างต้น กำหนด $\mathtt{MOTIFS}(Pattern, Dna)$ เป็นชุดของ k-mers ที่ทำให้ $d(Pattern, Motifs)$ มีค่าน้อยที่สุดสำหรับ $Pattern$ จำเพาะหนึ่งๆ โดยทดสอบกับชุดของ k-mers ทั้งหมดที่เป็นไปได้ในชุดของสายดีเอ็นเอ ลองพิจารณาชุดของดีเอ็นเอต่อไปนี้ บริเวณ 3-mer ที่มีการใส่สีในทั้ง 5 บรรทัดแสดง $\mathtt{MOTIFS}(\mathtt{AAA}, Dna)$



หยุดคิด	ถ้ามีข้อมูลเข้าเป็นชุดของสายดีเอ็นเอ *Dna* และ *Pattern* ที่มีขนาด k-mer จงออกแบบอัลกอริทึมที่สามารถสร้าง $\mathtt{MOTIFS}(Pattern, Dna)$ ได้อย่างรวดเร็ว

เหตุผลที่เราไม่ต้องพิจารณาชุดของ k-mers เพื่อสร้าง *Motifs* ทั้งหมดที่เป็นไปได้เนื่องจากเราสามารถเลือก k-mers ใน $\mathtt{MOTIFS}(Pattern, Dna)$ โดย k-mer ที่ถูกเลือกจากแต่ละ Dna_i เป็นอิสระต่อกัน ถ้า $\mathtt{d}(Pattern, Text)$ 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 ที่มีระยะทางแฮมมิงน้อยที่สุดเพื่อเทียบกับ *Pattern* ถูกแสดงด้วย

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

นิยาม $\mathtt{MOTIF}()$ นี้มีความคลุมเครือได้ ซึ่งหมายถึงอาจมีมากกว่าหนึ่ง k-mers ที่มีระยะทางแฮมมิงน้อย

ที่สุดเมื่อเทียบกับ *Pattern* ตัวอย่างเช่น `MOTIF(AAG, gcAATcctCAGc)` มีทั้ง **AAT** และ **CAG** เป็นต้น อย่างไรก็ตามความคลุมเครือนี้ไม่มีผลต่อการหาโมทิฟ ดังนั้นถ้ามีข้อมูลสายสตริงเสียงข้างมาก *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 $d(AAA, Dna) = 1 + 1 + 2 + 0 + 1 = 5$

	ttaccttAAC	1
	gATAtctgtc	1
Dna	ACGgcgttcg	2
	ccctAAAgag	0
	cgtcAGAggt	1

เป้าหมายของเราคือหา k-mer *Pattern* ที่ทำให้ $d(Pattern, Dna)$ มีค่าน้อยสุดจากชุดของ k-mer *Patterns* ทั้งหมดที่เป็นไปได้ ซึ่งเป็นปัญหาที่เทียบเท่ากับปัญหาการหาโมทิฟก่อนหน้า โดย k-mer *Pattern* นี้เรียกว่ามีเดียนสตริง (median string)

นิยามปัญหาที่ 4.3 ปัญหามีเดียนสตริง

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

สังเกตว่าการหามีเดียนสตริงนี้เป็นปัญหาการทำออปติไมเซชัน (optimization) 2 ชั้น ชั้นแรกคือการหา *Pattern* ที่ดีที่สุดจาก *Patterns* ทั้งหมดที่เป็นไปได้ และชั้นที่สองฟังก์ชัน $d(Pattern, Dna)$ เองต้องหา *Pattern* จากชุดของ k-mers ทั้งหมดที่เป็นไปได้ในชุดของสายดีเอ็นเอที่ใกล้เคียงกับ *Pattern* หนึ่งๆที่นำมาเปรียบเทียบ สูโดโคดที่ใช้หามีเดียนสตริงโดยวิธีการ Brute force ถูกแสดงในสูโดโคดที่ 4.2

เปรียบเทียบวิธีการหาโมทิฟข้างต้น

ถึงแม้ว่าตัวอย่างวิธีการแก้ปัญหาที่ 4.2 และ 4.3 จนถึงจุดนี้ยังใช้แนวทาง Brute force แต่ถ้าเราเปรียบเทียบเวลาที่ใช้ในการหาโมทิฟจากการวิเคราะห์การแก้ปัญหาผ่านมีเดียนสตริง กับการใช้วิธีการ Bruce force โดยใช้โมทิฟเมทริกซ์จะพบว่า วิธีการมีเดียนสตริงใช้เวลา $O(4^k \cdot n \cdot k \cdot t)$ ในขณะที่วิธีการก่อนหน้าใช้เวลา $O(n^t \cdot k \cdot t)$ ทั้งนี้ในกรณีของวิธีการมีเดียนสตริง $d(Pattern, Dna)$ จะถูกเรียกใช้งานสำหรับ *Pattern* ทั้งหมด 4^k รูปแบบ ใน

สูโดโคดที่ 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
```

ขณะที่ในการทดสอบ *Pattern* แต่ละรูปแบบจะสแกนสายดีเอ็นเอแต่ละเส้น 1 ครั้ง ซึ่งจะใช้เวลาในโดยรวมในการสแกนดีเอ็นเอทุกเส้นโดยประมาณเท่ากับ $k \cdot n \cdot t$ โดย k คือความยาวของโมทิฟที่ต้องการหา n คือความยาวของสายดีเอ็นเอแต่ละเส้น และ t คือจำนวนเส้นของสายดีเอ็นเอ ซึ่งจะเห็นว่าวิธีการหาโมทิฟโดยการหา มีเดียนสตริงนั้นมีประสิทธิภาพดีกว่า เนื่องจาก k มักมีขนาดสั้นไม่เกิน 20 นิวคลีโอไทด์ในขณะยีนที่แสดงออกในเงื่อนไขหนึ่งๆ อาจมีจำนวนหลายร้อยเส้น

บทเรียนที่ได้จากตัวอย่างนี้คือการเปลี่ยนมุมมองหรือแนวทางการวิเคราะห์ปัญหา (เปลี่ยนจากการหาผลรวมในแนวคอลัมน์มาเป็นการหาผลรวมในแนวแถว ซึ่งได้ผลรวมเท่ากัน) อาจนำมาซึ่งวิธีการในการแก้ปัญหาที่ดีกว่าถึงแม้ว่ายังใช้แนวทางในการแก้ปัญหาแบบเดิม (หาคะแนนรวมน้อยสุด) อย่างไรก็ตามวิธีการหาโมทิฟโดยการหามีเดียนสตริงยังมีประสิทธิภาพไม่เพียงพอในการนำมาใช้งานจริง เนื่องจากต้องมีการพิจารณาจำนวนรูปแบบที่เป็นไปได้ทั้งหมด 4^k รูปแบบ ถ้า k = 15 วิธีการนี้ก็จะใช้เวลานานในการหาคำตอบ

ถึงจุดนี้สมมติฐานหลักข้อนึงในการหาโมทิฟคือเราทราบความยาวของโมทิฟ (ค่า k) ที่ต้องการหาล่วงหน้า อย่างไรก็ตามสมมติฐานนี้ไม่เป็นจริงในทางปฏิบัติ ผลที่ตามมาคือเราจำเป็นต้องออกแบบให้อัลกอริทึมของเราสามารถทดสอบค่า k ต่างๆ ได้ รวมทั้งต้องสามารถอนุมานความยาว k ที่ถูกต้องได้

วิธีการหาโมทิฟแบบโลภ (Greedy Motif Search)

โปรไฟล์เมทริกซ์กับการโยนลูกเต๋า

หลายอัลกอริทึมมีลักษณะการทำงานแบบทำซ้ำและในแต่ละรอบที่ของการทำงานมักมีทางเลือกหรือตัวเลือกมากกว่า 1 ทาง อัลกอริทึมที่ใช้วิธีการแก้ปัญหาแบบโลภ (Greedy algorithms) จะเลือกเส้นทางที่ให้ผลคำตอบรวมดีที่สุดในรอบทำซ้ำนั้นๆ โดยยึดหลักทำให้เร็วแลกกับความถูกต้องที่ลดลงหรือได้คำตอบเพียงโดยประมาณ อย่างไรก็ตามการประยุกต์ใช้แนวทางการแก้ปัญหาแบบโลภกับปัญหาทางชีววิทยาหลายๆปัญหาพบว่าสามารถแก้ปัญหาได้ในระดับหนึ่ง ในหัวข้อนี้เราจะนำแนวทางการแก้ปัญหาแบบโลภมาใช้ในการหาโมทิฟ จาก `PROFILE(Motifs)`ในรูปที่ 4.5 จะเห็นว่าเราสามารถพิจารณาว่าโปรไฟล์เมทริกซ์นี้เป็นเสมือนลูกเต๋าที่มี 4 หน้าคือ “A”, “C”, “G”, และ “T” และเราสามารถสร้าง k-mer หนึ่งโดยการโยนลูกเต๋าโปรไฟล์เมทริกซ์นี้ k ครั้ง ตัวอย่างเช่นในคอลัมน์แรกโอกาสที่จะเป็นนิวคลีโอไทด์ “A”, “C”, “G” และ “T” เป็น 0.13, 0.87, 0.0, และ 0.0 ตามลำดับ ใน

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

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 แสดงการใช้โปรไฟล์เมทริกซ์ในการหาค่าความน่าจะเป็นในการเกิดลำดับเบส k-mer CGTATGTC เทียบกับสตริงสายหลัก โดยค่าความน่าจะเป็นเท่ากับผลคูณของค่าความน่าจะเป็นของการเกิดนิวคลีโอไทด์ในแต่ละตำแหน่ง (จากโปรไฟล์เมทริกซ์) ตามลำดับเบสใน k-mer

k-mer ใดๆ จะมีค่าความน่าจะเป็นสูงถ้าการเรียงตัวของลำดับเบสในสายมีความใกล้เคียงกับลำดับเบสของสายสตริงเสียงข้างมากของโปรไฟล์เมทริกซ์ จากรูปที่ 4.9 ข้างต้นถ้า k-mer ที่ทดสอบเป็นเส้นเดียวกับสายสตริงเสียงข้างมากค่าความน่าจะเป็นจะเท่ากับ 0.0205753 ตามการคำนวณต่อไปนี้ ซึ่งมีค่ามากกว่าค่าความน่าจะเป็นของ k-mer **CACTAATT** ข้างต้น

$$\Pr(CACTAATT|\,Profile) = .87 \cdot .0.46 \cdot 0.46 \cdot .6 \cdot .87 \cdot .97 \cdot 1 \cdot .53 = 0.049$$

ฝึกหัด	จงคำนวณค่าความน่าจะเป็น $\Pr(CCCCTAGC|\,Profile)$ โดยใช้โปรไฟล์เมทริกซ์ในรูปที่ 4.9

ถ้ามีโปรไฟล์เมทริกซ์เราจะสามารถหาได้ว่า k-mer ใดในสายดีเอ็นเอที่มีความใกล้เคียงกับสายสตริงเสียงข้างมากที่สุด หรือสอดคล้องกับโปรไฟล์ที่สุด ซึ่งเรียกว่าเป็น Profile-most probable k-mer ตัวอย่างเช่นถ้าเราใช้โปรไฟล์เมทริกซ์ข้างต้นในการสแกนหา 8-mer ในสายดีเอ็นเอ **CTCCTCATAAATTATCCGCC** จะได้ **CTCCTCATAAATTATCCGCC** เป็น 8-mer ที่ใกล้เคียงกับสายสตริงเสียงข้างมากที่สุด ซึ่งในตัวอย่างนี้ค่าความน่าจะเป็นของ 8-mer อื่นๆ จะมีค่าเป็น 0 โดยทั่วไปอาจมีมากกว่าหนึ่ง k-mer ที่ให้ค่าความน่าจะเป็นดีที่สุด ซึ่งในกรณีนี้เราจะเลือก k-mer แรกที่พบ

วิธีการหาโมทิฟโดยแนวทางการปัญหาแบบโลภอาจทำได้โดยในแต่ละรอบใช้ k-mer ใน Dna_1 เป็นโมทิฟแรก จากนั้นสร้างโปรไฟล์เมทริกซ์จาก k-mer นี้ และหา $Motif_2$ ใน Dna_2 ที่สอดคล้องกับโปรไฟล์เมทริกซ์ที่สร้างขึ้นที่สุด เมื่อได้ $Motif_2$ แล้วทำการอัพเดตโปรไฟล์เมทริกซ์โดยใช้ข้อมูลทั้ง $Motif_1$ และ $Motif_2$ และหา $Motif_3$ ใน Dna_3 ที่สอดคล้องกับโปรไฟล์เมทริกซ์ที่สร้างขึ้นและวนซ้ำการทำงานจนครบ t-1 รอบ ก็จะได้ โปรไฟล์เมทริกซ์

นิยามปัญหาที่ 4.4 ปัญหาการหา k-mer ที่สอดคล้องกับโปรไฟล์ที่สุด

ปัญหาการหา k-mer ที่สอดคล้องกับโปรไฟล์ที่สุด (Profile-most probable k-mer problem) หา k-mer ที่สอดคล้องกับโปรไฟล์ที่สุด	
ข้อมูลเข้า	สายสตริง *Text* จำนวนเต็ม *k* และโปรไฟล์เมทริกซ์ขนาด 4 x *k*
ผลลัพธ์	k-mer ในสายสตริง *Text* ที่สอดคล้องกับโปรไฟล์ที่สุด

ของ k-mers ที่ได้มาจากชุดของสายดีเอ็นเอ *Dna* ซึ่งโมทิฟชุดนี้จะถูกนำไปทดสอบคะแนนว่าดีกว่าชุดของโมทิฟเดิมหรือไม่ ถ้าดีกว่าก็ทำการอัพเดตชุดของโมทิฟให้เป็นชุดใหม่นี้ และทำการวนซ้ำเพื่อใช้ k-mer ถัดๆไปของ Dna_1 จนครบดังสูโดโคดที่ 4.3 ต่อไปนี้

สูโดโคดที่ 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
```

วิเคราะห์การทำงานของ Greedy Motif Search

ถ้าพิจารณาโดยผิวเผินอาจเห็นว่าวิธีการแก้ปัญหาโดย `GreedyMotifSearch()` ข้างต้นมีตรรกะการทำงานที่ยอมรับได้ อย่างไรก็ดีถ้าพิจารณาในรายละเอียดจะพบว่าวิธีการนี้ใช้ไม่ได้ ลองพิจารณาการหาโมทิฟ **ACGT** ที่ถูกต้องจากชุดของดีเอ็นเอต่อไปนี้ โดยมีสมมติฐานว่าอัลกอริทึมได้ทำการเลือก 4-mer **ACCT** ที่ถูกต้องแล้วจากดีเอ็นเอสายแรก

ttACCTtaac
gATGTctgtc
acgGCGTtag
ccctaACGAg
cgtcagAGGT

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

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

และพร้อมนำไปใช้ในการค้นหาโมทิฟในดีเอ็นเอสายที่สองที่สอดคล้องกับโปรไฟล์เมทริกซ์ข้างต้น อย่างไรก็ตาม เนื่องจากโปรไฟล์เมทริกซ์มีค่า 0 ในหลายตำแหน่งมาก ทำให้ค่าความน่าจะเป็นของ 4-mer ใดๆที่ไม่ใช่ **ACCT** มีค่าเป็น 0 ทั้งหมด ยกเว้นว่ามีลำดับเบส **ACCT** ปรากฏอยู่ในดีเอ็นเอทุกเส้น จากตัวอย่างนี้จะพบว่าโอกาสที่ `GreedyMotifSearch()` จะพบคำตอบที่ถูกต้องมีน้อยมาก การปรากฏของ 0 ในหลายๆตำแหน่งของโปรไฟล์เมทริกซ์นี้เป็นปัญหาที่จำเป็นต้องพิจารณาเพิ่มเติม

การหาโมทิฟจากมุมของโอลิเวอร์ ครอมเวลล์

มีค่าความน่าจะเป็นเท่าไหร่ที่จะไม่มีพระอาทิตย์ขึ้นในวันพรุ่งนี้

กฎของโอลิเวอร์ ครอมเวลล์ (Oliver Cromwell) กล่าวว่าเราไม่ควรใช้ค่าความน่าจะเป็นเท่ากับ 0 หรือ 1 นอกเสียจากว่าเรากำลังพิจารณาข้อมูลเชิงตรรกะที่มีค่าที่เป็นจริงหรือเท็จเท่านั้น อีกนัยยะหนึ่งคือเราควรเผื่อค่าความน่าจะเป็นเล็กๆให้กับการเกิดเหตุการณ์บางเหตุการณ์ที่แม้จะมีโอกาสเกิดขึ้นจริงน้อยมากๆ ตัวอย่างเช่น โอกาสที่จะไม่มีพระอาทิตย์ขึ้นในวันพรุ่งนี้ ซึ่งในศตวรรษที่ 18 นักคณิตศาสตร์ชาวฝรั่งเศสชื่อ ปิแอร์ ไซมอน ลาปลาซ (Pierre-Simon Laplace) ได้ประมาณค่าโอกาสที่จะไม่มีพระอาทิตย์ขึ้นในวันพรุ่งนี้ไว้เท่ากับ 1/1826251 โดยอาศัยข้อมูลก่อนหน้าว่าทุกวันใน 5000 ปีที่ผ่านมาพระอาทิตย์ขึ้นในตอนเช้าทุกวัน การประมาณโอกาสที่จะไม่มีพระอาทิตย์ขึ้นในเช้าวันพรุ่งนี้อาจดูเป็นเรื่องขำขัน อย่างไรก็ตามแนวคิดและวิธีการของลาปลาซมีบทบาทสำคัญในวิธีการทางสถิติปัจจุบัน

จากการพิจารณาชุดของข้อมูลหนึ่งๆ มีโอกาสที่บางเหตุการณ์อาจจะไม่เคยเกิดขึ้นถ้าข้อมูลมีขนาดใหญ่ไม่เพียงพอหรือเหตุการณ์นั้นเป็นเหตุการณ์ที่โดยพื้นฐานมีโอกาสเกิดขึ้นน้อยมากอยู่แล้ว ซึ่งหมายความว่าค่าความน่าจะเป็นจากการสังเกตการณ์จะเป็น 0 อย่างไรก็ตาม การกำหนดค่า 0 ให้กับเหตุการณ์ต่างๆ โดยดูจากข้อมูลเท่าที่มีอยู่เท่านั้นอาจจะไม่ตรงกับความเป็นจริง รวมทั้งอาจทำให้เกิดค่าความผิดพลาดได้ ลาปลาซแก้ปัญหานี้โดยการปรับค่าความน่าจะเป็นเทียมให้กับเหตุการณ์เหล่านี้

กฎการสืบทอดของลาปลาซ

กฎของครอมเวลล์มีความเกี่ยวข้องกับการคำนวณโอกาสการเกิด k-mer หนึ่งๆ โดยใช้โปรไฟล์เมทริกซ์ ลองพิจารณาโปรไฟล์ต่อไปนี้

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

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

โดยเบสที่หกทำให้ค่าความน่าจะเป็นของสาย k-mer นี้เป็น 0 ถึงแม้ว่า **CACTACTT** จะต่างจากสตริงสายหลักเพียงตำแหน่งนี้ตำแหน่งเดียวก็ตาม ซึ่งมีความน่าจะเป็นน้อยเสียกว่า **CGTATGTC** ข้างต้น เพื่อเป็นการเพิ่ม

ความยุติธรรมในการให้คะแนน นักชีวสารสนเทศมักจะแทนค่า 0 ในแต่ละตำแหน่งด้วยตัวเลขจำนวนเต็มที่มีค่าน้อยและเรียกค่านี้ว่าสูโดเคาท์ (pseudo-counts) และแนวทางที่ง่ายที่สุดในการเพิ่มสูโดเคาท์เรียกว่ากฎการสืบทอดของลาปลาซ ซึ่งเป็นไปในแนวทางเดียวกับที่ลาปลาซใช้ในการคำนวณค่าความน่าจะเป็นที่พระอาทิตย์จะไม่ขึ้นในวันพรุ่งนี้นั่นเอง ในกรณีของโมทิฟสูโดเคาท์มักเป็นค่า 1 ที่เพิ่มเข้าไปในผลของ `COUNT(Motifs)` ดังตัวอย่างต่อไปนี้

```
                    T  A  A  C
                    G  T  C  T
          Motifs    A  C  T  A
                    A  G  G  T

                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
COUNT(Motifs)   G: 1  1  1  0    Profile(Motifs) G: 1/4  1/4  1/4  0
                T: 1  1  1  2                    T: 1/4  1/4  1/4  2/4
```

และเมื่อมีการเพิ่ม 1 กับทุกค่าของ `COUNT(Motifs)`ตามกฎการสืบทอดของลาปลาซทั้งสองเมทริกซ์ข้างต้น 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
```

การหาโมทิฟแบบโลภที่ถูกปรับปรุง

เราสามารถปรับปรุงวิธีการหาโมทิฟแบบโลภได้โดยการกำจัดค่า 0 ทั้งหมดที่อยู่ในโปรไฟล์เมทริกซ์ โดยจากสูโดโคดที่ 4.3 ข้างต้นจะเปลี่ยนจาก `สร้างโปรไฟล์เมทริกซ์จาก Motif_1 ถึง Motif_i-1` เป็น

`ประยุกต์ใช้กฎการสืบทอดของลาปลาซในการสร้างโปรไฟล์เมทริกซ์จาก Motif_1 ถึง Motif_i-1`

พิจารณาตัวอย่างการหา 4-mer โมทิฟในตัวอย่างก่อนหน้า 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
```

และใช้โปรไฟล์เมทริกซ์ในการคำนวณค่าความน่าจะเป็นของ 4-mer ทั้งหมดที่อยู่ในดีเอ็นเอเส้นที่สอง ได้ผลดังต่อ ไปนี้

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

ซึ่งได้โมทิฟที่สอดคล้องกับโปรไฟล์เมทริกซ์มากที่สุดสองโมทิฟคือ **ATGT** และ **GTct** และถ้าสมมติว่าเราโชคดี ในการเลือก k-mer ที่ถูกต้องระหว่าง k-mers ที่ดีที่สุดสองสายนี้ จะได้โมทิฟเมทริกซ์และโปรไฟล์เมทริกซ์ใหม่ 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

และใช้โปรไฟล์เมทริกซ์นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mer ทั้งหมดที่อยู่ในดีเอ็นเอเส้นที่สาม ได้ผลดัง ต่อไปนี้

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

ซึ่งได้โมทิฟที่สอดคล้องกับโปรไฟล์เมทริกซ์มากที่สุดสองโมทิฟคือ acg**G** และ **GCGT** ถ้าในรอบนี้สมมติว่า acg**G** ถูกเลือกมาใช้ จะได้โมทิฟเมทริกซ์และโปรไฟล์เมทริกซ์ใหม่ดังต่อไปนี้

Motifs: **ACCT**
ATGT
acg**G**

	A:	3+1	0+1	0+1	0+1		A:	4/7	1/7	1/7	1/7
	C:	0+1	2+1	1+1	0+1		C:	1/7	3/7	2/7	1/7
COUNT(*Motifs*)	G:	0+1	0+1	2+1	1+1	PROFILE(*Motifs*)	G:	1/7	1/7	3/7	2/7
	T:	0+1	1+1	0+1	2+1		T:	1/7	2/7	1/7	3/7

และใช้โปรไฟล์เมทริกซ์นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mer ทั้งหมดที่อยู่ในดีเอ็นเอเส้นที่สี่จะได้ผลดัง ต่อไปนี้

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

ถึงแม้ว่าเราจะเลือก k-mer ที่ไม่ถูกต้องมาสร้างโมทิฟเมทริกซ์และโปรไฟล์เมทริกซ์จากดีเอ็นเอเส้นที่สาม โปรไฟล์เมทริกซ์ที่สร้างขึ้นก็ยังสามารถนำมาใช้ในการหาโมทิฟที่ถูกต้อง **ACGA** ในดีเอ็นเอสายที่สี่ และนำมาสร้างเป็นโมทิฟเมทริกซ์และโปรไฟล์เมทริกซ์ใหม่ในรอบถัดไปดังต่อไปนี้

```
                    Motifs:  ACCT
                             ATGT
                             acgG
                             ACGA

                A: 4+1 0+1 0+1 1+1                  A: 5/8 1/8 1/8 2/8
                C: 0+1 3+1 1+1 0+1                  C: 1/8 4/8 2/8 1/8
COUNT(Motifs)   G: 0+1 0+1 3+1 1+1  PROFILE(Motifs) G: 1/8 1/8 4/8 2/8
                T: 0+1 1+1 0+1 2+1                  T: 1/8 2/8 1/8 3/8
```

และใช้โปรไฟล์เมทริกซ์นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mer ทั้งหมดที่อยู่ในดีเอ็นเอเส้นที่ห้าจะได้ผลดัง ต่อไปนี้

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

และ 4-mer ที่สอดคล้องกับโปรไฟล์เมทริกซ์มากที่สุดในดีเอ็นเอสายที่ห้าคือ **AGGT** ซึ่งหมายความว่า `GreedyMotifSearch()`สร้างโมทิฟเมทริกซ์ต่อไปนี้ ที่นำไปสู่การสร้างสตริงเสียงข้างมากที่มีความถูกต้อง

```
                    Motifs:  ACCT
                             ATGT
                             acgG
                             ACGA
                             AGGT

CONSENSUS(Motifs): ACGT
```

จะเห็นว่าการใช้สูโดเคาท์สามารถทำให้ประสิทธิภาพการหาโมทิฟโดยแนวทางแบบโลภได้ผลที่ดีขึ้น อย่างไรก็ตาม การหาโมทิฟยังมีวิธีการอื่นที่มีประสิทธิภาพมากกว่า

การหาโมทิฟแบบสุ่ม

อัลกอริทึมแบบสุ่ม (randomized algorithms) เกือบทั้งหมด รวมทั้งอัลกอริทึมที่ใช้ในการหาโมทิฟแบบสุ่มที่จะกล่าวถึงต่อไปเป็นอัลกอริทึมในกลุ่มมอนติคาร์โล (Monte Carlo algorithms) ซึ่งจะมีคุณสมบัติไม่รับประกันว่า จะได้คำตอบที่ถูกต้อง 100% แต่จะสามารถหาคำตอบแบบประมาณได้เร็ว และเนื่องจากสามารถหาคำตอบได้เร็ว เราสามารถรันอัลกอริทึมเป็นพันๆครั้งและเลือกคำตอบที่มีการประมาณค่าที่ถูกต้องที่สุดได้

จากนิยามเกี่ยวกับ `PROFILE(Motifs)` ถ้ามีชุดของสายดีเอ็นเอ *Dna* และโปรไฟล์เมทริกซ์ (*Profile*) ที่สร้างมาจากชุดของ k-mers โดยแต่ละ k-mer มาจากแต่ละสายดีเอ็นเอใน *Dna* ถ้ามีข้อมูลเข้าเป็นชุดของสายดีเอ็นเอ *Dna* และโปรไฟล์เมทริกซ์ขนาด 4 x k เรานิยาม `MOTIFS (Profile,Dna)`เป็นชุดของ k-mers ที่มีความสอดคล้องกับ *Profile* มากที่สุดจากสายดีเอ็นเอแต่ละเส้น ตัวอย่างเช่นถ้าเรามีโปรไฟล์และชุดของสายดีเอ็นเอต่อไปนี้

```
                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 **PROPFILE()** ข้างต้นในการหา 4-mer จากดีเอ็นเอแต่ละเส้นที่สอดคล้องกับโปรไฟล์มากที่สุด จะได้ชุดของ k-mers ต่อไปนี้

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

ซึ่งโดยทั่วไปชุดของ k-mers ในรอบแรกที่จะถูกนำมาสร้างเป็นโมทิฟเมทริกซ์และโปรไฟล์เมทริกซ์อาจสุ่มเลือกมาจาก k-mer ใดๆ ของดีเอ็นเอแต่ละเส้นก็ได้ และโปรไฟล์เมทริกซ์นี้เองก็ถูกนำไปหาชุดของโมทิฟเพื่อสร้างเป็นโปรไฟล์เมทริกซ์ในรอบถัดๆไปดังสมการต่อไปนี้

$$\text{MOTIFS}(\text{PROFILE}(\text{Motifs}), Dna)$$

ซึ่งสมการนี้อยู่บนสมมติฐานว่าชุดของโมทิฟที่ได้จาก **MOTIFS()**จะให้คะแนนที่ดีกว่าโมทิฟชุดแรกที่เลือกมาแบบสุ่มจากดีเอ็นเอแต่ละเส้น และหลังจากนั้นเราสามารถสร้างโปรไฟล์เมทริกซ์ในรอบถัดไปตามสมการ

$$\text{PROFILE}(\text{MOTIFS}(\text{PROFILE}(Motifs), Dna))$$

และใช้โปรไฟล์นี้ในการหาชุดโมทิฟที่มาจากดีเอ็นเอแต่ละเส้นที่สอดคล้องกับโปรไฟล์นี้ที่สุด

$$\text{MOTIFS}(\text{PROFILE}(\text{MOTIFS}(\text{PROFILE}(Motifs), Dna)), Dna)$$

และทำวนซ้ำเรื่อยๆถ้าคะแนนรวมของโททิฟเมทริกซ์ยังน้อยลงเรื่อยๆ

โดยวิธีการที่ยกตัวอย่างนี้เป็นการทำงาน ของการหาโมทิฟแบบสุ่ม (randomized motif search) (สูโดโคดที่ 4.4) เนื่องจากการรัน **RandomizedMotifSearch()** เพียงครั้งเดียวอาจไม่ได้คำตอบที่ดี โดยทั่วไปนักชีวสารสนเทศจะทำรันฟังก์ชันซ้ำเป็นหลักหลายพันครั้ง โดยในแต่ละรอบของการรันชุดของ k-mers จะถูกเลือกมาแบบสุ่มและจะทำการเลือกชุดของ k-mers ที่ให้ผลลัพธ์ดีที่สุดจากการรันทั้งหมดนั้น

ทำไมการหาโมทิฟแบบสุ่มถึงให้ผลลัพธ์ที่ถูกต้องได้

อาจมีคำถามว่าการหาโมทิฟแบบสุ่มจะให้ผลลัพธ์ที่ถูกต้องได้อย่างไร ในเมื่อในแต่ละรอบของการทำงานก็เริ่มจากการสุ่มเลือกชุดของ k-mers มาพิจารณาชุดของสายดีเอ็นเอที่มีโมทิฟ (4,1) **ACGT** แทรกอยู่ และสมมติว่าในรอบแรกชุดของ k-mers (อักษรสีแดง) ที่ถูกสุ่มเลือกมาจากดีเอ็นเอแต่ละเส้นแทบไม่ถูกต้องเลย และตัวอักษรใหญ่คือ

สูโดโคดที่ 4.4 การหาโมทิฟแบบสุ่ม

```
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
```

โมทิฟที่แทรกเข้าไปในดีเอ็นเอแต่ละสาย

	ttACCT**taac**
	gAT**GTct**gtc
Dna	**ccgG**CGTtag
	c**acta**ACGAg
	cgtcag**AGGT**

และจาก k-mers สีแดงที่สุ่มเลือกมาสามารถนำมาสร้างเป็นโมทิฟเมทริกซ์และโปรไฟล์เมทริกซ์ดังต่อไปนี้

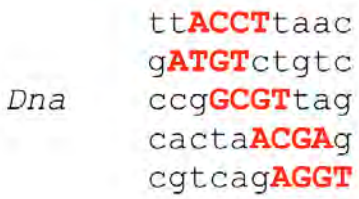
Motifs
t a a c
G T c t
c c g G
a c t a
A G G T

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

เราสามารถใช้โปรไฟล์เมทริกซ์ที่สร้างขึ้นนี้ในการหา k-mer จากดีเอ็นเอแต่ละเส้นที่สอดคล้องกับโปรไฟล์นี้มากที่สุดดังผลคะแนนความน่าจะเป็นที่แสดงต่อไปนี้

```
.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
```

โดย k-mer ที่สอดคล้องกับโปรไฟล์ที่สุดในดีเอ็นเอแต่ละเส้นถูกแสดงด้วยคะแนนสีแดง เมื่อนำ 4-mer ชุดนี้ไปแสดงในชุดของดีเอ็นเอตั้งต้นจะพบว่าเป็นโมทิฟ (4,1) ที่ถูกแทรกไว้ทั้งหมดนั่นเอง



หยุดคิด	จงอธิบายว่าทำไมการเลือก k-mer แบบสุ่มจะได้ให้ผลที่ถูกต้องได้ อาจลองชุดของ k-mers เริ่มต้นใดๆก็ได้

สำหรับปัญหาก่อนหน้าที่มีการแทรก 15-mer โมทิฟ-(15,4) ของ **AAAAAAAAGGGGGG** เข้าไปยังดีเอ็นเอ 10 เส้นโดยที่แต่ละเส้นมีความยาว 600 นิวคลีโอไทด์ ถ้าเราใช้วิธีการหาโมทิฟแบบสุ่มเพื่อแก้ปัญหานี้โดยการรันทั้งสิ้น 100,000 ครั้งโดยที่การรันทุกครั้ง k-mers ชุดแรกจะถูกเลือกแบบสุ่ม รูปที่ 4.10 แสดงชุดของโมทิฟที่มีคะแนนรวมน้อยที่สุด (43) จากการรัน 100,000 ครั้ง โดยได้สตริงเสียงข้างมากเป็น **AAAAAAAAacaGGGG** โมทิฟเหล่านี้มีความอนุรักษ์น้อยกว่าชุดของโมทิฟที่ถูกแทรกเข้าไปเพียงเล็กน้อยโดยโมทิฟชุดนั้นมีคะแนนรวมเท่ากับ 40 หรือ 41 สำหรับชุดของโมทิฟที่เป็นผลลัพธ์ของการรัน `GreedyMotifSearch()` อย่างไรก็ตาม `RandomizedMotifSearch()` สามารถรันในจำนวนรอบที่มากกว่าซึ่งเพิ่มโอกาสในการพบโมทิฟที่ถูกต้องมากขึ้นเรื่อยๆ

หยุดคิด	ลองเขียนโคดของฟังก์ชัน `RandomizedMotifSearch()` และตรวจสอบดูว่าได้ผลลัพธ์ของสตริงเสียงข้างมากที่คล้ายกับตัวอย่างข้างต้นหรือไม่ และต้องรันโคดทั้งหมดกี่รอบเพื่อให้ได้โมทิฟ-(15,4) เมทริกซ์ ที่มีคะแนนรวมเท่ากับ 40

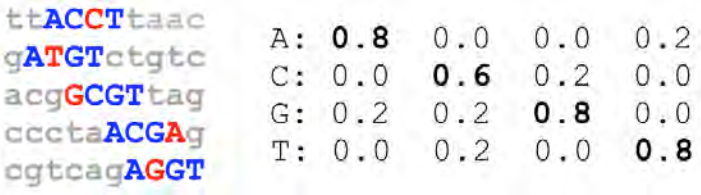
รูปที่ 4.10 ชุดของโมทิฟที่เป็นผลลัพธ์จากวิธีการหาโมทิฟแบบสุ่มที่มีคะแนนรวมน้อยที่สุดจากการรัน 100,000 ครั้ง และสายสตริงเสียงข้างมากที่อนุมานจากโมทิฟเมทริกซ์

(ที่มา: รูปที่ 2.7 ของ [21])

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

ทำไมการหาโมทิฟแบบสุ่มถึงให้ผลลัพธ์ที่ดี

ในหัวข้อที่ผ่านมาเราเริ่มด้วยการสร้างโปรไฟล์เมทริกซ์จากชุดของโมทิฟ-(4,1) ที่นำมาแทรกในชุดของสายดีเอ็นเอ โดยมีสตริงเสียงข้างมากเป็น ACGT ดังต่อไปนี้



ถ้าแต่ละตำแหน่งในสายดีเอ็นเอแต่ละเส้นมีโอกาสเป็นนิวคลีโอไทด์ “A”, “C”, “G” หรือ “T” เท่าๆกัน เราจะสามารถคาดหวังว่าโปรไฟล์เมทริกซ์ที่สร้างจากชุดของ 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
```

ซึ่งเป็นโปรไฟล์ที่ยูนิฟอร์ม (uniform) แสดงโอกาสในการเกิดนิวคลีโอไทด์แบบหนึ่งๆเท่าๆกัน ในทุกๆตำแหน่ง และเป็นโปรไฟล์ที่ไม่มีประโยชน์ ในทางกลับกันถ้าชุดของชุดของโมทิฟที่นำมาสร้างโปรไฟล์มีโอกาสของการปรากฏนิวคลีโอไทด์ที่จำเพาะในตำแหน่งหนึ่งๆไม่เท่ากันตัวอย่างเช่น

```
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
```

ก็จะสามารถนำไปสู่การพบโมทิฟที่เป็นคำตอบหรือเข้าใกล้คำตอบได้ ซึ่งการทำงานของ `RandomizedMotifSearch()` ได้ถูกออกแบบมาให้การทำงานในรอบถัดๆไปได้ผลที่เข้าใกล้คำตอบมากขึ้น จากตัวอย่างของโปรไฟล์เมทริกซ์ข้างต้นที่จะถูกใช้ในการหาชุดของโมทิฟในรอบถัดไปก็มีทิศทางจำเพาะของโมทิฟที่จะสอดคล้องตามค่าความถี่ที่แตกต่างกันของแต่ละนิวคลีโอไทด์ในแต่ละตำแหน่ง ทั้งนี้อยู่บนสมมติฐานว่ามีโมทิฟที่ต้องการค้นหาแทรกอยู่ชุดในของสายดีเอ็นเอนั่นเอง

ฝึกหัด	ถ้ามีการสุ่มเลือก 15-mer มาจากแต่ละดีเอ็นเอยาว 600 นิวคลีโอไทด์ จำนวน 10 เส้น จงคำนวณค่าความน่าจะเป็นที่มี 15-mer อย่างน้อยหนึ่งเส้น เป็นโมทิฟที่แทรกเข้าไป

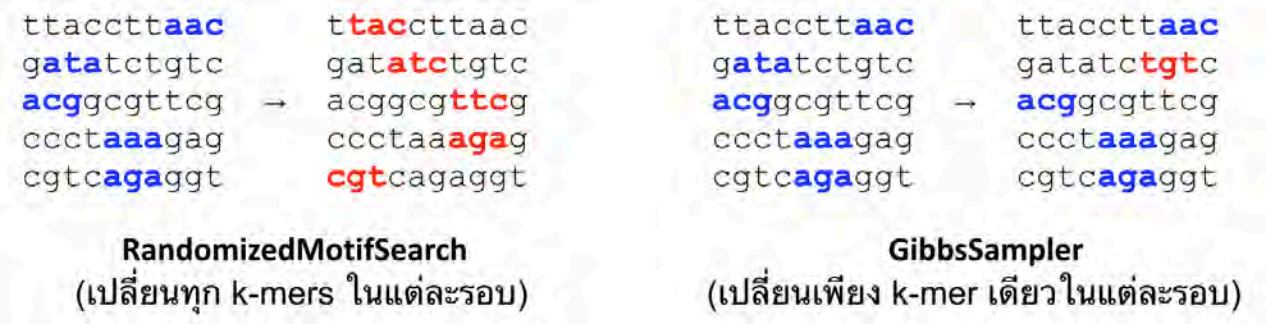
ถึงแม้ว่ามีค่าความน่าจะเป็นน้อยมากๆที่ชุดของ k-mers ที่เลือกมาแบบสุ่มจะเป็นชุดเดียวกับโมทิฟที่แทรกเข้าไปทั้งหมด อย่างไรก็ตามค่าความน่าจะเป็นที่มีอย่างน้อยหนึ่ง k-mer ที่ถูกเลือกมาแบบสุ่มนั้นตรงกับโมทิฟที่แทรกเข้าไป ถึงแม้ว่าจะเป็นค่าที่น้อยก็มีนัยยะสำคัญ เนื่องจากเราสามารถรัน `RandomizedMotifSearch()` หลายๆครั้ง ซึ่งเป็นการเพิ่มโอกาสที่จะพบบาง k-mer ที่เป็นโมทิฟที่แทรกเข้าไป และนำไปสู่การสร้างโปรไฟล์ เมทริกซ์ที่มีทิศทางจำเพาะต่อโมทิฟที่ต้องการหา

อย่างไรก็ตามการพบหนึ่ง k-mer ที่เป็นโมทิฟจริงมักจะไม่เพียงพอในการที่จะทำให้ `RandomizedMotifSearch()`หาคำตอบที่ดีที่สุดได้ เนื่องจากตำแหน่งเริ่มต้นของ k-mer ที่จะสุ่มเลือกมาจากดีเอ็นเอเส้นหนึ่งๆ ก็มีจำนวนมากมาย การเลือกชุดของโมทิฟแบบสุ่มในทุกๆรอบที่รันมักไม่ได้คำตอบที่ดีเท่าตัวอย่างข้างต้น เพราะมีโอกาสน้อยที่ชุดของ k-mers ที่เกิดจากการสุ่มเลือกใหม่ในทุกรอบจะนำไปสู่โปรไฟล์เมทริกซ์ที่มีทิศทางความจำเพาะต่อโมทิฟที่ต้องการหานั่นเอง

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

กิปส์แซมปลิง

ในขณะที่ `RandomizedMotifSearch()` จะทำการเลือก k-mers แบบสุ่มใหม่ทั้งหมดในแต่ละรอบของการทำงาน ซึ่งก็มีโอกาสเป็นไปได้ว่าอาจมี k-mers ที่เป็นโมทิฟที่ถูกต้องแล้วถูกละทิ้งไปทั้งหมดในรอบใหม่ กิปส์แซมเปลอร์ (Gibbs Sampler) ได้ปรับเปลี่ยนแนวทางข้างต้นโดยในแต่ละรอบจะเลือกทิ้งเพียง k-mer เดียวจากรอบที่ผ่านมา ดังแสดงในตัวอย่างต่อไปนี้



โดย `GibbsSampler()` เริ่มการทำงานรอบแรกในลักษณะเดียวกับ `RandomizedMotifSearch()` คือในรอบแรกชุดของสาย k-mers จะถูกเลือกแบบสุ่มมาจากดีเอ็นเอแต่ละสาย แต่ในรอบหลังจากนั้น `GibbsSampler()` จะทำการเลือกแบบสุ่มค่าระหว่าง 1 ถึง t และทำการกำหนด k-mer ที่เลือกใหม่เฉพาะบรรทัดที่ถูกสุ่มเลือกมานั้น สูโดโคดที่ 4.5 แสดงการทำงานของ `GibbsSampler()`โดยมีการทำซ้ำ N ครั้ง ในทางปฏิบัติการหยุดการทำงานของ `GibbsSampler()`มีได้หลายเงื่อนไขซึ่งจะไม่ได้กล่าวถึงในบทเรียนนี้

สูโดโคดที่ 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
```

หยุดคิด	ในขณะที่ค่าคะแนนที่เป็นผลลัพธ์จากการรัน `RandomizedMotifSearch()` จะลดลงในในการรันแต่ละรอบ ในกรณีของ `GibbsSampler()` มีโอกาสเป็นไปได้ที่คะแนนในการรันอาจแกว่งเพิ่มมากขึ้น คำถามคือปรากฏการณ์นี้สมเหตุสมผลไหม

ขั้นตอนการทำงานของกิปส์แซมปลิง

หัวข้อนี้แสดงขั้นตอนการทำงานของกิปส์แซมปลิงในรายละเอียดโดยใช้ชุดของสายดีเอ็นเอและ k-mers แบบสุ่มรอบแรก (ตัวอักษรสีแดง) เป็นชุดเดียวกับที่ใช้ในการอธิบายการหาโมทิฟแบบสุ่ม `RandomizedMotif Search()` ในหัวข้อก่อนหน้า และในรอบที่สองของการทำงาน k-mer ของดีเอ็นเอเส้นที่สามถูกเลือกออกจากกลุ่ม

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

ซึ่งได้ผลเป็นโมทิฟเมทริกซ์ เคาท์เมทริกซ์ และโปรไฟล์เมทริกซ์ต่อไปนี้

```
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
```

จะเห็นว่าโปรไฟล์เมทริกซ์มีความอนุรักษ์มากกว่าโปรไฟล์แบบยูนิฟอร์ม (uniform) เพียงเล็กน้อย ซึ่งอาจทำให้ไม่แน่ใจว่าโปรไฟล์เมทริกซ์นี้จะนำไปสู่การหาโมทิฟที่เป็นคำตอบได้อย่างไร ลองใช้โปรไฟล์เมทริกซ์นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mers ทั้งหมดในสายดีเอ็นเอที่ถูกคัดออกได้ผลดังนี้

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

จะเห็นว่าค่าความน่าจะเป็นของ 4-mers แทบทุกแบบมีค่าเป็น 0 ซึ่งจะเป็นปัญหาเดียวกับที่พบตอนที่พิจารณาวิธีการโมทิฟแบบโลภ ซึ่งสามารถแก้ปัญหาโดยใช้สูโดเคาท์และได้เคาท์เมทริกซ์ และโปรไฟล์เมทริกซ์ใหม่ ดังต่อไปนี้

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

หลังจากมีการเพิ่มสูโดเคาท์ และใช้โปรไฟล์เมทริกซ์ใหม่นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mers ทั้งหมดในสายดีเอ็นเอที่ถูกคัดออก (สายที่สาม) ได้ผลดังนี้

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

ดีเอ็นเอสายที่สามจะถูกนำกลับเข้ามาและ 4-mer **GCGT** จะถูกเลือกแทน 4-mer เดิม ccgG ที่ถูกเลือกแบบสุ่มไว้จากรอบที่ผ่านมาและและเริ่มรอบการรันถัดไปโดยการสุ่มเลือกสายดีเอ็นเอที่จะถูกคัดออกใหม่โดยเป็นสายที่หนึ่งในตัวอย่างต่อไปนี้

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 และโปรไฟล์เมทริกซ์ต่อไปนี้

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

จะเห็นว่าโปรไฟล์เมทริกซ์เข้าใกล้โมทิฟจริงมากกว่ารอบแรก และเมื่อทำการเพิ่มสูโดเคาท์ จะได้ผลดังต่อไปนี้

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

หลังจากมีการเพิ่มสูโดเคาท์ และใช้โปรไฟล์เมทริกซ์ใหม่นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mers ทั้งหมดในสายดีเอ็นเอที่ถูกคัดออก (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 ที่ถูกเลือกแบบสุ่มไว้จากรอบที่ผ่านมาและเริ่มรอบการรันถัดไปโดยการสุ่มเลือกสายดีเอ็นเอที่จะถูกคัดออก (เส้นที่สี่) ในรอบถัดไป ดังตัวอย่างต่อไปนี้

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

และได้โมทิฟเมทริกซ์ เคาท์เมทริกซ์ และโปรไฟล์เมทริกซ์ที่มีสูโดเคาท์แล้วดังต่อไปนี้

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

ใช้โปรไฟล์เมทริกซ์ใหม่นี้ในการคำนวณค่าความน่าจะเป็นของ 4-mers ทั้งหมดในสายดีเอ็นเอที่ถูกคัดออก (สายที่ 4) ได้ผลดังนี้

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

โดยดีเอ็นเอสายที่สี่จะถูกนำกลับเข้ามาและ 4-mer **ACGA** นี้จะถูกเลือกแทน 4-mer acta เดิมที่ถูกเลือกแบบสุ่มจากรอบที่ผ่านมาดังต่อไปนี้

Dna	
	tt**ACCT**taac
	gAT**GTct**gtc
	ccg**GCGT**tag
	cacta**ACGA**g
	cgtcag**AGGT**

จะเห็นได้ว่าชุดของโมทิฟ (ตัวอักษรใหญ่) ในรอบนี้มีทิศทางที่เข้าใกล้ชุดของโมทิฟจริงมากแล้ว อาจลองสมมติว่ารอบถัดไปดีเอ็นเอเส้นที่สองจะถูกคัดออก ลองสร้างโปรไฟล์เมทริกซ์ และคำนวณค่าความน่าจะเป็นของ 4-mers ทั้งหมดที่เป็นไปได้ ในดีเอ็นเอเส้นที่สองและดูว่า 4-mer ที่ถูกเลือกแบบสุ่มจากโปรไฟล์เป็นโมทิฟจริงหรือไม่

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

ถึงแม้ว่า `GibbsSampler()` จะให้ผลลัพธ์ที่ดีในหลายๆกรณี แนวคิดและวิธีการแก้ปัญหาผ่าน `GibbsSampler()` จะพิจารณาคำตอบที่เป็นไปได้แค่บางส่วนจึงมีข้อจำกัดในเรื่องของคำตอบโดยอาจไม่ใช่คำตอบที่ดีที่สุดเนื่องจากติด local minimum ด้วยเหตุผลนี้ การใช้งาน `GibbsSampler()` ก็ควรจะมีการรันด้วยจำนวนครั้งมากๆ ด้วยหวังว่าผลการรันในครั้งใดครั้งหนึ่งอาจได้คำตอบที่ดีที่สุด

บทส่งท้าย

เชื้อวัณโรคที่อยู่ในเจ้าบ้าน (host) หลบซ้อนจากยาปฏิชีวนะได้อย่างไร

โรควัณโรค หรือ Tuberculosis (TB) เป็นโรคติดเชื้อ (infectious disease) โดยเชื้อก่อโรคคือ *Mycobacterium tuberculosis* (MTB) และเป็นโรคที่ให้มีผู้เสียชีวิตเป็นหลักล้านคนในแต่ละปี ถึงแม้ว่าในระยะหลังความรุนแรงและการระบาดของโรคลดลงเนื่องจากมียาปฏิชีวนะที่มีประสิทธิภาพ ก็มีสายพันธุ์ MTB ใหม่ๆที่ดื้อยา ในความเป็นจริงแล้ว MTB เป็นเชื้อก่อโรคที่ประสบความสำเร็จในการอยู่ร่วมกับมนุษย์มาหลายทศวรรษ โดยมีการประมาณการว่ามีประชากรจำนวน 1 ใน 3 ของโลกที่มีเชื้อ MTB อยู่ในร่างกายแต่ไม่แสดงออก (latent MTB infections) ซึ่งสำหรับหลายๆคนเชื้อ MTB อาจไม่ก่อโรคเลย ในขณะที่ผู้ป่วยวัณโรคเป็นกลุ่มที่เชื้อมีการทำงาน ด้วยเหตุนี้นักชีววิทยาและนักวิทยาศาสตร์มีความสนใจเป็นอย่างมากว่าอะไรเป็นตัวกระตุ้นให้เชื้อ MTB ที่แอบซ้อนอยู่ในร่างกายเริ่มการทำงาน

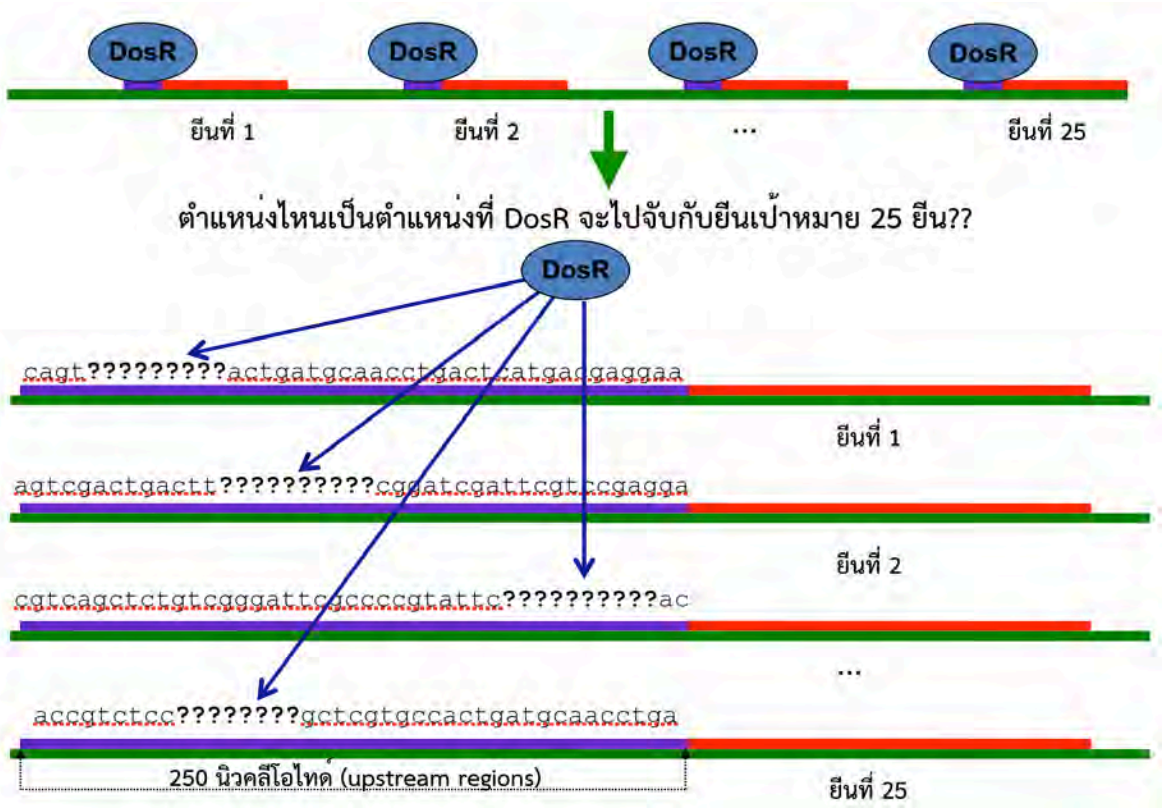
ภาวะพร่องออกซิเจน (Hypoxia) ถูกรายงานว่ามีความเกี่ยวข้องกับ MTB ที่อยู่ในรูปแบบที่ไม่แสดงออก โดยนักชีววิทยาพบว่า MTB จะไม่ทำงานในภาวะพร่องออกซิเจน โดยจะทำงานอีกครั้งและแพร่เชื้อเมื่อปอดของผู้ป่วยฟื้นฟูขึ้น เนื่องจากเชื้อ MTB มีความสามารถในการอยู่รอดเป็นเวลาหลายๆปีในภาวะพร่องออกซิเจน จึงมีความสนใจในการค้นหาว่ามียีนใดบ้างใน MTB ที่เกี่ยวข้องที่เกี่ยวข้องหรือควบคุมให้เกิดสภาวะที่ไม่แสดงออกของเชื้อ โดยนักชีววิทยาต้องการค้นหาทรานสคริปชันแฟตเตอร์ที่สามารถรับสัญญาณความพร่องของออกซิเจน และสามารถควบคุมการแสดงออกของยีนอื่นๆ เพื่อให้ปรับตัวและพร้อมเข้าสู่สภาวะไม่แสดงออก

ในปีค.ศ. 2003 นักชีววิทยาพบว่ายีน dormancy survival regulator (DosR) เป็นทรานสคริปชันแฟคเตอร์ที่ควบคุมการแสดงออกของยีนอื่นๆหลายยีนในภาวะพร่องออกซิเจน อย่างไรก็ตามในการศึกษานั้นยังไม่

ทราบว่า DosR ควบคุมการแสดงออกของยีนเหล่านั้นอย่างไร รวมทั้งไม่ทราบบริเวณที่เป็นที่จับหรือไบน์ดิงไซต์ของ DosR ในยีนกลุ่มเป้าหมาย เพื่อตอบคำถามนี้นักชีววิทยาได้ทำการทดลองโดยใช้ดีเอ็นเออะเรย์ และพบว่ามี 25 ยีนที่มีการเปลี่ยนแปลงการแสดงออกอย่างชัดเจนในสภาวะพร่องออกซิเจน ในส่วนของงานทางชีวสารสนเทศเพื่อเป็นการหาไบน์ดิงไซต์ของ DosR ข้อมูลเข้าคือ upstream regions ของทั้ง 25 ยีนนี้โดยแต่เส้นมีความยาว 250 นิวคลีโอไทด์ (รูปที่ 4.11) โดยต้องการหาโมทิฟที่เป็นไบน์ดิงไซต์ของ DosR จาก upstream regions เหล่านี้

เพื่อให้สามารถรันอัลกอริทึมได้เร็วขึ้น upstream regions ถูกเลือกมาวิเคราะห์เพียง 10 เส้น ทั้งนี้ในการหาโมทิฟ ของโจทย์นี้ไม่มีความรู้มาก่อนว่าความยาวของโมทิฟควรเป็นเท่าไหร่ รูปที่ 4.12 แสดงผลการทำงานของ `MedianString()` และ `RandomizedMotifSearch()` โดยมีการลองเปลี่ยนค่าความยาวของโมทิฟในช่วง 8-12 นิวคลีโอไทด์

หยุดคิด	จากตัวอย่างผลการทำงานของ `MedianString()` เราจะสามารถอนุมานไบน์ดิงไซต์ของ DosR ได้หรือไม่ และคิดว่าความยาวของโมทิฟที่ถูกต้องเป็นเท่าไหร่



รูปที่ 4.11 โจทย์ทางชีววิทยาที่ต้องการหาไบน์ดิงไซต์ของทรานสคริปชันแฟคเตอร์ DosR ใน upstream regions ของยีนเป้าหมาย 25 ยีนใน 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() จาก 10 upstream regions ของยีนเป้าหมายของ DosR

ถึงแม้ว่าผลของสตริงเสียงข้างมากที่ได้จาก `RandomizedMotifSearch()` จะมีความแปรผัน และคะแนนมากกว่า `MedianString()` อย่างไรก็ตาม `RandomizedMotifSearch()` ก็มีประโยชน์ในการหาโมทิฟที่มีขนาดยาวเนื่องจากถ้าใช้ `MedianString()` ใช้เวลานานมาก โดย `Randomized MotifSearch()` สามารถหาโมทิฟที่ความยาว 20 นิวคลีโอไทด์คือ **CGGGACCTACGTCCCTAGCC** ด้วยคะแนนเท่ากับ 57 โดยสตริงเสียงข้างมากที่ทีความยาว 12 นิวคลีโอไทด์ (12-mer) ของทั้ง `MedianString()` และ `RandomizedMotifSearch()`เป็นส่วนของสตริงเสียงข้างมากของ 20-mer นี้

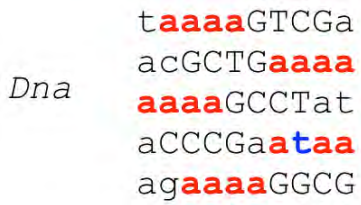
GGACTTCCGGCC
CGGGACCTACGTCCCTAGCC
GGACCAACGGCC

ท้ายสุด ถ้ามีการรัน `GibbsSampler()` โดยใช้ *N*=200 จะได้สตริงเสียงข้างมากเส้นเดียวกับที่รันโดยใช้ `RandomizedMotifSearch()`ข้างต้น โดยมีชุดของโมทิฟที่นำมาสร้างโมทิฟเมทริกซ์ที่แตกต่างกันด้วยคะแนนที่น้อยกว่าคือ 55 จะเห็นว่าอัลกอริทึมในการหาโมทิฟที่แตกต่างกันให้ผลลัพธ์ที่ไม่เหมือนกัน สำหรับการหาโมทิฟของ DosR ข้างต้น ยังไม่มีคำตอบที่ชัดเจนว่าไบน์ดิงไซต์จริงของ DosR มีรูปแบบอย่างไร และอีกคำถามที่น่าสนใจคือถ้าเราได้โปรไฟล์เมทริกซ์ที่เป็นตัวแทนของไบน์ดิงไซต์ที่ถูกต้องแล้ว เราจะสามารถหาว่ามียีนอื่นๆอีกหรือไม่ที่อาจจะถูกควบคุมโดยทรานสคริปชันแฟคเตอร์เดียวกัน ได้อย่างไร

ความท้าทายของการหาโมทิฟ

การหาโมทิฟที่ถูกต้องจะยากและซับซ้อนถ้าในสายดีเอ็นเอเส้นหนึ่งๆมีความถี่ของนิวคลีโอไทด์จำเพาะหนึ่งๆ มากผิดปกติ โดยวิธีการค้นหาชุด k-mers โดยให้ได้คะแนนรวมน้อยที่สุดจะใช้ไม่ได้ผล ตัวอย่างเช่น ถ้าในดีเอ็นเอเส้นหนึ่งมีนิวคลีโอไทด์ A มากเป็นพิเศษถึง 85% ในขณะที่มีนิวคลีโอไทด์ T, C, และ G อย่างละ 5% จากอัลกอริทึม

ข้างต้น k-mer **AA...AA** มีโอกาสที่จะเป็นโมทิฟที่มีคะแนนรวมน้อยสุดและไม่สามารถหาโมทิฟที่เป็นคำตอบ ที่ถูกต้องอย่าง **GCCG** ที่มีคะแนนเท่ากับ 5 โดยที่ aaaa มีคะแนนรวมเท่ากับ 1 ในดังตัวอย่างต่อไปนี้ได้



แนวทางแก้ปัญหานี้ สามารถทำได้โดยการใช้เอนโทรปีสัมพัทธ์ (relative entropy) ซึ่งจะกล่าวถึงในหัวข้อถัดไป นอกจากปัญหานี้แล้ว อีกปัญหาหนึ่งคือมีโมทิฟจำนวนมากที่มีการใช้อักขระอื่นในการแสดงผลเพื่อให้ได้ข้อมูลครบถ้วนกว่า 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 ถ้าพิจารณาโมทิฟต่อไปนี้ **CSKWYWWATKWATYYK** ซึ่งแสดงโมทิฟของ CSRE ในยีสต์ที่มีการกล่าวถึงในช่วงต้นของบทเรียน ซึ่งวิธีการที่อธิบายในบทเรียนไม่สามารถหาโมทิฟลักษณะนี้ได้

เอนโทรปีสัมพัทธ์

ถ้ามีข้อมูลเข้าเป็นชุดของสายดีเอ็นเอ เรากำหนดเอนโทรปีสัมพัทธ์ของโปรไฟล์เมทริกซ์ขนาด 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 ทั้งนี้ในหัวข้อก่อนหน้าเราพยายามหาผลรวม เอนโทรปีที่น้อยที่สุด แต่ในกรณีของเอนโทรปีสัมพัทธ์นี้เราต้องการหาผลรวมที่มีค่ามากที่สุด โดยพจน์ต่อไปนี้ เรียกว่า **cross-entropy** ของโปรไฟล์เมทริกซ์ และเอนโทรปีสัมพัทธ์ก็คือผลต่างระหว่างค่าครอสเอนโทรปีกับค่าเอนโทรปีนั่นเอง

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

ถ้าคำนวณค่าเอนโทรปีสัมพัทธ์ของโมทิฟ **GCCG** ข้างต้น จะได้ค่าเป็น 9.85-3.53 = 6.32 ดังแสดงต่อไปนี้ โดยในตัวอย่างนี้ 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 ที่มีความอนุรักษ์มากกว่ามีคะแนนรวมน้อยกว่าแต่ไม่มีความหมายในทางชีววิทยา มีค่าเอนโทรปีสัมพัทธ์เท่ากับ 4.18-0.72 = 3.46 ดังแสดงต่อไปนี้ ดังนั้นการใช้เอนโทรปีสัมพัทธ์ก็ทำให้เราสามารถหาโมทิฟที่ถูกต้องได้อีกครั้ง

Motifs	a	a	a	a
	a	a	a	a
	a	a	a	a
	a	t	a	a
	a	a	a	a

PROFILE *(Motifs)*	A: 1.0	0.8	1.0	1.0
	C: 0.0	0.0	0.0	0.0
	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) เป็นแบบจำลองที่ใช้ในการแสดงไบน์ดิงไซต์ของทรานสคริปชันแฟคเตอร์หนึ่งๆ แทนการใช้สตริงเสียงข้างมาก (consensus string) PWM นี้ถูกนำเสนอเป็นครั้งแรกโดย Gary Stormo และคณะในปีค.ศ. 1982 [86] ซึ่งในเวลานั้นถูกนำมาใช้เป็นแบบจำลองแสดงบริเวณในสายอาร์เอ็นเอที่น่าจะเป็นจุดเริ่มของการแปลรหัสจากอาร์เอ็นเอไปเป็นโปรตีน (translation start site) ในเชื้ออีโคไล (*E. coli*) โดยในผลงานนั้นประโยชน์หลักของ PWM คือถูกใช้เป็นเครื่องมือในการสแกนหาว่ายีนอื่นๆ ที่ไม่ได้ถูกพิจารณามีจุดเริ่มของการแปลรหัสจากอาร์เอ็นเอไปเป็นโปรตีนที่บริเวณไหน

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 ตัวอย่างขั้นตอนการสร้าง Position Weight Matrix (PWM) จากชุดของโมทิฟ
(ที่มา: ดัดแปลงจาก https://en.wikipedia.org/wiki/Position_weight_matrix)

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) ของ $b_r = 1/|r|$ โดย $|r|$ คือจำนวนอักขระทั้งหมดที่เป็นไปได้ ในกรณีของดีเอ็นเอ $b_r = 1/|r| = ¼ = 0.25$ ในกรณีของกรดอะมิโน $b_r = 1/|r| = 1/20 = 0.05$ ทั้งนี้ค่า b_r โดยปริยายนี้อยู่บนสมมติฐานว่าโอกาสในการเกิดนิวคลีโอไทด์แต่ละแบบในสายดีเอ็นเอหนึ่งๆหรือโอกาสในการเกิดกรดอะมิโนแต่ละแบบในสายโปรตีนหนึ่งๆมีจำนวนใกล้เคียงกัน ถ้ามีนิวคลีโอไทด์หรือกรดอะมิโนบางแบบปรากฏมากเป็นพิเศษค่า b_r สามารถคำนวณได้โดยใช้ตัวอย่างในหัวข้อเอนโทรปีสัมพัทธ์ข้างต้น

หลังจากได้ PWM เราสามารถนำ PWM ไปสแกนหาโมทิฟใน upstream regions ของยีนอื่นๆ หรือยีนทั้งจีโนมเพื่อดูว่ามี upstream regions ของยีนเหล่านั้นมีโมทิฟที่เราสนใจแทรกอยู่หรือไม่ ทั้งนี้การสแกนหานี้คือการนำ PWM ไปทาบกับ upstream region ทีละ *k* เบส (ตัวอักษรสีแดง) ในตัวอย่างต่อไปนี้ และหาผลบวกของ *k* เบสนั้นโดยใช้ค่าของแต่ละเบสตามตำแหน่งที่ปรากฏใน 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
```

โดยตัวอย่างข้างต้นนี้คะแนนของ ACCGTAAGT มากกว่าคะแนนของ CCGTAAGTT ซึ่งหมายถึง ACCGTAAGT มีโอกาสเป็นโมทิฟที่กำลังค้นหามากกว่า

การหาดีเอ็นเอโมทิฟในบทเรียนนี้สามารถนำไปประยุกต์ใช้กับการแก้ปัญหาอื่นๆที่เกี่ยวข้องเช่นการหาไบน์ดิงไซต์ของไมโครอาร์เอ็นเอจำเพาะหนึ่งๆ บนชุดของดีเอ็นเอหรืออาร์เอ็นเอเป้าหมาย การหาไบน์ดิงไซต์ของโปรตีนบนชุดของอาร์เอ็นเอเป้าหมายเป็นต้น

ตัวอย่างโปรแกรมที่ใช้ในการหาโมทิฟที่มีการใช้งานกันอย่างแพร่หลาย

MEME [87, 88] เป็นชุดของเครื่องมือที่ช่วยในการวิเคราะห์โมทิฟของทั้งโปรตีน ดีเอ็นเอ และอาร์เอ็นเอ โดยชุดของเครื่องสามารถแบ่งได้ออกเป็น 4 กลุ่มหลักๆ ประกอบด้วย (1) ชุดเครื่องมือที่ใช้ในการหา *de novo* โมทิฟ ซึ่งมีเป้าหมายเดียวกันกับเนื้อหาหลักของบทเรียนนี้ (2) ชุดเครื่องมือทางสถิติที่ใช้โมทิฟที่อยู่ในฐานข้อมูลที่มีการรายงานมาก่อนเพื่อทดสอบการปรากฏของโมทิฟเหล่านี้อย่างมีนัยยะสำคัญในชุดของข้อมูลเข้าจากผู้ใช้ (3) เครื่องมือที่ใช้ในการค้นหาโมทิฟ (การค้นหาโดยนำ PWM ไปสแกน upstream regions ของยีนต่างๆ ที่มีการอธิบายในบทเรียนนี้เป็นตัวอย่างการค้นหาโมทิฟในเครื่องมือกลุ่มนี้) โดยแต่ละเครื่องมือในชุดนี้มีเป้าหมายจำเพาะแตกต่างกันไป เช่น MAST อัลกอริทึม รับข้อมูลเข้าเป็นชุดของโมทิฟจากผู้ใช้งานและฐานข้อมูลที่ผู้ใช้เลือก โดยผลของการทำงานคือการให้คะแนนแต่ละสายข้อมูลในฐานข้อมูลที่ถูกเลือกตามการปรากฏของชุดของโมทิฟที่เป้นข้อมูลเข้า ในขณะที่ MCAST อัลกอริทึมเหมาะกับการใช้สแกนจีโนมโดยเน้นการหาชุดของไบน์ดิงไซต์ หรือซิสเรกูลาทอรีโมดูล (*cis*-regulatory modules: CRMs) ซึ่งอาจถูกจับโดยชุดของทรานสคริปชันแฟคเตอร์ที่มีการรายงานมาก่อน เป็นต้น (4) เครื่องมือเพื่อการเปรียบเทียบ *de novo* โมทิฟที่หาได้โดย MEME กับที่หาได้จากเครื่องมืออื่นๆ ชุดเครื่องมือ MEME นี้ นอกจาก MEME suite ที่มีการใช้งานอย่างแพร่หลายและมีการพัฒนาเพิ่มเติมอย่างต่อเนื่องแล้ว ยังมีผลงานวิจัยอื่นๆอีกหลายผลงานที่ใช้แนวทางในการหาโมทิฟแตกต่างกันไปตามตัวอย่างที่รีวิวใน [89] และตัวอย่างผลการเปรียบเทียบประสิทธิภาพของบางเครื่องมือเหล่านี้สามารถศึกษาเพิ่มเติมได้จาก [90] เป็นต้น

ตัวอย่างฐานข้อมูลโมทิฟ

ตัวอย่างฐานข้อมูลโมทิฟที่มีการใช้งานอย่างแพร่หลาย เช่น JASPAR (http://jaspar.genereg.net/) [91-99] เป็นฐานของมูลของไบน์ดิงไซต์ของทรานสคริปชันแฟคเตอร์ในรูปแบบของ Position Frequency Matrices หรือ PFM ที่สามารถนำไปสร้างเป็น Position Weight Matrices (PWM) ต่อได้ โดยรวบรวมจากผลการทดลองโดยวิธีการทางแล็บแบบต่างๆ ข้อมูลในฐานข้อมูลมาจากสิ่งมีชีวิตที่หลากหลายทั้งสัตว์เลี้ยงลูกด้วยนม สัตว์มีกระดูกสันหลัง พืช และแมลง เป็นต้น โดยมีการปรับเพิ่มข้อมูลอย่างต่อเนื่องจนปัจจุบัน Cis-BP (http://cisbp.ccbr.

utoronto.ca) [100] เป็นฐานไบน์ดิงไซต์ของโปรตีนที่จับกับดีเอ็นเอในรูปแบบ Position Weight Matrices (PWM) ของสิ่งมีชีวิตกลุ่มยูแคริโอตจำนวนมาก ทั้งนี้ข้อมูลหลักส่วนหนึ่งของ Cis-BP มาจากการทำนาย UniPROBE (http://thebrain.bwh.harvard.edu/uniprobe/) [101] เน้นการเก็บข้อมูลไบน์ดิงไซต์จากการทดลอง protein binding microarray (PBM) โดยเน้นการวัดการจับของชุดโปรตีนที่สกัดจากสิ่งมีชีวิตต่างๆ ทั้งกลุ่มที่เป็นโพแคริโอตเช่น เชื้อ *Vibrio harveyi* และกลุ่มยูแคริโอตเช่นยีสต์ (*Saccharomyces cerevisiae*) หนอน (*Caenorhabditis elegans*) หนูและมนุษย์ โดยข้อมูลอยู่ในรูปแบบ PWM ฐานข้อมูล TFBSshape [102] เป็นผลงานวิจัยที่ได้รับการจัดกลุ่มโดยวารสาร Nucleic Acid Research ให้อยู่ในกลุ่มที่ช่วยทำให้เกิดความก้าวหน้าอย่างมาก (breakthrough) โดยผลงานวิจัยได้ใช้ข้อมูลไบน์ดิงไซต์จากฐานข้อมูลและข้อมูลเกี่ยวกับโครงสร้างรูปทรงของดีเอ็นเอมาพัฒนาเป็นเครื่องมือและฐานข้อมูลทางชีวสารสนเทศที่ช่วยในการวิเคราะห์คำนวณคุณลักษณะในเชิงโครงสร้างดีเอ็นเอจากชุดของสายนิวคลีโอไทด์ที่รวบรวมจากฐานข้อมูล JASPAR และ UniPROBE ซึ่งสามารถนำไปวิเคราะห์ความจำเพาะในการจับของทรานสคริปชันแฟคเตอร์ ฐานข้อมูล TRANSFAC [103-105] เป็นฐานข้อมูลหลักในการเก็บข้อมูลของทรานสคริปชันแฟคเตอร์และยีนเป้าหมายในงานวิจัยสมัยแรกๆ อย่างไรก็ตามข้อมูลหลังปี ค.ศ. 2005 อยู่ในฐานข้อมูล TRANSFAC®Professional ซึ่งเป็นฐานข้อมูลปิดเชิงพาณิชย์ โดยข้อมูลก่อนปี ค.ศ. 2006 ยังเปิดอยู่เป็นสาธารณะ นอกจากฐานข้อมูลที่เก็บไบน์ดิงไซต์มาจากสิ่งมีชีวิตที่หลากหลายแล้ว ยังมีฐานข้อมูลอีกกลุ่มที่เน้นการเก็บข้อมูลจำเพาะสิ่งมีชีวิตหรือกลุ่มของสิ่งมีชีวิต เช่น ฐานข้อมูล HOCOMOCO [106-108] เน้นการเก็บข้อมูลไบน์ดิงไซต์ของทรานสคริปชันแฟคเตอร์ในมนุษย์ในรูปแบบ PWM ที่เน้นความถูกต้องและคุณภาพของข้อมูล โดยในเวอร์ชันถัดๆมาจะรวมข้อมูลของหนูด้วย ฐานข้อมูล PRODORIC (http://www.prodoric.de) [109] ที่เป็นข้อมูลไบน์ดิงไซต์กลุ่มโพแคริโอต เป็นต้น ข้อมูลสรุปเกี่ยวกับแต่ละฐานข้อมูลสามารถศึกษาเพิ่มเติมได้จาก [110] สำหรับเทคโนโยลีทางแล็บที่ใช้ในการศึกษาการจับกันระหว่างโปรตีนกับดีเอ็น รวมทั้งผลกระทบต่อการออกแบบอัลกอริทึมสามารถศึกษาเพิ่มได้จาก [111, 112] เป็นต้น ตัวอย่างของเครื่องมือที่ใช้ในการแสดงไบน์ดิงไซต์ที่มีการใช้งานอย่างแพร่หลายคือ Sequence Logo [113], WebLogo [114] และมีเครื่องมือที่ช่วยในการแปลงจากซีเควนโลโก้กลับเป็น PWM ด้วยตัวอย่างเช่น [115] เป็นต้น

แบบฝึกหัดบทที่ 4

1. ให้เขียนโปรแกรมเพื่อแก้ปัญหาที่เกี่ยวข้องกับการหาโมทิฟโดยใช้โจทย์ที่โรซาลินด์ (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) *(Optional)* Implement GibbsSampler (http://rosalind.info/problems/ba2g/)

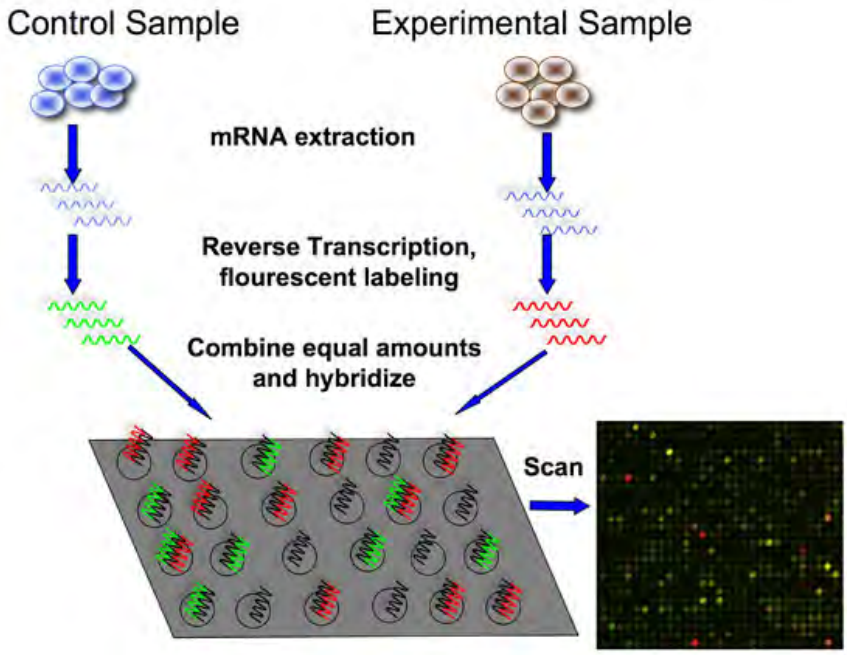
2. เราจะประยุกต์ใช้วิธีการข้างต้นในการหาโมทิฟในชุดของสายโปรตีนได้อย่างไร
3. ถ้าเรามีโปรไฟล์เมทริกซ์ของทรานสคริปชันแฟคเตอร์ ABC แล้ว ต้องการหาว่ายังมียีนใดอีกบ้างในจีโนมที่อาจจะเป็นยีนเป้าหมายของ ABC เราต้องการข้อมูลอะไรบ้างและต้องทำอย่างไร
4. จากข้อมูล Position Frequency Matrix ของ SPI1 ใน JASPAR (MA0080.4) สร้าง Position Weight Matrix (PWM) ตามตัวอย่างต่อไปนี้
   https://davetang.org/muse/2013/10/01/position-weight-matrix/

ภาคผนวกบทที่ 4

ดีเอ็นเออะเรย์

ดีเอ็นเออะเรย์ (DNA array) เป็นชุดของดีเอ็นเอโมเลกุลที่ถูกนำมาติดไว้กับชิปซิลิคอนหรือแผ่นแก้วที่มีลักษณะเป็นช่องๆ เหมือนตารางสองมิติ โดยในแต่ละช่องจะมีลำดับเบสดีเอ็นเอที่มีความจำเพาะเรียกว่าโพรบ (probe) ติดอยู่เพื่อใช้ในการวัดปริมาณการแสดงออกของยีนเป้าหมายหรือทาร์เกต (target) ที่จำเพาะกับโพรบนั้นๆ ในเทคโนโลยีอะเรย์ที่มีอยู่ โพรบจะถูกสังเคราะห์ขึ้นให้มีความจำเพาะกับแต่ละยีนในสิ่งมีชีวิตที่ต้องการนำมาทดสอบ ในการออกแบบการทดลองมักจะมีการสกัดเมสเซ็นเจอร์อาร์เอ็นเอจากเซลล์มาตรฐานเพื่อใช้เป็นข้อมูลอ้างอิงกับเมสเซ็นเจอร์อาร์เอ็นเอของเซลล์ที่มีเงื่อนไขจำเพาะ เช่นเซลล์ของ MTB ในสภาวะปกติกับเซลล์ของ MTB ภาวะพร่องออกซิเจนเป็นต้น โดยเมสเซ็นเจอร์อาร์เอ็นเอเหล่านี้ถูกแปลงย้อนกลับเป็นซีดีเอ็นเอ (cDNA) และจะติดแท็กด้วยฟลูออเรสเซนต์ด้วยสีที่แตกต่างกันเช่นซีดีเอ็นเอที่มาจากเซลล์ปกติติดสีแท็กสีเขียว ส่วนซีดีเอ็นเอที่มีจากเซลล์ที่อยู่ในเงื่อนไขที่สนใจติดสีแท็กสีแดง หลังจากนั้นจะนำซีดีเอ็นเอติดแท็กเหล่านี้ไปทดสอบการจับกับโพรบบนชิป

ถ้าซีดีเอ็นเอเป็นคู่สบกับโพรบใดก็จะมีการปล่อยแสงฟลูออเรสเซนต์ออกมา โดยความเข้มของแสงขึ้นอยู่กับจำนวนของซีดีเอ็นเอหรืออาร์เอ็นเอของยีนจำเพาะหนึ่งๆที่แสดงออกมานั่นเอง การทดลองหา evening elements ที่มีการกล่าวถึงในตอนต้นบทเรียนนั้นใช้ดีเอ็นเออะเรย์ชิปที่สามารถวัดการแสดงออกของยีนใน *Arabidopsis thaliana* ได้ 8,000 ยีนพร้อมกัน



รูปที่ 4.14 ดีเอ็นเออะเรย์

(ที่มา: http://bitesizebio.s3.amazonaws.com/content/uploads/2011/07/cDNA-microarray-experiment.jpg)

บทที่ 5 การเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีน (Sequence alignment)

วัตถุประสงค์

- เพื่อให้นิสิตได้เห็นแนวทางในการวิเคราะห์ข้อมูลแนวทางหนึ่งที่มีความสำคัญและเกี่ยวข้องกับการแก้ปัญหาทางชีววิทยาหลายปัญหาทั้งการเปรียบเทียบเทียบยีน/โปรตีนเพื่ออนุมานฟังก์ชัน การเปรียบเทียบยีน/โปรตีนเพื่อหาโดเมนหรือส่วนของสายข้อมูลที่มีความอนุรักษ์เพื่ออนุมานความสำคัญและฟังก์ชัน การเปรียบเทียบยีน/โปรตีนเพื่อการวิเคราะห์ความสัมพันธ์ในเชิงวิวัฒนาการเบื้องต้น เป็นต้น
- เพื่อให้นิสิตคุ้นเคยกับข้อมูลและองค์ความรู้ที่เกี่ยวข้องและเข้าใจการพัฒนาการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีน
- เพื่อให้นิสิตได้เห็นตัวอย่างงานวิจัยและผลงานวิจัย รวมทั้งตัวอย่างโปรแกรมที่ใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนทั้งแบบเปรียบเทียบระหว่างคู่ของสายข้อมูล ชุดของสายข้อมูล และการสืบค้นสายข้อมูลกับฐานข้อมูลขนาดใหญ่
- เพื่อให้นิสิตได้เห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทาย รวมทั้งงานวิจัยอื่นๆ ที่เกี่ยวข้อง

ผลลัพธ์ที่คาดหวัง

- นิสิตเห็นที่มาของโจทย์ทางชีววิทยาที่ต้องการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นและหรือโปรตีน
- นิสิตเข้าใจคุณลักษณะของข้อมูลเข้า
- นิสิตสามารถอธิบายการทำงานของอัลกอริทึมหลักๆ ที่ใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนได้ และเข้าใจความสำคัญและการใช้งานเมทริกซ์คะแนนแบบต่างๆ เช่นเมทริกซ์คะแนนแพมและบลอสซัม เป็นต้น
- นิสิตสามารถเขียนโปรแกรมที่ใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนได้

- นิสิตสามารถยกตัวอย่างโปรแกรมที่ใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีน
- นิสิตสามารถยกตัวอย่างความท้าทายที่ยังมีอยู่และสามารถนำเสนอแนวทางในการพัฒนาวิธีการแก้ปัญหาเหล่านี้ได้ รวมทั้งสามารถประยุกต์องค์ความรู้จากบทเรียนเพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้

เนื้อหาโดยสรุป

การเปรียบเทียบความคล้ายคลึงระหว่างสายดีเอ็นเอและหรือโปรตีนเป็นขั้นตอนพื้นฐานขั้นตอนหนึ่งในวิธีการทางชีวสารสนเทศที่มักใช้ในการอนุมานฟังก์ชันของสายข้อมูลเข้าโดยเปรียบเทียบกับฐานข้อมูลของโปรตีน การศึกษาความเกี่ยวเนื่องกันของสายดีเอ็นเอและหรือโปรตีนซึ่งนำไปสู่การอนุมานความสัมพันธ์ในเชิงวิวัฒนาการ ทั้งนี้ปัญหาการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนนี้เป็นปัญหาเชิงอัลกอริทึม เรียกรวมๆ ว่าการทำ sequence alignment ซึ่งประกอบด้วย (1) การเปรียบเทียบความคล้ายคลึงกันระหว่างสายข้อมูลสองเส้น (pair-wise alignment) โดยอัลกอริทึมพื้นฐานจะใช้กำหนดการพลวัต (dynamic programming) ในการขยับลำดับเบสหรือกรดอะมิโนให้ตรงกันมากที่สุดในองค์รวม (global alignment) หรือเฉพาะบริเวณ (local alignment (2) การเปรียบเทียบความคล้ายคลึงกันของสายข้อมูลมากกว่าสองสาย (multiple sequence alignment) โดยการทำ pair-wise alignment ระหว่างสายข้อมูลแต่ละเส้นกับสายข้อมูลเส้นอื่นๆภายในชุด ซึ่งคู่ของสายข้อมูลที่คล้ายคลึงกันมากที่สุดจะถูกนำมารวมกันและแสดงด้วยสายข้อมูลตัวแทน และทำการเปรียบเทียบสายข้อมูลตัวแทนนี้กับสายข้อมูลเส้นอื่นๆ ภายในชุด และวนทำซ้ำจนกว่าจะครบจำนวนสายข้อมูลภายในชุด (3) การเปรียบเทียบสายข้อมูลเข้ากับสายข้อมูลภายในฐานข้อมูลขนาดใหญ่ เช่น ฐานข้อมูลโปรตีนของ NCBI และ UniProt เป็นต้น โดยอัลกอริทึมหลักถูกพัฒนาในโปรแกรม BLAST [1] ที่มีการใช้งานกันอย่างแพร่หลายมาก

บทที่ 5 การเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีน (Sequence alignment)

ในปีค.ศ. 1983 ดูลิตเติล (Doolittle) และคณะ [116] และวอเตอร์ฟิลด์ (Waterfield) และคณะ [117] ได้ทำการเปรียบเทียบลำดับกรดอะมิโนของยีน platelet derived growth factor (PDGF) ที่ถูกถอดรหัสออกมากับลำดับลำดับกรดอะมิโนของยีนอื่นๆที่มีข้อมูลอยู่ในช่วงเวลานั้นโดยผลการเปรียบเทียบนี้ทำให้นักชีววิทยาโรคมะเร็งประหลาดใจเป็นอย่างมาก เนื่องจากลำดับเบสของยีน PDGF มีความคล้ายคลึงกับลำดับเบสของยีน v-sis มากความคล้ายกันมากนี้เป็นเรื่องพิศวงเพราะยีนทั้งสองนี้มีฟังก์ชันการทำงานต่างกันมาก โดยยีน PDGF จะแปลรหัสไปเป็นโปรตีนที่ทำงานเกี่ยวกับการกระตุ้นการเติบโตของเซลล์ ในขณะที่ยีน v-sis เป็นยีนที่ก่อมะเร็ง (oncogene) หลังจากการค้นพบของดูลิตเติลนักวิทยาศาสตร์ได้ตั้งสมมติฐานว่ามะเร็งบางรูปแบบอาจเกิดจากการยีนที่ดีทำงานปกติในเวลาที่ไม่ถูกต้อง ความเชื่อมโยงระหว่างยีน PDGF และ v-sis สร้างกระบวนทัศน์ใหม่ในการศึกษาเกี่ยวกับสายนิวคลีโอไทด์ที่เป็นข้อมูลใหม่ซึ่งจะถูกนำมาเปรียบเทียบความคล้ายคลึงกับสายดีเอ็นเอหรือสายของโปรตีนในฐานข้อมูลเป็นขั้นตอนแรกในงานทางด้านจีโนมิกส์ในปัจจุบัน การได้นิวคลีโอไทด์สายใหม่จำนวนมากในปัจจุบันมักเกิดจากโครงการถอดรหัสจีโนมของสิ่งมีชีวิตใหม่ๆ เช่น ตัวอย่างผลงานวิจัยของผู้เขียนในโครงการถอดรหัสจีโนมเชื้อรา [52] ที่เมื่อทำนายบริเวณในจีโนมที่เป็นยีนได้ทั้งหมดแล้ว ลำดับนิวคลีโอไทด์ของยีนที่ทำนายได้เหล่านี้จะถูกนำมาเทียบเคียงความคล้ายคลึงกับสายโปรตีนที่มีอยู่ในฐานข้อมูลเปิดสาธารณะผ่านโปรแกรม BLAST เช่น ฐานข้อมูล nr (Non-Redundant) ซึ่งเป็นฐานข้อมูลที่เก็บสายข้อมูลโปรตีนจำนวนมากมายของสิ่งมีชีวิตชนิดต่างๆ โดยผลของการเทียบเคียงจะถูกนำมาใช้เพื่ออนุมานฟังก์ชันของยีนต่างๆ ที่ทำนายได้ เป็นต้นสำหรับจำนวนสายโปรตีนที่อยู่ในฐานข้อมูลโปรตีน (nr) ที่ NCBI เข้าถึงเมื่อวันที่ 2 กุมภาพันธ์ พ.ศ. 2561 นั้นมีจำนวนทั้งสิ้น 478,964,146 รายการ

ทำความรู้จักการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีน

การเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนในมุมมองของการเล่นเกมส์

ตัวอย่างปัญหาเริ่มต้นจะเป็นการเปรียบเทียบกันระหว่างสายข้อมูลเพียงสองเส้น โดยการวัดความเหมือนกันระหว่างสายข้อมูลจะใช้ระยะทางแฮมมิง (Hamming distance) ซึ่งจะนับจำนวนเบสที่แตกต่างกันระหว่างสายข้อมูลสองเส้น โดยอาจตั้งเงื่อนไขเข้มงวดว่าต้องเทียบอักขระต่ออักขระที่อยู่ในลำดับเดียวกันระหว่างสายข้อมูลทั้งสองเส้น อย่างไรก็ตามสายดีเอ็นเอมักเกิดการเพิ่มหรือลดเบส ดังนั้นเงื่อนไขข้างต้นจึงไม่เหมาะสม และมีการปรับวัตถุประสงค์โดยเป็นการหาอักขระในอีกสายที่ตรงกับอักขระตัวปัจจุบันในดีเอ็นเอสายแรก ตัวอย่างเช่น มีสาย ดีเอ็นเอ **ATGCATGG** และ **TGCATGCA** ที่ไม่มีตำแหน่งใดเลยที่มีอักขระตรงกัน และมีระยะทางแฮมมิงเท่ากับ 8

อย่างไรก็ตามถ้ามีปรับเลื่อนสายดีเอ็นเอให้เหมาะสมเราสามารถทำให้ได้ระยะทางแฮมมิงลดลงเหลือ 2 ดังตัวอย่างต่อไปนี้

ATGCATGG-
-TGCATGCA

หรือตัวอย่างที่ดีเอ็นเอสองสายอาจมีความเหมือนในลำดับเบสย่อย เช่น

ATGC-TTA-
-TGCATTAA

ตัวอย่างข้างต้นนี้นำไปสู่สมมติฐานของวิธีการวัดผลการเทียบดีเอ็นเอสองสายที่เหมาะสม โดยพยายาม เลื่อนลำดับเบสในดีเอ็นเอสายหนึ่งให้มีจำนวนเบสที่ตรงกับเบสในดีเอ็นเอสายที่สองมากที่สุด โดยวิธีการพื้นฐานเพื่อให้ได้จำนวนเบสที่ตรงกันมากที่สุดนี้สามารถทำได้โดยนำเบสแรกของดีเอ็นเอทั้งสองเส้นออกไปจากสายถ้าเป็นเบสเดียวกันและเพิ่มให้ 1 คะแนน แต่ถ้าไม่ตรงกันให้นำเบส 1 ตัวออกจากสายดีเอ็นเอเส้นใดเส้นหนึ่งเพื่อให้ได้เบสถัดไปตรงกับเบสปัจจุบันของดีเอ็นเออีกสาย และทำไปเรื่อยๆจนกว่าดีเอ็นเอเส้นใดเส้นหนึ่งจะไม่มีเบสเหลือ โดยมีเป้าหมายว่าจะได้จำนวนของเบสที่ตรงกันจำนวนมากที่สุด

ปัญหาการหาสตริงย่อยร่วมที่ยาวที่สุด

เรากำหนดการเทียบความคล้ายคลึงกันของดีเอ็นเอสาย *v* และ *w* โดยใช้เมทริกซ์สองแถวโดยแถวแรกเก็บลำดับเบสของดีเอ็นเอสาย *v* และแถวที่สองเก็บลำดับเบสของดีเอ็นเอสาย *w* ตามลำดับโดยเรียงซ้ายไปขวา และอาจมีการแทรกอักขระ '-' ในตำแหน่งที่เบสไม่ตรงกัน ตัวอย่างต่อไปนี้เป็นการเลื่อนเบสให้สอดคล้องกันโดยใช้ข้อมูลสายดีเอ็น 2 เส้นคือ ATGTTATA และ ATCGTCC

v: A T – G T T A T A
w: A T C G T – C – C

ในตัวอย่างข้างต้นนี้ในแต่ละคอลัมน์ที่มีเบสที่ตรงกันเรียกว่าแมช (matches) และแสดงถึงเบสที่อนุรักษ์ร่วมกันระหว่างดีเอ็นเอสองสาย สำหรับคอลัมน์ที่มีเบสต่างกันเรียกว่ามิสแมช (mismatches) และคอลัมน์ที่มี '-' แสดงการเกิดอินเดล (indels) โดยคอลัมน์ที่มี '-' ในบรรทัดบน (*v*) เรียกว่าเกิด insertion หมายถึงมีการเพิ่ม '-' เข้าไป เพื่อให้ *v* มีความคล้ายกับ *w* มากขึ้นในขณะที่คอลัมน์ที่มี '-' ในบรรทัดล่าง (*w*) เรียกว่าเกิด deletion ในการเพิ่ม *v* ให้มีความคล้ายกับ *w* มากขึ้น ในตัวอย่างนี้มี 4 แมช 2 มิสแมช 1 insertion และ 2 deletions แมชที่เกิดขึ้นระหว่างดีเอ็นเอสองสายเป็นตัวกำหนดส่วนของลำดับเบสที่เกิดขึ้นร่วมกัน (common sub sequences) โดยไม่จำเป็นต้องอยู่ติดกันทั้งหมด โดยในตัวอย่างข้างต้น **ATGT** เป็นส่วนของลำดับเบสที่เกิดขึ้นร่วมกันของทั้ง **ATGT**TATA และ **AT**C**GT**CC และเนื่องจากการขยับเบสระหว่างสายดีเอ็นเอมีเป้าหมายเพื่อให้ได้จำนวนเบสที่ตรงกันมากที่สุดดังนั้น **ATGT** ซึ่งเป็นผลจากการขยับเบสข้างต้นจึงเป็นตัวแทนของส่วนของลำดับเบสที่เกิดขึ้นร่วมกันยาวที่สุด (longest common substring) ด้วย ทั้งนี้คู่ของสายดีเอ็นเอใดๆอาจมีส่วนของลำดับเบสที่เกิดขึ้นร่วมกันที่ยาวที่สุดมากกว่า 1 เส้น

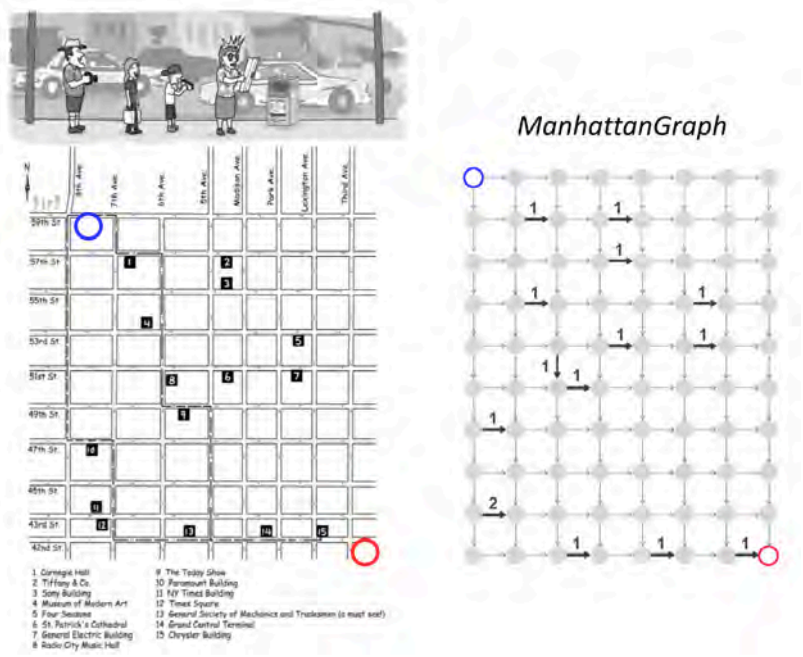
นิยามปัญหาที่ 5.1 ปัญหาการหาสตริงย่อยร่วมที่ยาวที่สุด

ปัญหาการหาสตริงย่อยร่วมที่ยาวที่สุด หาส่วนของสตริงที่เกิดขึ้นร่วมกันที่ยาวที่สุดระหว่างสายสตริงสองเส้น	
ข้อมูลเข้า	สายสตริง 2 เส้น
ผลลัพธ์	ส่วนของสตริงที่เกิดขึ้นร่วมกันที่ยาวที่สุด

ปัญหาการหาเส้นทางเดินชมเมืองแมนแฮตตัน

วางแผนการเดินชมเมืองอย่างไรให้ผ่านจุดท่องเที่ยวได้มากที่สุด

การหาวิธีการเดินชมเมืองแมนแฮตตันให้ผ่านจุดท่องเที่ยวได้มากที่สุดนี้เรียกว่า Manhattan Tourist Problem โดยแผนที่ของเมืองสามารถแสดงได้โดยกราฟแบบมีทิศทาง (directed graph) เรียกว่า *Manhattan Graph* โดยแยกต่างๆจะถูกแสดงด้วยโหนด และเส้นเชื่อมระหว่างโหนดจะมีค่าน้ำหนักซึ่งแสดงจำนวนจุดท่องเที่ยวที่อยู่ในเส้นทางเดินนั้น สำหรับเส้นเชื่อมที่ไม่มีค่าหมายถึงไม่มีจุดท่องเที่ยวในบล็อกทางเดิน



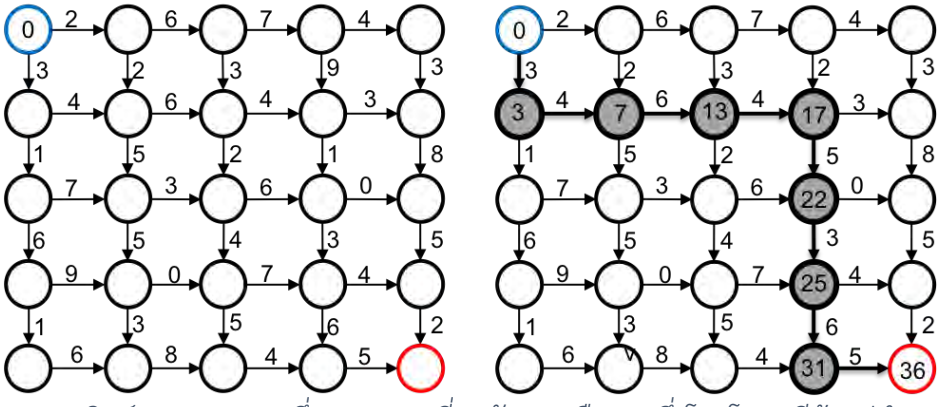
รูปที่ 5.1 (ซ้าย) ตัวอย่างแผนที่ใจกลางเมืองแมนฮัตตันที่มีจุดท่องเที่ยว (กล่องสีดำเล็กๆ) ในถนนสายต่างๆ และ (ขวา) กราฟแบบมีทิศทาง ManhattanGraph ที่แต่ละเส้นเชื่อมจะมีจำนวนจุดท่องเที่ยวในเส้นทางเดินนั้น (ที่มา: รูปที่ 5.2 ของ [21])

วงสีน้ำเงินในรูปที่ 5.1 แสดงจุดตั้งต้นของการเดินเที่ยวเรียกว่าโหนดต้นทาง (source node) และจุดสิ้นสุดการเดินจะเป็นวงสีแดงเรียกว่าโหนดปลายทาง (sink node) โดยสามารถเดินได้เพียงสองทิศทางคือเดินลงล่างหรือเดินไปทางขวามือเท่านั้น จาก *ManhattanGraph* ในโจทย์นี้เราต้องการหาเส้นทางเดินที่ผ่านจุดท่องเที่ยวมากที่สุดหรือ

ต้องการเส้นทางเดินที่มีผลรวมของค่าน้ำหนักมากที่สุดนั่นเอง รูปที่ 5.2 แสดงเมทริกซ์ที่ทำให้เป็นทั่วไปในการแก้ปัญหาโจทย์เดียวกัน

นิยามปัญหาที่ 5.2 ปัญหาการหาเส้นทางเดินชมเมืองแมนฮัตตัน

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



รูปที่ 5.2 (ซ้าย) เมทริกซ์ขนาด n x m ซึ่งแสดงแผนที่จุดตัดของเมืองๆหนึ่งโดยโหนดสีฟ้าอยู่ตำแหน่ง (0,0) และโหนดสีแดงอยู่ที่ตำแหน่ง (4,4) (ขวา) เส้นทางเดินจากโหนดตั้งต้นไปยังโหนดปลายทางโดยวิธีการเลือกเส้นทางแบบโลภ

(ที่มา: ดัดแปลงจากรูปที่ 5.3 ของ [21])

ฝึกหัด	มีเส้นทางเดินที่เป็นไปได้ทั้งหมดกี่เส้นทางในเมทริกซ์ทางซ้ายของรูปที่ 5.2

จากนิยามปัญหาที่ 5.2 และมีเมทริกซ์ตัวอย่างดังในรูปที่ 5.2 (ซ้าย) จะพบว่าวิธีการอย่าง Brute force จะต้องทดลองเส้นทางเดินทั้งหมดที่เป็นไปได้จำนวนมากซึ่งไม่มีประสิทธิภาพ ในขณะที่วิธีการหาเส้นทางแบบโลภใช้เวลาน้อยแต่มักไม่ได้คำตอบที่ดีที่สุดดังตัวอย่างในรูปที่ 5.2 (ขวา) ที่หาเส้นทางเดินได้ค่าความยาวเส้นทางหรือน้ำหนักรวมเท่ากับ 36 ซึ่งไม่ใช่ค่าที่ดีที่สุด

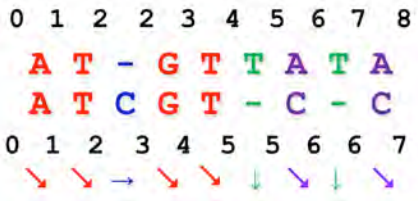
หยุดคิด	เส้นทางที่ยาวที่สุดหรือมีน้ำหนักมากที่สุดในรูปที่ 5.2 (ซ้าย) เป็นเท่าไหร่

นิยามปัญหาที่ 5.2 ข้างต้นสามารถประยุกต์ใช้ได้กับกราฟที่มีทิศทางใด โดยมีเงื่อนไขว่าต้องไม่มีลูปหรือไซเคิล ในกราฟ (directed acyclic graph: DAG)

การเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนกับปัญหาการหาเส้นทางเดินชมเมืองแมนฮัตตัน

ในรูปที่ 5.3 ได้มีการเพิ่มอะเรย์ของค่าจำนวนเต็มแสดงตำแหน่งของการเทียบเบสระหว่างดีเอ็นเอสองสาย โดยอะเรย์ [0 1 2 2 3 4 5 6 7 8] และอะเรย์ [0 1 2 3 4 5 5 6 6 7] แสดงจำนวนของเบสของ **ATGTTATA** และ **ATCGTCC** ที่ถูกใช้ไปแล้ว ณ คอลัมน์นั้นๆ ตามลำดับ นอกจากนี้อะเรย์ที่สาม [↘ ↘ → ↘ ↘ ↓ ↘ ↓↘] แสดงการผลการเทียบเบสว่าเป็นแมชหรือมิสแมช (mathch/mismatch: ↘/↘) เกิด insertion (→) หรือเกิด deletion (↓) โดยอะเรย์นี้แสดงเส้นทางจากโหนดตั้งต้นไปยังโหนดปลายทางในเมทริกซ์ 8 x 7 ที่แสดงในภาพที่ 5.4 (ซ้าย) นั่นเอง โดยโหนดที่ i ของเส้นทางนี้ประกอบด้วยค่าในตำแหน่งที่ i ของอะเรย์ [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)` ↘ `(4,5)` ↓ `(5,5)` ↘ `(6,6)` ↓ `(7,6)` ↘ `(8,7)`



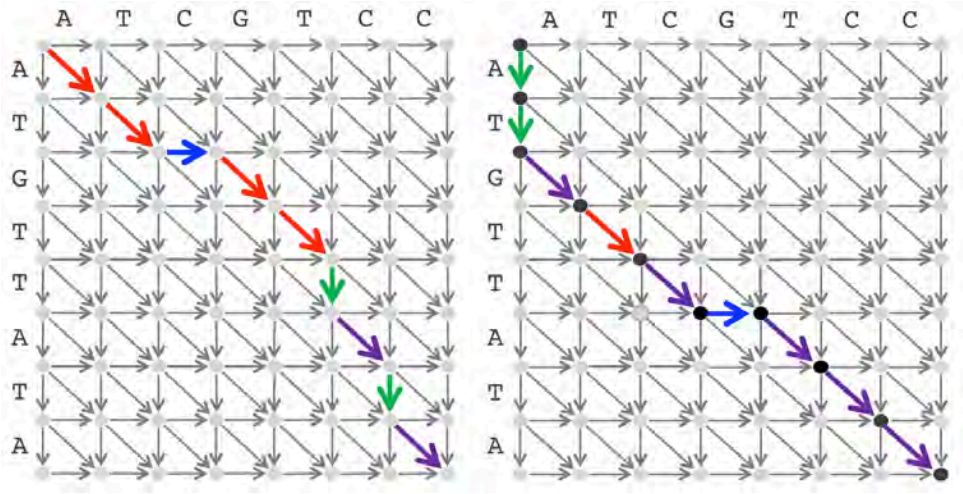
รูปที่ 5.3 การเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอ ATGTTATA และ ATCGTCC

โดยอะเรย์ของตัวเลขแถวบนสุดและแถวที่สี่แสดงจำนวนเบสของสายดีเอ็นเอ ATGTTATA และ ATCGTCC ที่ถูกใช้ไปแล้วในคอลัมน์หนึ่งๆ ตามลำดับ สำหรับอะเรย์ในบรรทัดสุดท้ายแสดงผลของการเปรียบเทียบในแต่ละคอลัมน์ว่าเป็นแมช มิสแมชหรืออินเดล

(ที่มา: รูปที่ 5.5 ของ [21])

การเปรียบเทียบความคล้ายคลึงกันระหว่างดีเอ็นเอสองสายสามารถแสดงได้ด้วยเส้นทางหนึ่งๆในกราฟแสดงการเปรียบเทียบลำดับเบส (alignment graph) ในรูปที่ 5.4 ทางซ้ายเส้นทางที่แสดงคือ `(0,0)` ↘ `(1,1)` ↘ `(2,2)` → `(2,3)` ↘ `(3,4)` ↘ `(4,5)` ↓ `(5,5)` ↘ `(6,6)` ↓ `(7,6)` ↘ `(8,7)` ซึ่งเป็นเส้นทางที่สอดคล้องกับตัวอย่างการเปรียบเทียบดีเอ็นเอสองสายในรูปที่ 5.3 รูปทางขวาแสดงตัวอย่างเส้นทางอื่นที่เป็นเส้นทางที่การเลื่อนเบสได้จำนวนเบสที่แมชเพียง 1 เบส

ฝึกหัด	แสดงผลการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอโดยอ้างอิงจากเส้นทางในกราฟรูป 5.4 ทางขวามือ



รูปที่ 5.4 (ซ้าย) เส้นทางในกราฟแสดงการเปรียบเทียบลำดับเบสแสดงการเปรียบเทียบการคล้ายคลึงกันระหว่างดีเอ็นเอสองสายคือ ATGTTATA และ ATCGTCC (ขวา) ตัวอย่างเส้นทางอื่นซึ่งแสดงการเลื่อนเบสระหว่างสายดีเอ็นเออีกแบบซึ่งมีเพียง 1 เบสที่แมช

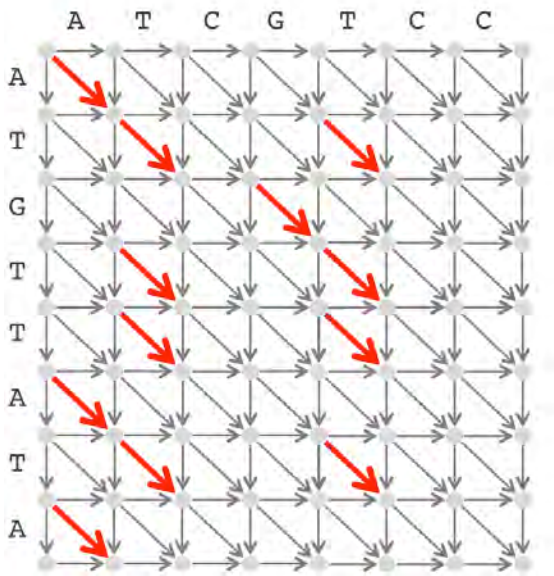
(ที่มา: รูปที่ 5.6 ของ [21])

หยุดคิด	เราสามารถใช้กราฟแสดงการเปรียบเทียบลำดับเบส (alignment graph) ในการหาส่วนของสตริงที่ยาวที่สุดที่ปรากฏอยู่ในสตริงทั้งสองสาย (Longest Common Substring: LCS) ได้ หรือไม่

รูปที่ 5.5 แสดงกราฟแสดงการเปรียบเทียบลำดับเบสระหว่าง **ATGTTATA** และ **ATCGTCC** หรือ `AlignmentGraph`(**ATGTTATA**, **ATCGTCC**) ที่มีการเน้นการแมช (↘) ระหว่างเบสที่เป็นไปได้ทั้งหมด โดยแต่ละเส้นเชื่อมที่แสดงการแมชนี้จะมีคะแนนเท่ากับ 1 ในขณะที่เส้นเชื่อมอื่นๆทั้งหมดมีคะแนนเป็น 0 กราฟแสดงการเปรียบเทียบลำดับเบสในรูปที่ 5.5 นี้ สามารถนำไปใช้ในการออกแบบอัลกอริทึมที่ใช้ในการหาเส้นทางที่ยาวที่สุดในกราฟแบบมีทิศทางและไม่มีลูป (DAG) โดยอัลกอริทึมหลักที่ใช้ในการแก้ปัญหานี้คือไดนามิกโปรแกรมมิง (dynamic programming)

ไดนามิกโปรแกรมมิงกับกราฟแบบมีทิศทางและไม่มีลูป

ถ้ามีโหนด *b* อยู่ในกราฟแบบมีทิศทางและไม่มีลูป (DAG) และ s_b เป็นเส้นทางที่ยาวที่สุดจากโหนดตั้งต้นมายังโหนด *b* เราเรียกโหนด *a* ว่าเป็นพรีดีเซสเซอร์ (predecessor) ของ *b* ถ้ามีเส้นเชื่อมจาก *a* มายัง *b* ใน DAG และจำนวนเส้นทางเข้า (indegree) ของโหนดหนึ่งๆ จะเท่ากับจำนวนพรีดีเซสเซอร์ของโหนดนั้นๆ คะแนน s_b ของโหนด *b* โดยมีจำนวนเส้นทางเข้าเท่ากับ *k* คำนวณได้จากสมการต่อไปนี้



รูปที่ 5.5 AlignmentGraph(ATGTTATA, ATCGTCC) ที่แสดงการแมช (↘) ทั้งหมดที่เป็นไปได้
(ที่มา: รูปที่ 5.7 ของ [21])

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

จากกราฟแสดงการเปรียบเทียบลำดับเบสในรูปที่ 5.5 สามารถคำนวณเส้นทางที่ยาวที่สุดได้โดยใช้สมการต่อไปนี้

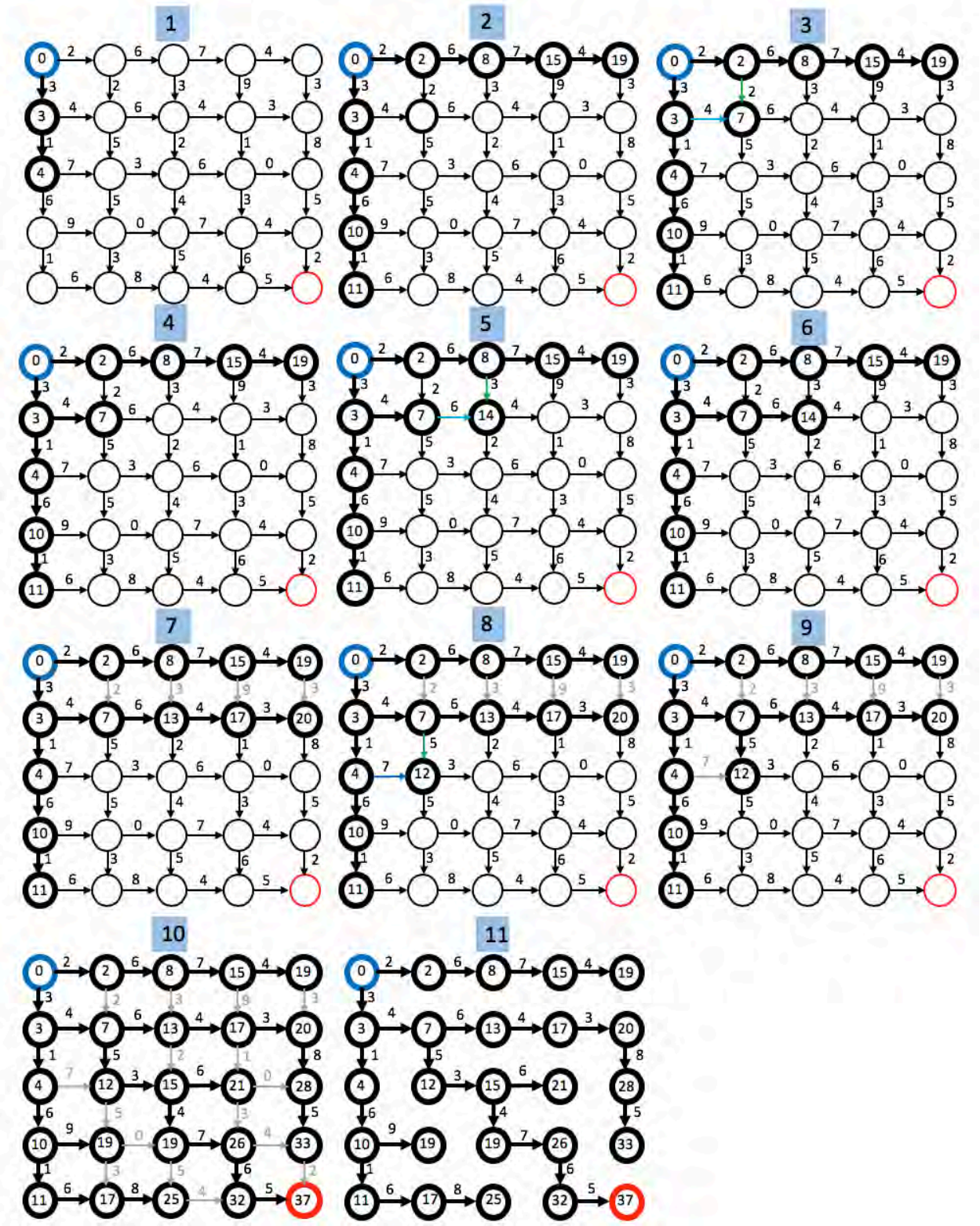
$$s_{i,j} = \max \begin{cases} s_{i-1,j} + 0 \\ s_{i,j-1} + 0 \\ s_{i-1,j-1} + 1\ if\ v_i = w_j \end{cases}$$

และแสดงโดยอนุญาตให้ค่าน้ำหนักมีความเป็นทั่วไปมากขึ้นโดยใช้สมการการต่อไปนี้

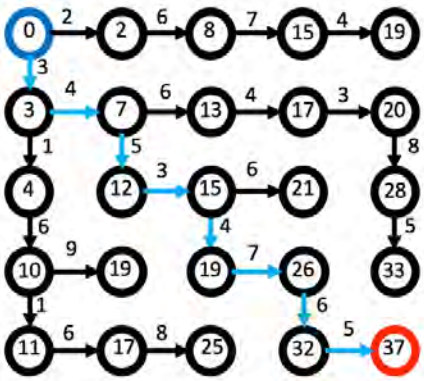
$$s_{i,j} = \max \begin{cases} s_{i-1,j} + weight\ of\ edge\ \downarrow between\ (i-1,j)\ and\ (i,j) \\ s_{i,j-1} + weight\ of\ edge\ \rightarrow between\ (i,j-1)\ and\ (i,j) \\ s_{i-1,j-1} + weight\ of\ edge\ \searrow between\ (i-1,j-1)\ and\ (i,j) \end{cases}$$

หยุดคิด	สมการ recurrence ข้างต้น แม้ไม่การพิจารณาค่ามิสแมช (mismatch) เป็นส่วนหนึ่งสมการแต่ก็ยังใช้ในการหาเส้นทางที่ยาวที่สุดซึ่งอนุมานถึงจำนวนเบสที่ตรงกันมากที่สุดได้ เพราะอะไร

รูปที่ 5.6 แสดงขั้นตอนการหาเส้นทางที่ยาวที่สุดโดยใช้สมการ recurrence ข้างต้น และใช้เมทริกซ์เดียวกับการหาเส้นทางแบบโลภในรูปที่ 5.2 และรูปที่ 5.7 เป็นเส้นทางที่ยาวที่สุดที่เป็นผลลัพธ์สุดท้ายของรูปที่ 5.6



รูปที่ 5.6 แสดงขั้นตอนการหาเส้นทางที่ยาวที่สุดสำหรับกราฟแสดงการเปรียบเทียบลำดับเบสในรูปที่ 5.5 โดยใช้ไดนามิกโปรแกรมมิง



รูปที่ 5.7 เส้นทางที่มีผลรวมค่าน้ำหนักเส้นเชื่อมมากที่สุดจากผลลัพธ์ในรูปที่ 5.6

การเดินย้อนกลับในกราฟแสดงการเปรียบเทียบลำดับเบส

เราสามารถใช้แนวคิดของการเดินย้อนกลับจากโหนดปลายทางไปยังโหนดต้นทางเพื่อแสดงส่วนของดีเอ็นเอที่ยาวที่สุดร่วมกัน (LCS) ระหว่างสายดีเอ็นเอ v และ w จากรูปที่ 5.5 ข้างต้นถ้าเรากำหนดค่าน้ำหนักของเส้นเชื่อมที่แสดงแมช (match) เท่ากับ 1 และเส้นเชื่อมที่เหลือทั้งหมดเป็น 0 ค่าของ $s_{|v|,|w|}$ จะเท่ากับ LCS ของสายดีเอ็นเอ v และ w อัลกอริทึมต่อไปนี้มีการเก็บข้อมูลเส้นเชื่อม (backtracking pointer) ที่ถูกใช้ในระหว่างการคำนวณค่า $s_{i,j}$ โดยมีค่าที่เป็นไปได้ 3 ค่าคือ ↓, → และ↘

สูโดโคดที่ 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 ที่สร้างขึ้นจะสามารถแสดงผลเป็นสตริงของผลการเทียบเบสในแต่ละคอลัมน์โดยใช้ตัวอย่างสูโคโคด `OutputLCS` ต่อไปนี้

สูโดโคดที่ 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	จากตัวอย่างสูโดโคด `OutputLCS` ข้างต้นซึ่งจะแสดงผลของ LCS เพียงสายเดียว ให้ปรับโคด `OutputLCS` และ `LCSBrackTrack` ข้างต้น ให้สามารถหา LCS ทั้งหมดที่มีอยู่ในสายสตริงทั้งสองเส้น

การให้คะแนนของความคล้ายคลึงกัน

ข้อจำกัดในการให้คะแนนความคล้ายคลึงกันระหว่างดีเอ็นเอสองเส้นโดยให้ค่าแมช (match) เป็น 1 ในขณะที่ค่ามิสแมช (mismatch) และอินเดล (indel) ที่เหลือทั้งหมดเป็น 0 อาจทำให้เราสามารถปรับสายดีเอ็นเอเพื่อให้ได้จำนวนแมชมากที่สุดโดยไม่สนใจว่าจะต้องเพิ่มอินเดลเข้าไปเท่าไหร่ อย่างไรก็ตามอินเดลมีความหมายในทางชีววิทยาซึ่งหลายครั้งเกี่ยวข้องกับกระบวนการวิวัฒนาการ ดังนั้นการเพิ่มลดเบสในระหว่างการเทียบลำดับเบสควรต้องมีการพิจารณาการลงโทษในกรณีที่เกิดอินเดลด้วย

เมทริกซ์คะแนน

เพื่อให้การให้คะแนนการเปรียบเทียบความเหมือนระหว่างสายดีเอ็นเอมีการนำเรื่องอินเดลเข้ามาพิจารณา เรายังให้คะแนนแมชของแต่ละเบสเป็น 1 แต่เพิ่มส่วนของการลงโทษโดยมีการกำหนดค่า μ ซึ่งเป็นค่าคงที่ที่เป็นบวกเพื่อนำมาคูณกับจำนวนมิสแมชที่เกิดขึ้นและค่า σ จะเป็นตัวคูณของจำนวนอินเดลที่เกิดขึ้น ซึ่งคะแนนของการเปรียบเทียบความเหมือนนี้สามารถแสดงได้โดย

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

จากตัวอย่างการเปรียบเทียบความเหมือนของดีเอ็นเอสองเส้นต่อไปนี้ 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
```

นักชีววิทยาได้เพิ่มรายละเอียดของการลงโทษเพิ่มเติมโดยใช้องค์ความรู้ที่มีมาก่อนว่า โอกาสหรือความถี่ของการเกิดความไม่ตรงกันเหล่านี้มีไม่เท่ากันสำหรับนิวคลีโอไทด์และกรดอะมิโนแต่ละตัว ดังนั้นคะแนนลงโทษของการเกิดมิสแมชและอินเดลสำหรับแต่ละนิวคลีโอไทด์หรือกรดอะมิโนที่จำเพาะจะมีค่าแตกต่างกันไป โดยคะแนนลงโทษที่แตกต่างกันนี้สามารถกำหนดอยู่ในรูปแบบเมทริกซ์คะแนน ตัวอย่างเช่น ถ้าสายสตริงมีอักขระที่เป็นไปได้ทั้งหมด k แบบ จะมีการสร้างเมทริกซ์คะแนนขนาด (k+1) x (k+1) โดยเก็บคะแนนของการเทียบระหว่างทุกคู่ของอักขระ สำหรับเมทริกซ์คะแนนของดีเอ็นเอที่มีนิวคลีโอไทด์ 4 ประเภท (k=4) โดยกำหนดว่ามิสแมช (mismatches) ทั้งหมดจะมีคะแนนลงโทษเท่ากันคือ μ และอินเดล (indels) ทั้งหมดเป็น σ จะได้เมทริกซ์คะแนนของนิวคลีโอไทด์ที่จะถูกนำไปใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอดังต่อไปนี้

	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 อย่างไรก็ตามเมทริกซ์คะแนนที่ใช้ในการเปรียบเทียบสายโปรตีนจะมีรายละเอียดมากกว่ามากตามจำนวนกรดอะมิโนและตามวิธีการที่ได้มาซึ่งคะแนนในเมทริกซ์ โดยเมทริกซ์คะแนนหลักที่ใช้ในการเปรียบเทียบสายโปรตีนประกอบด้วยเมทริกซ์คะแนนแพม (PAM) และเมทริกซ์คะแนนบลอสซัม (BLOSUM) (ภาคผนวกบทที่ 5)

การเปรียบเทียบความคล้ายคลึงกันแบบภาพรวมและแบบจำเพาะบริเวณ

การเปรียบเทียบความคล้ายคลึงกับแบบภาพรวม

วิธีการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนที่ผ่านมาเป็นการเปรียบเทียบในเชิงภาพรวม (global alignment) โดยมีนิยามปัญหาดังต่อไปนี้

นิยามปัญหาที่ 5.3 ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนแบบภาพรวม

ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนแบบภาพรวม (Global Alignment Problem) หาคะแนนที่มากที่สุดในการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนโดยใช้เมทริกซ์คะแนน	
ข้อมูลเข้า	สายสตริงสองเส้นซึ่งเป็นตัวแทนของสายดีเอ็นเอหรือโปรตีนและเมทริกซ์คะแนน
ผลลัพธ์	ผลการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนโดยมีคะแนนรวมมากที่สุด

เพื่อเป็นการแก้ปัญหาข้างต้น เราจะหาเส้นทางที่ยาวที่สุดในกราฟแสดงการเปรียบเทียบลำดับเบสหลังจากมีการปรับค่าเส้นเชื่อมต่างๆในกราฟโดยใช้เมทริกซ์คะแนนแล้ว โดยสมการที่ใช้ในการคำนวณคะแนนสำหรับแต่ละโหนดในกราฟจะถูกคำนวณด้วยสมการต่อไปนี้

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

ข้อจำกัดของการเปรียบเทียบความคล้ายคลึงกันแบบภาพรวม

การวิเคราะห์ชุดของยีนในกลุ่มโฮมีโอบอกซ์ (homeobox genes) ถูกนำมาใช้แสดงข้อจำกัดของการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนแบบภาพรวม (global alignment) ซึ่งไม่สามารถค้นพบความหมายทางชีววิทยาที่ซ้อนอยู่ ยีนในกลุ่มโฮมีโอบอกซ์นี้มีหน้าที่ในการควบคุมการพัฒนาเอ็มบรีโอและถูกพบในสิ่งมีชีวิตหลายชนิดรวมทั้งแมลงวันและมนุษย์ (รูปที่ 5.8) ยีนกลุ่มโฮมีโอบอกซ์มีขนาดยาวและมีความแตกต่างกันค่อนข้างมากระหว่างสิ่งมีชีวิต อย่างไรก็ตามมีบริเวณย่อยๆในยีนเหล่านี้ที่มีความอนุรักษ์มากระหว่างสิ่งมีชีวิตเรียกว่าโฮมีโอโดเมน (homeodomain) คำถามคือเราจะสามารถหาบริเวณเหล่านี้ซึ่งเป็นเพียงส่วนสั้นๆ ในสายของยีนที่มีความยาวมากกว่ามากได้อย่างไรเพราะคะแนนความคล้ายคลึงในภาพรวมมักมีค่าน้อย เนื่องจากการเปรียบเทียบแบบภาพรวมจะพยายามหาความเหมือนกันตลอดความยาวของสายข้อมูลทั้งสองเส้น อย่างไรก็ตามถ้าเราต้องการหาความคล้ายคลึงกันเฉพาะบริเวณ เช่น บริเวณที่เป็นโฮมีโอโดเมนข้างต้น เราจะต้องพยายามหาความคล้ายคลึงกันเฉพาะในบริเวณที่จำเพาะโดยไม่สนใจความคล้ายคลึงกันในภาพรวม

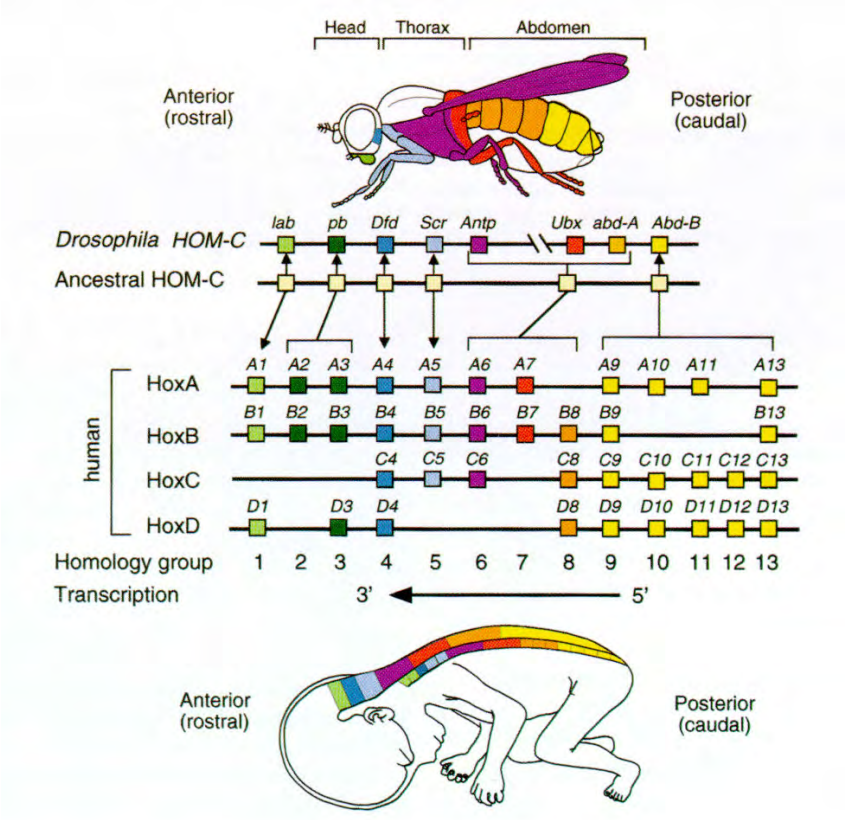
พิจารณาตัวอย่างการเปรียบเทียบดีเอ็นเอสองสายต่อไปนี้โดยพิจารณาคะแนนในภาพรวมจะมี 22 แมช 18 อินเดล และ 2 มิสแมช ซึ่งทำให้ได้คะแนนรวมเป็น 22-18-2 = 2 คะแนน (โดยสมมติว่า μ และ σ มีค่าเป็น 1 ทั้งคู่)

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

อย่างไรก็ตามคู่ของสายดีเอ็นเอนี้มีเส้นทางอื่นในกราฟแสดงการเปรียบเทียบลำดับเบส โดยการขยับลำดับเบสเน้นให้เกิดชุดของเบสที่แมชอยู่ติดกันดังตัวอย่างต่อไปนี้

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

เช่นมี 17 แมชและ 32 อินเดลและได้คะแนนรวมเท่ากับ -15 ถึงแม้ว่าบริเวณที่มีความอนุรักษ์จะให้คะแนนถึง 12-2 = 10 คะแนน ซึ่งไม่น่าจะเกิดโดยบังเอิญ เส้นทางที่สองนี้เน้นให้เกิดชุดของเบสที่แมชอยู่ติดกันซึ่งถ้าดูคะแนนรวมข้างต้นก็จะได้น้อยกว่าคะนนรวมของเส้นทางแรกมาก เพราะด้านซ้ายและขวาของเส้นทางด้านบนนี้เกิดอินเดลจำนวนมาก ในขณะที่ผลของการคำนวณคะแนนแบบภาพรวมที่มากกว่ากลับไม่สามารถสื่อความหมายในเชิง



รูปที่ 5.8 ยีนโฮมีโอบอกซ์ที่พบในมนุษย์เปรียบเทียบกับแมลง

(ที่มา: https://media.nature.com/full/nature-assets/pr/journal/v42/n4/images/pr19972506f1.jpg)

ชีววิทยาได้ในกรณีนี้ ดังนั้นในกรณีที่ความคล้ายคลึงกันจะเกิดเฉพาะบางบริเวณของสายดีเอ็นเอหรือโปรตีนนักชีววิทยาจะไม่สนใจการเปรียบเทียบในภาพรวม (global alignment) แต่จะเน้นการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอหรือโปรตีนเฉพาะบริเวณ (local alignment) ที่มีคะแนนของการเปรียบเทียบแบบภาพรวมเฉพาะบริเวณมากที่สุดตามนิยามปัญหาที่ 5.4

นิยามปัญหาที่ 5.4 การเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนเฉพาะบริเวณ

ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนเฉพาะบริเวณ (Local Alignment Problem) หาคะแนนที่มากที่สุดในการเปรียบเทียบความคล้ายคลึงกันระหว่างดีเอ็นเอและหรือโปรตีนสองสายแบบเฉพาะบริเวณ	
ข้อมูลเข้า	สายสตริงสองเส้น *v* และ *w* ซึ่งเป็นตัวแทนของสายดีเอ็นเอหรือโปรตีนและเมทริกซ์คะแนน
ผลลัพธ์	ผลการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนแบบเฉพาะบริเวณที่มีคะแนนรวมมากที่สุด

วิธีการพื้นฐานที่ใช้ในการแก้ปัญหานี้คือการหาเส้นทางที่มีคะแนนรวมมากที่สุดจากกราฟแสดงการเปรียบเทียบลำดับเบสที่มีการเชื่อมต่อคู่ของโหนดทุกคู่

หยุดคิด	เวลาที่ใช้ในการแก้ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนเฉพาะบริเวณโดยใช้วิธีการพื้นฐานข้างต้นเป็นเท่าไหร่

การนั่งแท็กซี่ฟรีกับกราฟแสดงการเปรียบเทียบลำดับเบส

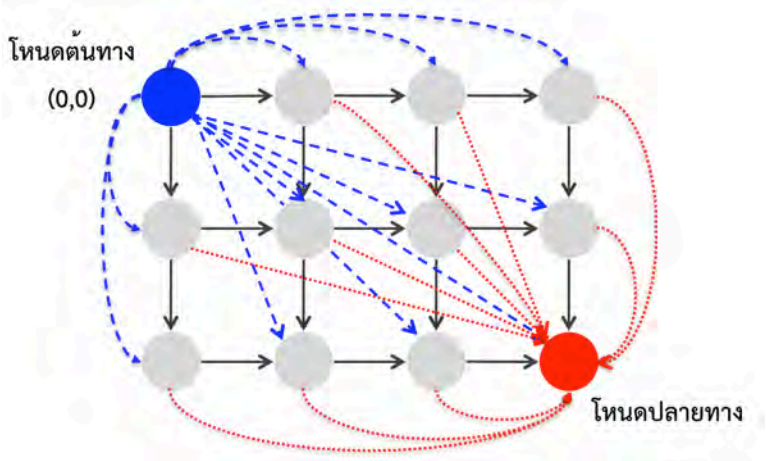
เพื่อให้สามารถหาคำตอบได้เร็วขึ้น อาจลองนึกถึงการนั่งแท็กซี่ฟรีจากโหนดเริ่มต้นที่ตำแหน่ง (0,0) ตรงไปยังโหนดที่เป็นจุดเริ่มต้นของส่วนของสายข้อมูลที่มีความอนุรักษ์ ถ้ามีโหนดนั้นอยู่และเริ่มนับจำนวนแมชที่ปรากฏไปเรื่อยๆ จนพบโหนดสุดท้ายที่เป็นส่วนของสายข้อมูลที่มีความอนุรักษ์และจากนั้นจะนั่งแท็กซี่ฟรีอีกครั้งโดยตรงไปยังโหนดปลายทาง โดยคะแนนของการเปรียบเทียบความคล้ายคลึงกันของสตริงสองสายนี้จะเท่ากับคะแนนของการเปรียบเทียบเฉพาะบริเวณที่เกิดความอนุรักษ์ระหว่างสตริงสองสาย จากตัวอย่างของการนั่งแท็กซี่ฟรีข้างต้น เทียบได้กับการเพิ่มเส้นเชื่อมจากโหนดตั้งทาง (0,0) ไปยังโหนดอื่นๆ ทั้งหมดโดยมีน้ำหนักเป็น 0 และเพิ่มเส้นเชื่อมจากโหนดใดๆที่ไม่ใช้โหนดต้นทางไปยังโหนดปลายทางโดยมีน้ำหนักเป็น 0 เช่นกัน ซึ่งจะได้กราฟแบบมีทิศทางและไม่เกิดลูปดังแสดงในรูปที่ 5.9 และเหมาะสมในการนำไปแก้ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอหรือโปรตีนเฉพาะบริเวณข้างต้น ทั้งนี้ด้วยแนวคิดของแท็กซี่ฟรีเราไม่จำเป็นต้องหาเส้นทางที่ยาวที่สุด (มีน้ำหนักรวมมากที่สุด) ระหว่างทุกคู่ของโหนดในกราฟ เนื่องจากเส้นทางที่ยาวที่สุดจากโหนดต้นทางไปยังโหนดปลายทางจะเป็นเส้นทางที่ดีที่สุดแล้ว

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

ในการคำนวณค่า $s_{i,j}$ โดยการเพิ่มเส้นเชื่อมน้ำหนัก 0 จากโหนดต้นทางไปยังทุกโหนดในกราฟ ทำให้โหนดต้นทางเป็นพรีดีเซสเซอร์ (predecessor) ของทุกโหนดดังนั้นสมการในการคำนวณคะแนนในสมการก่อนหน้าต้องปรับเพิ่มเติมดังแสดงในสมการต่อไปนี้

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

และเนื่องจากโหนดปลายทางเองก็มีเส้นเชื่อมตรงจากโหนดก่อนหน้าทุกโหนด การหาเส้นทางข้างต้นก็จะครอบคลุมความยาวรวมของสายดีเอ็นหรือโปรตีนทั้งเส้นแล้วนั่นเอง



รูปที่ 5.9 กราฟแสดงการเปรียบเทียบลำดับเบสที่มีการเพิ่มเส้นเชื่อมที่มีค่าน้ำหนักเป็น 0 (เส้นประสีน้ำเงิน) ที่เชื่อมโหนดตั้งต้นสีน้ำเงิน (0,0) ไปยังทุกโหนดที่อยู่ในกราฟและเพิ่มเส้นเชื่อมที่มีค่าน้ำหนักเป็น 0 (เส้นไข่ปลาสีแดง) ที่เชื่อมทุกโหนดใดๆ ที่ไม่ใช่โหนดตั้งต้นไปยังโหนดปลายทางสีแดง

(ที่มา: ปรับจากรูปที่ 5.21 ของ [21])

การประยุกต์ใช้การเปรียบเทียบความคล้ายคลึงกันของสายสตริงกับปัญหาอื่นๆ

Edit distance

ในปีค.ศ. 1966 Vladimir Levenshtein ได้นิยามปัญหาของ edit distance หรือระยะทางระหว่างสตริงสองสาย ว่าเป็นการหาจำนวน edit operations ที่จะต้องใช้ในการแปลงสตริงเส้นหนึ่งให้เป็นสตริงอีกเส้นหนึ่ง โดย edit operations ประกอบด้วย insertion, deletion หรือ substitution (การเปลี่ยนค่าอักขระในสายสตริง) ในตำแหน่งหนึ่งๆ ตัวอย่างเช่น สตริง **TGCATAT** สามารถแปลงได้เป็น **ATCCGAT** โดยใช้ 5 โอเปอร์เรชัน ซึ่งอนุมานได้ว่าระยะทางที่มากที่สุดระหว่างสตริงสองสายเท่ากับ 5 ดังแสดงในขั้นตอนต่อไปนี้

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

หยุดคิด	เราสามารถเปลี่ยนสตริง TGCATAT ให้เป็น ATCCGAT โดยใช้จำนวนโอเปอร์เรชันน้อยกว่า 5 ได้ไหม

ในความเป็นจริงแล้วระยะทางระหว่างสตริงสองสายนี้มีค่าเท่ากับ 4 ดังแสดงต่อไปนี้

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

Levenshtein ได้นิยามปัญหาการหาระยะทางระหว่างสตริงสองสายไว้แต่ไม่ได้อธิบายอัลกอริทึมในการหา

ฝึกหัด	จงเขียนอัลกอริทึมที่ใช้ในการหาระยะทางระหว่างสตริงสองสาย

Fitting alignment

สมมติว่าเรามีสายโปรตีนยาว 20,000 กรดอะมิโน (v) ในเชื้อ *Bacillus brevis* และต้องการหาส่วนของโปรตีนที่มีความคล้ายคลึงกับโปรตีนโดเมน A (A-domain) ที่มีความยาว 600 กรดอะมิโน (w) ในเชื้ออื่นๆ การเปรียบเทียบความคล้ายคลึงกันของสายสตริงแบบภาพรวม (global alignment) จะไม่สามารถให้ผลลัพธ์ตามที่คาดหวังเพราะวิธีการนี้จะพยายามเทียบ 600 กรดอะมิโนกับทั้ง 20,000 กรดอะมิโนและหาคะแนนรวมที่มากที่สุด แต่จะไม่มีความหมายทางชีววิทยาที่คาดหวังข้างต้น ในขณะที่การเปรียบเทียบความคล้ายคลึงกันเฉพาะบริเวณก็จะพยายามหาส่วนของสตริงของทั้ง v และ w ที่มีความอนุรักษ์ร่วมกันที่ให้คะแนนมากที่สุด ดังนั้นจึงจำเป็นต้องมีอัลกอริทึมจำเพาะที่จะทำการเทียบส่วนของสตริง v' ใดๆของ v ซึ่งทำให้ได้คะแนนรวมของการเปรียบเทียบความเหมือนแบบภาพรวมระหว่าง 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 ในขณะที่คะแนนของการเปรียบเทียบความเหมือนแบบภาพรวม (global alignment) เท่ากับ 6-9 = -3 ในขณะที่คะแนนของ Fitting alignment มีค่าเท่ากับ 5-1-2 = 2

Overlap alignment

ในบทที่ 2 เรายกตัวอย่างการประกอบร่างจีโนมโดยใช้กราฟแสดงความคาบเกี่ยว (overlap graph) ซึ่งความซับซ้อนของความคาบเกี่ยวนี้มีเพิ่มมากขึ้นเมื่อรีดที่อ่านมามีความผิดพลาด การเปรียบเทียบความคล้ายคลึงกันระหว่างซัฟฟิกซ์ของรีดที่ 1 กับพรีฟิกซ์ของรีดที่ 2 แสดงโดยตัวอย่างต่อไปนี้

ATGCATGCCGG
T-CC-GAAAC

การเปรียบเทียบความคล้ายคลึงกันในส่วนของสายข้อมูลที่คาบเกี่ยวกัน (overlap alignment) ของสตริง $v = v_1 \ldots v_n$ และ $w = w_1 \ldots w_m$ เป็นการเปรียบเทียบความคล้ายคลึงของสตริงแบบภาพรวมเฉพาะบริเวณที่คาบเกี่ยวกัน

การให้คะแนนลงโทษในกรณีที่เกิด Insertion หรือ Deletion

Affine gap penalties

การให้คะแนนลงโทษของการเกิด indels โดยใช้ σ ข้างต้น ถึงแม้ว่าจะทำให้การให้คะแนนความคล้ายคลึงกันมีความหมายทางชีววิทยามากขึ้น แต่ก็ยังมีรายละเอียดของการเพิ่มหรือลดเบสที่ต้องพิจารณาเพิ่มเติม ตัวอย่างเช่น ในการคำนวณคะแนนลงโทษของอินเดล ณ จุดนี้แต่ละตำแหน่งที่เกิดอินเดลถือว่าเป็นอิสระต่อกันซึ่งหมายความว่าถ้าในการเปรียบเทียบสายดีเอ็นเอสองเส้นเกิด k ตำแหน่ง คะแนนลงโทษส่วนอินเดลนี้จะเท่ากับ $\sigma \cdot k$ อย่างไรก็ตาม หลายๆครั้งอินเดลที่เกิดขึ้นในสายดีเอ็นเอนั้นเกิดจากการเพิ่มหรือลดชุดของเบส ดังนั้นคะแนนลงโทษ $\sigma \cdot k$ ข้างต้นก็จะเป็นการลงโทษมากเกินไป ในตัวอย่างการเปรียบเทียบสายดีเอ็นเอต่อไปนี้ พบว่าทั้งด้านซ้ายและขวาได้คะแนนเท่ากัน อย่างไรก็ตามในเชิงปฏิบัติจะพบว่าผลการเทียบในด้านขวานั้นมีความเหมาะสมกว่าในทางชีววิทยา

GATCCAG GATCCAG
GA-C-AG GA--CAG

จากตัวอย่างข้างต้น จึงได้มีการเพิ่มตัวแปรอีกหนึ่งตัวเรียกว่าแกป (gap) ซึ่งเป็นจำนวนอินเดลที่เกิดขึ้นติดต่อกันในผลการเปรียบเทียบคู่ของสายดีเอ็นเอหรือโปรตีน โดยได้มีการนำเสนอการกำหนดคะแนนลงโทษในกรณีที่เกิดแกปความยาว k เบสโดยใช้สมการ เช่น $\sigma + \epsilon \cdot (k-1)$ โดยที่ σ แสดงค่า gap opening penalty หรือค่าเริ่มต้นการเกิดแกป โดยคำนวณจากเบสแรกของแกป ในขณะที่ ϵ หรือ gap extension penalty แสดงค่าสัมประสิทธิ์ของจำนวนอินเดลที่เหลือในแกป โดยทั่วไป ϵ จะมีค่าน้อยกว่า σ เพื่อแสดง affine penalty ที่การลงโทษในกรณีที่เกิดอินเดลต่อเนื่องจะมีโทษน้อยกว่าการเกิดอินเดลเดี่ยวๆ ตัวอย่างเช่น ถ้า $\sigma = 5$ และ $\epsilon = 1$ จะคำนวณค่า

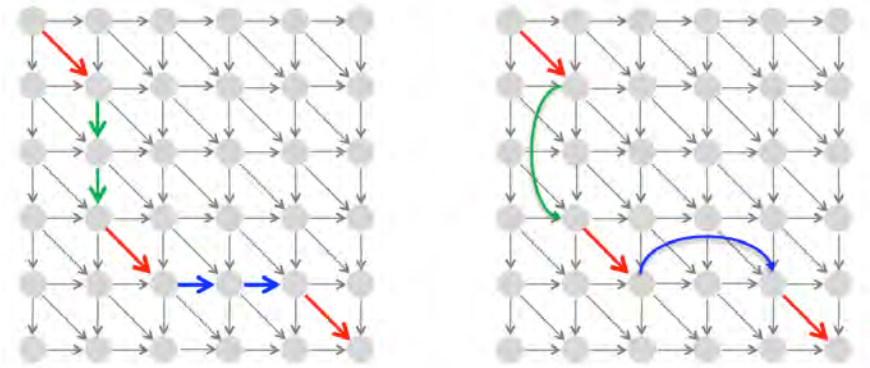
คะแนนลงโทษของการเปรียบเทียบคู่ของสายดีเอ็นเอข้างต้นทางซ้ายได้เท่ากับ $2\sigma = 10$ ในขณะที่ด้านขวาจะเท่ากับ $\sigma + \epsilon = 6$

นิยามปัญหาที่ 5.5 ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนโดยใช้ Affine Gap Penalties

ปัญหาการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนโดยใช้ Affine Gap Penalties สร้างผลของการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนแบบภาพรวมที่มีคะแนนมากสุดและใช้ Affine Gap Penalty	
ข้อมูลเข้า	สายสตริงสองเส้นซึ่งเป็นตัวแทนของสายดีเอ็นเอหรือโปรตีน เมทริกซ์คะแนน σ และ ϵ
ผลลัพธ์	ผลของการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีนแบบภาพรวมที่คำนวณจากเมทริกซ์คะแนน ค่าเริ่มต้นของการเกิดแกป σ และ ค่าสัมประสิทธิ์ของจำนวนอินเดลที่อยู่ในแกปโดยไม่รวมอินเดลแรกสุด ϵ

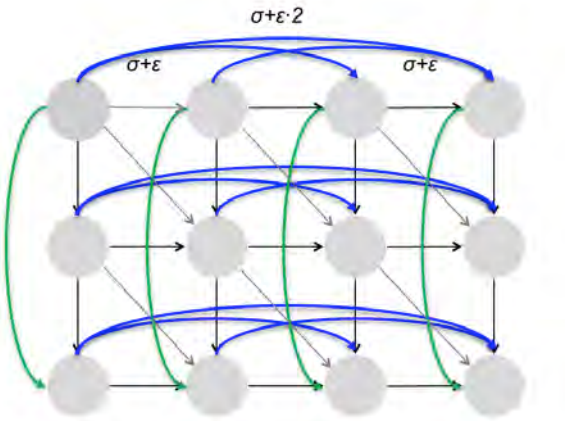
หยุดคิด	เราต้องปรับแต่งกราฟแสดงการเปรียบเทียบลำดับเบส (alignment graph) อย่างไรให้สามารถแสดงแกปในกราฟ

รูปที่ 5.10 (ขวา) แสดงวิธีการปรับแต่งกราฟแสดงการเปรียบเทียบลำดับเบสที่มีการนำแกปเข้ามาร่วมแสดงผลโดยมีการเพิ่มเส้นเชื่อมที่มีความยาวตามขนาดของแต่ละแกป และเนื่องจากเราไม่สามารถทราบล่วงหน้าว่าจะมีแกปเกิดขึ้นในบริเวณไหนบ้างในผลของการเปรียบเทียบ จึงจำเป็นต้องเพิ่มเส้นเชื่อมที่แสดงทุกแกปที่เป็นไปได้ ซึ่งหมายถึงเราจะเพิ่มเส้นเชื่อมระหว่างโหนด (i,j) ไปยังโหนด $(i+k,j)$ และ $(i,j+k)$ โดยเส้นเชื่อมเหล่านี้มีน้ำหนักเท่ากับ $\sigma + \epsilon \cdot (k-1)$ สำหรับทุกค่า k ที่เป็นไปได้ ดังแสดงในรูปที่ 5.11 สำหรับสายดีเอ็นเอสองเส้นที่แต่ละเส้นยาวมีความยาวเท่ากับ n จำนวนของเส้นเชื่อมที่มีการพิจารณาเรื่อง Affine Gap Penalty จะเพิ่มจาก $O(n^2)$ เป็น $O(n^3)$



รูปที่ 5.10 (ซ้าย) กราฟแสดงการเปรียบเทียบลำดับเบส (alignment graph) ปกติ (ขวา) การปรับแต่งกราฟแสดงการเปรียบเทียบลำดับเบสโดยนำแกปเข้ามาแสดงเป็นส่วนหนึ่งของกราฟ

(ที่มา: ปรับจากรูปที่ 5.22 ของ [21])



รูปที่ 5.11 จำนวนของเส้นเชื่อมที่เพิ่มขึ้นเมื่อมีการพิจารณาเรื่อง Affine Gap Penalty

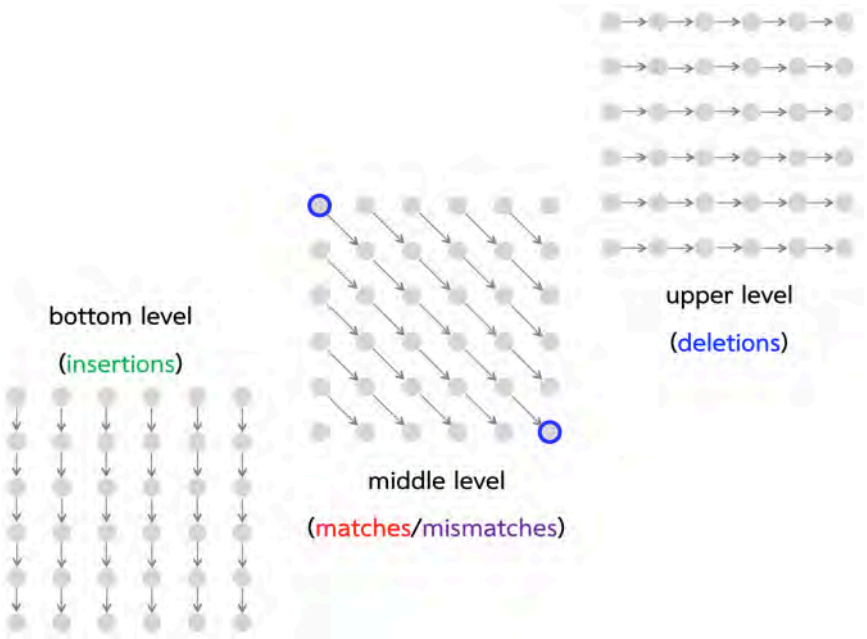
(ที่มา: ปรับจากรูปที่ 5.23 ของ [21])

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

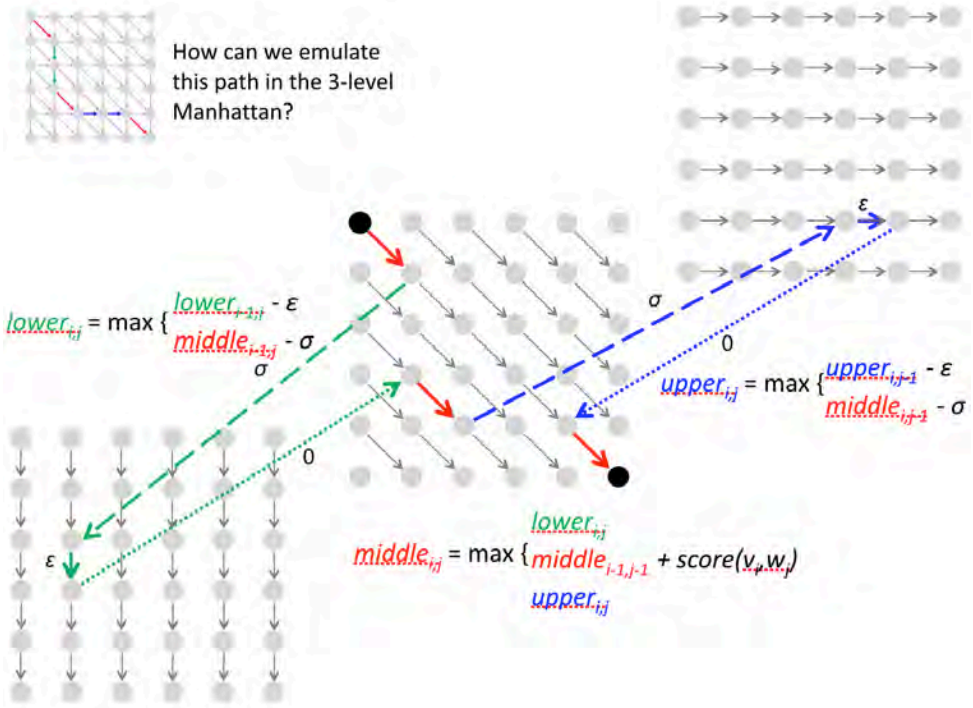
สร้างแผนที่สามระดับของเมืองแมนฮัตตัน

ในการสร้าง DAG ที่มีเส้นเชื่อม $O(n^2)$ โดยสามารถใช้แก้ปัญหาที่ 5.5 ได้นั้นสามารถทำได้โดยการเพิ่มจำนวนของโหนด ผ่านการสร้างกราฟแสดงการเปรียบเทียบลำดับเบส 3 ระดับ (รูปที่ 5.12) โดยแต่ละโหนด (i,j) ในกราฟเดิมจะถูกจำลองออกมาเป็น 3 ระดับคือ $(i,j)_{lower}$, $(i,j)_{middle}$, และ $(i,j)_{upper}$ โดยระดับกลาง (middle) จะเก็บเฉพาะเส้นเชื่อมในแนวเส้นทแยงมุมซึ่งแสดงสถานะแมชหรือมิสแมช โดยมีน้ำหนักของเส้นเชื่อมเป็น Score(v_i, w_j) กราฟระดับล่าง (lower) จะเก็บเฉพาะเส้นเชื่อมที่ชี้ลงในแนวดิ่งซึ่งแสดงสถานะ gap extension ใน v โดยมีน้ำหนักของเส้นเชื่อมเป็น $-\epsilon$ และกราฟระดับบนที่จะเก็บเฉพาะเส้นเชื่อมในแนวนอนซึ่งแสดงสถานะ gap extension ใน w โดยมีน้ำหนักของเส้นเชื่อมเป็น $-\epsilon$

เพื่อให้สามารถนำเรื่องของแกปมาร่วมพิจารณา แต่ละโหนด $(i,j)_{middle}$ จะมีเส้นเชื่อมไปยังโหนด $(i+1,j)_{lower}$ และ $(i,j+1)_{upper}$ ซึ่งเส้นเชื่อมทั้งสองนี้จะมีค่าน้ำหนักเท่ากับ $-\sigma$ สำหรับอินเดลแรกและ $-\epsilon$ สำหรับอินเดลถัดๆไป แกป และ 0 สำหรับตำแหน่งที่ปิดแกปซึ่งทำให้ได้คะแนนการลงโทษเป็น $\sigma + \epsilon \cdot (k-1)$ ตามที่ต้องการ รูปที่ 5.13 แสดงเส้นทางในการเดินของรูปที่ 5.10 (รูปขวา) โดยใช้กราฟแสดงการเปรียบเทียบลำดับเบส 3 ระดับ รูป DAG ที่แสดงในรูปที่ 5.13 นี้ถึงแม้จะดูซับซ้อนแต่จำนวนเส้นเชื่อมที่ใช้จะเท่ากับ $O(nm)$ สำหรับคู่ของสายดีเอ็นเอหรือโปรตีนที่มีความยาว n และ m ตามลำดับและเส้นทางที่ยาวที่สุดก็ยังเป็นไปวิธีการเปรียบเทียบความคล้ายคลึงกันโดยใช้ Affine Gap Penalty ทั้งนี้กราฟแสดงการเปรียบเทียบลำดับเบส 3 ระดับสามารถแปลงให้เป็นชุดของความสัมพันธ์เวียนเกิด (recurrence relations) ตามที่แสดงในรูปที่ 5.13



รูปที่ 5.12 กราฟแสดงการเปรียบเทียบลำดับเบส 3 ระดับเพื่อลดจำนวนเส้นเชื่อมที่ต้องใช้ในการแก้ปัญหาที่ 5.5 (ที่มา: รูปที่ 5.24 ของ [21])



รูปที่ 5.13 กราฟแสดงการเปรียบเทียบลำดับเบส 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 (ที่มา: รูปที่ 5.24 ของ [21])

บทส่งท้าย

วิธีการในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนข้างต้นเป็นการเปรียบเทียบระหว่างสายข้อมูลสองเส้นหรือที่เรียกว่า pairwise alignment ซึ่งสามารถขยายหรือนำไปประยุกต์ใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายของดีเอ็นหรือโปรตีนในชุดของข้อมูลซึ่งเรียกว่า multiple sequence alignment

การเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอหรือโปรตีนหลายเส้น

การเปรียบเทียบความคล้ายคลึงกันของสายสตริงจำนวน t เส้นคือ v^1, ..., v^t เรียกว่าการทำ Multiple sequence alignment หรือ t-way alignment ซึ่งสามารถแสดงโดยเมทริกซ์จำนวน t แถวโดยแถวที่ i จะมีลำดับอักขระของสายสตริงเส้นที่ i และอาจมีการแทรกช่องว่างในบางตำแหน่ง โดยมีสมมติฐานว่าไม่มีคอลัมน์ใดเลยที่มีแต่ช่องว่าง ในตัวอย่างของ 3-way alignment ต่อไปนี้ อักขระที่พบมากสุดในแต่ละตำแหน่งจะถูกแสดงด้วยตัวอักษรใหญ่

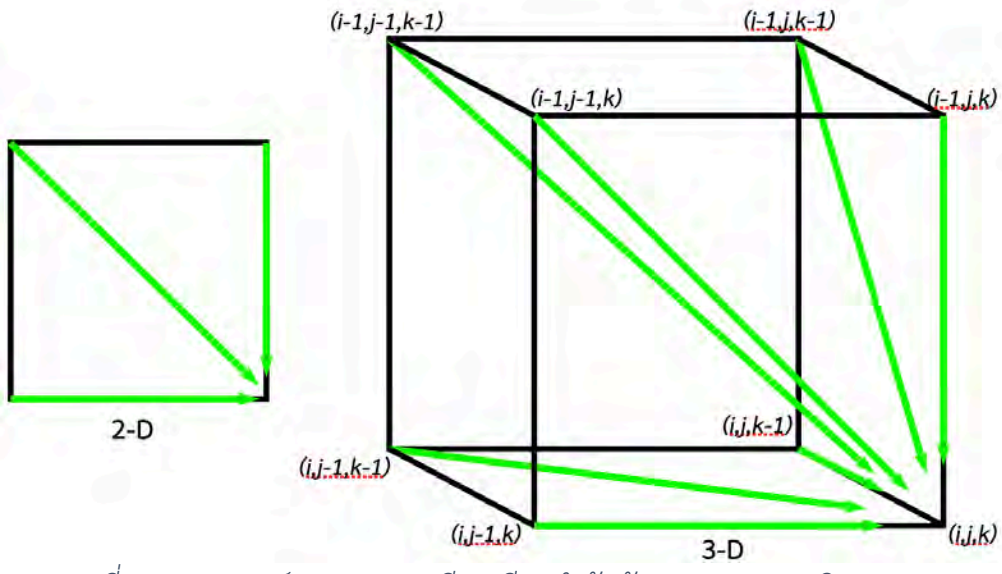
```
A  T  -  G  T  T  a  T  A
A  g  C  G  a  T  C  -  A
A  T  C  G  T  -  C  T  c
0  1  2  2  3  4  5  6  7  8
0  1  2  3  4  5  6  7  7  8
0  1  2  3  4  5  5  6  7  8
```

โดยเมทริกซ์นี้เป็นเมทริกซ์แบบทั่วไปของเมทริกซ์ที่แสดงผล pairwise alignment ส่วนอะเรย์ของค่าที่ตามมาอีก 3 บรรทัดแสดงจำนวนของอักขระที่ถูกใช้ไปแล้ว ณ คอลัมน์นั้นๆ ของสตริงสายที่ 1, 2 และ 3 ตามลำดับ โดยอะเรย์ของค่าเหล่านี้สอดคล้องกับเส้นทางในกริดสามมิติต่อไปนี้

(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 กราฟที่แสดงการเปรียบเทียบของดีเอ็นเอหรือโปรตีนสามสายสามารถแสดงโดยกริดในกล่องสามมิติหรือที่เรียกว่าลูกบาศก์หรือคิวบ์ (cube) ดังแสดงในรูปที่ 5.14 การคำนวณคะแนนของเส้นทางหนึ่งๆในลูกบาศก์สามารถขยายจากการคำนวณคะแนนในการเปรียบ เทียบระหว่างดีเอ็นเอหรือโปรตีนสองสายโดยการใช้ไดนามิกโปรแกรมมิงดังชุดของสมการต่อไปนี้

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



รูปที่ 5.14 ลูกบาศก์แสดงกราฟเปรียบเทียบลำดับอักขระของสายสตริงสามสาย

(ที่มา: ปรับจากรูปที่ 5.31 ของ [21])

โดยในกรณีที่มีสายข้อมูลจำนวน t เส้นและแต่ละเส้นยาว n อักขระ กราฟแสดงการเปรียบเทียบลำดับเบสจะประกอบด้วย n^t โหนด และแต่ละโหนดจะมีเส้นเชื่อมเข้ามากสุดจำนวน 2^t -1 เส้น ซึ่งหมายถึงต้องใช้เวลาในรันเท่ากับ $O(n^t 2^t)$ ถ้า t มีจำนวนมากอัลกอริทึมที่ใช้ไดนามิกโปรแกรมมิงข้างต้นจะไม่มีประสิทธิภาพดีพอในทางปฏิบัติ ซึ่งได้มีการนำเสนออัลกอริทึมต่างๆที่ใช้ฮิวริสติกในการหาคำตอบในลักษณะที่ใกล้เคียงแต่อาจจะไม่ใช่คำตอบที่ดีที่สุดโดยเน้นการลดเวลาในรันอัลกอริทึมเพื่อหาคำตอบ

การเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอหรือโปรตีนหลายเส้นแบบโลภ

จากตัวอย่างของการเปรียบเทียบความคล้ายคลึงกันระหว่างดีเอ็นเอสามสายต่อไปนี้

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

สามารถนำไปสู่การทำการเปรียบเทียบความคล้ายคลึงกันระหว่างคู่ของดีเอ็นเอสองสาย (pairwise alignment) จำนวนสามคู่

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

คำถามคือเราสามารถรวมผลการเปรียบเทียบความคล้ายคลึงกันระหว่างคู่ของสายดีเอ็นเอไปเป็นผลของการเปรียบเทียบชุดของสายดีเอ็นข้างต้นได้หรือไม่

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

จากตัวอย่างของการเปรียบเทียบชุดของสายดีเอ็นเอ **CCCCTTTT**, **TTTTGGGG** และ **GGGGCCCC** นี้แสดงให้เห็นว่าเรา *ไม่* สามารถรวมชุดผลของการเปรียบเทียบคู่ของสายดีเอ็นเอให้เป็นผลของการเปรียบเทียบชุดของสายดีเอ็นเอได้เสมอไป ดังในตัวอย่าง “หยุดคิด” ข้างต้น ผลของการเปรียบเทียบคู่ของสายดีเอ็นเอทางซ้ายมืออนุมานได้ว่า ลำดับเบส **CCCC** จะมาก่อน **TTTT** ในผลของเปรียบเทียบชุดของสายดีเอ็นเอ ในขณะที่ผลของการเปรียบเทียบคู่ของสายดีเอ็นเอทางขวามืออนุมานได้ว่า ลำดับเบส **TTTT** จะมาก่อน **GGGG** ในผลของเปรียบเทียบชุดของสายดีเอ็นเอ ในขณะที่ผลของการเปรียบเทียบคู่ของสายดีเอ็นเอตรงกลางอนุมานได้ว่า ลำดับเบส **GGGG** จะมาก่อน **CCCC** ดังนั้น **CCCC** จะต้องมาก่อน **TTTT** และ **TTTT** จะต้องมาก่อน **GGGG** ซึ่ง **GGGG** จะต้องมาก่อน **CCCC** ซึ่งเกิดความขัดแย้งกันเอง

เพื่อเป็นการหลีกเลี่ยงการเกิดความขัดแย้งกันเองนี้ อัลกอริทึมแบบโลภบางอัลกอริทึมจะพยายามสร้างผลการเปรียบเทียบชุดของสายดีเอ็นเอจากชุดของผลการเปรียบเทียบระหว่างคู่ของสายดีเอ็นเอ โดยจะอาศัยฮิวริสติกเลือกคู่ของสายดีเอ็นเอที่มีความคล้ายคลึงกันที่สุดออกมาและใช้ผลการเปรียบเทียบของดีเอ็นเอคู่นี้เป็นจุดตั้งต้น โดยในแต่ละรอบหลังจากนั้นจะทำการเลือกดีเอ็นเอมา 1 สายจากชุดของดีเอ็นเอที่ยังเหลืออยู่ที่มีความคล้ายคลึงกับผลการเปรียบเทียบคู่ของสายดีเอ็นเอตั้งต้นที่สุด มาสร้างเป็นผลการเปรียบเทียบสายดีเอ็นเอ 3 สาย ทำการเลือกดีเอ็นเอ 1 สายถัดไปและทำซ้ำกระบวนการข้างต้นจนกว่าจะไม่มีสายดีเอ็นเอเหลือในชุด คำถามคือการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอกับผลการเปรียบเทียบชุดของสายดีเอ็นเอที่กำลังสร้างอยู่นั้นต้องทำอย่างไร

ทั้งนี้ผลการเปรียบเทียบลำดับเบสนิวคลีโอไทด์จำนวน k คอลัมน์สามารถแสดงโดยโปรไฟล์เมทริกซ์ต่อไปนี้ (4 x k ในกรณีของดีเอ็นเอ และ 20 x k ในกรณีของโปรตีน) โดยอัลกอริทึมแบบโลภจะทำการเพิ่มดีเอ็นเอเข้าไป 1 เส้นที่ใกล้เคียงกับโปรไฟล์ปัจจุบันมากที่สุด สร้างโปรไฟล์ใหม่โดยรวมเส้นที่เพิ่มเข้ามาและทำซ้ำจนหมดชุดของสายดีเอ็นเอ ดังนั้นปัญหาการเปรียบเทียบชุดของสายดีเอ็นเอหรือโปรตีนก็จะกลายเป็นปัญหาของการเปรียบเทียบคู่ของสายดีเอ็นเอหรือโปรตีนจำนวน t-1 ครั้ง

วิธีการเปรียบเทียบความคล้ายคลึงกันของชุดของสายดีเอ็นเอหรือโปรตีนแบบโลภข้างต้นจะทำงานได้ดีในกรณีที่สายดีเอ็นเอหรือโปรตีนมีความคล้ายคลึงกันมาก อย่างไรก็ตามถ้าสายดีเอ็นเอหรือโปรตีนที่นำมาเปรียบเทียบมีความแตกต่างกันมากประสิทธิภาพของอัลกอริทึมจะลดลงอย่างมาก โดยเฉพาะในกรณีสายดีเอ็นเอหรือ

```
                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
```

โปรตีนที่ถูกเลือกเป็นคู่ตั้งต้นไม่ใช่ตัวแทนของคำตอบที่ดีที่สุด และจะมีผลต่อเนื่องไปยังการเพิ่มความผิดพลาดในการเลือกสายดีเอ็นมาสร้างเป็นโปรไฟล์เมทริกซ์ในรอบถัดๆไป วิธีการอื่นๆที่มีการนำเสนอ เช่น การทำ progressive alignment ซึ่งเป็นวิธีการที่ใช้ในโปรแกรม CLUSTAL [118] โดยขั้นตอนหลักประกอบด้วยการเปรียบเทียบความคล้ายคลึงกันระหว่างทุกคู่ของสายข้อมูลโดยใช้อัลกอริทึม Needleman-Wunsch [119] ซึ่งทำการเปรียบเทียบคู่ของสายข้อมูลแบบภาพรวม (global alignment) และสร้างผลการเปรียบเทียบเป็นเมทริกซ์แสดงระยะห่างระหว่างทุกคู่ของสายข้อมูล ข้อมูลในเมทริกซ์นี้จะถูกนำไปใช้ในการสร้างต้นไม้แสดงความสัมพันธ์ระหว่างสายข้อมูล (guide tree) จากต้นไม้นี้สายข้อมูลสองเส้นที่มีความคล้ายคลึงกันมากที่สุดจะถูกเลือกมาเปรียบเทียบกันอีกครั้งโดยใช้อัลกอริทึม Needleman-Wunsch แบบข้างต้น และผลของการเปรียบเทียบจะถูกแปลงให้เป็นสายสตริงเสียงข้างมาก (consensus string) ซึ่งถูกนำไปใช้เปรียบเทียบกับสายข้อมูลอื่นๆ ต่อ เสมือนกับเป็นสายข้อมูลหนึ่งเส้น โดยสายข้อมูลเส้นถัดไปที่มีความคล้ายคลึงมากสุดจะถูกนำเทียบกับสายสตริงเสียงข้างมากข้างต้น และผลของการเปรียบเทียบจะถูกนำมาสร้างเป็นสายสตริงเสียงข้างมากเส้นใหม่และทำซ้ำจนหมดชุดของสายข้อมูล

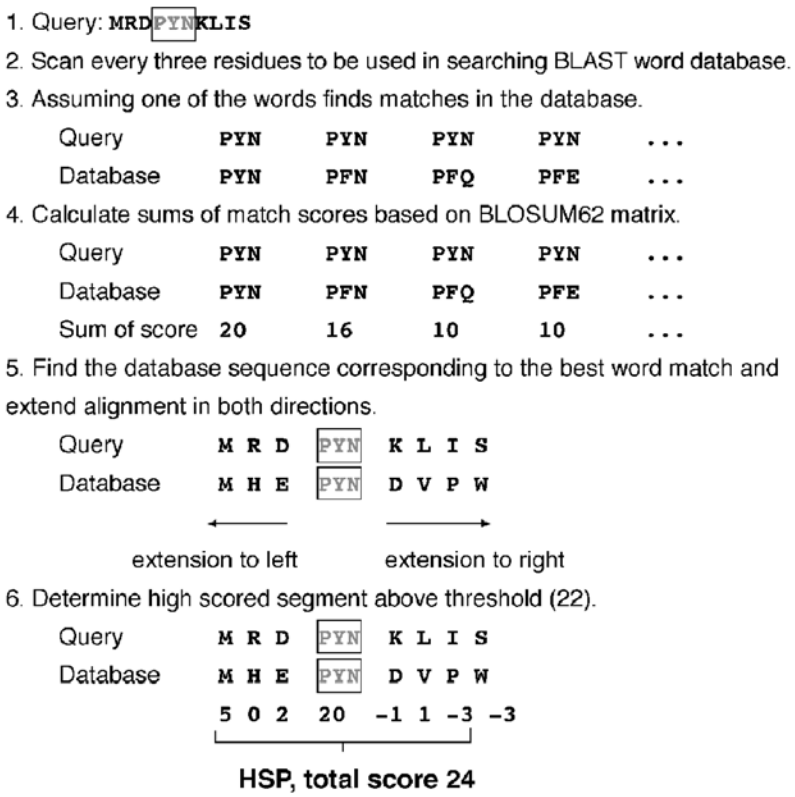
การเปรียบเทียบสายดีเอ็นเอหรือโปรตีนกับฐานข้อมูลขนาดใหญ่

โปรแกรม BLAST (**B**asic **L**ocal **A**lignment **S**earch Tool) [1] ถูกพัฒนาโดย Steven Altschul และคณะ ที่ NCBI ในปี ค.ศ. 1990 และกลายมาเป็นโปรแกรมที่ถูกใช้งานกันอย่างแพร่หลายและต่อเนื่องมาจนปัจจุบัน (ถ้าวัดความแพร่หลายของโปรแกรม BLAST โดยดูจากจำนวนผลงานตีพิมพ์ที่มีการอ้างอิงถึงจะพบว่าโปรแกรม BLAST ปี ค.ศ.

1990 ถูกอ้างอิงถึง 70,106 ครั้ง จากผลการสืบค้นกูเกิลเมื่อวันที่ 10 ก.พ. ค.ศ. 2018) โดยโปรแกรม BLAST ทำหน้าที่ในการรับสายดีเอ็นเอหรือโปรตีนเป็นข้อมูลเข้าและใช้ฮิวริสติกในการเทียบสายข้อมูลเข้านั้นกับสายข้อมูลทั้งหมดที่อยู่ในฐานข้อมูล โดยมีวัตถุประสงค์หลักเพื่อหาส่วนของสายข้อมูลในฐานข้อมูลที่มีความอนุรักษ์ร่วมกับสายข้อมูลเข้า ส่วนของสายข้อมูลที่พบความอนุรักษ์ในฐานข้อมูลนั้นบ่งชี้ความคล้ายคลึงกันมากกว่าความคล้ายคลึงกันโดยบังเอิญ รูปที่ 5.15 การทำงานของโปรแกรม BLAST โดยขั้นตอนหลักๆประกอบด้วยสร้างรายการของคำจากสายข้อมูลเข้า ในตัวอย่างนี้แต่ละคำประกอบด้วยลำดับกรดอะมิโน 3 ตัว (ในกรณีที่เป็นสายดีเอ็นเอ แต่ละคำจะประกอบด้วยลำดับเบสนิวคลีโอไทด์ 11 ตัว) ชุดของคำนี้เรียกว่า seeding ขั้นตอนที่สองตรวจสอบในฐานข้อมูลว่ามีสายโปรตีนใดบ้างที่ประกอบด้วยคำที่อยู่ในสายข้อมูลเข้า การตรวจสอบความตรงกันของคำนี้จะให้คะแนนโดยใช้เมทริกซ์คะแนน BLOSUM62 (สามารถเปลี่ยนได้) ในการคำนวณคะแนนการแทนที่ของกรดอะมิโนหนึ่งๆด้วยกรดอะมิโนอื่น การตัดสินว่าแต่ละคำที่นำมาเทียบนั้นตรงกับคำในฐานข้อมูลหรือไม่ ขึ้นอยู่กับเกณฑ์คะแนนที่กำหนดไว้ ขั้นถัดไปจะทำการเปรียบเทียบสายข้อมูลเข้ากับสายโปรตีนหนึ่งๆที่มีคำที่ตรงกันโดยการทำ pairwise alignment การเปรียบเทียบนี้จะเริ่มจากส่วนของคำที่เหมือนกันและทำการขยายการเปรียบเทียบโดยใช้ลำดับกรดอะมิโนทั้งซ้ายและขวาของคำตั้งต้น โดยการเปรียบเทียบนี้จะทำการขยายจำนวนกรดอะมิโนทั้งซ้ายและขวาไปเรื่อยๆจนกว่าคะแนนที่ได้จะลดลงจนต่ำกว่าค่าเกณฑ์ที่กำหนดเนื่องจากการเกิดมิสแมช (ค่าเกณฑ์คะแนนในการหยุดการเปรียบเทียบนี้เท่ากับ 22 สำหรับโปรตีนและ 20 สำหรับดีเอ็นเอ) และส่วนของสายโปรตีนที่ตรงกันและไม่มีแกปนี้เรียกว่า High-scoring segment pair (HSP) โดยโปรแกรม BLAST เวอร์ชันแรกนั้นจะรายงาน HSP ที่มีคะแนนสูงที่สุด ในเวอร์ชันถัดๆมาได้มีการอนุญาตให้มีแกปในการเปรียบเทียบได้เรียกว่า gapped BLAST ซึ่งในกรณีนี้ HSP ที่มีคะแนนสูงสุดจะถูกเลือกมาเปรียบเทียบกับสายข้อมูลเข้าโดยใช้ไดนามิกโปรแกรมมิงและอนุญาตให้มีแกปได้ โดยมีการขยายจำนวนกรดอะมิโนที่จะนำมาเปรียบเทียบทั้งซ้ายและขวาเป็นเรื่อยๆจนกว่าค่าคะแนนจะลดลงเหมือนในเวอร์ชันก่อนหน้า อย่างไรก็ตามโปรแกรม BLAST อนุญาตให้คะแนนลดลงได้ชั่วคราวถ้าหลังจากนั้นยังสามารถเพิ่มคะแนนให้กลับมาสูงกว่าเกณฑ์ได้

ชุดของโปรแกรม BLAST

โปรแกรม BLAST ได้ถูกออกแบบและพัฒนามาเป็นชุด โดยโปรแกรมภายชุดประกอบด้วย BLASTN, BLASTP, BLASTX, TBLASTN, และ TBLASTX โดยความแตกต่างหลักของแต่ละโปรแกรมภายในชุดคือประเภทของสายข้อมูลเข้า โปรแกรม BLASTN จะทำการเทียบสายข้อมูลเข้าที่เป็นลำดับเบสนิวคลีโอไทด์กับฐานข้อมูลลำดับเบสนิวคลีโอไทด์ (เช่น ฐานข้อมูลนิวคลีโอไทด์ที่ NCBI: nt:- 254,826,802 เส้น เข้าถึงเมื่อวันที่ 10 ก.พ. พ.ศ. 2561) โปรแกรม BLASTP จะทำการเทียบสายข้อมูลเข้าที่เป็นลำดับกรดอะมิโนกับฐานข้อมูลโปรตีน (เช่น Non-redundant protein database ที่ NCBI: NR:- 482,053,249 เส้น เข้าถึงเมื่อวันที่ 10 ก.พ. พ.ศ. 2561) โปรแกรม BLASTX จะทำการเทียบสายข้อมูลเข้าที่เป็นลำดับเบสนิวคลีโอไทด์กับฐานข้อมูลโปรตีนโดยก่อนนำไปเทียบจะทำการแปลงลำดับเบสนิวคลีโอไทด์ให้เป็นลำดับกรดอะมิโนทั้ง 6 เฟรม ในขณะที่โปรแกรม TBLASTN จะทำการ



รูปที่ 5.15 แสดงขั้นตอนหลักในการทำงานของโปรแกรม BLAST โดยคะแนนที่ใช้ในตัวอย่างเป็นเมทริกซ์คะแนน BLOSUM62 โดยตัวอย่างของคำที่ตรงกับส่วนของสายข้อมูลเข้าจะถูกแสดงไว้ในกล่อง

(ที่มา: รูปที่ 4.1 ของ [120])

เทียบสายข้อมูลเข้าที่เป็นลำดับกรดอะมิโนกับฐานข้อมูลนิวคลีโอไทด์โดยก่อนนำไปเทียบจะทำการแปลงลำดับกรดอะมิโนเป็น 6 เฟรมในระดับนิวคลีโอไทด์ก่อน โปรแกรม TBLASTX จะทำการเทียบสายข้อมูลเข้าที่เป็นลำดับเบสนิวคลีโอไทด์กับฐานข้อมูลนิวคลีโอไทด์โดยก่อนเทียบจะทำการแปลงลำดับเบสนิวคลีโอไทด์ทั้ง 6 เฟรมเป็นลำดับกรดอะมิโนเพื่อเทียบกับนิวคลีโอไทด์ในฐานข้อมูลที่ถูกแปลงทั้ง 6 เฟรมเป็นกรดอะมิโนก่อนเทียบเช่นกัน

ตัวอย่างโปรแกรมที่มีการใช้งานกันอย่างแพร่หลาย

นอกจากโปรแกรม BLAST เป็นโปรแกรมหลักที่ใช้ในการเทียบสายดีเอ็นเอหรือโปรตีนกับฐานข้อมูลของสายข้อมูลขนาดใหญ่ (database search) เช่นฐานข้อมูลโปรตีนที่ NCBI (https://www.ncbi.nlm.nih.gov) หรือฐานข้อมูลโปรตีนยูนิพรอต UniProt (http://www.uniprot.org) แล้ว ก็ยังมีชุดของโปรแกรมที่ใช้การเปรียบเทียบความคล้ายคลึงกันระหว่างคู่ของสายข้อมูล (pair-wise alignment) เช่น โปรแกรม NEEDLE (EMBOSS) ที่ใช้อัลกอริทึม Needleman-Wunsch ซึ่งเป็นการทำงานแบบ global alignment และโปรแกรม Water ที่ใช้อัลกอริทึม Smith-Waterman [121] และโปรแกรม LALIGN ที่ EMBL-EBI (European Molecular Biology Laboratory-

European Bioinformatics Institute) (https://www.ebi.ac.uk/Tools/psa/) ซึ่งเป็นการทำงานแบบ local alignment สำหรับตัวอย่างโปรแกรมที่ใช้ในการทำ multiple sequence alignment เช่น CLUSTAL W [122], T-Coffee [123], MAFFT [124], MUSCLE [125], Clustal Omega [126, 127] ซึ่งหลายๆเครื่องมือในกลุ่มนี้ก็ได้มีเวอร์ชันที่เป็นเว็บที่ EMBL-EBI (https://www.ebi.ac.uk/Tools/msa/) ด้วย เช่นกัน

แบบฝึกหัดบทที่ 5

จงเขียนโปรแกรมเพื่อแก้ปัญหาที่เกี่ยวข้องกับการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีน 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 กรดอะมิโนแต่ละตัวสามารถถูกแทนที่ด้วยกรดอะมิโนอื่นด้วยความถี่ที่แตกต่างกัน เมทริกซ์คะแนนแพม (Pointe Accepted Mutation: PAM scoring metrics) เป็นเมทริกซ์ที่แต่ละแถวและคอลัมน์เป็นตัวแทนของแต่ละกรดอะมิโน ในกรณีของชีวสารสนเทศเมทริกซ์คะแนนแพมถูกใช้เป็นเมทริกซ์ที่ให้คะแนนการเกิดมิสแมช (การแทนที่กรดอะมิโนหนึ่งด้วยอีกกรดอะมิโนหนึ่ง) หรือที่เรียกว่า substitution matrix ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายของโปรตีน ค่าในแต่ละช่องของเมทริกซ์คะแนนแพม $M(i,j)$ บ่งชี้โอกาสที่กรดอะมิโนแถว i จะถูกแทนที่ด้วยกรดอะมิโนในแต่ละคอลัมน์ ผ่านกระบวนการแปรผันในธรรมชาติในวิวัฒนาการของสิ่งมีชีวิต โดยไม่ใช่การแทนที่แบบสุ่ม เมทริกซ์คะแนนแพมแต่ละเมทริกซ์ เช่น PAM_1, PAM_{50} ตัวเลข 1 และ 50 เกี่ยวเนื่องกับระยะเวลาในขบวนการวิวัฒนาการของลำดับกรดอะมิโนในสายของโปรตีนที่นำมาใช้ในการสร้างเมทริกซ์ PAM_n หมายถึงในลำดับกรดอะมิโนทุก 100 ตำแหน่งจะมี n การแปรผัน เมทริกซ์คะแนนแพม ถูกเสนอครั้งแรกโดย มากาเร็ต เดย์ฮอฟ (Margaret Dayhoff) ในปี ค.ศ. 1978 โดยเมทริกซ์เหล่านี้ถูกสร้างจาก การแปรผันของกรดอะมิโนจำนวน 1572 ตำแหน่งที่พบในต้นไม้วิวัฒนาการ (phylogenetic trees) ที่สร้างจากชุดของสายโปรตีนที่มีความคล้ายคลึงและเกี่ยวเนื่องกันมากจำนวน 71 แฟมมิลี (families) โปรตีนที่ถูกนำมาเปรียบเทียบเคียงความคล้ายคลึงกันจะต้องมีความอนุรักษ์โดยรวมอย่างน้อย 85%

เมทริกซ์คะแนน PAM_1 ถูกสร้างขึ้นจากผลของการเปรียบเทียบชุดของโปรตีนที่มีความเหมือนกัน 99%โดย $M(i, j)$ หรือ mutation matrix จะเก็บจำนวนครั้งที่กรดอะมิโน i และ j ปรากฏในคอลัมน์เดียวกันหารด้วยจำนวน

ครั้งทั้งหมดที่กรดอะมิโน *i* ปรากฏในตำแหน่งต่างๆ ของทุกสายโปรตีนที่นำมาเปรียบเทียบ ถ้าให้ *f(j)* เป็นความถี่ของการปรากฏกรดอะมิโน *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 [128])

เมทริกซ์คะแนนบลอสซัม

เมทริกซ์คะแนนบลอสซัม (BLOSUM scoring metrics หรือ **BLO**cks **SU**bstitution **M**atrix) เป็นเมทริกซ์ที่มีลักษณะเดียวกับเมทริกซ์คะแนนแพม โดยแต่ละแถวและคอลัมน์เป็นตัวแทนของแต่ละกรดอะมิโน โดยแตกต่างจากเมทริกซ์คะแนนแพมในการได้มาซึ่งคะแนนในแต่ละช่อง เมทริกซ์คะแนนบลอสซัมได้ถูกนำเสนอเป็นครั้งแรกโดยสตีเวน เฮนนิคอฟ (Steven Henikoff) และโจรา เฮนนิคอฟ (Jora Henikoff) ในปีค.ศ. 1992 [129] โดยการพิจารณาลำดับกรดอะมิโน ในบริเวณที่มีความอนุรักษ์สูง (ไม่มีแกป) ระหว่างชุดโปรตีนที่อยู่ในแฟมิลีเดียวกันในฐานข้อมูล BLOCKS [130] และทำการนับความถี่ของการเกิดกรดอะมิโนแต่ละตัวและความน่าจะเป็นที่กรดอะมิโนหนึ่งจะถูกแทนที่ด้วยอีกกรดอะมิโนหนึ่ง จากนั้นคำนวณคะแนนโดยใช้ log-odds ทุกเมทริกซ์บลอสซัมถูกคำนวณมาจากการสังเกตผลของการเปรียบเทียบความคล้ายคลึงกันระหว่างสายโปรตีนภายในชุดที่มีความคล้ายกันที่

เปอร์เซ็นต์หนึ่งๆ ค่าเมทริกซ์จำเพาะอย่าง BLOSUM80 ถูกสร้างจากชุดของโปรตีนที่มีความคล้ายคลึงกันโดยเฉลี่ย 80% โดยสายโปรตีนที่อยู่ในชุดไม่ได้ถูกอ้างอิงจากความใกล้เคียงกันในเชิงวิวัฒนาการอย่างในกรณีของเมทริกซ์แพม ตัวอย่างของเมทริกซ์ BLOSUM62 ถูกแสดงในรูปที่ 5.15

BLOSUM62 matrix

		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
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

รูปที่ 5.17 เมทริกซ์คะแนนบลอสซัม 62 (BLOSUM62)

(ที่มา: รูปที่ 6.10 ของ [128])

ที่มาของหัวข้อนี้เรียบเรียงจาก [128]

ความแตกต่างระหว่างเมทริกซ์คะแนนแพมและบลอสซัม

สำหรับเมทริกซ์คะแนนแพมค่าตัวเลขที่มากเช่น PAM250 จะใช้งานเป็นเมทริกซ์คะแนนเพื่อการเปรียบเทียบความคล้ายคลึงกันของสายโปรตีนได้ดีในกรณีที่สายโปรตีนที่นำมาเปรียบเทียบกันนั้นค่อนข้างห่างกันในเชิงวิวัฒนาการ ในขณะที่ค่าที่น้อยลงจะใช้ได้ดีกับการเปรียบเทียบความคล้ายคลึงกันของสายโปรตีนที่มีความใกล้เคียงกันในเชิงวัฒนาการ สำหรับเมทริกซ์บลอสซัมค่าตัวเลขต่อท้ายแต่ละเมทริกซ์มีความหมายในเชิงตรงกันข้ามกับแพม โดยค่าตัวเลขมากจะเหมาะกับการนำไปใช้ในการเปรียบเทียบความคล้ายคลึงกันของสายโปรตีนที่มีความใกล้เคียงกัน ในขณะที่ตัวเลขต่อท้ายเมทริกซ์ที่มีค่าน้อยเช่น BLOSUM45 จะเหมาะกับการนำไปใช้ในการเปรียบเทียบชุดของโปรตีนที่แตกต่างกันมากกว่า สตีเวนและโจรา เฮนนิคอฟได้เปรียบเทียบสรุปความเกี่ยวเนื่องกันระหว่างเมทริกซ์คะแนนสองแบบดังต่อไปนี้

PAM250 ≈ BLOSUM45

PAM120 ≈ BLOSUM60

PAM160 ≈ BLOSUM62

โดยเมทริกซ์คะแนน BLOSUM62 ถูกใช้มากที่สุดในการให้คะแนนการแทนค่ากรดอะมิโนเมื่อเกิดมิสแมช (mismatch) ในการเปรียบเทียบความคล้ายกันระหว่างสายโปรตีน รวมทั้งถูกใช้เป็นเมทริกซ์หลักโดยโปรแกรม BLAST ในเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอหรือโปรตีนที่เป็นข้อมูลเข้ากับฐานข้อมูลของสายโปรตีน โดยสรุปเมทริกซ์คะแนนแพมและบลอสซัมมีความแตกต่างกันดังต่อไปนี้

1. เมทริกซ์คะแนนแพมถูกสร้างโดยอ้างอิงกับความเกี่ยวข้องของโปรตีนในเชิงวิวัฒนาการ อัตราในการถูกแทนที่ของกรดอะมิโนในแต่ละตำแหน่งด้วยกรดอะมิโนอื่นๆ ถูกประมาณจากจากต้นไม้วิวัฒนาการ (phylogenetic tree) ที่สร้างขึ้นและสายของโปรตีนที่เป็นบรรพบุรุษ ในขณะที่เมทริกซ์บลอสซัมได้มาจากการสังเกตข้อมูลที่เป็นผลมาจากการเทียบเคียงชุดของสายโปรตีนในบริเวณที่ไม่มีแกปและมีความอนุรักษ์ร่วมกันสูง ดังนั้นในทางปฏิบัติเมทริกซ์คะแนนแพมมักถูกใช้ในการสร้างเปรียบเทียบชุดของสายโปรตีนเพื่อสร้างต้นไม้วิวัฒนาการ ในขณะที่เมทริกซ์คะแนนบลอสซัมจะเหมาะสมในการนำใช้ในการเปรียบเทียบสายของโปรตีนแบบเฉพาะส่วน (local alignment)
2. เมทริกซ์คะแนนแพมถูกสร้างจากความยาวโดยรวมของสายของโปรตีนซึ่งรวมทั้งบริเวณที่อนุรักษ์และไม่อนุรักษ์ดังนั้นมีความเกี่ยวเนื่องกับการเปรียบเทียบสายของโปรตีนแบบภาพรวม (global alignment) ในขณะที่เมทริกซ์บลอสซัม ถูกสร้างจากผลการเปรียบเทียบความคล้ายคลึงกันของสายโปรตีนเฉพาะส่วน (local alignment) จำเพาะบริเวณที่มีความอนุรักษ์ระหว่างสายของโปรตีน

ถึงแม้ว่าเครื่องมือทางชีวสารสนเทศมักมีการเตรียมเมทริกซ์คะแนนหลักไว้ให้ใช้ เช่นโปรแกรม BLAST ใช้ BLOSUM62 เป็นเมทริกซ์คะแนนหลัก อย่างไรก็ตามในมุมมองของนักชีวสารสนเทศการเข้าใจคุณลักษณะ ข้อดีและข้อจำกัดของแต่ละเมทริกซ์ และสามารถเลือกใช้ได้ถูกต้องเหมาะสมตามวัตถุประสงค์เป็นสิ่งที่สำคัญ โดยมีคำแนะนำในการเลือกใช้เมทริกซ์คะแนนแสดงดังต่อไปนี้

PAM100 ≈ BLOSUM90 (สำหรับเปรียบเทียบสายโปรตีนที่มีความใกล้เคียงกันมาก)
PAM120 ≈ BLOSUM80 (สำหรับเปรียบเทียบสายโปรตีนทั่วไปแทบทั้งหมด)
PAM160 ≈ BLOSUM62 (สำหรับเปรียบเทียบสายโปรตีนทั่วไปแทบทั้งหมด)
PAM200 ≈ BLOSUM52 (สำหรับเปรียบเทียบสายโปรตีนทั่วไปแทบทั้งหมด)
PAM250 ≈ BLOSUM45 (สำหรับเปรียบเทียบสายโปรตีนที่แตกต่างกันมาก)

รูปแบบไฟล์ที่เกี่ยวข้อง

CLUSTAL

มีรูปแบบไฟล์ที่ใช้ในการแสดงผลการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนมีหลายรูแบบ แต่รูปแบบที่เป็นที่รู้จักและมีการใช้งานกันโดยทั่วไปคือรูปแบบ clustal ซึ่งเป็นรูปแบบไฟล์แสดงผลจากโปรแกรม CLUSTAL W โดยไฟล์รูปแบบ clustal (รูปที่ 5.16) จะเป็นเท็กซ์ไฟล์ที่มีรูปแบบจำเพาะ บรรทัดแรกสุด

จะเริ่มด้วยคำว่า CLUSTAL W แล้วตามด้วยเลขเวอร์ชันในวงเล็บ ส่วนบรรทัดที่แสดงผลการเปรียบเทียบจะมีแสดงความยาวของสายข้อมูลไม่เกิน 60 อักขระต่อบรรทัด โดยที่สายข้อมูลที่นำมาเทียบจะอยู่ในบรรทัดถัดไป และบรรทัดลำดับที่สามจะเป็นสัญลักษณ์แสดงผลการเปรียบเทียบของแต่ละตำแหน่ง โดยสัญลักษณ์ ‘*’ แสดงสถานะแมช สัญลักษณ์ ‘:’ แสดงว่ากรดอะมิโนไม่เหมือนกันแต่ก็พบมาก่อนว่าเกิดการแทนกันได้ สัญลักษณ์ ‘.’ แสดงว่ากรดอะมิโนไม่เหมือนกันแต่ก็พบว่ามีโอกาสที่จะเกิดการแทนกันได้บ้าง ส่วนสัญลักษณ์ ‘-’ แสดงแกป และช่องว่าง ‘ ’ แสดงว่าไม่แมช สำหรับเลขต่อท้ายแต่ละบรรทัดเช่น 60 ในสายข้อมูลส่วนแรก และ 120 ในสายข้อมูลส่วนที่สอง อาจมีหรือไม่มีก็ได้ ถ้ามีจะแสดงจำนวนอักขระในบรรทัดนั้นๆ 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 ตัวอย่างรูปแบบไฟล์ CLUSTAL

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

สำหรับรายละเอียดของรูปแบบไฟล์อื่นๆที่เป็นไปได้สามารถอ่านเพิ่มเติมได้จาก

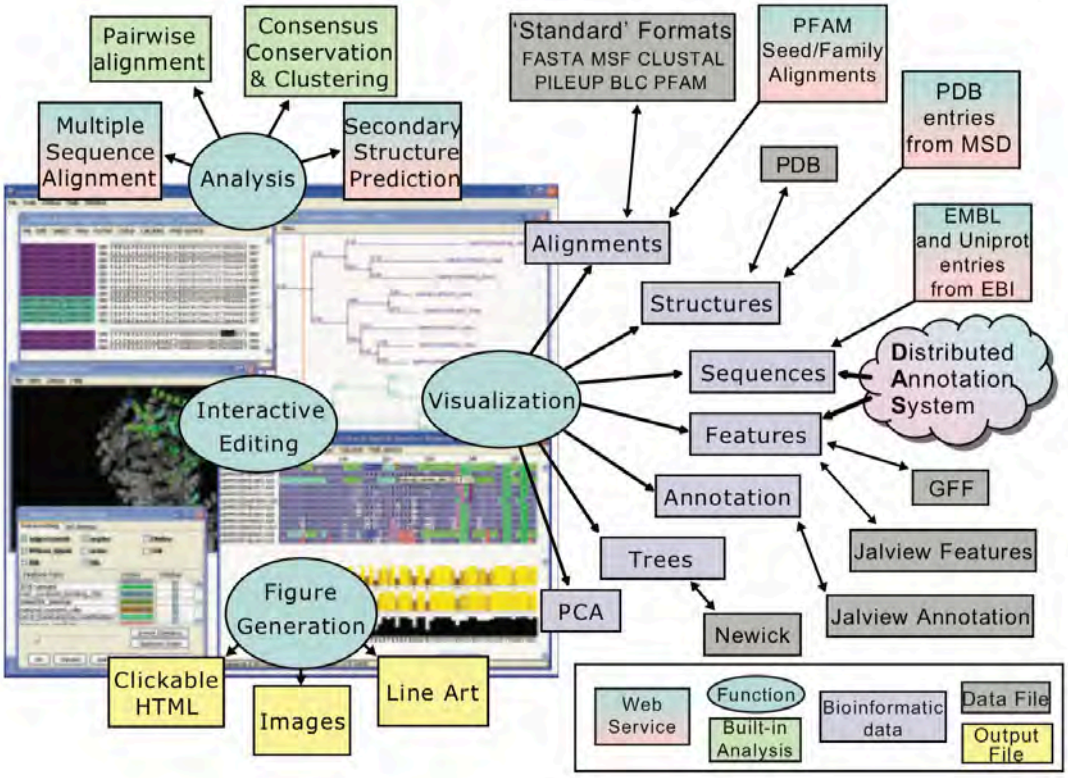
http://emboss.open-bio.org/html/use/ch05s04.html

โปรแกรมที่ใช้ในการแก้ไข แสดงผล และเปรียบเทียบความคล้ายคลึงกันของชุดของสายดีเอ็นเอและหรือโปรตีน

Jalview

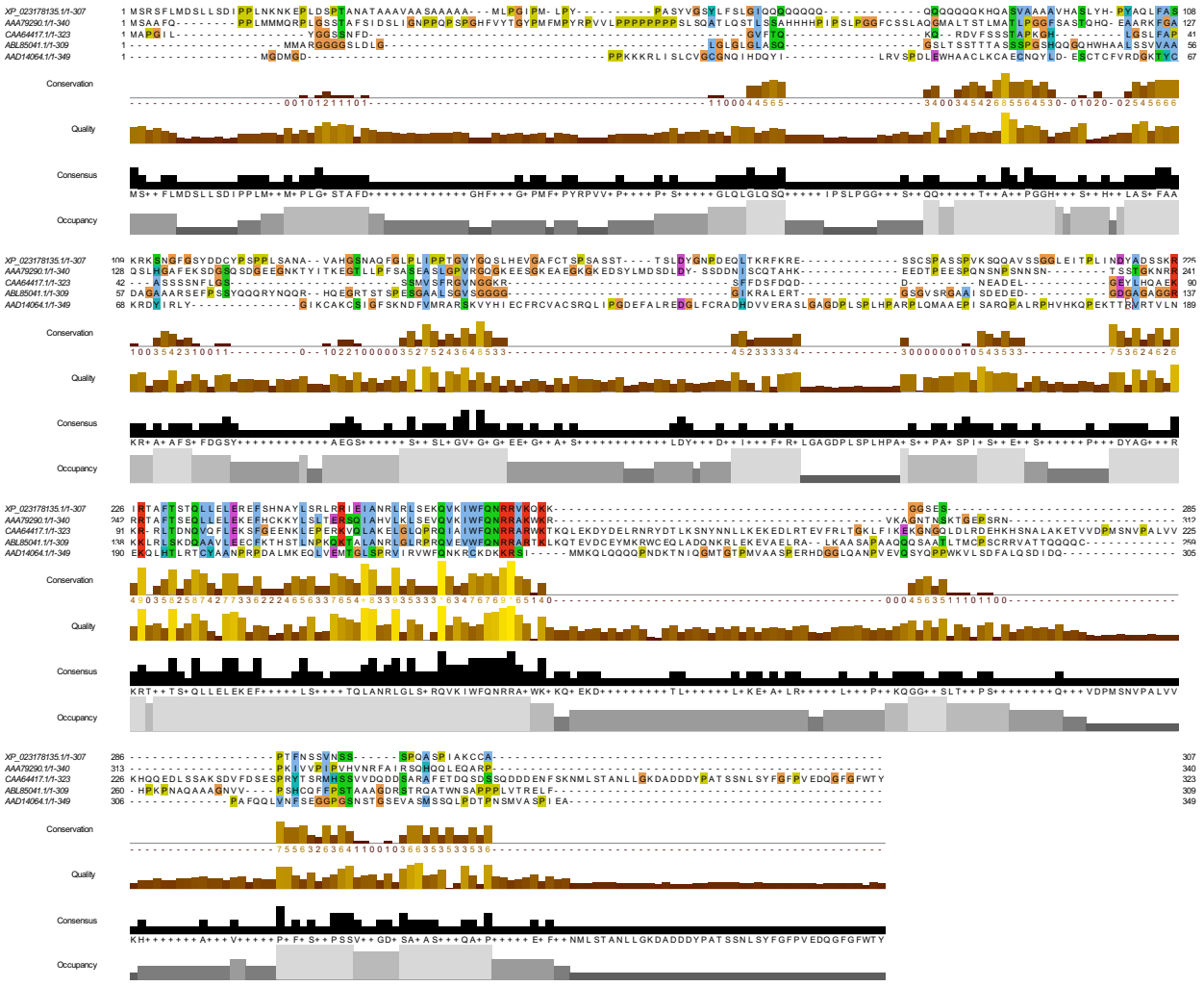
โปรแกรม Jalview [131, 132] (รูปที่ 5.19) เป็นโปรแกรมโอเพนซอร์ส (open source) ที่ใช้ในการแก้ไข แสดงผล และเปรียบเทียบความคล้ายคลึงกันระหว่างสายข้อมูลประเภทเดียวกันภายในชุด ซึ่งอาจเป็นชุดของ

สายข้อมูลดีเอ็นเอ ชุดของสายข้อมูลอาร์เอ็นเอ หรือชุดของสายข้อมูลโปรตีน นอกจากนี้โปรแกรม Jalview ยังช่วยเชื่อมโยงข้อมูลอื่นๆที่เกี่ยวข้องเข้ามาในการวิเคราะห์และแสดงผลได้โดยอัตโนมัติ เช่นเชื่อมโยงโครงสร้างสามมิติของแต่ละสาย ข้อมูลโปรตีน (ถ้ามีโครงสร้างในฐานข้อมูลเปิด) ข้อมูลโปรตีนโดเมนจาก Pfam ข้อมูลอาร์เอ็นเอแฟมิลีจากฐานข้อมูล Rfam เป็นต้น รูปที่ 5.20 โปรแกรม Jalview แสดงผลของการรันโปรแกรม MUSCE ซึ่งทำ multiple sequence alignment โดยมีข้อมูลเข้าเป็นโปรตีนโฮมีโอบอกซ์ 5 เส้นจากมนุษย์ (*Homo sapiens*) แมลงวัน (*Drosophila hydei*) คางคก (*Xenopus laevis)* มะเขือเทศ (*Solanum lycopersicum)* และหญ้าชนิดหนึ่ง (*Brachypodium sylvaticum*) โดยพบว่ามีบริเวณจำเพาะมากกว่า 1 บริเวณที่มีความอนุรักษ์ร่วมกันระหว่าง 5 สิ่งมีชีวิต



รูปที่ 5.19 ตัวอย่างหน้าจอของโปรแกรม Jalview ฟังก์ชันการทำงาน และความสามารถในการเชื่อมโยงข้อมูลกับฐานข้อมูลสาธารณะ

(ที่มา: รูปที่ 1 ของ [132])



รูปที่ 5.20 โปรแกรม Jalview แสดงผลการรันโปรแกรม MUSCLE ในการเปรียบเทียบความคล้ายคลึงกันของโปรตีนโฮมีโอบอกซ์ของสิ่งมีชีวิต 5 ชนิดคือ (1) มนุษย์ (2) Xenopus laevis (3) Drosophila hydei (4) Solanum lycopersicum และ (5) Brachypodium sylvaticum

บทที่ 6 การจำแนกฟีโนไทป์ของไวรัสเอชไอวี (Classifying HIV phenotypes)

วัตถุประสงค์

- เพื่อให้นิสิตเห็นตัวอย่างของปัญหาทางชีววิทยาที่อัลกอริทึมในบทเรียนก่อนหน้ายังไม่สามารถตอบปัญหาได้ดีพอ
- เพื่อให้นิสิตได้เห็นแนวทางที่แตกต่างในการเปรียบเทียบความคล้ายคลึงกันของสายข้อมูลดีเอ็นเอและหรือโปรตีน
- เพื่อให้นิสิตเข้าใจองค์ประกอบพื้นฐานของ Hidden Markov Model (HMM) แผนภาพ HMM ปัญหา Decoding อัลกอริทึมวิเทอบิ อัลกอริทึมฟอร์เวิร์ดแบคเวิร์ด อัลกอริทึม Baum-Welch
- เพื่อให้นิสิตได้เห็นตัวอย่างงานวิจัยและผลงานวิจัย รวมทั้งตัวอย่างโปรแกรมที่ใช้ HMM
- เพื่อให้นิสิตได้เห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทาย รวมทั้งงานวิจัยอื่นๆ ที่เกี่ยวข้อง

ผลลัพธ์ที่คาดหวัง

- นิสิตสามารถอธิบายความแตกต่างของการเปรียบเทียบความคล้ายคลึงกันระหว่างสายข้อมูลโดยวิธีการทำ sequence alignment ในบทที่ 5 และการใช้โปรไฟล์ HMM ในบทเรียนนี้
- นิสิตสามารถอธิบายองค์ประกอบหลักของ HMM สามารถเขียนแผนภาพ HMM กราฟวิเทอบิ สามารถหาค่าพารามิเตอร์ที่เกี่ยวข้องเช่น ค่าความน่าจะเป็นในการเปลี่ยนสถานะ ค่าความน่าจะเป็นในการส่งออกอักขระในสายลำดับสายข้อมูลที่ส่งออก การทำงานของอัลกอริทึมหลักๆ ที่เกี่ยวข้อง
- นิสิตสามารถเขียนโปรแกรมเพื่อใช้ HMM ในการแก้ปัญหาอย่างง่ายได้
- นิสิตสามารถยกตัวอย่างโปรแกรมและฐานข้อมูลที่เกี่ยวข้องกับ HMM เพื่อการเปรียบเทียบความคล้ายคลึงกันระหว่างสายโปรตีนได้
- นิสิตสามารถประยุกต์องค์ความรู้จากบทเรียนนี้เพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้

เนื้อหาโดยสรุป

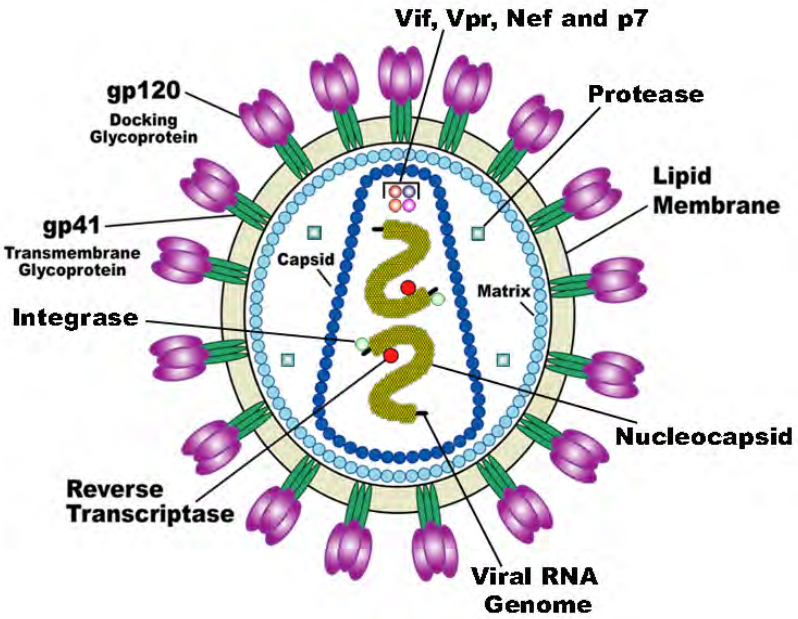
บทเรียนนี้ยกตัวอย่างปัญหาทางชีววิทยาในเรื่องของการเปรียบเทียบลำดับกรดอะมิโนของโปรตีน gp120 ในบริเวณ V3 loop ของไวรัสเอชไอวีที่มีความแปรผันค่อนข้างมากและมีเพียงตำแหน่ง 11 และ 25 ที่ถูกใช้ระบุว่าเกี่ยวข้องกับลักษณะการแสดงออกหรือฟีโนไทป์ที่จำเพาะของเชื้อ ซึ่งการเปรียบเทียบความคล้ายคลึงกันของสายข้อมูลในบทเรียนก่อนหน้าไม่มีการพิจารณาค่าความน่าจะเป็นของการส่งอักขระหนึ่งๆในลำดับจำเพาะหนึ่งๆ ในบทเรียนนี้ได้อธิบายแนวทางการเปรียบเทียบความคล้ายคลึงกันของสายข้อมูลโดยการประยุกต์ใช้แบบจำลองมาร์คอฟซ้อนเร้น (Hidden Markov Model: HMM) องค์ประกอบของ HMM แผนภาพ HMM กราฟวิเทอบิ และอัลกอริทึมที่เกี่ยวข้องเช่น อัลกอริทึมวิเทอบิ อัลกอริทึมฟอร์เวิร์ดแบคเวิร์ด อัลกอริทึม Baum-Welch เป็นต้น ตัวอย่างโปรแกรมและฐานข้อมูลที่ประยุกต์ใช้โปรไฟล์ HMM ในการเปรียบเทียบความคล้ายคลึงกันของสายโปรตีน รวมทั้งตัวอย่างการประยุกต์ใช้ HMM ในการแก้ปัญหาอื่นๆทางชีววิทยา

บทที่ 6 การจำแนกฟีโนไทป์ของไวรัสเอชไอวี
(Clasifying HIV phenotypes)

ไวรัสเอชไอวีหลบเลี่ยงระบบภูมิคุ้มกันในร่างกายมนุษย์ได้อย่างไร

ในปีค.ศ. 1984 มาร์กาเร็ต เฮคเคลอร์ (Margaret Heckler) ซึ่งเป็นรัฐมนตรีประจำกระทรวงสาธารณสุข (US Health and Human Services) ของประเทศสหรัฐอเมริกาในขณะนั้นได้ประกาศว่าจะมีวัคซีนเพื่อป้องกันโรคเอชไอวีภายในสองปีและในปี ค.ศ. 1997 ประธานาธิบดีบิล คลินตัน (Bill Clinton) ในขณะนั้นได้อนุมัติศูนย์วิจัยใหม่ภายใต้สถาบันสาธารณสุขแห่งชาติ (National Institute of Health: NIH) โดยมีเป้าหมายเพื่อพัฒนาวัคซีนเอชไอวี ในปี ค.ศ. 2005 บริษัทเมิร์ก (Merck) ได้เริ่มการทดลองวัคซีนเอชไอวีทางคลินิกและหยุดการทดลองหลังผ่านไป 2 ปีหลังจากที่พบว่าวัคซีนที่ทดลองนั้นเพิ่มความเสี่ยงของการติดเชื้อเอชไอวีในผู้รับวัคซีนบางราย

ในปัจจุบันยังไม่มีวัคซีนเอชไอวีที่ได้รับการรับรองถึงแม้จะมีการลงทุนอย่างมากมายรวมทั้งยังมีการทดลองเชิงคลินิกอย่างต่อเนื่อง และมีประชากร 36.7 ล้านคนทั่วโลกที่เป็นผู้ติดเชื้อ (ที่มา: https://www.hiv.gov/hiv-basics/overview/data-and-trends/global-statistics เข้าถึงออนไลน์เมื่อวันที่ 11 ก.พ. พ.ศ. 2561) ทั้งนี้งานวิจัยและพัฒนาในเชิงการรักษาได้มีความก้าวหน้าเป็นอย่างมากและประสบความสำเร็จในการพัฒนายาต้านรีโทรไวรัสประกอบด้วยชุดของยาที่ทำให้อาการของผู้ป่วยติดเชื้อคงที่ อย่างไรก็ตามรักษาด้วยวิธีนี้ไม่สามารถทำให้หายขาดจากโรครวมทั้งไม่สามารถควบคุมการแพร่กระจายเชื้อ



รูปที่ 6.1 ไวรัสเอชไอวี

(ที่มา: US National Institute of Health/Wikipedia)

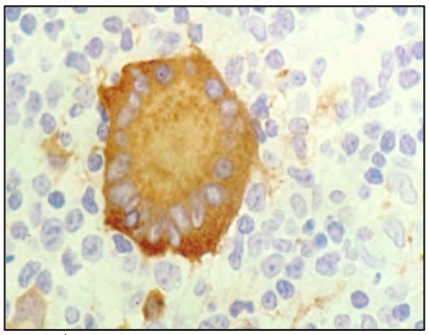
วัคซีนที่ใช้ในการป้องกันเชื้อไวรัสมักถูกสร้างจากโปรตีนที่ผนังเซลล์ของไวรัส (รูปที่ 6.1) โดยวัคซีนเหล่านี้จะกระตุ้นภูมิคุ้มกันของมนุษย์ในรู้จักโปรตีนที่ผิวเซลล์ของไวรัสว่าเป็นโปรตีนแปลกปลอม ต้องทำลาย และจดจำไว้เพื่อคอยคุ้มกันเซลล์ภายในร่างกายจากไวรัส อย่างไรก็ตามโปรตีนที่ผิวเซลล์ของไวรัสเอชไอวีมีความแปรผันของลำดับกรดอะมิโนมากเนื่องจากไวรัสเองต้องพยายามเปลี่ยนแปลงตัวเองให้รวดเร็วเพื่อความอยู่รอด ประชากรของเชื้อเอชไอวีในผู้ติดเชื้อรายหนึ่งๆ มีการเปลี่ยนแปลงตัวไปอย่างรวดเร็วเพื่อให้สามารถหลีกเลี่ยงภูมิคุ้มกันของผู้ติดเชื้อได้ รูปที่ 6.2 แสดงผลของการทำ multiple alignment ส่วนของโปรตีน gp120 ที่เก็บมาจากผู้ติดเชื้อ 1 รายใน 9 ช่วงเวลาที่แตกต่างกัน โดยคอลัมน์ที่เป็นสีเข้มแสดงส่วนที่แตกต่างกันระหว่างช่วงเวลา และคอลัมน์สีฟ้าแสดงกรดอะมิโนที่แตกต่างจากกรดอะมิโนหลักของคอลัมน์นั้นๆ ทั้งนี้เพื่อแสดงให้เห็นว่าเชื้อเอชไอวีมีการปรับเปลี่ยนตัวเองได้รวดเร็วมาก ซึ่งถ้านำเชื้อเอชไอวีของผู้ติดเชื้อหลายๆ คนมาเทียบกันก็จะมีความแปรผันมากขึ้นไปอีก ดังนั้นวัคซีนเอชไอวีที่มีประสิทธิภาพจะต้องมีความครอบคลุมในการรู้จักโปรตีนของไวรัสเอชไอวีที่มีรูปแบบที่หลากหลาย โดยอาจสร้างสายเปปไทด์โดยจำลองส่วนที่มีความแปรผันน้อยที่สุดของโปรตีนที่ผิวเซลล์ของไวรัสเอชไอวีและใช้เปปไทด์นี้เป็นวัคซีน อย่างไรก็ตามนอกจากการแปรผันที่รวดเร็วของโปรตีนที่ผิวเซลล์แล้ว โปรตีนเหล่านี้ยังสามารถหลบเลี่ยงโดยกระบวนการไกลโคซีเลชัน (glycosylation) ซึ่งเป็นดัดแปลงโมเลกุลของโปรตีนหลังการแปลรหัส (post-translational modification) เหมือนการใส่หน้ากากซึ่งทำให้สามารถหลบซ่อนได้จากระบบภูมิคุ้มกันของผู้ติดเชื้อ ผลคือยังไม่วัคซีนเอชไอวีที่ใช้งานได้อย่างมีประสิทธิภาพ

รูปที่ 6.2 ผลของการทำ multiple alignment ส่วนของโปรตีน gp120 ที่เก็บมาจากผู้ติดเชื้อ 1 รายใน 9 ช่วงเวลาที่แตกต่างกัน

(ที่มา: รูปที่ 10.1 ของ [21])

ไวรัสเอชไอวีประกอบด้วย 9 ยีนโดยในบทเรียนนี้เราจะมุ่งเป้าไปที่ยีน *env* ที่มีอัตราการเปลี่ยนแปลงลำดับเบสสูงมากคือประมาณ 1-2% ต่อนิวคลีโอไทด์ต่อปี โดยโปรตีนที่แปลรหัสมาจากยีน *env* นี้จะถูกตัดออกเป็นสองโปรตีนคือ ไกลโคโปรตีน จีพี 120 (glycoprotein gp 120) ที่มีความยาวประมาณ 480 กรดอะมิโน และไกลโคโปรตีน จีพี 41 (glycoprotein gp 41) ที่มีความยาวประมาณ 345 กรดอะมิโน โดยโปรตีนทั้งสองนี้จะจับกันเป็น envelope spike ซึ่งทำให้สามารถนำไวรัสเอชไอวีเข้าสู่เซลล์เจ้าบ้านได้

นอกจากนี้เนื่องจากไวรัสเอชไอวีมีการกลายพันธุ์อย่างรวดเร็ว เชื้อเอชไอวีที่แยกออกมาได้อาจมีลักษณะที่ปรากฏหรือฟีโนไทป์ (phenotype) ที่แตกต่างกันซึ่งการรักษาต้องใช้ชุดของยาที่แตกต่างกัน ตัวอย่างเช่นเชื้อเอชไอวีอาจแบ่งออกเป็นกลุ่ม ที่สามารถเพิ่มจำนวนได้อย่างรวดเร็วและเป็น syncytium-inducing (SI) และกลุ่มที่เพิ่มจำนวนช้าและเป็น non-syncytium-inducing (NSI) โดยในกลุ่มที่เป็น SI นั้น โปรตีน gp120 ในไวรัสจะถูกเคลื่อนย้ายไปที่ผิวเซลล์ของไวรัสโดยโปรตีนนี้สามารถทำให้ผิวเซลล์หลายๆเซลล์ของผู้ติดเชื้อรวมเข้าด้วยกันกลายเป็นเซลล์ขนาดใหญ่มีหลายนิวเคลียสและทำงานไม่ได้เรียกว่า syncytium ดังแสดงในรูปที่ 6.3 โดยกระบวนการนี้ทำให้ไวรัสเอชไอวีที่เป็น SI สามารถฆ่าเซลล์ของผู้ติดเชื้อได้หลายๆเซลล์พร้อมกันโดยการเข้าสู่เซลล์เพียงครั้งเดียว



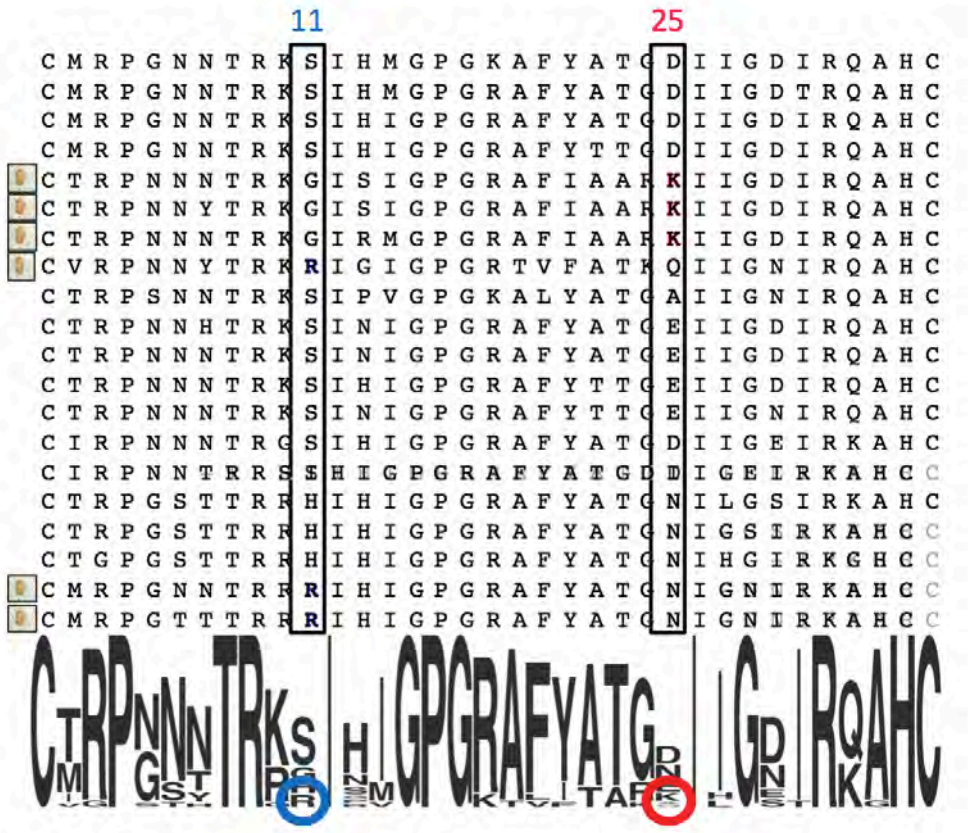
รูปที่ 6.3 Syncytium ที่พบในผู้ป่วยเอชไอวี โดยมีหลายนิวเคลียสอยู่ภายใน
(ที่มา: รูปที่ 10.2 ของ [21])

เนื่องจาก gp120 มีความสำคัญในการนำมาจำแนกการแสดงออกของไวรัสเอชไอวีเป็น SI และ NSI นักชีววิทยาจึงมีความสนใจในการตวรจสอบและตัดสินใจว่ากรดอะมิโนใดและในตำแหน่งไหนของ gp120 ที่สามารถนำมาใช้ในการจำแนกฟีโนไทป์นี้

ในปีค.ศ. 1992 Jean-Jacques De jong ได้ทำการวิเคราะห์ผลของการเปรียบเทียบชุดของลำดับกรดอะมิโนในบริเวณที่เป็นส่วน V3 loop ของโปรตีน gp120 ดังแสดงในรูปที่ 6.4 และได้สร้างกฎ 11/25 โดยระบุว่าเชื้อเอชไอวีที่มีโอกาสแสดงฟีโนไทป์ SI กรดอะมิโนที่ตำแหน่ง 11 หรือ 25 ในบริเวณที่เป็น V3 loop จะเป็นกรดอะมิโนอาร์จินีน (arginine: Arg: R) หรือ ไลซีน (lysine: Lys: K) ในการศึกษาหลังจากนั้นพบว่ามีอีกหลายตำแหน่งที่มีผลต่อการมี SI/NSI ฟีโนไทป์

ข้อจำกัดของการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีน

ก่อนที่นักชีววิทยาจะสามารถสร้างกฎเพื่อใช้ในการจำแนกฟีโนไทป์ของไวรัสเอชไอข้างต้น ปัญหาพื้นฐานที่พบคือการได้มาซึ่งผลของการเปรียบเทียบบริเวณ V3 loop ของโปรตีน gp120 จากคนไข้หลายๆคน ที่มีความถูกต้องสูง ซึ่งการเปรียบเทียบความคล้ายคลึงกันของชุดของโปรตีน (multiple alignment) นั้น ความผิดพลาดในการขยับลำดับกรดอะมิโนเพียงตำแหน่งเดียวจะมีผลต่อการชนิดของกรดอะมิโนที่จะปรากฏในตำแหน่ง 11 และ 25 ที่มีผลต่อการจำแนก SI/NSI ฟีโนไทป์ รูปที่ 6.4 (ล่าง) แสดงโมทิฟโลโก้ของ V3 loop ซึ่งจะเห็นว่ามีบางคอลัมน์



รูปที่ 6.4 ผลการเปรียบเทียบลำดับกรดอะมิโนบริเวณที่เป็น V3 loop ของโปรตีน gp120 จากผู้ป่วยเอชไอวี 20 ราย โดยคอลัมน์ที่ 11 และ 25 ของผู้ป่วยที่มี SI ฟีโนไทป์จะมีกรดอะมิโนเป็น arginine (R) หรือ lysine (K)
(ที่มา: รูปที่ 10.3 ของ [21])

ที่มีความอนุรักษ์สูงในขณะที่คอลัมน์อื่นๆมีความแปรผันสูง นอกจากนี้โมทิฟโลโก้ของ V3 loop นี้ไม่มีส่วนของ insertions/deletions ซึ่งมักปรากฏในส่วนอื่นๆที่ความอนุรักษ์น้อยกว่าบริเวณ V3 loop ซึ่ง insertions/ deletions นี้จะทำให้การวิเคราะห์โปรตีน gp120 โดยรวมมีความซับซ้อนขึ้นไปอีก และเนื่องจากแต่ละคอลัมน์มีความอนุรักษ์ของชุดของกรดอะมิโนมากน้อยแตกต่างกันไป คำถามที่ตามมาคือการใช้เมทริกซ์คะแนนอย่างบลอสซัมหรือแพม รวมทั้งคะแนนลงโทษอินเดลเดียวกันสำหรับทุกคอลัมน์จะสามารถตอบโจทย์การเทียบบริเวณ V3 loop ข้างต้น ที่คาดหวังให้ลำดับกรดอะมิโนหนึ่งๆอยู่ในคอลัมน์ที่ถูกต้องได้หรือไม่ ด้วยคำถามเหล่านี้จึงมีการเสนอแนวทางที่จะใช้เกณฑ์การให้คะแนนที่แตกต่างกันสำหรับแต่ละคอลัมน์ ตัวอย่างเช่น กรดอะมิโนที่ไม่ใช่อาร์จินีน (R) ในคอลัมน์ที่ 3 ในรูปที่ 6.4 ข้างต้น ควรมีคะแนนลงโทษมากกว่ากรดอะมิโนที่ไม่ใช่กรดอะมิโน S ในคอลัมน์ที่ 11 เป็นต้น โดยสรุปคือการเปรียบเทียบความคล้ายคลึงกันของชุดของโปรตีนในบทที่ 5 ไม่สามารถตอบโจทย์การเทียบ V3 loop ในบทนี้ได้ จึงต้องมีการออกแบบและพัฒนาอัลกอริทึมเพิ่มเติมในการเทียบชุดของสายโปรตีนโดยจะมีการนำวิธีการทางสถิติเข้ามาร่วมพิจารณา

เล่นพนันกับยากูซ่า

เกมส์พนันเกมส์หนึ่งที่เป็นที่นิยมคือการทายว่าลูกเต๋าสองลูกที่เจ้ามือเขย่าอยู่นั้น ถ้าเปิดออกมาแล้วผลรวมของหน้าลูกเต๋าจะมีค่าคู่หรือคี่ ซึ่งเกมส์ที่มีลักษณะเดียวกันแต่ซับซ้อนน้อยกว่าคือเกมส์ทายหัวหรือก้อย ในเกมส์ทายหัวก้อยนี้ถ้าในรอบการทายนึงๆ มีคนทายก้อยมากกว่าหัว เจ้ามือขี้โกงอาจเปลี่ยนเหรียญให้เป็นเหรียญที่เมื่อโยนแล้วมีโอกาสที่จะออกหัวเป็น ¾ แทนเหรียญปกติที่โอกาสที่จะออกหัวและก้อยเป็น ½ เท่ากัน

หยุดคิด	ถ้ามีการโยนเหรียญ 100 ครั้ง และออกหัว 63 ครั้ง คำถามคือเจ้ามือขี้โกงหรือไม่ และใช้เหรียญถ่วงน้ำหนักหรือไม่

เราไม่สามารถตอบปัญหาข้างต้นได้ชัดเจนเนื่องจากตัวคำถามเองไม่ได้กำหนดรายละเอียดเป็นทางการ เหรียญแต่ละเหรียญเมื่อโยนแต่ละครั้งก็มีโอกาสออกทั้งหัวและก้อย อย่างไรก็ตามสิ่งที่เราสามารถตอบได้คือเหรียญที่ใช้นั้นน่าจะเป็นเหรียญปกติ (Fair coin: F) หรือเหรียญถ่วงน้ำหนัก (Biased coin: B) เราเขียนค่าความน่าจะเป็นในการออกหัวและก้อยสำหรับเหรียญปกติ ได้ดังต่อไปนี้

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

และค่าความน่าจะเป็นในการออกหัวและก้อยสำหรับเหรียญถ่วงน้ำหนักคือ

$$\Pr_{\mathrm{B}}(\text{"H"}) = 3/4 \quad \Pr_{\mathrm{B}}(\text{"T"}) = 1/4$$

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

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

ในขณะที่เหรียญถ่วงน้ำหนักจะมีค่าความน่าจะเป็นในการเกิดลำดับ *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 $\Pr(x|F) > \Pr(x|B)$ เจ้ามือก็น่าจะใช้เหรียญปกติ ถ้า $\Pr(x|F) < \Pr(x|B)$ เจ้ามือก็น่าจะใช้เหรียญถ่วงน้ำหนัก เนื่องจากค่าของ $(1/2)^n$ และค่า $\frac{3^k}{4^n}$ มีค่าน้อยมากสำหรับ n ที่มีค่ามาก ดังนั้นเราจึงเลือกใช้ log-odds ratio ในการเปรียบเทียบเปรียบเทียบแทน ดังต่อไปนี้

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

ฝึกหัด	จงแสดงว่าค่า $\Pr(x\|F)$ มีค่ามากกว่า $\Pr(x\|B)$ ถ้าค่า log-odds ratio มีค่าเป็นบวกและน้อยกว่า $\Pr(x\|B)$ ค่า log-odds ratio มีค่าลบ

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

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

ซึ่งถ้าพิจารณาจากการเปรียบเทียบค่าความน่าจะเป็นข้างต้นจะพบว่า เจ้ามือน่าจะใช้เหรียญปกติ ถึงแม้ว่า 63 เข้าใกล้ 75 มากกว่า 50

เจ้ามือแอบใช้เหรียญถ่วงน้ำหนักสลับกับเหรียญปกติ

ถ้าเรามีสมมติฐานว่าก่อนโยนเหรียญในแต่ละรอบเจ้ามือมีโอกาสเปลี่ยนเหรียญที่โยนจากเหรียญปกติไปเป็นเหรียญถ่วงน้ำหนักด้วยความน่าจะเป็น 0.1 ถ้าเราเห็นผลของการโยนเหรียญในแต่ละรอบ เราจะสามารถทราบได้อย่างไรว่ารอบไหนเจ้ามือใช้เหรียญปกติและรอบไหนเจ้ามือใช้เหรียญถ่วงน้ำหนัก (นิยามปัญหาที่ 6.1) นิยามปัญหานี้ไม่ชัดเจนในเชิงการคำนวณ ทั้งนี้ทั้งเหรียญปกติและเหรียญถ่วงน้ำหนักสามารถออกหัวหรือก้อยก็ได้ สิ่งที่เราต้องการหาคือความน่าจะเป็นของลำดับการใช้เหรียญของเจ้ามือ

หยุดคิด	เราจะนิยามปัญหาคาสิโนข้างต้นใหม่ได้อย่างไร

นิยามปัญหาที่ 6.1 ปัญหาคาสิโน

ปัญหาคาสิโน (Casino Problem) ถ้ามีผลของการโยนเหรียญในแต่ละรอบให้ สามารถบอกได้ว่าเมื่อไหร่ที่เจ้ามือใช้เหรียญปกติและเมื่อไหร่เจ้ามือใช้เหรียญถ่วงน้ำหนัก	
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

การนิยามปัญหาคาสิโนข้างต้นจะต้องอยู่ในลักษณะที่เราสามารถเปรียบเทียบและเลือก $\boldsymbol{\pi}$ ใดๆ ที่น่าจะเป็นคำตอบที่ดีที่สุดได้

การหา CG-islands

ก่อนที่จะย้อนกลับไปเรื่องการโยนเหรียญ เราจะลองมาพิจารณาปัญหาทางชีววิทยาอีก 1 ปัญหา ซึ่งสามารถเทียบเคียงกับการโยนเหรียญข้างต้น ซึ่งวิธีการแก้ปัญหาสามารถนำไปประยุกต์ใช้กับการแก้ปัญหาทางชีววิทยาได้อีกหลากหลายปัญหา รวมทั้งการปัญหาการจำแนกฟีโนไทป์ของไวรัสเอชไอวี

ย้อนกลับไปเมื่อต้นคริสต์ศตวรรษที่ 20 ฟีบัส เลวีน (Phoebus Levene) ได้ค้นพบนิวคลีโอไทด์ 4 ตัวที่ประกอบกันเป็นสายดีเอ็นเอ ที่เวลานั้นอย่างไรก็ตามความรู้เกี่ยวกับดีเอ็นเอยังมีไม่มากนัก (ผลงานวิจัยของวัตสันและคริกเกี่ยวกับดีเอ็นเอสายคู่ได้รับการตีพิมพ์หลังจากนั้นประมาณ 50 ปี) ผลคือเลวีนยังมีความสงสัยว่าดีเอ็นเอเก็บข้อมูลทางพันธุกรรมโดยใช้เพียง 4 ตัวอักษรได้อย่างไร และได้ตั้งสมมติฐานว่าดีเอ็นเอประกอบด้วย 4 นิวคลีโอไทด์นี้ด้วยจำนวนเท่าๆกัน ศตวรรษถัดมาเรามีองค์ความรู้เพิ่มเติมว่าคู่สบของแต่ละนิวคลีโอไทด์ที่อยู่สแตรนตรงข้ามประกอบเป็นดีเอ็นเอสายคู่นั้นมีสัดส่วนของแต่นิวคลีโอไทด์ในปริมาณเท่าๆกัน อย่างไรก็ตามสมมติฐานที่ว่าสายดีเอ็นเอสายเดี่ยวประกอบด้วย 4 นิวคลีโอไทด์จำนวนเท่าๆกันนั้นไม่เป็นความจริง นอกจากนี้จีโนมของสิ่งมีชีวิตที่แตกต่างกันจะมีองค์ประกอบของนิวคลีโอไทด์ **G** และ **C** (GC-content) ไม่เท่ากัน ตัวอย่างเช่นมนุษย์มี GC-content ประมาณ 42% และเมื่อพิจารณาเฉพาะส่วนของ GC-content เราอาจคาดว่าคู่ของนิวคลีโอไทด์ (dinucleotide) ที่อยู่ติดกันเช่น **CC**, **CG**, **GC** และ **GG** จะพบในจีโนมมนุษย์ด้วยความถี่ 0.21 x 0.21 = 4.41% อย่างไรก็ตามความถี่ของ **CG** ในจีโนมมนุษย์มีเพียงประมาณ 1% คู่นิวคลีโอไทด์ **CG** ที่พบได้น้อยกว่าที่คาดการณ์นี้เกิดจากมีกระบวนการเมธิลเลชัน (Methylation) ซึ่งเป็นกระบวนการที่พบเป็นปกติในธรรมชาติในการดัดแปลงดีเอ็นเอโดยการเติมกลุ่มเมธิล (CH_3) ให้กับนิวคลีโอไทด์ไซโตซีน (**C**) ที่อยู่ติดเป็นคู่กับกวานีน (**G**) ผลคือไซโตซีนที่ถูกเติมกลุ่มเมธิลเข้าไปนั้นมีโอกาสที่จะเปลี่ยนไปเป็นไทมีน (**T**) และเป็นที่มาว่าทำไมความถี่ของการเกิดนิวคลีโอไทด์คู่ **CG** จึงต่ำกว่าการเกิดนิวคลีโอไทด์คู่อื่นๆ ในจีโนมของสิ่งมีชีวิตหลายชนิด อย่างไรก็ตามกระบวนการเมธิลเลชันมักถูกยับยั้งในบริเวณรอบๆยีนที่เรียกว่า CG-island ซึ่งเป็นบริเวณที่มีความถี่ของ **CG** สูงกว่าบริเวณอื่น ดังนั้นการหาบริเวณที่น่าจะเป็นยีนในจีโนมนั้นวิธีการหนึ่งคือการหาบริเวณที่เป็น CG-island

วิธีการแบบง่ายในการหาบริเวณที่เป็น CG-islands ในจีโนมคือการสร้างไล่ดูลำดับเบสในสายของจีโนมโดยใช้สไลดิงวินโดว์ (sliding window) (กำหนดความยาวของสายจีโนมที่ต้องการพิจารณาในครั้งหนึ่งๆ ด้วยขนาดของสไลดิงวินโดว์) และกำหนดว่าส่วนของจีโนมใดที่อยู่ในช่วงของวินโดว์และมีความถี่ของ **CG** มากจีโนมบริเวณนั้นก็น่าจะเป็น CG-island และน่าจะมียีนอยู่ใกล้ๆ ข้อจำกัดของวิธีการนี้คือเราไม่ทราบขนาดของวินโดว์ที่เหมาะสม ทั้งนี้ในขณะที่เลื่อนวินโดว์ไป ในบริเวณที่คาบเกี่ยวกันอาจถูกระบุว่าเป็น CG island บ้าง ไม่เป็น CG island บ้างก็ได้

แบบจำลองมาร์คอฟซ้อนเร้น

จากการโยนเหรียญมาเป็นแบบจำลองมาร์คอฟซ้อนเร้น

เป้าหมายของเราคือพัฒนาแนวคิดและแบบจำลองที่สามารถนำไปประยุกต์ใช้ในการแก้ทั้งปัญหาคาสิโนและการหา CG-island ข้างต้น อาจลองจินตนาการโดยเปลี่ยนเจ้ามือในคาสิโนที่เป็นคนให้เป็นเครื่องจักรเครื่องหนึ่งซึ่งเราไม่ทราบว่าเครื่องจักรนี้ถูกสร้างขึ้นมาและมีกลไกการทำงานภายในอย่างไร อย่างไรก็ตามสิ่งที่เราทราบคือเครื่องจักรนี้มีการทำงานเป็นรอบ โดยในแต่ละรอบนั้นเครื่องจักรจะอยู่ในสถานะใดสถานะหนึ่งที่ซ้อนเร้นอยู่ระหว่าง F

และ B และจะแสดงผลออกมาให้เห็นเป็น “H” หรือ “T” เท่านั้น โดยในแต่ละรอบที่เครื่องจักรทำงาน เครื่องจักรจะต้องตัดสินใจสองอย่าง

- จะย้ายไปที่สถานะซ้อนเร้นใดระหว่าง F และ B
- จะแสดงผลออกไปเป็น “H” หรือ “T”

เครื่องจักรตอบคำถามแรกโดยการเลือกแบบสุ่มระหว่างสองสถานะโดยมีค่าความน่าจะเป็นที่จะอยู่ในสถานะเดิม 0.9 และย้ายไปอีกสถานะหนึ่งเท่ากับ 0.1 เครื่องจักรตอบคำถามที่สองโดยเลือกว่าจะส่งออกตัวอักษร “H” หรือ “T” ด้วยความน่าจะเป็นที่ถูกกำหนดไว้จำเพาะสำหรับแต่ละสถานะซ้อนเร้นที่อยู่ในรอบนั้นๆ จากตัวอย่างของการโยนเหรียญในปัญหาคาสิโน ค่าความน่าจะเป็นในการส่งออกตัวอักษร “H” หรือ “T” เป็น 0.5 เท่ากันสำหรับสถานะซ้อนเร้น F และเป็น 0.75 และ 0.25 สำหรับสถานะซ้อนเร้น B *เป้าหมายของเราคือการอนุมานลำดับของสถานะการทำงานภายในของเครื่องจักรโดยการวิเคราะห์จากข้อมูลที่ส่งออกมา*

จากแนวคิดข้างต้นเราได้ทำการแปลงเจ้ามือที่เป็นมนุษย์ไปเป็นเครื่องจักรที่เรียกว่า แบบจำลองมาร์คอฟซ้อนเร้น (Hidden Markov Models: HMMs) ความแตกต่างอย่างเดียวระหว่างเครื่องจักรโยนเหรียญข้างต้นกับแบบจำลองมาร์คอฟซ้อนเร้นแบบทั่วไปคือแบบจำลองมาร์คอฟซ้อนเร้นทั่วไปไม่ได้จำกัดจำนวนสถานะซ้อนเร้น ไม่ได้จำกัดค่าความน่าจะเป็นในการย้ายจากสถานะซ้อนเร้นหนึ่งไปยังสถานะซ้อนเร้นอื่นๆ และไม่ได้จำกัดค่าความน่าจะเป็นที่ในการส่งออกผลแต่ละแบบของแต่ละสถานะซ้อนเร้น โดยทั่วไป HMM มีองค์ประกอบหลัก 4 ส่วนคือ Σ, States, Transition, และ Emission โดยถูกกำหนดไว้ดังต่อไปนี้

- Σ เป็นอักขระซึ่งเป็นผลที่แสดงออก
- States คือชุดของสถานะซ้อนเร้นทั้งหมด
- เมทริกซ์ขนาด |States| x |States| แสดงค่าความน่าจะเป็นในการเปลี่ยนจากสถานะซ้อนเร้นหนึ่งไปยังอีกสถานะซ้อนเร้นหนึ่ง (transition probability) โดยเมทริกซ์นี้เรียกว่าเมทริกซ์ทรานสิชัน (transition matrix) ค่าในเมทริกซ์เช่น $transition_{l,,k}$ แสดงค่าความน่าจะเป็นในการเปลี่ยนจากสถานะซ้อนเร้น *i* ไปยังสถานะซ้อนเร้น *j*
- เมทริกซ์ขนาด |States| x |Σ| แสดงค่าความน่าจะเป็นในการแสดงผลเป็นอักขระจำเพาะหนึ่งๆ ของแต่ละสถานะซ้อนเร้น (emission probability) โดยเมทริกซ์นี้เรียกว่าเมทริกซ์อิมิสชัน (emission matrix) ค่าในเมทริกซ์เช่น $emission_k(b)$ แสดงค่าความน่าจะเป็นในการแสดงผลเป็นอักขระ *b* ที่เป็นสมาชิกของ Σ เมื่อ HMM อยู่ในสถานะซ้อนเร้น *k*

และสำหรับแต่ละสถานะซ้อนเร้น *l*

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

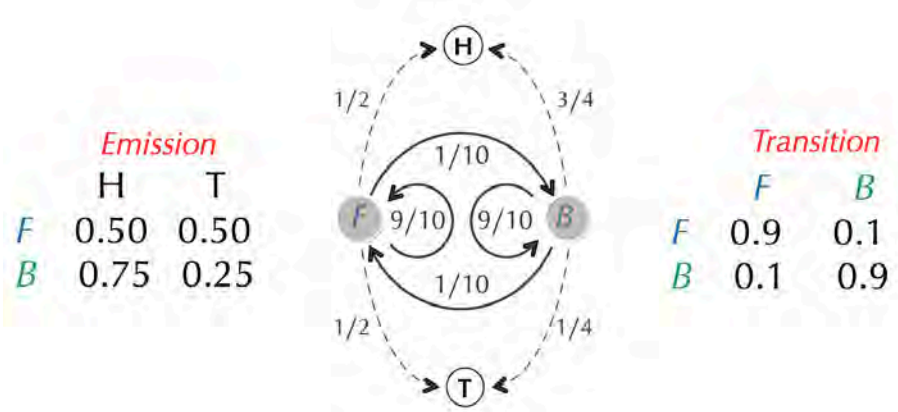
และ

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

ฝึกหัด	จงแสดงองค์ประกอบทั้งสี่ส่วนของ HMM ในการจำลองปัญหาคาสิโน

แผนภาพ HMM

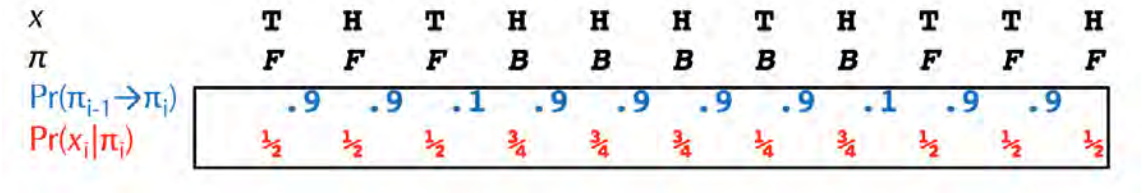
แบบจำลอง HMM สามารถแสดงในรูปแผนภาพ HMM หรือ HMM diagram ซึ่งในแผนภาพแต่ละโหนดที่เป็นสีเทาแสดงสถานะซ้อนเร้น เส้นเชื่อมทึบที่ชี้ออกจากโหนด *l* ไปยังอีกโหนด *k* แสดงการเปลี่ยนสถานะจากสถานะซ้อนเร้น *l* ไปยังสถานะซ้อนเร้น *k* โดยมีค่าความน่าจะเป็นในการเปลี่ยนสถานะแสดงบนเส้นเชื่อม นอกจากนี้โหนดที่เป็นเส้นประแสดงแต่อักขระที่สามารถแสดงออก โดยค่าบนเส้นประที่ชี้ออกจากโหนดสถานะซ้อนเร้นหนึ่งๆ มายังโหนดอักขระนี้แสดงค่าความน่าจะเป็นที่โหนดสถานะซ้อนเร้นนี้จะแสดงผลเป็นอักขระนี้ ในรูปที่ 6.5 ซึ่งจำลองปัญหาคาสิโนข้างต้น ประกอบด้วยสองสถานะซ้อนเร้นคือใช้เหรียญปกติ (F) กับใช้เหรียญถ่วงน้ำหนัก (B) โดยค่าความน่าจะเป็นในการเปลี่ยนสถานะเป็น 0.1 เท่ากันและค่าความน่าจะเป็นในการอยู่ในสถานะเดิมเป็น 0.9 และสถานะซ้อนเร้นเหรียญปกติมีค่าความน่าจะเป็นในการแสดงผลเป็นหัว (“H”) และก้อย (“T”) เท่ากันคือ 0.5 ในขณะที่สถานะซ้อนเร้นเหรียญถ่วงน้ำหนักค่าความน่าจะเป็นในการแสดงผลเป็นหัว (“H”) และก้อย (“T”) เป็น 0.75 และ 0.25 ตามลำดับ



รูปที่ 6.5 แผนภาพ HMM ที่จำลองปัญหาคาสิโน
(ที่มา: รูปที่ 10.5 ของ [21])

เส้นทางที่ซ้อนอยู่ (*hidden path*) $\pi = \pi_1\pi_{2\ldots}\pi_n$ ใน HMM คือลำดับของสถานะซ้อนเร้นที่ HMM ได้เดินผ่านในแต่ละรอบ โดยทางเดินในเส้นทางที่ซ้อนอยู่นี้คือชุดของเส้นทึบในแผนภาพ HMM นั่นเอง รูปที่ 6.6 แสดงตัวอย่างของเครื่องจักร HMM ที่ทำหน้าที่เป็นเจ้ามือขี้โกง โดย $Pr(\pi_{(i-1)} \longrightarrow \pi_i)$ แสดงค่าความน่าจะเป็นในการ

เปลี่ยนสถานะและ $Pr(x_i|\pi_i)$ แสดงค่าความน่าจะเป็นในการแสดงผลเป็น x_i โดยสถานะซ่อนเร้น π_i และผลที่แสดงออกคือลำดับการออกหน้าของเหรียญ x = “THTHHHTHTTH” โดยมีลำดับของการใช้เหรียญหรือลำดับของสถานะซ่อนเร้นเป็น π = *FFFBBBBBFFF* โดยเหรียญปกติถูกใช้ในการโยน 3 ครั้งแรกและ 3 ครั้งหลังสุดโดยอีก 5 ครั้งตรงกลางใช้เหรียญถ่วงน้ำหนัก



รูปที่ 6.6 ตัวอย่างลำดับการออกหน้าของเหรียญและสถานะซ่อนเร้นที่ถูกใช้ในแต่ละลำดับ

(ที่มา: รูปที่ 10.6 ของ [21])

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

กำหนดวิธีการแก้ปัญหาคาสิโนใหม่

จากปัญหาคาสิโนข้างต้นซึ่งมีเป้าหมายเพื่อหา π ที่สอดคล้องกับลำดับการออกหัวก้อย x มากที่สุด เราจะพิจารณาการแก้ปัญหานี้โดยใช้ HMM เป็นเครื่องมือ โดยเริ่มพิจารณาปัญหาง่ายกว่าคือให้หาค่าความน่าจะเป็น $\Pr(x, \pi)$ ที่ HMM จะใช้เส้นทาง $\pi = \pi_1\pi_{2\ldots}\pi_n$ และแสดงผลลำดับของอักขระเป็น $x = x_1x_{2\ldots}x_n$ โดยที่

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

สำหรับแต่ละสายสตริง x ที่แสดงออกมานั้นมีค่าความน่าจะเป็นเท่ากับ $\Pr(x)$ ซึ่งเป็นอิสระจากเส้นทางแสดงลำดับของสถานะซ่อนเร้นที่ถูกเลือกโดย HMM โดย

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

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

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

เหตุการณ์ที่ HMM ใช้เส้นทางแสดงลำดับของสถานะซ่อนเร้นและแสดงออกสายสตริง x ออกมานั้นสามารถมองได้ว่าเกิดจากการรวมสองเหตุการณ์เข้าด้วยกันคือ

- **เหตุการณ์ที่ 1:** HMM เลือกเส้นทางแสดงลำดับของสถานะซ้อนเร้น π ซึ่งมีค่าความน่าจะเป็นเท่ากับ $\Pr(\pi)$
- **เหตุการณ์ที่ 2:** HMM แสดงออกสายสตริง x โดยกำหนดให้ HMM ใช้เส้นทาง π โดยเราเรียกค่าความน่าจะเป็นในลักษณะนี้ว่าความน่าจะเป็นแบบมีเงื่อนไข (conditional probability) ของ x เมื่อกำหนด π ให้และแสดงสัญลักษณ์ได้เป็น $\Pr(x|\pi)$

โดยทั้งสองเหตุการณ์ข้างต้นจะต้องเกิดในกรณีที่ HMM ใช้เส้นทาง π และให้ผลเป็นสายสตริง x ซึ่งสามารถเขียนสมการได้เป็น

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

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

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

สำหรับค่าความน่าจะเป็นที่จะแสดงผลเป็นสายสตริง 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 ก็จะสามารถเขียนสมการใหม่ได้ดังต่อไปนี้

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

$$= \prod_{i=1}^{n} \Pr(x_i|\pi_i) \cdot \Pr(\pi_{i-1} \rightarrow \pi_i)$$

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

ฝึกหัด	จงคำนวณ $\Pr(x, \pi)$ สำหรับ x และ π ในรูปที่ 6.6 และคิดว่ามีเส้นทาง π อื่นที่ดีกว่าที่ไม่ใช่ FFFBBBBBFF ที่ให้ผลการแสดงออกเป็น x = “THTHHHTHTTH” หรือไม่ ถ้ามีเป็นเส้นทางไหน

หยุดคิด	หลังจากทราบองค์ประกอบหลักของ HMM แล้ว เราสามารถนำ HMM นี้ไปแก้ปัญหาการหา CG-islands ในจีโนมที่มีกล่าวถึงก่อนหน้าได้อย่างไร และมีข้อจำกัดอะไรบ้างหรือไม่

The Decoding Problem

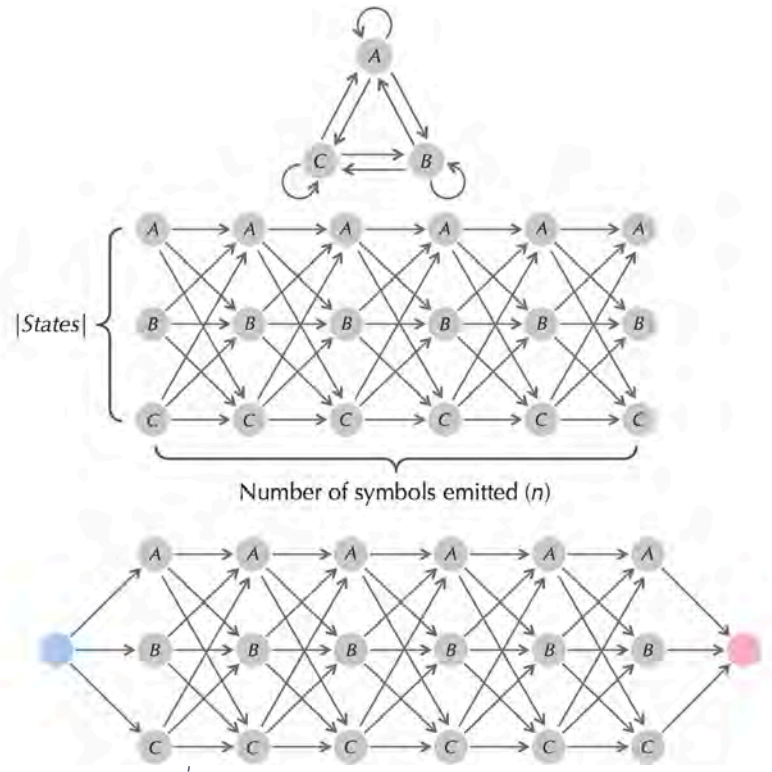
กราฟวิเทอบิ

ดังได้กล่าวในข้างต้นทั้งปัญหาคาสิโนและปัญหาการหา CG-islands ในจีโนม เป้าหมายคือการหาเส้นทางแสดงลำดับของสถานะซ่อนเร้น π ที่มีโอกาสแสดงผลออกเป็น x มากสุด หรืออีกนัยยะหนึ่งคือพยายามทำให้ได้ $\Pr(x, \pi)$ ที่มีค่ามากที่สุด

นิยามปัญหาที่ 6.2 ปัญหา Decoding

Decoding Problem หาเส้นทางแสดงลำดับของสถานะซ่อนเร้น π ที่ดีที่สุดใน HMM โดยให้ผลออกมาเป็นสายสตริง x	
ข้อมูลเข้า	สายสตริง $x = x_1x_2\ldots x_n$ ที่ส่งออกจาก HMM โดยที่ HMM ประกอบด้วย Σ, States, Transition และ Emission
ผลลัพธ์	เส้นทางแสดงลำดับของสถานะซ่อนเร้น π ที่ให้คะแนน $\Pr(x, \pi)$ ที่มีค่ามากที่สุด

ในปีค.ศ. 1967 แอนดรู วิเทอบิ (Andrew Viterbi) ใช้ HMM ที่มีการจัดรูปแบบให้เหมือนตาราง 2 มิติที่แสดงแผนที่ของเมืองแมนฮัตตัน (ดังตัวอย่างในบทที่ 5 เรื่องการเปรียบเทียบความคล้ายคลึงกันของสายดีเอ็นเอและหรือโปรตีน) ในการแก้ปัญหา Decoding สำหรับ HMM ที่มีการส่งออกสายข้อมูลเป็น $x = x_1x_2\ldots x_n$ เมื่อนำมาจัดให้อยู่ในรูปแบบกราฟวิเทอบิ (รูปที่ 6.7) สถานะซ่อนเร้นแต่ละสถานะจะถูกนำมาเรียงกันแถวละ 1 สถานะ ซึ่งหมายถึงจำนวนแถวจะเท่ากับ |States| และในแต่ละแถวจะประกอบด้วย n คอลัมน์ตามจำนวนของอักขระที่ HMM ส่งออก โหนด (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 $\text{emission}_k(x_i)$ ดังนั้นทุกเส้นทางที่เชื่อมต่อโหนดในคอลัมน์แรกของ



รูปที่ 6.7 (บน) แผนภาพ HMM ที่ประกอบด้วยสถานะซ้อนเร้น 3 สถานะ โดยไม่ได้แสดงค่าในส่วนของ Σ, Transition และ Emission (กลาง) HMM ข้างต้นที่ส่งออกสายข้อมูลเป็น x= x_1 x_2... x_n ถูกนำมาปรับให้อยู่ในรูปแบบกราฟวิเทอบิ (ล่าง) กราฟวิเทอบิที่มีการเพิ่มโหนดต้นทางและโหนดปลาย

(ที่มา: รูปที่ 10.7 ของ [21])

กราฟวิเทอบิไปยังโหนดในคอลัมน์สุดท้ายแสดง π ทั้งหมดที่เป็นไปได้ โดยค่าน้ำหนักของเส้นเชื่อมระหว่างโหนดแต่ละเส้นกำหนดโดย

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

และกำหนดผลคูณของค่าน้ำหนักของเส้นทาง 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 $\Pr(x,\pi)$ ที่แสดงข้างต้นเพียงอย่าง

เดียวคือสมการแสดงผลคูณค่าน้ำหนักยังไม่มีการคูณ $transition_{\pi_0,\pi_1} \cdot emission_{\pi_1}(x_1)$ ซึ่งเป็นการเปลี่ยนสถานะจากสถานะเริ่มต้น π_0 ไปยังสถานะ π_1 และส่งออกอักขระแรก เพื่อให้วิเทอบิกราฟมีสถานะเริ่มต้นด้วย จึงมีการเพิ่มโหนดตั้งต้น (source) ทางซ้ายสุดและทำการเชื่อมโหนดตั้งต้นนี้ไปยังทุกโหนดในคอลัมน์แรกโดยมีน้ำหนักของเส้นเชื่อมเท่ากับ $WEIGHT_0(source, k) = transition_{\pi_0,k} \cdot emission_k(x_1)$ นอกจากนี้เรายังสามารถเพิ่มโหนดปลายทาง (sink) และเพิ่มเส้นเชื่อมจากทุกโหนดให้คอลัมน์สุดท้ายไปยังโหนดปลายทางนี้โดยมีค่าน้ำหนักของเส้นเชื่อมเหล่านี้เท่ากับ 1 ด้วยกราฟวิเทอบินี้เราสามารถแก้ปัญหา Decoding โดยการเส้นทางในกราฟที่เชื่อมระหว่างโหนดต้นทางไปยังโหนดปลายทางที่ให้ผลคูณของค่าน้ำหนักมากที่สุด

ฝึกหัด	จงหาเส้นทาง π ที่ให้ผลคูณของค่าน้ำหนักมากที่สุดสำหรับสตริงที่ส่งออกจาก HMM เป็น x = “HHTT”

อัลกอริทึมวิเทอบิ

เราสามารถใช้ไดนามิกโปรแกรมมิงในการแก้ปัญหา Decoding ในหัวข้อที่ผ่านมา โดยกำหนดให้ $s_{k,i}$ เป็นผลคูณของค่าน้ำหนักที่มากที่สุดจากโหนดต้นทางมายังโหนด (k,i) โดยอัลกอริทึมวิเทอบิ (Viterbi algorithm) อาศัยสมมติฐานว่าเส้นเชื่อมจำนวน i-1 เส้นแรกของเส้นทางที่ดีที่สุดจากโหนดต้นทางมายังโหนด (k,i) นั้น มาจากเส้นทางที่ดีที่สุดจากโหนดต้นทางมายังโหนด (l, i-1) สำหรับสถานะซ้อนเร้น l ใดๆ ซึ่งสมมติฐานนี้ทำให้เราสามารถสร้างสมการแสดงความสัมพันธ์เวียนเกิดต่อไปนี้

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

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

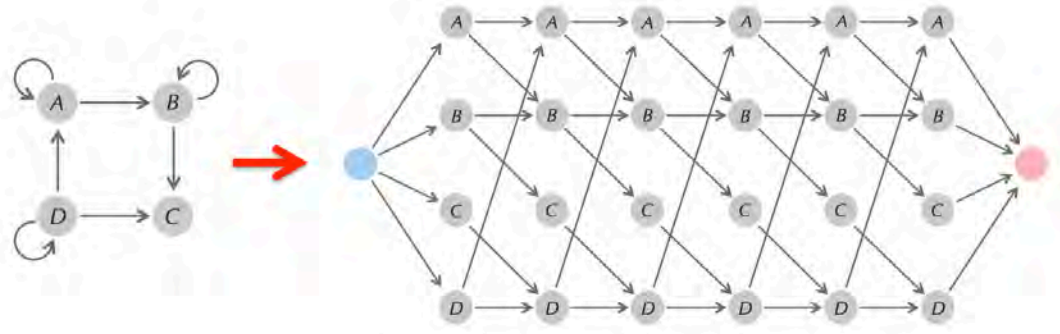
$$s_{sink} = \max_{all\ states\ l} s_{l,n}$$

ประสิทธิภาพของอัลกอริทึมวิเทอบิ

เราสามารถแก้ปัญหา Decoding โดยจำลองปัญหาเป็นการหาเส้นทางที่ยาวที่สุดในกราฟที่มีทิศทาง (Direct Acyclic Graph: DAG) แบบเดียวกับปัญหาการเปรียบเทียบความเหมือนของสายดีเอ็นหรือโปรตีนในบทที่ 5 โดยการพยายามหาเส้นทางที่มีผลคูณของค่าน้ำหนักที่มากที่สุดในสมการ $\prod_{i=1}^{n} WEIGHT_i(\pi_{i-1}, \pi_i)$ ซึ่งถ้ามีการใส่ลอการิทึมสำหรับผลคูณในสมการ จะสามารถเปลี่ยนรูปของสมการเป็น $\sum_{i=1}^{n} \log(WEIGHT_i(\pi_{i-1}, \pi_i))$ ซึ่งถ้าทำการแทนค่าน้ำหนักของแต่ละเชื่อมด้วยค่าลอการิทึมของน้ำหนักก็จะสามารถหาเส้นทางที่ดีที่สุดโดยหาเส้นทางที่มีผลบวกมากที่สุด ทั้งนี้เวลาที่ใช้ในการทำงานของอัลกอริทึมวิเทอบิจะเป็นสมการเส้นตรงแปรผันตามจำนวนเส้นเชื่อมในกราฟโดยมีค่าเป็น $O(|States^2| \cdot n)$ โดย n คือจำนวนอักขระที่ส่งออก

ฝึกหัด	จงประยุกต์ใช้ HMM ที่ใช้ในการแก้ปัญหา Decoding ข้างต้น เพื่อแก้ปัญหาการหา CG-islands ใน 1 ล้านแรกของโครโมโซม X ของมนุษย์ โดยสมมติว่าการเปลี่ยนสถานะจาก CG-islands ไปเป็น non CG-islands เกิดขึ้นน้อยที่ค่าความน่าจะเป็นเท่ากับ 0.001 ในขณะที่การเปลี่ยนสถานะจาก non CG-islands มาเป็น CG-islands ยิ่งน้อยกว่าที่ความน่าจะเป็น 0.0001 จาก 1 ล้านนิวคลีโอไทด์ที่เป็นข้อมูลเข้า พบ CG-islands ทั้งหมดกี่บริเวณ

การเปลี่ยนสถานะใน HMM จากสถานะซ้อนเร้นหนึ่งไปยังอีกสถานะซ้อนเร้นหนึ่งในทางปฏิบัติอาจไม่มีทางเกิดขึ้นจริง (forbidden transitions) ดังนั้นเส้นเชื่อมระหว่างสถานะเหล่านี้จะถูกลบออกไปจากแผนภาพ HMM และกราฟวิเทอบิดังแสดงในรูปที่ 6.8 (ซ้าย) ที่ไม่มีการเปลี่ยนสถานะจาก A ไป D หรือการเปลี่ยนสถานะจาก C ไปยัง C เอง เส้นเชื่อมเหล่านี้จะถูกนำออกไปจากแผนภาพกราฟวิเทอบิที่จำลองจากแผนภาพ HMM ทางขวาทั้งสิ้น 6 อักขระ (รูปที่ 6.8 ขวา) ซึ่งการลดจำนวนเส้นเชื่อมที่ไม่เกิดขึ้นจริงนี้จะทำให้อัลกอริทึมวิเทอบิทำงานได้เร็วขึ้น



รูปที่ 6.8 (ซ้าย) แผนภาพ HMM ที่ประกอบด้วยสถานะซ้อนเร้น 4 สถานะและมีการเปลี่ยนสถานะเพียงบางแบบ (ขวา) การลดเส้นเชื่อมในกราฟวิเทอบิที่ไม่มีทางเกิดขึ้น

(ที่มา: รูปที่ 10.8 ของ [21])

การหาสายข้อมูลส่งออกที่มีโอกาสเกิดขึ้นมากที่สุด

ในหัวข้อที่ผ่านเราศึกษาว่าไดนามิกโปรแกรมมิงช่วยในการหาค่า $\Pr(\pi)$ ที่มีค่ามากที่สุดได้อย่างไร ในหัวข้อนี้เราสนใจว่าค่าความน่าจะเป็นที่ HMM จะส่งออกสตริง x หนึ่งๆ เป็นเท่าไหร่หรือเพื่อคำนวณหา $\Pr(x)$ นั่นเอง

ฝึกหัด	ในปัญหาคาสิโนก่อนหน้า จะมีโอกาสพบสายสตริงส่งออกสายไหนมากกว่ากันระหว่าง “HHTT” และ “HTHT”

นิยามปัญหาที่ 6.3 ปัญหาการหาความน่าจะเป็นที่ HMM จะส่งออกสายสตริงหนึ่งๆ

ปัญหาการหาความน่าจะเป็นที่ HMM จะส่งออกสายสตริงหนึ่งๆ (Outcome Likelihood Problem) หาค่าความน่าจะเป็นที่ HMM จะส่งออกสายสตริงหนึ่งๆ	
ข้อมูลเข้า	สายสตริง $x = x_1x_2\ldots x_n$ ที่ถูกส่งออกจาก HMM ที่ประกอบด้วย Σ, States, Transition และ Emission
ผลลัพธ์	ค่าความน่าจะเป็น $\Pr(x)$ ที่ HMM จะส่งออกสายสตริง x

หยุดคิด	เราสามารถดัดแปลงสมการแสดงความสัมพันธ์เวียนเกิดของวิเทอบิต่อไปนี้ ในการแก้ปัญหา 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 อย่างไรก็ตามจำนวนของเส้นทางจะเพิ่มมากขึ้นแบบเอกซ์โปเนนเชียลตามจำนวนอักขระที่ส่งออกในสตริง x ดังนั้นการเลือกใช้ไดนามิกโปรแกรมมิงก็จะเป็นแนวทางที่ดีกว่าในการคำนวณ $\Pr(x)$

กำหนดให้ผลรวมของผลคูณของทุกเส้นทางจากโหนดตั้งต้น (source) ถึงโหนด (k,i) ในกราฟวิเทอบิเป็น $forward_{k,i}$ และ $forward_{sink}$ เท่ากับ $\Pr(x)$ ในการคำนวณ $forward_{k,i}$ เราแบ่งเส้นทางทั้งหมดที่เชื่อมจากโหนดตั้งต้น source ถึง (k, i) ออกเป็น |States| ซับเซ็ต โดยแต่ละซับเซ็ตก็จะมีเส้นทางที่ผ่านโหนด (l, i-1) ซึ่งมีผลรวมของผลคูณเป็น $forward_{l,i-1}$ ก่อนที่จะมาถึงโหนด (k, i) สำหรับบาง l ที่เป็นสถานะซ้อนเร้นใดๆ 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

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

คือสัญลักษณ์แสดงการหาค่าที่มากที่สุดในอัลกอริทึมวิเทอบิก่อนหน้ากลายมาเป็นสัญลักษณ์แสดงการหาผลรวมในสมการนี้ ซึ่งเราสามารถแก้ปัญหา Outcome Likelihood โดยการคำนวณ $forward_{sink}$ ซึ่งแสดงโดยสมการต่อไปนี้

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

จากสมการนี้เราสามารถคำนวณค่าความน่าจะเป็น $\Pr(x)$ ในการส่งออกสายสตริง x หนึ่งๆ โดยถ้าต้องการหาสายสตริงส่งออกที่มีโอกาสเกิดขึ้นมากที่สุด เราสามารถนิยามปัญหาได้ดังต่อไปนี้

นิยามปัญหาที่ 6.4 ปัญหาการหาสายสตริงที่มีโอกาสถูกส่งออกมากที่สุด

ปัญหาการหาสายสตริงที่มีโอกาสถูกส่งออกมากที่สุด (Most Likely Outcome Problem) หาสายสตริงที่มีโอกาสถูกส่งออกมากที่สุด	
ข้อมูลเข้า	โมเดล HMM ที่ประกอบด้วย Σ, States, Transition และ Emission และจำนวนเต็ม n
ผลลัพธ์	สายสตริง $x = x_1x_2\ldots x_n$ ที่ถูกส่งออกจาก HMM โดยเป็นสายสตริงที่ทำให้ค่าความน่าจะเป็น $\Pr(x)$ มากสุด

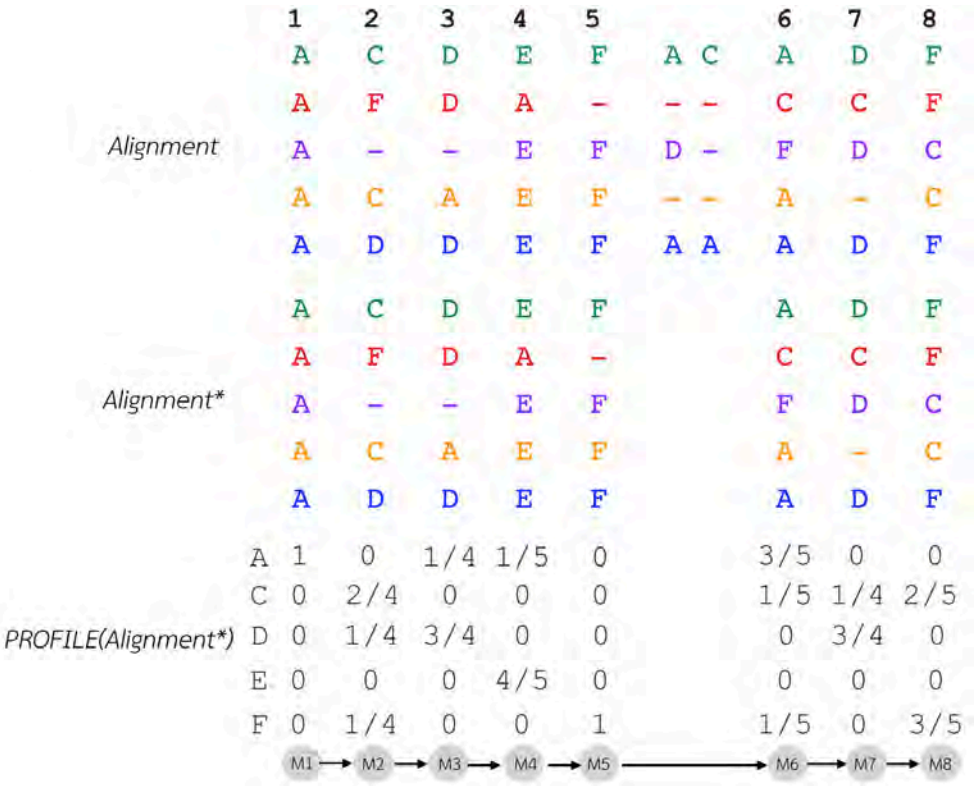
การสร้างโปรไฟล์ HMM เพื่อใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นและหรือโปรตีน

HMMs เกี่ยวข้องกับการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนอย่างไร

ถึงจุดนี้หลายๆคนยังอาจสงสัยว่า HMM มีความเกี่ยวข้องกับโจทย์ตั้งต้นของบทเรียนที่กล่าวถึงการเปรียบเทียบความคล้ายคลึงกันระหว่างโปรตีนในส่วนที่เป็น V3 loop ของโปรตีน gp120 อย่างไร ซึ่งในหัวข้อนี้จะอธิบายว่า HMM จะสามารถเข้ามาแก้ปัญหาการเปรียบเทียบสายโปรตีนในโจทย์นี้ได้ดีกว่าได้อย่างไร ถ้าเรามีสายของโปรตีนที่อยู่ในแฟมิลีเดียวกัน เราจะสามารถตรวจสอบได้ว่าสายโปรตีนที่เข้ามาใหม่อยู่ในแฟมิลีนี้ด้วยหรือไม่โดยการเปรียบเทียบความคล้ายคลึงกันระหว่างสายโปรตีนที่เข้ามาใหม่กับแต่ละสายโปรตีนที่อยู่ในแฟมิลี ถ้าผลของการเปรียบเทียบกับอย่างน้อยหนึ่งสายโปรตีนในแฟมิลีผ่านเกณฑ์คะแนนความเหมือน ก็สามารถอนุมานได้ว่าโปรตีนเส้นใหม่นี้น่าจะอยู่ในแฟมิลีนี้ อย่างไรก็ตามการเปรียบเทียบในแนวทางนี้จะให้คำตอบที่ไม่ถูกต้องถ้าโปรตีนที่นำมาเปรียบเทียบนั้นแตกต่างกันค่อนข้างมากอย่างในกรณี V3 loop ของโปรตีน gp120 ที่ได้มาจากเชื้อเอชไอวีที่แยกออกมาในแต่ละครั้งซึ่งอาจมาจากผู้ป่วยคนเดียวกัน

จากรูปที่ 6.9 (บน) แสดงการเปรียบความคล้ายคลึงกันระหว่างส่วนของโปรตีน 5 เส้น โดยแต่ละเส้นมีความยาวของสายโปรตีนเท่ากับ 10 กรดอะมิโน โดยคอลัมน์ที่ 6 และ 7 ในรูปนี้มีสัญลักษณ์ที่แสดงช่องว่าง (‘-’) หลายบรรทัด ซึ่งไม่น่ามีความหมายเชิงการเกิดความอนุรักษ์ร่วมกันภายในแฟมิลี ทำให้นักชีววิทยามักตัดคอลัมน์เหล่านี้ออกจากผลการเปรียบเทียบความคล้ายคลึงกันระหว่างสายข้อมูลภายในชุด ถ้าคอลัมน์มีจำนวนบรรทัดที่มีช่องว่างมากกว่าหรือเท่ากับค่า column removal threshold (θ) ดังแสดงในรูปที่ 6.9 (กลาง) และผลในรูปนี้เรียกว่า seed alignment ซึ่งเราคาดหวังว่าเราจะสามารถสร้าง HMM ได้จาก seed alignment และโปรไฟล์เมทริกซ์ของมัน ดังแสดงในรูปที่ 6.9 (ล่าง) ดังนั้นแทนการเทียบ seed alignment กับสายโปรตีนเส้นใหม่ (Text) เราจะคำนวณค่าความน่าจะเป็นที่ HMM จะส่งออกสายโปรตีน Text นี้แทน โดยถ้า HMM ถูกออกแบบไว้ดีแล้ว Text ที่มีความคล้ายกับสายข้อมูลใน Alignment* มากกว่า ก็มีโอกาสที่จะถูกส่งออกโดย HMM ได้มากกว่า

เราสามารถสร้าง HMM จากลำดับของคอลัมน์ของ *Alignment** ในรูปที่ 6.9 (กลาง) โดย HMM จะประกอบด้วยสถานะแมชที่เรียงต่อกัน k สถานะ โดยเมื่อ HMM เข้าสู่สถานะ MATCH(i) จะส่งออกอักขระ x_i โดยมี



รูปที่ 6.9 (บน) ผลการเปรียบเทียบความคล้ายคลึงกันระหว่างสายโปรตีน 5 เส้นในชุด (กลาง) ผลการเปรียบเทียบความคล้ายคลึงกันระหว่างสายโปรตีน 5 เส้นในชุดโดยตัดคอลัมน์ที่มี ‘-’ มากกว่าค่า (θ =0.35) ออก (ล่าง) HMM ที่แสดงสถานะแมช (match)

(ที่มา: รูปที่ 10.9 ของ [21])

ค่าความน่าจะเป็นเท่ากับค่าความถี่ของการเกิดอักขระนั้นๆในคอลัมน์ที่ i ของ PROFILE(*Alignment**) โดย HMM มีค่าความน่าจะเป็นในการเปลี่ยนจากสถานะที่ i ไปยังสถานะที่ i+1 เท่ากับ 1 โดยค่าคะแนนความคล้ายคลึงกันระหว่าง *Alignment** กับ Text ที่เป็นสายโปรตีนเส้นใหม่ มีค่าเท่ากับค่าความน่าจะเป็น $\Pr(Text)$ ที่ HMM จะส่งออก Text และคะแนนนี้มีค่าเท่ากับผลคูณของค่าความถี่ของอักขระที่แมชในแต่ละคอลัมน์ ใน PROFILE (*Alignment**) ตัวอย่างเช่นค่าความน่าจะเป็นที่ HMM ในรูปที่ 6.9 จะส่งออกสายอักขระ **ADDAFFDF** เป็น

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

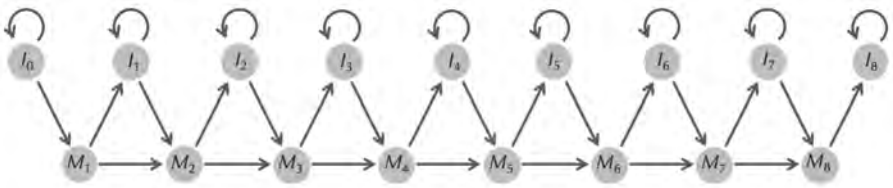
หยุดคิด	อะไรคือข้อจำกัดของ HMM ในรูปที่ 6.9

ถึงแม้ว่า HMM ข้างต้นจะมีการให้คะแนนในแต่ละคอลัมน์แตกต่างกัน โดย Text ที่มีความคล้ายคลึงกับ *Alignment** มากจะมีค่าคะแนนความคล้ายคลึงมากกว่า Text ที่แตกต่างจาก *Alignment** อย่างไรก็ตาม HMM ข้างต้นยังขาดส่วนที่เป็นหัวใจของ HMM เนื่องจากมีเส้นทางของสถานะซ้อนเร้นเพียง 1 เส้นทาง และไม่สามารถใช้แสดงแบบจำลองของการเกิด insertion และ deletion รวมทั้งจะสามารถใช้ HMM กับข้อมูลเข้าที่มีความยาวของสายโปรตีนเท่ากับจำนวนคอลัมน์ใน *Alignment** เท่านั้น

การสร้างโปรไฟล์ HMM

เพื่อกำจัดข้อจำกัดของ HMM ข้างต้น ได้มีการเสนอ HMM ในรูปแบบที่เรียกว่าโปรไฟล์ HMM (profile HMM) โดยถ้าข้อมูลเข้าคือผลของการเปรียบเทียบความคล้ายคลึงกันของชุดของสายดีเอ็นเอหรือโปรตีน *Alignment* และมีการลบคอลัมน์ที่มีค่าเป็นช่องว่าง (‘-’) โดยอาศัยค่า θ ข้างต้น ก็จะได้เป็นผลใหม่คือ *Alignment** ซึ่งเรียกว่า seed alignment โดยโปร์ไฟล์ HMM จะถูกสร้างจากข้อมูล seed alignment และแสดงโดย HMM(*Alignment**) โดยถ้ามีข้อมูลเข้าเป็นโปรตีนสายใหม่ Text เป้าหมายของเราคือการหาเส้นทางแสดงลำดับสถานะซ้อนเร้นที่ให้ค่าความน่าจะเป็นที่มากที่สุด โดยการแก้ปัญหา Decoding สำหรับโปรไฟล์ HMM ที่ส่งออกสายอักขระ Text นั่นเอง

จากโปรไฟล์ HMM ที่แสดงเฉพาะลำดับของสถานะซ้อนเร้นแมชในรูปที่ 6.9 เพื่อให้สามารถนำ Text ที่มีความยาวที่แตกต่างมาเปรียบเทียบได้ เราจะต้องทำการเพิ่มสถานะซ้อนเร้นอื่นๆนอกเหนือจากสถานะแมชจำนวน k สถานะข้างต้น โดยขั้นแรกทำการเพิ่มสถานะ insertion จำนวน k+1 สถานะแสดงโดย INSERTION(0),…INSERTION(k) (รูปที่ 6.10) การเพิ่มสถานะ INSERTION(i) นี้ทำให้โปรไฟล์ HMM สามารถส่งออกอักขระเพิ่มเติมหลังจากผ่านคอลัมน์ที่ i ของ PROFILE(*Alignment**) และก่อนเข้าสู่สถานะที่ i+1 ดังนั้นเราจะต้องลากเส้นเชื่อมจาก MATCH(i) ไปยัง INSERTION(i) และจาก INSERTION(i) ไปยัง MATCH(i+1) นอกจากนี้



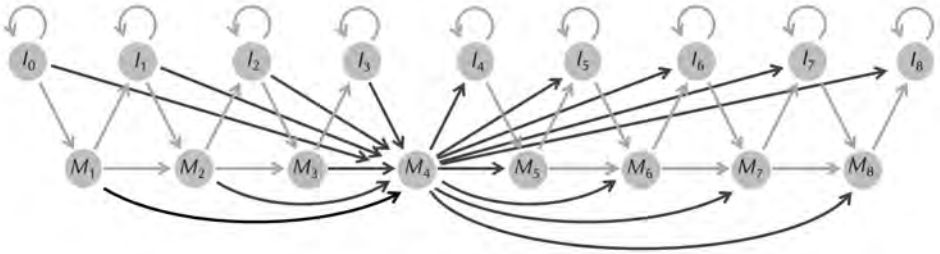
รูปที่ 6.10 แผนภาพ HMM ที่มีการเพิ่มสถานะซ้อนเร้น insertion จำนวน k+1 สถานะ จากรูปที่ 6.9

(ที่มา: รูปที่ 10.11 ของ [21])

เพื่อให้สามารถแทรกอักขระได้มากกว่า 1 ตัวในระหว่างคอลัมน์ใน PROFILE(*Alignment**) จะต้องเพิ่มเส้นเชื่อมที่โหนด INSERTION(i) จะชี้เข้าตัวเองด้วย

หยุดคิด	เราสามารถใช้ HMM ในรูปที่ 6.10 ในการเปรียบเทียบความคล้ายคลึงกับสายโปรตีนเข้าที่ยาวน้อยกว่า 8 กรดอะมิโนได้หรือไม่

หลังจากมีการปรับปรุง HMM ให้สามารถรองรับการเปรียบเทียบกับสายโปรตีนเข้าที่ยาวกว่าจำนวนลำดับกรดอะมิโน k ใน HMM ตั้งต้น โดยการเพิ่มสถานะซ้อนเร้น insertions เข้ามา ในส่วนนี้จะมีการปรับปรุง HMM เพิ่มเติมให้รองรับสายข้อมูลเข้าที่อาจเกิด deletions โดยจะอนุญาตให้โปรไฟล์ HMM สามารถข้ามบางคอลัมน์ใน PROFILE(*Alignment**) ไป ทั้งนี้วิธีการหนึ่งที่เป็นไปได้ในการสร้าง HMM ใหม่นี้คือการเพิ่มเส้นเชื่อมจากแต่ละสถานะที่มีอยู่ไปยังสถานะที่อยู่ทางขวาอื่นๆทั้งหมดดังตัวอย่างที่แสดงในรูปที่ 6.11

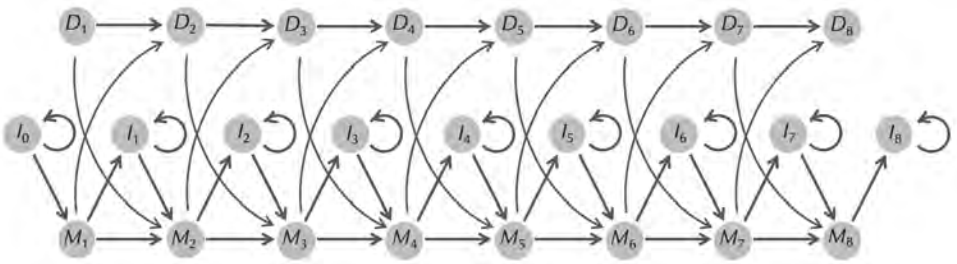


รูปที่ 6.11 ตัวอย่างของการปรับ HMM เพื่อให้รองรับสถานะ deletions โดยการลากเส้นเชื่อมเพิ่มเติมจากสถานะหนึ่งๆ ไปยังสถานะอื่นๆทั้งหมดทางขวา โดยใน แผนภาพ HMM นี้ แสดงเส้นเชื่อมเพิ่มเติม (สีดำ) ทั้งหมดที่ชี้เข้าและชี้ออกจากโหนด MATCH(4)

(ที่มา: รูปที่ 10.12 ของ [21])

หยุดคิด	จากตัวอย่างของการเพิ่มเส้นเชื่อม (สีดำทั้งหมด) รวมทั้งที่ชี้เข้าและออกจากโหนด MATCH(4) เพื่อรองรับการเกิด deletions ถ้าให้ลากเส้นเชื่อมจนครบจะมีเส้นเชื่อมเพิ่มเติมทั้งหมดกี่เส้น

จากตัวอย่างการปรับ HMM เพื่อให้รองรับการเกิด deletions ข้างต้น จะมีจำนวนเส้นเชื่อมที่ต้องลากเพิ่มขึ้นมาก คำถามคือเราจะสามารถลดจำนวนเส้นเชื่อมเหล่านี้ลงได้ไหม อย่างไร รูปที่ 6.12 แสดงผลของการปรับ HMM เพื่อรองรับการเกิด deletions ในแนวทางของการเพิ่มสถานะซ้อนเร้นใหม่เรียกว่าสถานะ deletion จำนวน k สถานะ ซึ่งแทนด้วย DELETION(i),..., DELETION(k) ตามรูปที่ 6.12 ตัวอย่างเช่น แทนการลากเส้นเชื่อมจากโหนด MATCH(i-1) ไปยังโหนด MATCH(i+1) เราสามารถลากเส้นเชื่อมจากในเส้นทาง MATCH(i-1)→ DELETION(i) → MATCH(i+1) ทั้งนี้การเข้าสู่สถานะ DELETION(i) จะอนุญาตให้ HMM ข้ามบางคอลัมน์ไปโดยไม่มีการส่งออกอักขระในคอลัมน์นั้นๆ



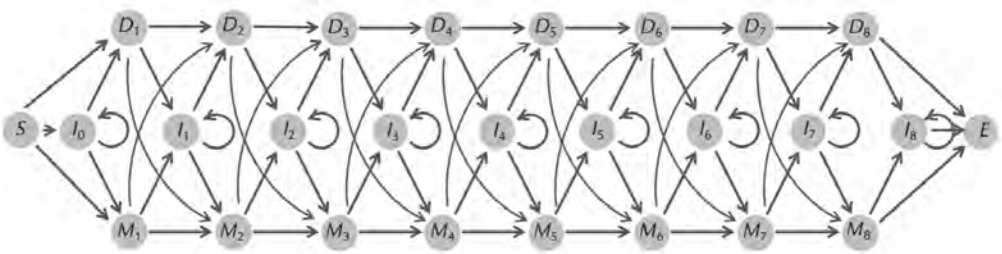
รูปที่ 6.12 ตัวอย่างการปรับ HMM โดยการเพิ่มสถานะ deletion (ตัวย่อคือ D_i)

(ที่มา: รูปที่ 10.13 ของ [21])

หยุดคิด	จาก HMM ในรูปที่ 6.12 คิดว่า HMM นี้เพียงพอหรือมีความครบถ้วนหรือยังในการจำลองการเกิดทั้ง insertions และ deletions

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

หยุดคิด	ลองพิจารณาคำถามต่อไปนี้ 1. แผนภาพ HMM ในรูปที่ 6.13 มีจำนวนเส้นเชื่อมทั้งหมดเท่าไหร่ และแตกต่างจากจำนวนเส้นเชื่อมทั้งหมดในรูปที่ 6.11 อย่างไร 2. แผนภาพ HMM ในรูปที่ 6.13 นี้มีกราฟวิเทอบิหน้าตาอย่างไร และมีจำนวนโหนดและเส้นเชื่อมจำนวนเท่าไหร่



รูปที่ 6.13 โปรไฟล์ HMM ที่ถูกปรับปรุงเสร็จเรียบร้อยแล้ว โดยเพิ่มส่วนที่รองรับการเปลี่ยนสถานะระหว่าง insertion และ deletion รวมทั้งมีการเพิ่มโหนดตั้งต้น S และโหนดปลายทาง E เข้ามาด้วย

(ที่มา: รูปที่ 10.14 ของ [21])

ค่าความน่าจะเป็น Transition และ Emission ของโปรไฟล์ HMM

รูปที่ 6.14 แสดงเส้นทางในโปรไฟล์ HMM ของลำดับกรดอะมิโนในแต่ละบรรทัดของ *Alignment* ในรูปที่ 6.9 โดยแต่ละเส้นทางมีการแยกสีตามบรรทัดของสายข้อมูลโปรตีน กรดอะมิโนที่อยู่ใน seed alignment (*Alignment**) (ไม่รวมคอลัมน์ที่มีพื้นหลังสีเทา) อาจอยู่ในสถานะแมช (เป็นอักขระแสดงกรดอะมิโน) หรือสถานะ deletion (เป็นอักขระ '-') สำหรับอักขระที่ไม่ได้อยู่ใน seed alignment (คอลัมน์ที่มีพื้นหลังเป็นสีเทา) ถ้าเป็นอักขระ '-' จะไม่ถูกนำมาพิจารณา แต่ถ้าเป็นอักขระอื่นจะหมายถึงอักขระที่ถูกส่งออกโดยสถานะ insertion หนึ่งๆ

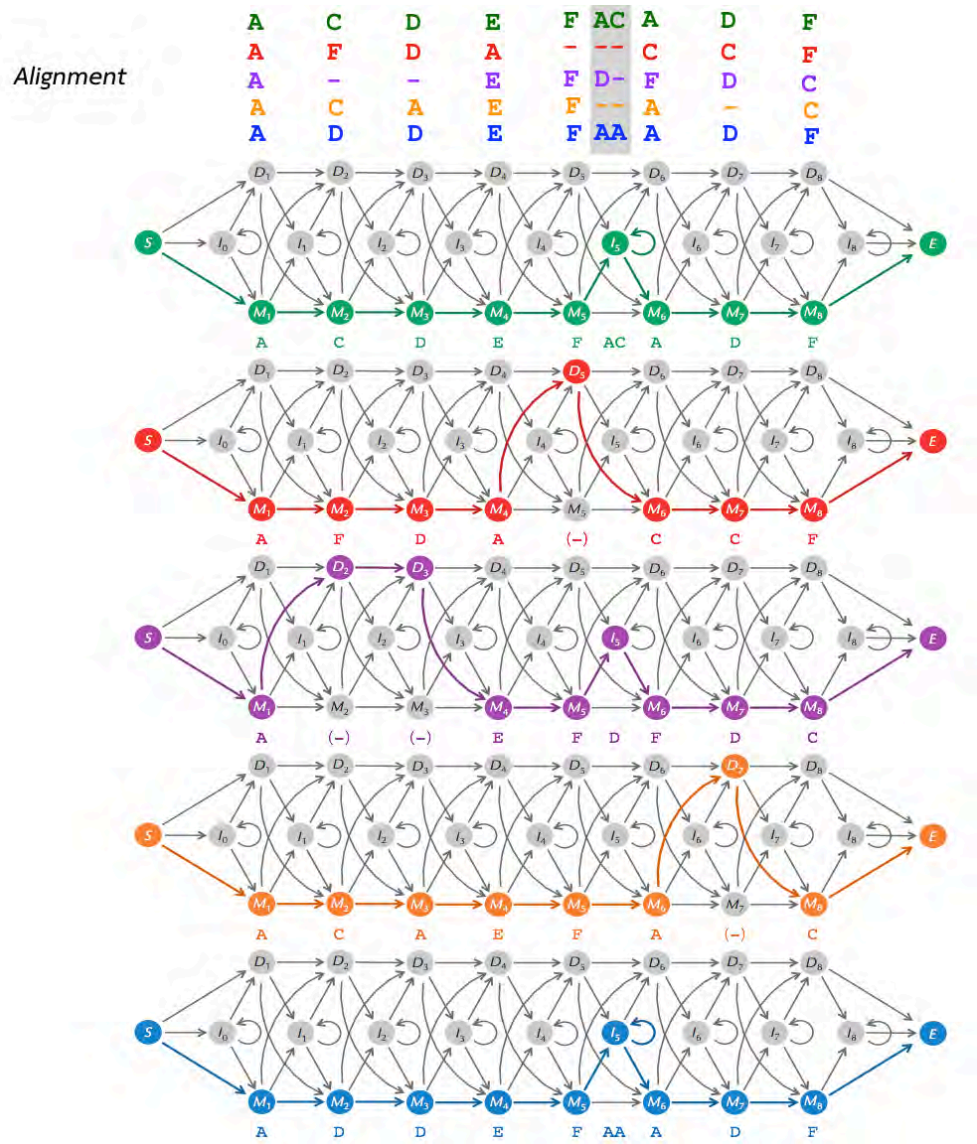
หยุดคิด	เราจะกำหนดค่าความน่าจะเป็นในการเปลี่ยนจากสถานะหนึ่งไปยังอีกสถานะหนึ่ง (transition) และค่าความน่าจะเป็นในการส่งออกอักขระหนึ่งๆ (emission) ในแต่ละลำดับจำเพาะของโปรไฟล์ HMM ในรูปที่ 6.14 ได้อย่างไร

เราสามารถกำหนดค่าความน่าจะเป็นในการเปลี่ยนสถานะ $\text{transition}_{l,k}$ ได้จากการนับความถี่ในการเปลี่ยนจากสถานะ l ไปยังสถานะ k ในห้าเส้นทางในรูปที่ 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 เส้นทางในโปรไฟล์ HMM ที่แสดงลำดับกรดอะมิโนในแต่ละบรรทัดของ Alignment ในรูปที่ 6.9 อักขระ '-' ภายในวงเล็บใต้แต่ละแผนภาพ HMM แสดงถึงสถานะ deletion ซึ่งจะไม่มีการส่งออกอักขระโดย HMM

(ที่มา: รูปที่ 10.15 ของ [21])

และในรูปที่ 6.14 นี้ความน่าจะเป็นที่จะเปลี่ยนจากสถานเริ่มต้น (initial state: S) มาเป็น MATCH(1) เท่ากับ 1 สำหรับกรณีโปรไฟล์ HMM ทั่วไป สถานเริ่มต้นยังสามารถเข้าสู่สถานะ INSERTION(0) และ DELETION(1) ได้อีกสองสถานะ

เราสามารถกำหนดค่าความน่าจะเป็นในการส่งออกอักขระ $\text{emission}_k(b)$ โดยการหารจำนวนอักขระ b ที่ส่งออกโดยสถานะ k ด้วยจำนวนอักขระที่ถูกส่งออกทั้งหมดโดยสถานะ k ในตัวอย่างรูปที่ 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 ดังนั้นเราสามารถอนุมานค่าความน่าจะเป็นในการส่งออกอักขระใดๆในสถานะจำเพาะหนึ่งๆ ดังต่อไปนี้

$$emission_{INSERTION(5)}(A) = \frac{3}{5} \qquad emission_{MATCH(2)}(A) = 0$$
$$emission_{INSERTION(5)}(C) = \frac{1}{5} \qquad emission_{MATCH(2)}(C) = 2/4$$
$$emission_{INSERTION(5)}(D) = \frac{1}{5} \qquad emission_{MATCH(2)}(D) = 1/4$$
$$emission_{INSERTION(5)}(E) = 0 \qquad emission_{MATCH(2)}(E) = 0$$
$$emission_{INSERTION(5)}(F) = 0 \qquad emission_{MATCH(2)}(F) = 1/4$$

ถึงจุดนี้เราก็พร้อมที่จะสร้างโปรไฟล์ HMM เพื่อใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายของโปรตีนจากผลของ multiple alignment ใดๆ

นิยามปัญหาที่ 6.5 ปัญหาโปรไฟล์ HMM

ปัญหาโปรไฟล์ HMM (Profile HMM Problem) สร้างโปรไฟล์ HMM จากผล multiple alignment	
ข้อมูลเข้า	ผล multiple alignment *Alignment* และค่า column removal threshold (θ)
ผลลัพธ์	HMM(*Alignment,* θ)

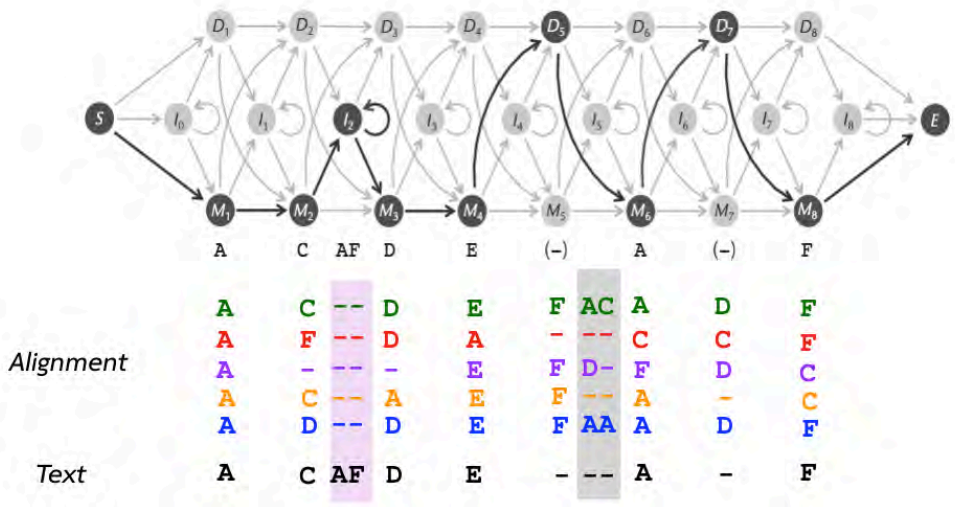
ฝึกหัด	สร้างโปรไฟล์ HMM โดยใช้ชุดของลำดับกรดอะมิโนในตำแหน่ง V3 loop ของโปรตีน gp120 จากเชื้อเอชไอวีในรูปที่ 6.2

การจำแนกโปรตีนโดยใช้โปรไฟล์ HMM

การเทียบสายโปรตีนกับโปรไฟล์ HMM

ถ้ามีข้อมูลชุดของสายโปรตีนในแฟมิลีหนึ่งที่มีผลการเปรียบเทียบความคล้ายคลึงกันในรูปแบบ *Alignment* เราสามารถกลับมาพิจารณาปัญหาในการตัดสินว่าสายโปรตีนเข้าสายใหม่ Text นั้นน่าจะเป็นสมาชิกของแฟมิลีนี้หรือไม่ เพื่อตอบคำถามนี้ในขั้นแรกเราต้องสร้าง HMM(*Alignment,* θ) รูปที่ 6.15 แสดงเส้นทางลำดับของสถานะที่สอดคล้องกับผลการเทียบสายโปรตีนเข้าสายใหม่ Text กับ *Alignment* โดยสองอักขระแรกใน Text อยู่ในสถานะแมช สองอักขระถัดไปอยู่ในสถานะ insertion และมีคอลัมน์เป็นของตัวเอง (คอลัมน์พื้นหลังสีชมพู) อักขระ '-' ในคอลัมน์ที่ 7 และ 11 แสดงสถานะ deletion ซึ่งไม่มีการแสดงอักขระใดโดย HMM ส่วนอักขระ '-' ใน

คอลัมน์พื้นหลังสีเทาไม่ได้นำมาพิจารณาเนื่องจากถูกตัดออกตั้งแต่ตอนสร้าง HMM ตามเงื่อนไข column removal threshold (θ)



รูปที่ 6.15 (บน) เส้นทางผ่าน HMM(Alignment, 0.35) สร้างจาก multiple alignment ในรูปที่ 6.9 และส่งออกสายของอักขระ Text = ACAFDEAF (ล่าง) สายอักขระที่ถูกส่งออกโดยการเทียบ Text กับ Alignment

(ที่มา: รูปที่ 10.17 ของ [21])

ในการเทียบ Text กับ *Alignment* เราสามารถใช้อัลกอริทึมวิเทอบิเพื่อหาเส้นทางที่ดีที่สุดที่ส่งออก Text จาก HMM(*Alignment*, θ) หรือถ้าผลคูณของค่าน้ำหนักคะแนนในเส้นทางหนึ่งๆ มีค่าเกินกว่าเกณฑ์ที่กำหนด เราก็อาจตัดสินว่าโปรตีนเส้นใหม่ Text นี้น่าจะเป็นสมาชิกของโปรตีนแฟมิลีนี้ และถ้า Text ถูกตัดสินว่าเป็นสมาชิกของแฟมิลี เราสามารถขยายจำนวนข้อมูลใน seed alignment โดยเพิ่มโปรตีนสายใหม่นี้เข้าไป ทำการอัพเดตพารามิเตอร์ที่เกี่ยวข้องใน HMM การขยายจำนวนสมาชิกใน seed alignment ทำให้สามารถจำแนกโปรตีนในแฟมิลีนี้ (ที่อาจมีความหลากหลาย) ได้ครอบคลุมมากยิ่งขึ้น ถึงจุดนี้เราควรพบว่าโปรไฟล์ HMM สามารถใช้เป็นเครื่องมือในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายของโปรตีน โดยสามารถบรรลุเป้าหมายที่ต้องการให้แต่ละคอลัมน์ซึ่งเป็นผลของการทำ multiple alignment สามารถมีคะแนนที่แตกต่างกันขึ้นอยู่กับความถี่ของแต่ละอักขระที่ถูกส่งออกในแต่ละคอลัมน์

หยุดคิด	ถ้าค่าผลคูณของค่าน้ำหนักคะแนนเกินกว่าเกณฑ์ที่กำหนดในโปรตีนมากกว่า 1 แฟมิลีของโปรตีน เราจะจำแนกโปรตีนเข้าสายใหม่นี้อย่างไร

สูโดเค้าท์

ถ้าลองสร้างเมทริกซ์ของค่าความน่าจะเป็นในการเปลี่ยนจากสถานะหนึ่งไปยังอีกสถานะหนึ่ง (transition probability matrix) หรือเมทริกซ์ของค่าความน่าจะเป็นในการส่งออกอักขระจำเพาะหนึ่งๆของทุกคอลัมน์

(emission probability matrix) จะพบว่ามีหลายช่องในเมทริกซ์ที่มีค่าเป็น 0 ซึ่งค่า 0 เหล่านี้อาจทำให้เกิดปัญหาได้ ตัวอย่างเช่น เส้นทางในรูปที่ 6.15 น่าจะเป็นเส้นทางที่ดีที่สุดของ Text = **ACAFDEAF** อย่างไรก็ตามถ้ามีการคำนวณค่า $\Pr(x, \pi)$ จะได้ค่าเป็น 0 เนื่องจากค่าความน่าจะเป็นในการเปลี่ยนสถานะจาก MATCH(2) ไปยัง INSERTION(2) ในโปรไฟล์ HMM นี้มีค่าเป็น 0 (ชุดของสายข้อมูลตั้งต้น 5 สายที่นำมาสร้างโปรไฟล์ HMM นี้ไม่มีสายใดเลยที่มีการเปลี่ยนสถานะจาก MATCH(2) ไปเป็น INSERTION(2))

ถ้ายังจำได้ในบทที่ 4 ที่ศึกษาเรื่องการหา regulatory motif เราก็พบปัญหานี้ในการสร้างโปรไฟล์เช่นกัน และแนวทางในการแก้ปัญหาที่ทำได้ง่ายคือการเพิ่มสูโดเค้าท์ (pseudo count) โดยในกรณีนี้เราจะบวกค่าตามที่ระบุในตัวแปร σ (โดยมีค่าน้อยๆ) ให้กับช่องในเมทริกซ์เฉพาะที่อาจเกิดขึ้นได้เพียงแต่เรายังอาจมีข้อมูลตั้งต้นไม่เพียงพอ เช่นจากสถานะ MATCH(i) ไป MATCH(i+1) จาก MATCH(i) ไป INSERTION(i) จาก MATCH(i) ไป DELETION(i+1) จาก DELETION(i) ไปยัง INSETION(i) จาก INSERTION(i) ไปยัง DELETION(i+1) และ MATCH(i+1) เป็นต้น และเมื่อมีการเพิ่มค่า σ ให้กับช่องเหล่านี้แล้ว ก็ต้องมีการปรับค่าให้เป็นมาตราฐาน (normalize) โดยที่ผลรวมของแต่ละบรรทัดต้องมีค่าเท่ากับ 1

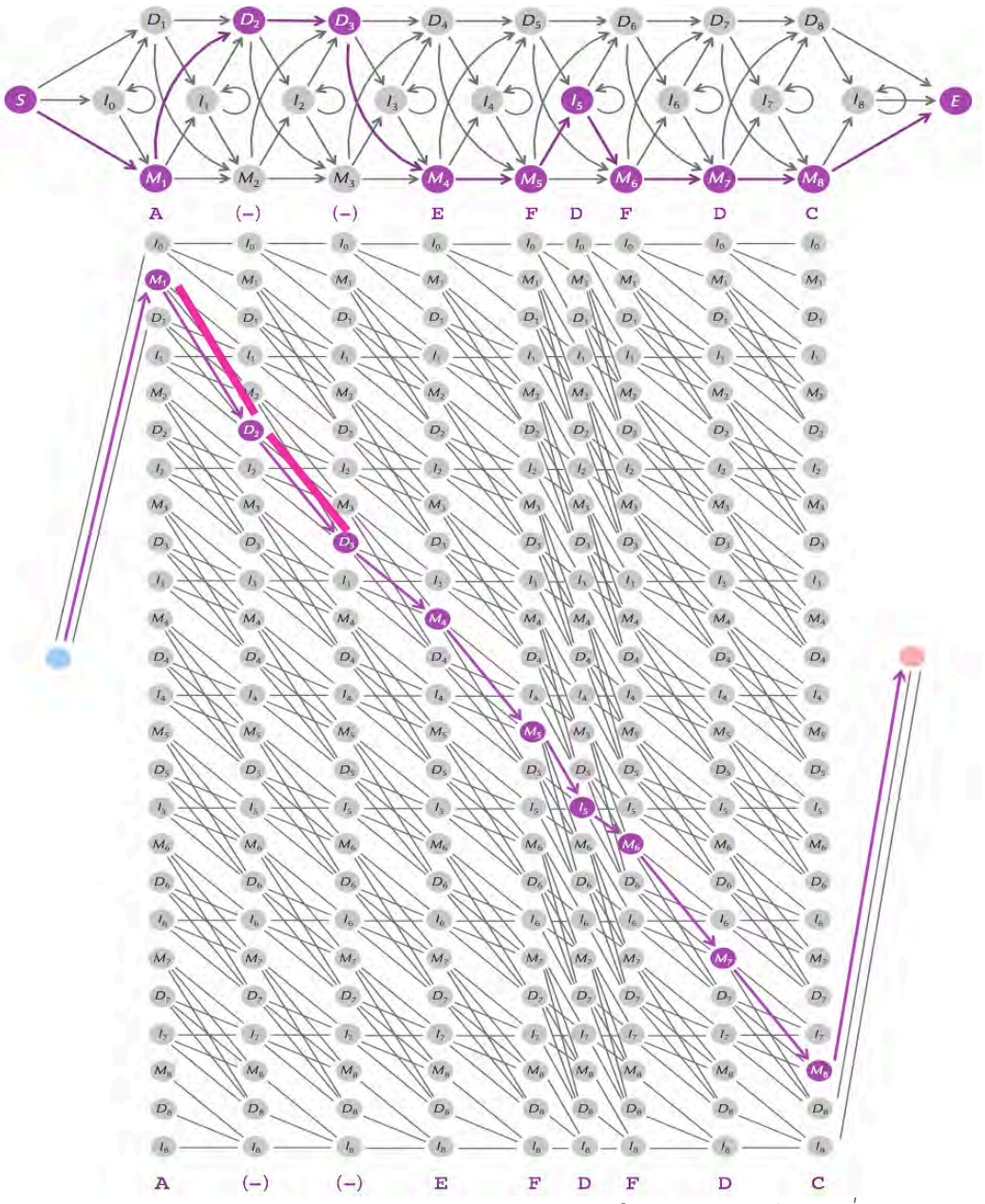
เราสามารถเพิ่มสูโดเค้าท์ ตามด้วยการปรับค่าให้เป็นมาตรฐาน ในเมทริกซ์ที่เก็บค่าความน่าจะเป็นในการส่งออกอักขระใดๆในแต่ละคอลัมน์ได้เช่นกัน ทั้งนี้เราสามารถอ้างถึงโปรไฟล์ HMM ที่มีการใส่สูโดเค้าท์และปรับค่าให้เป็นมาตรฐานแล้วด้วยสัญลักษณ์ HMM(Alignment, θ, σ)

นิยามปัญหาที่ 6.6 ปัญหาโปรไฟล์ HMM ที่เพิ่มสูโดเค้าท์

ปัญหาโปรไฟล์ HMM ที่เพิ่มสูโดเค้าท์ (Profile HMM with Pseudo counts Problem) สร้างโปรไฟล์ HMM ที่มีการเพิ่ม pseudo counts จากผล multiple alignment	
ข้อมูลเข้า	ผล multiple alignment *Alignment* และค่า column removal threshold (θ) และค่าสูโดเค้าท์ σ
ผลลัพธ์	HMM(*Alignment,* θ, σ)

หยุดคิด	เนื่องจากแผนภาพ HMM ในรูปที่ 6.15 มี 25 โหนด (ไม่รวมโหนดตั้งต้นและโหนดปลายทาง) กราฟวิเทอบิเพื่อการจำลองการส่งออกอักขระจะประกอบด้วย 25 แถว คำถามคือกราฟวิเทอบินี้จะมีทั้งหมดกี่คอลัมน์

ถึงจุดนี้เราก็พร้อมที่จะเทียบสายข้อมูลเข้า Text กับชุดของสายข้อมูลที่เป็นผลจากการทำ multiple alignment มาก่อนหน้า โดยเริ่มจากการสร้างกราฟวิเทอบิของสายข้อมูลเข้านี้ (รูปที่ 6.16) และแก้ปัญหา Decoding เพื่อหาเส้นทางแสดงลำดับสถานะที่น่าจะเป็นมากที่สุด



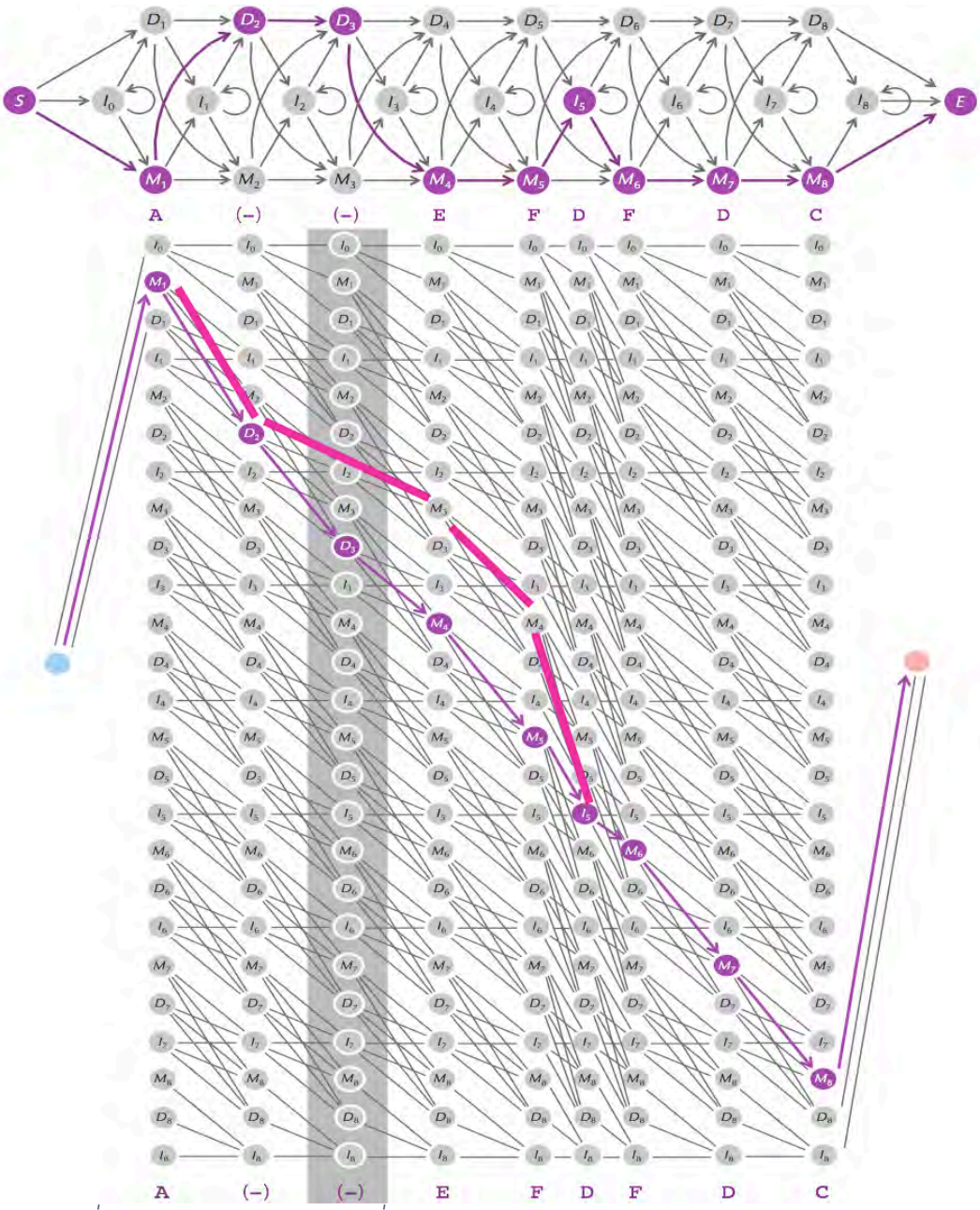
รูปที่ 6.16 กราฟวิเทอบิสำหรับ HMM(Alignment, θ) และเส้นทางในกราฟ (เส้นสีม่วง) ที่สอดคล้องกับสายอักขระที่ส่งออก AEFDFDC จากรูปที่ 6.14 เส้นเชื่อมระหว่างคอลัมน์แสดงถึงการเปลี่ยนสถานะที่เป็นไปได้ซึ่งมีทิศทางมุ่งไปทางขวา เส้นเชื่อมที่ชี้ไปยังโหนดที่แสดงสถานะ deletion จะมีเส้นสีชมพูกำกับ ส่วนด้านล่างสุดแสดงอักขระที่ส่งออกในแต่ละคอลัมน์

(ที่มา: รูปที่ 10.18 ของ [21])

หยุดคิด	ถ้าต้องหาเส้นทางแสดงลำดับสถานะผ่านกราฟวิเทอบิสำหรับเส้นทางสีเขียว แดง เหลืองและฟ้าในรูปที่ 6.14 จะเกิดอะไรขึ้น

ปัญหาของไซเลนท์เสตท

การหาเส้นทางแสดงลำดับสถานะที่ดีที่สุดโดยการประยุกต์ใช้วิธีการแก้ปัญหา Decoding ไม่ตรงไปตรงมาเหมือนตัวอย่างในตอนต้นบทเรียน เนื่องจากในความเป็นจริงแล้วกราฟในรูปที่ 6.16 ไม่ใช่กราฟวิเทอบิ ลองพิจารณาเส้นทางในรูปที่ 6.17 ซึ่งส่งออกสายอักขระเดียวกันกับที่ส่งออกในรูปที่ 6.16 แต่เส้นทางในรูปที่ 6.17 นี้เดินผ่าน



รูปที่ 6.17 เส้นทางที่แตกต่างจากเส้นทางในรูปที่ 16.6 แต่ส่งออกสายอักขระ AEFDFDC เดียวกัน คอลัมน์ที่มีพื้นหลังสีเทาจะถูกตัดออกไป ดังนั้นจำนวนคอลัมน์รวมจะลดลงไปหนึ่งคอลัมน์

(ที่มา: รูปที่ 10.18 ของ [21])

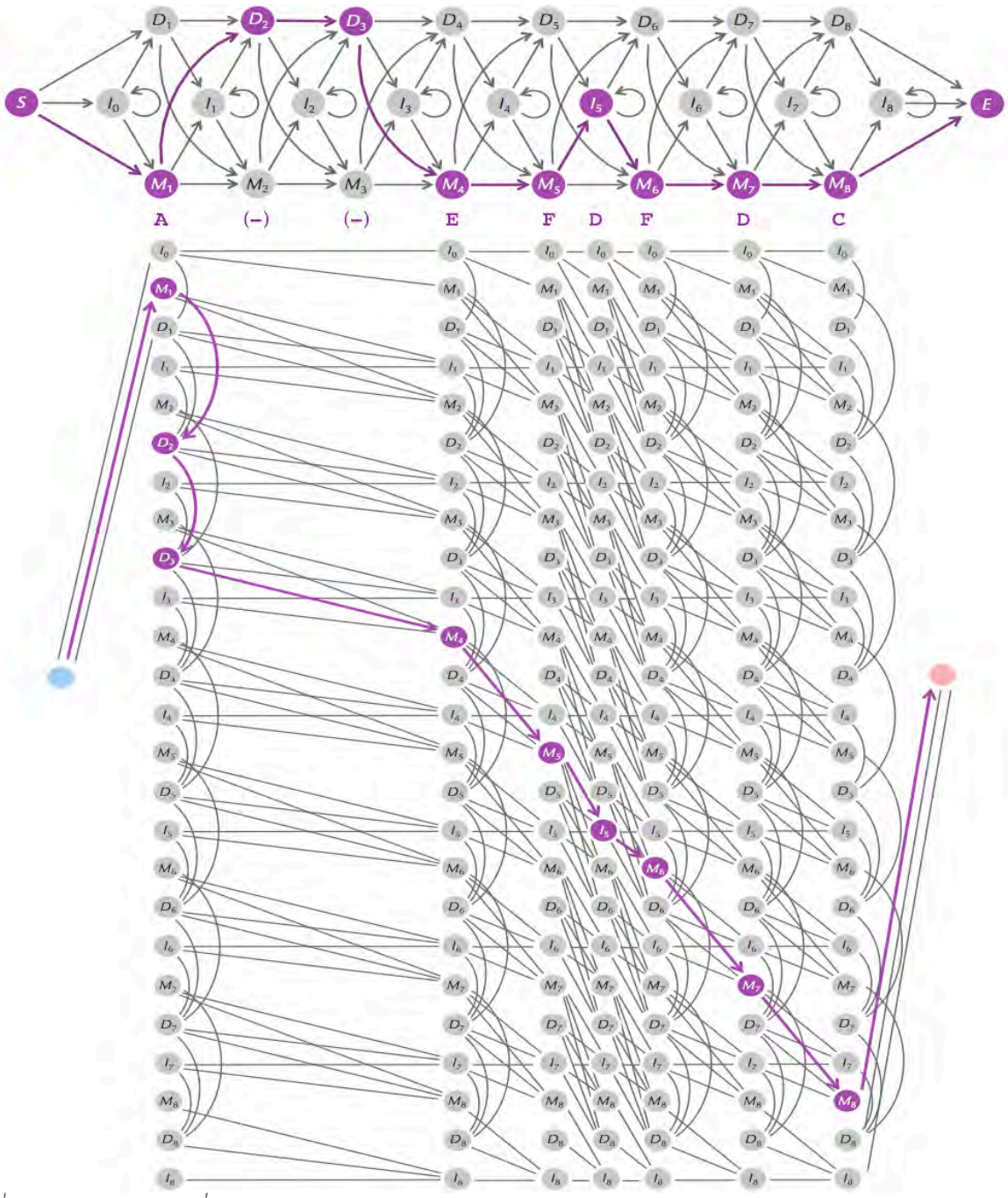
โหนดที่เป็นไซเลนท์สเตท (silent state) หรือโหนดที่เป็นสถานะ deletion เพียงโหนดเดียว (ในรูปที่ 6.16 เดินผ่านสองโหนด) ทำให้จำนวนคอลัมน์ในรูปที่ 6.17 ลดลงไปหนึ่งคอลัมน์ (คอลัมน์ที่มีพื้นหลังเป็นสีเทาจะถูกตัดออกไป) อย่างไรก็ตาม เราไม่สามารถเปลี่ยนแปลงกราฟวิเทอบิให้สอดคล้องกับเส้นทางแสดงสถานะซ้อนเร้น π ใดๆได้ เนื่องจากเราจะไม่ทราบเส้นทางเหล่านี้ล่วงหน้า ในทางกลับกันจำนวนคอลัมน์ของกราฟวิเทอบิจะต้องเท่ากับความยาวของสายอักขระที่ส่งออก ซึ่งเงื่อนไขนี้ไม่เป็นจริงสำหรับกราฟวิเทอบิทั้งในรูปที่ 6.16 และ 6.17

หยุดคิด	เราจะสามารถปรับแก้กราฟวิเทอบิสำหรับ HMM ที่มีไซเลนท์ได้อย่างไร

ในกรณีทั่วไปอัลกอริทึมวิเทอบิจะไม่อนุญาตให้มีไซเลนท์สเตทอื่นๆ นอกเหนือสถานะตั้งต้นและสถานะปลายทาง หรืออีกนัยยะหนึ่งคือตัวอัลกอริทึมมีสมมติฐานว่าโหนด (k,l) ในกราฟวิเทอบิอธิบายเหตุการณ์ที่ HMM จะส่งออกอักขระ x_i เมื่ออยู่ในสถานะ k อย่างไรก็ตามถ้า k เป็นไซเลนท์สเตท บทบาทของโหนด (k,i) ในกราฟวิเทอบิก็จะไม่ชัดเจน เพราะไม่สามารถระบุค่าความน่าจะเป็นที่ของการเปลี่ยนจากสถานะใดๆ มายังสถานะนี้ อย่างไรก็ตามในกรณีของโปรไฟล์ HMM เราสามารถแก้ปัญหานี้ได้โดยกำหนดกราฟวิเทอบิที่มีจำนวนแถวเท่ากับจำนวนสถานะ หรือ |States| และมีจำนวนคอลัมน์เท่ากับความยาวของ Text หรือ |Text| โดยทุกครั้งที่ HMM มีการเปลี่ยนจากสถานะใดๆมายังสถานะ deletion จะไม่มีขยับคอลัมน์ไปทางขวาในกราฟวิเทอบิ โดยจะมีการเปลี่ยนสถานะภายในคอลัมน์เดิม แต่ถ้า HMM เปลี่ยนไปยังสถานะแมชหรือ insertion จะมีการขยับไปทางขวาในคอลัมน์ถัดไป ผลที่ตามมาคือทุกคอลัมน์ในกราฟวิเทอบิจะส่งออกอักขระใดอักขระหนึ่งถึงแม้ว่าเส้นทางแสดงลำดับสถานะอาจผ่านมากกว่าหนึ่งสถานะในคอลัมน์หนึ่งๆ (รูปที่ 6.18)

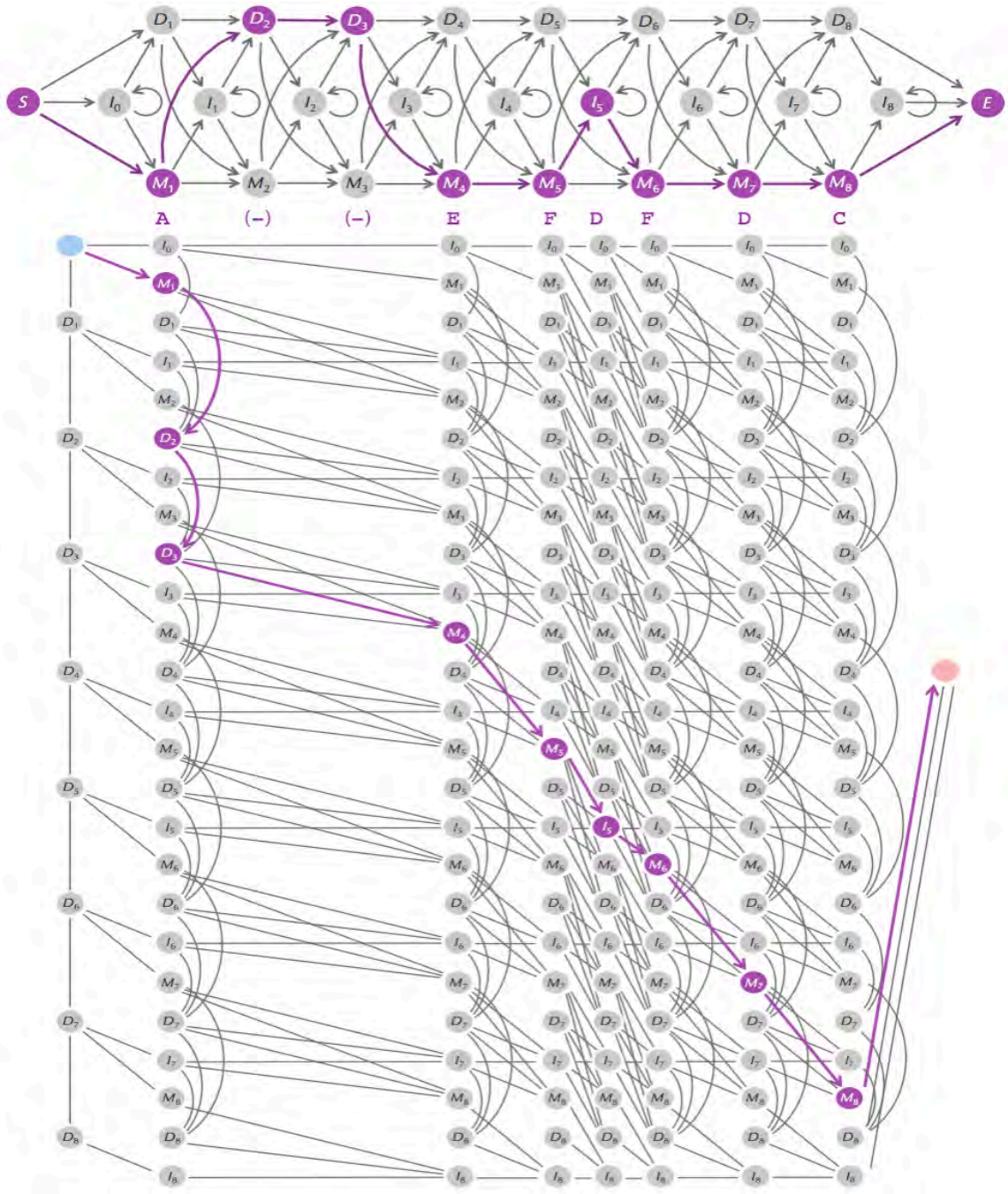
หยุดคิด	จากกราฟวิเทอบิในรูปที่ 6.18 ยังมีประเด็นที่ต้องพิจารณาเพิ่มเติมอีกหรือไม่

รูปที่ 6.18 ยังมีข้อจำกัดอยู่บ้าง โดยถ้า HMM มีการเปลี่ยนจากสถานะตั้งต้นไปยังสถานะ DELETION(1) จะไม่มีการส่งออกอักขระใดๆในคอลัมน์ที่ 1 ดังนั้นเราจะต้องมีการเปลี่ยนรูปแบบของโหนดตั้งต้นไปเป็นคอลัมน์ของไซเลนท์สเตทซึ่งมีทั้งโหนดตั้งต้นและสถานะ deletion ทั้งหมด (รูปที่ 6.19) ด้วยการปรับเปลี่ยนนี้ถ้า HMM เข้าสู่สถานะ DELETION(1) โดยเข้ามาจากโหนดตั้งต้น HMM ก็ยังสามารถผ่านสถานะ deletion อื่นๆ ก่อนเข้าสู่สถานะแมชหรือ insertion ในคอลัมน์ที่ 1 ต่อไป



รูปที่ 6.18 กราฟวิเทอบิที่มีจำนวนแถวเท่ากับ |States| และจำนวนคอลัมน์เท่ากับ |Text| ของโปรไฟล์ HMM ที่ส่งออกสายอักขระ AEFDFDC โดยเส้นเชื่อมที่แสดงการเปลี่ยนจากสถานะใดๆมายังสถานะ deletion จะอยู่ภายคอลัมน์เดียวกัน

(ที่มา: รูปที่ 10.20 ของ [21])



รูปที่ 6.19 กราฟวิเทอบิห้ายสุดของโปรไฟล์ HMM โดยจะส่งออกอักขระจำนวน 7 ตัว โดยเส้นเชื่อมในคอลัมน์เดียวกันจะมีทิศทางชี้ลง ในขณะที่เส้นเชื่อมระหว่างคอลัมน์จะมีทิศทางชี้ไปทางขวามือ ทั้งนี้เส้นทางเส้นสีม่วงแสดงเส้นทางใน HMM ที่ส่งออกอักขระ AEFDFDC

(ที่มา: รูปที่ 10.21 ของ [21])

ตกลงโปรไฟล์ HMM มีประโยชน์ไหม

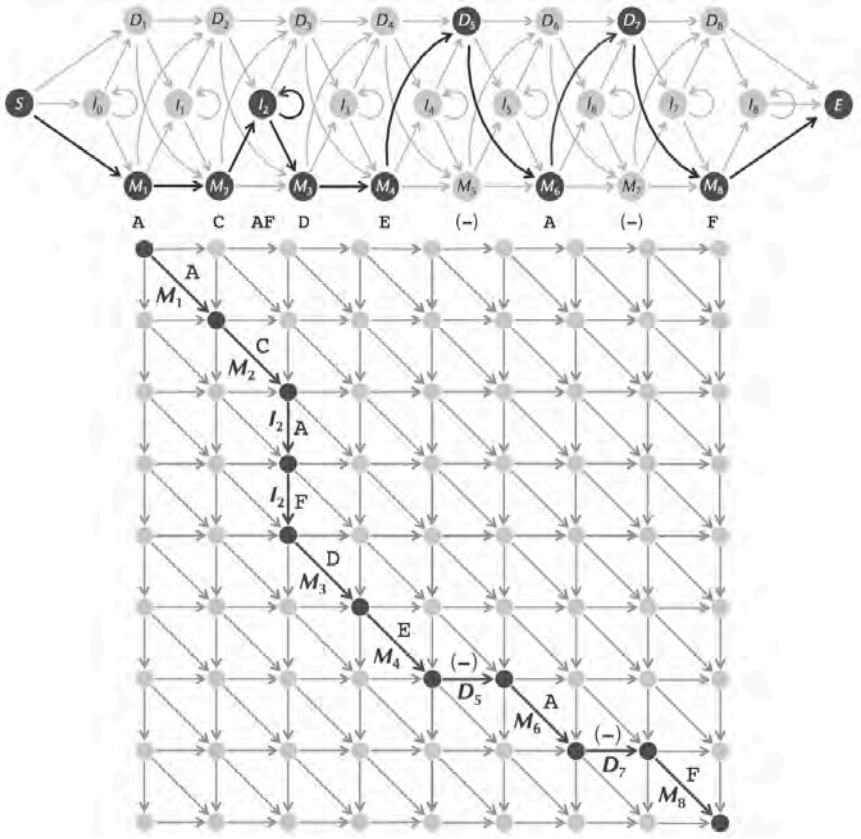
อัลกอริทึมวิเทอบิสามารถประยุกต์ใช้ได้กับ HMM ใดๆ ในหัวข้อนี้เราจะอธิบายว่าอัลกอริทึมวิเทอบิทำงานกับโปรไฟล์ HMM อย่างไร กำหนด $S_{MATCH(j),i}$ เป็นค่าความน่าจะเป็นของเส้นทางแสดงลำดับสถานะซ้อนเร้นที่น่าจะดีที่สุดของสายอักขระที่ส่งออก $x_1 \dots x_i$ ของ x โดยปิดท้ายด้วยสถานะซ้อนเร้น $MATCH(j)$ และกำหนด $S_{INSERTION(j),i}$ และ $S_{DELETION(j),i}$ ในลักษณะเดียวกัน เนื่องจากจะมีเส้นเชื่อมเพียง 3 เส้นที่ชี้เข้าสู่สถานะแมช $MATCH(j)$ ใดๆ สถานะของแบบเวียนเกิดของวิเทอบิสามารถแสดงได้ด้วยสมการต่อไปนี้

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

และถ้ามีการใส่ลอการิทึมทั้งสองฝั่งจะได้สมการชุดใหม่ดังต่อไปนี้ซึ่งมีความคล้ายคลึงกับชุดของสมการในการเปรียบเทียบความคล้ายคลึงกันระหว่างสายดีเอ็นเอและหรือโปรตีนสองเส้นแบบองค์รวม
(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 (ล่าง) แสดงเส้นทางในกราฟที่มีลักษณะใกล้เคียงกับกราฟแมนฮัตตัน โดยเส้นทางนี้สอดคล้องกับเส้นทางเดียวกันในโปรไฟล์ HMM เส้นเชื่อมทแยงมุม เส้นเชื่อมแนวตั้งชี้ลง และเส้นเชื่อมแนวนอนชี้ไปทางขวา ในกราฟนี้แสดงสถานะแมช สถานะ insertion และสถานะ deletion ตามลำดับ รูปที่ 6.20 นี้อาจทำให้บางคนรู้สึกว่าเสียเวลากับบทเรียนเรื่อง HMM และโปรไฟล์ HMM เนื่องจากถ้าดูจากเส้นทางแสดงสถานะซ้อนเร้นในรูปเส้นทางในโปรไฟล์ HMM (รูปที่ 6.20 บน) ก็เป็นเส้นทางเดียวกับเส้นทางของการเปรียบเทียบความคล้ายคลึงกันของข้อมูลสองสายแบบองค์รวม (รูปที่ 6.20 ล่าง) อย่างไรก็ตามถ้ามีการพิจารณาในรายละเอียดแล้วจะพบว่าการเลือกเส้นเชื่อมในรูปที่ 6.20 (บน) นี้จะมีความแตกต่างกันไปตามค่าความน่าจะเป็นของทั้งการเปลี่ยนจากสถานะซ้อนเร้นหนึ่งไปยังอีกสถานะซ้อนเร้นหนึ่งและค่าความน่าจะเป็นในการส่งออกแต่ละอักขระในแต่ละลำดับของการส่งออกของสายอักขระ โดยการได้มาซึ่งค่าพารามิเตอร์ของแต่ละคอลัมน์ในเมทริกซ์ที่แสดงผลการเปรียบเทียบความคล้ายคลึงกันของสายข้อมูล (alignment matrix) และสร้างเป็นโปรไฟล์ HMM ทำให้เราสามารถตรวจสอบความคล้ายคลึงกันของสายข้อมูลได้ละเอียดกว่า



รูปที่ 6.20 (บน) เส้นทางแสดงลำดับสถานะซ้อนเร้นผ่านโปรไฟล์ HMM และส่งออกสายของอักขระ ACAFDEAF (ล่าง) เส้นทางผ่านกราฟที่มีลักษณะใกล้เคียงกับกราฟแมนฮัตตันที่สอดคล้องกับเส้นทางแสดงลำดับสถานะซ้อนเร้นด้านบน

(ที่มา: รูปที่ 10.23 ของ [21])

การเรียนรู้พารามิเตอร์ใน HMM

การประมาณค่าพารามิเตอร์ใน HMM โดยทราบเส้นทางซ้อนเร้น

สมมติฐานหลักในการอธิบายเกี่ยวกับ HMM ในหัวข้อก่อนคือเราทราบค่าพารามิเตอร์ต่างๆ ของ HMM เช่นค่าความน่าจะเป็นในการเปลี่ยนสถานะซ้อนเร้น Transition และค่าความน่าจะเป็นในการส่งออกอักขระหนึ่งๆ เป็นต้น ซึ่งในความเป็นจริงแล้วความซับซ้อนหลักในการประยุกต์ใช้ HMM เพื่อตอบโจทย์ทางชีววิทยาคือการประมาณค่าพารามิเตอร์เหล่านี้จากข้อมูลที่มีอยู่ ถ้าเปรียบเทียบกับปัญหาคาสิโนก่อนหน้าก็คือเราทราบว่าเจ้ามือมีทั้งเหรียญปกติและเหรียญถ่วงน้ำหนัก แต่เราไม่ทราบว่าเหรียญถ่วงน้ำหนักนั้นจะถูกใช้ในการโยนรอบไหนบ้าง (ไม่ทราบโอกาสในการเปลี่ยนจากเหรียญปกติไปใช้เหรียญถ่วงน้ำหนักรวมทั้งกลับกัน) และไม่ทราบค่าความน่าจะเป็นในการออกหน้าเหรียญหัวหรือก้อยของเหรียญถ่วงน้ำหนัก

หยุดคิด	สมมติว่าลำดับหน้าของเหรียญที่ปรากฏคือ $x =$ “HHTHHHTHHTTTH” เราจะคาดเดาจำนวนครั้งที่ใช้เหรียญถ่วงน้ำหนักและความน่าจะเป็นในการเปลี่ยนเหรียญที่โยนได้หรือไม่ และถ้าทราบว่าเส้นทางของสถานะซ้อนเร้นเป็น π = FFFBBFFFFFBBB จะเปลี่ยนผลการคาดเดาหรือไม่

เราอ้างถึงเมทริกซ์ Transition และ Emission ว่าเป็นชุดของพารามิเตอร์ เป้าหมายของเราคือหาค่าพารามิเตอร์เหล่านี้และ π โดยทราบเพียงสายข้อมูล x ที่ส่งออกจาก HMM ทั้งนี้เพื่อให้สามารถบรรลุเป้าหมายได้เรามีสมมติฐานเพิ่มเติมว่านอกจากทราบสายข้อมูล x แล้วยังทราบค่าพารามิเตอร์หรือ π อย่างใดอย่างหนึ่งด้วย ในหัวข้อที่ผ่านมาจะถ้าเราทราบสายข้อมูล x ที่ส่งออกกับค่าพารามิเตอร์ เราจะสามารถหาเส้นทางสถานะซ้อนเร้น π ที่มีโอกาสเกิดมากที่สุดโดยใช้อัลกอริทึมวิเทอบิ อย่างไรก็ตามเรายังไม่พิจารณาว่าถ้าทราบสายข้อมูล x และเส้นทางสถานะซ้อนเร้น π เราจะประมาณค่าพารามิเตอร์ต่างๆ ได้อย่างไร

นิยามปัญหาที่ 6.7 ปัญหาการประมาณค่าพารามิเตอร์ของ HMM

ปัญหาการประมาณค่าพารามิเตอร์ของ HMM (HMM Parameter Estimation Problem) หาชุดของค่าพารามิเตอร์ที่เหมาะที่สุดในการอธิบายการส่งออกสายสตริง x และเส้นทางแสดงลำดับสถานะซ้อนเร้น π ของ HMM	
ข้อมูลเข้า	สายสตริง $x = x_1x_2 \ldots x_n$ ที่ถูกส่งออกและเส้นทางแสดงลำดับสถานะซ้อนเร้น $\pi = \pi_1 \ldots \pi_n$ ของ HMM โดยไม่ทราบค่าความน่าจะเป็นในเมทริกซ์ Transition และ Emission
ผลลัพธ์	ค่าความน่าจะเป็นในเมทริกซ์ Transition และ Emission ที่ทำให้ค่า $\Pr(x, \pi)$ มากที่สุด สำหรับทุกเมทริกซ์ Transition และ Emission

ถ้าเราทราบทั้ง x และ π เราจะสามารถประมาณค่าความน่าจะเป็น Transition ได้จากการทดลองคำนวณ เช่นถ้า $T_{l,k}$ แสดงจำนวนการเปลี่ยนสถานะจากสถานะซ้อนเร้น l มาเป็นสถานะซ้อนเร้น k ในเส้นทางแสดงลำดับสถานะซ้อนเร้น π เราสามารถคำนวณค่าความน่าจะเป็น $transition_{l,k}$ โดยคำนวณอัตราส่วนของ $T_{l,k}$ เทียบกับจำนวนของการเปลี่ยนจากสถานะซ้อนเร้น l ไปยังสถานะซ้อนเร้นอื่นๆทั้งหมด ดังสมการต่อไปนี้

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

เช่นเดียวกัน สำหรับค่าความน่าจะเป็น Emission ถ้าเรากำหนดว่า $E_k(b)$ แสดงจำนวนของการส่งออกอักขระ b ในสถานะซ้อนเร้น k โดยมีเส้นทางเป็น π เราจะสามารถประมาณค่าความน่าจะเป็น $emission_k(b)$ จากอัตราส่วนของ $E_k(b)$ เทียบกับจำนวนครั้งทั้งหมดที่ส่งออกอักขระใดๆในสถานะซ้อนเร้น k ดังสมการต่อไปนี้

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

จากสมการทั้งสองนี้ก็ทำให้เราสามารถประมาณค่าพารามิเตอร์ Transition และ Emission ได้

การเรียนรู้วิเทอบิ

จากความรู้ที่ว่าถ้าเราทราบสายข้อมูล x และชุดข้อมูลพารามิเตอร์ (Parameters) เราสามารถสร้างเส้นทางแสดงสถานะซ้อนเร้นที่มีโอกาสเกิดมากสุด π โดยใช้อัลกอริทึมวิเทอบิในการแก้ปัญหา Decoding

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

ในทางกลับกันถ้าเราทราบ x และ π เราก็สามารถประมาณค่าพารามิเตอร์ได้

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

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 เพื่ออธิบายการส่งออกสายสตริง x	
ข้อมูลเข้า	สายสตริง $x = x_1x_2...x_n$ ที่ถูกส่งออกโดย HMM โดยไม่ทราบค่าความน่าจะเป็นในเมทริกซ์ Transition และ Emission
ผลลัพธ์	ค่าความน่าจะเป็นในเมทริกซ์ Transition และ Emission ที่ทำให้ค่า $\Pr(x, \pi)$ มากที่สุด สำหรับทุกเมทริกซ์ Transition และ Emission และทุก π ที่เป็นไปได้

ปัญหาการเรียนรู้พารามิเตอร์ของ HMM นี้เป็นปัญหาที่ยากในการหาคำตอบ ทั้งนี้จึงมีการนำฮิวริสติกมาช่วยโดยข้อมูลพารามิเตอร์ในรอบแรกมาจากการเดาแบบสุ่ม และใช้ข้อมูลพารามิเตอร์นี้กับ x ในการหา π และเมื่อได้ π มาแล้วก็ย้อนกลับมาพิจารณาค่าพารามิเตอร์ โดยใช้ค่า x และ π และทำวนซ้ำระหว่างสองขั้นตอนนี้ โดยหวังว่าค่าประมาณพารามิเตอร์จะเข้าใกล้คำตอบของปัญหาการเรียนรู้ค่าพารามิเตอร์ของ HMM โดยแนวทางเรียนรู้ค่าพารามิเตอร์ของ HMM นี้เรียกว่า การเรียนรู้วิเทอบิ (Viterbi learning)

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

ทั้งนี้เรายังไม่ได้ระบุว่าการเรียนรู้วิเทอบิจะหยุดการวนซ้ำเพื่อประมาณค่าพารามิเตอร์ต่างๆเมื่อไหร่ ในทางปฏิบัติมีกฎในการหยุดหลายแนวทาง เช่น หยุดการทำงานเมื่อจำนวนรอบในการวนซ้ำเกินค่าที่กำหนดไว้ หรือหยุดเมื่อค่าความน่าจะเป็น $\Pr(x, \pi)$ มีความเปลี่ยนแปลงระหว่างรอบน้อยกว่าค่าที่กำหนด นอกจากนี้เนื่องจากการเรียนรู้วิเทอบิขึ้นอยู่กับการค่าพารามิเตอร์ที่เกิดจากการเดาสุ่มในรอบแรก ผลการเรียนรู้วิเทอบิอาจติดอยู่ในค่า local optimum (ค่าต่ำสุดสัมพัทธ์) เช่นเดียวกับการแก้ปัญหาอื่นๆโดยการใช้ฮิวริสติก เราจำเป็นต้องมีการรันอัลกอริทึมซ้ำหลายๆครั้ง เพื่อหาชุดของค่าพารามิเตอร์ที่ดีที่สุด

ฝึกหัด	ประยุกต์ใช้การเรียนรู้วิเทอบิในการเรียนรู้พารามิเตอร์สำหรับ HMM ของ CG-islands และโปรไฟล์ HMM สำหรับ gp120 HIV

การประมาณค่าพารามิเตอร์ของ HMM แบบยืดหยุ่น

ปัญหา Soft Decoding

เส้นทางแสดงลำดับสถานะซ้อนเร้นที่ดีที่สุดโดยอัลกอริทึมวิเทอบิจะให้คำตอบเพียงใช่หรือไม่ใช่สำหรับคำถามว่าที่เวลา i นั้นสถานะซ้อนเร้นคือสถานะ k ใช่หรือไม่ อย่างไรก็ตามในความเป็นจริงแล้วเราสามารถมั่นใจได้มากน้อยแค่ไหนสำหรับคำตอบนี้ ลองพิจารณาปัญหาคาสิโนอีกครั้ง สมมติว่าการโยนเหรียญรอบที่ i ออกหัว ถ้าหัวที่ออกนี้อยู่ในลำดับตรงกลางของการออกหัว 10 ครั้งติดกัน เราจะมีความมั่นใจมากขึ้นว่าเจ้ามือน่าจะใช้เหรียญถ่วงน้ำหนักในรอบการโยนเหรียญเหล่านั้น อย่างไรก็ตาม ถ้าผลการออกหน้าเหรียญใน 10 ครั้งนั้น 6 ครั้งออกหัวและ 4 ครั้งออกก้อย ในกรณีนี้เราควรมีความมั่นใจลดลงว่าเจ้ามือใช้เหรียญถ่วงน้ำหนักในรอบการโยนเหล่านั้น

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

นิยามปัญหาที่ 6.9 ปัญหา Soft Decoding

ปัญหา Soft Decoding หาค่าความน่าจะเป็นที่ HMM อยู่ในสถานะซ้อนเร้นและเวลาที่จำเพาะ โดยทราบสายสตริงส่งออก x	
ข้อมูลเข้า	สายสตริง $x = x_1x_2\ldots x_n$ ที่ถูกส่งออกโดย HMM
ผลลัพธ์	ค่าความน่าจะเป็นแบบมีเงื่อนไข $\Pr(\pi_i = k\|x)$ ที่ HMM อยู่ในสถานะซ้อนเร้น k ที่เวลาหรือรอบที่ i โดยมีการส่งออกสายสตริง x

ค่าความน่าจะเป็นแบบ *ไม่มี* เงื่อนไขที่เส้นทางแสดงลำดับสถานะซ้อนเร้นจะผ่านสถานะ k ที่เวลา i และส่งออกสตริง x สามารถคำนวณจากสมการผลรวมต่อไปนี้

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

และค่าความน่าจะเป็นแบบ *มี* เงื่อนไข $\Pr(\pi_i = k|x)$ เท่ากับสัดส่วนของเส้นทางที่ผ่านสถานะ k ที่เวลา i และส่งออกสายสตริง x เทียบกับจำนวนเส้นทางทั้งหมดที่สามารถส่งออกสายสตริง 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 $\Pr(\pi_i = F|x)$

อัลกอริทึม forward-backward

จากหัวข้อที่แล้วเรากำหนด $\Pr(\pi_i = k, x)$ มีค่าเท่ากับผลรวมของผลคูณค่าน้ำหนักคะแนน $\Pr(\pi, x)$ ของทุกเส้นทาง π ในกราฟวิเทอบิ ที่ผ่านโหนด (k,i) และแสดงออกสายสตริง x ดังแสดงในรูปที่ 6.21 (บน) เราสามารถแบ่งแต่ละเส้นทางออกเป็นเส้นทางย่อยสีฟ้าที่เริ่มจากโหนดต้นทาง (source) ไปยังโหนด (k,i) ซึ่งแสดงโดยสัญลักษณ์ π_{blue} และเส้นทางย่อยสีแดงจากโหนด (k,i) ไปยังโหนดปลายทาง (sink) แสดงโดยสัญลักษณ์ π_{red} โดยที่ $WEIGHT(\pi_{blue})$ และ $WEIGHT(\pi_{red})$ เป็นผลคูณค่าน้ำหนักคะแนนของเส้นทางย่อยดังแสดงในสมการการเวียนเกิดต่อไปนี้

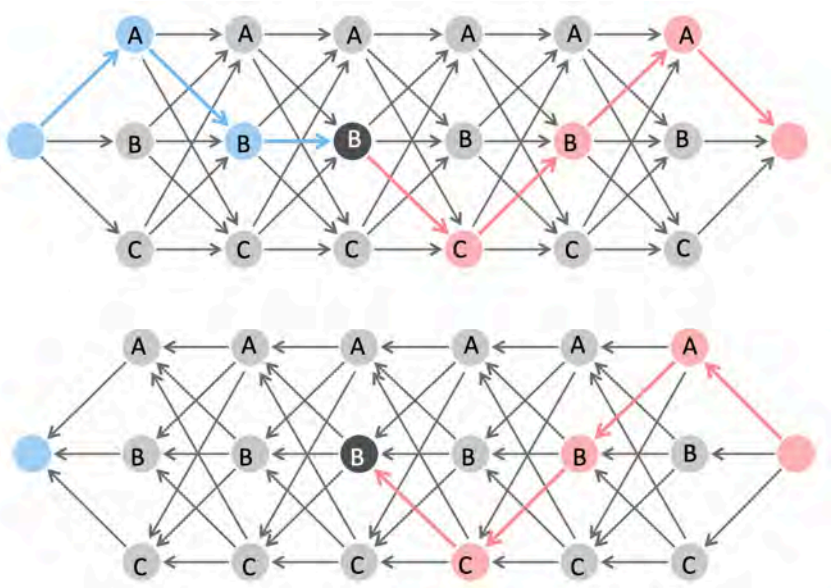
$$\begin{aligned}\Pr(\pi_i = k, x) &= \sum_{all\ paths\ \pi\ with\ \pi_i=k} \Pr(x, \pi) \\ &= \sum_{all\ paths\ \pi_{blue}} \sum_{all\ paths\ \pi_{red}} WEIGHT(\pi_{blue}) \cdot WEIGHT(\pi_{red}) \\ &= \sum_{all\ paths\ \pi_{blue}} WEIGHT(\pi_{blue}) \cdot \sum_{all\ paths\ \pi_{red}} WEIGHT(\pi_{red})\end{aligned}$$

ผลบวกของผลคูณค่าน้ำหนักคะแนนของเส้นทางย่อยสีฟ้าทั้งหมดหรือค่า $forward_{k,i}$ เป็นค่าที่มีการกล่าวถึงมาก่อนหน้าในการแก้ปัญหา Outcome Likelihood สำหรับหัวข้อนี้นอกจาก $forward_{k,i}$ แล้ว เราต้องคำนวณผลบวกของผลคูณค่าน้ำหนักคะแนนของเส้นทางย่อยสีแดงทั้งหมดซึ่งถูกเรียกว่า $backward_{k,i}$ ดังนั้นสมการข้างต้นสามารถเขียนใหม่ได้เป็น

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

โดย $backward_{k,i}$ ได้มาจากการคำนวณค่าโดยกลับทิศทางกราฟวิเทอบิ (รูปที่ 6.21 (ล่าง)) และประยุกต์ใช้อัลกอริทึมไดนามิกโปรแกรมมิงเช่นเดียวกับการหาค่า $forward_{k,i}$ เนื่องจากการกลับทิศทางเส้นเชื่อมจากโหนด (l, i+1) มายัง (k,i) มีค่าน้ำหนักคะแนนเป็น $WEIGHT_i(k,l) = transition_{k,l} \cdot emission_l(x_{i+1})$ ดังนั้นจึงเขียนสมการได้เป็น

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



รูปที่ 6.21 (บน) แสดงเส้นทางจากโหนดต้นทาง (source) ไปยังโหนดปลายทาง (sink) โดยผ่านโหนดสีดำ (k,i) ในกราฟวิเทอบิ โดยแบ่งออกเป็นเส้นทางย่อยสีฟ้าจากโหนดต้นทางมายังโหนด (k,i) และเส้นทางย่อยสีแดงจากโหนด (k,i) ไปยังโหนดปลายทาง (ล่าง) กราฟวิเทอบิกลับด้าน (reversed Viterbi graph) โดยเส้นเชื่อมทุกเส้นถูกกลับทิศทางโดยมีเส้นทางจากโหนดปลายทางมายังโหนด (k,i)

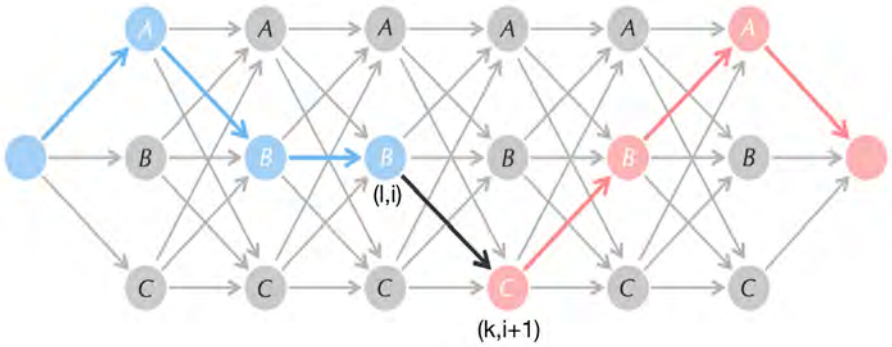
(ที่มา: รูปที่ 10.24 ของ [21])

การใช้ไดนามิกโปรแกรมมิงในการคำนวณค่าความน่าจะเป็น $\Pr(\pi_i = k, x)$ นี้เรียกว่าอัลกอริทึม forward-backward และการรวมอัลกอริทึม forward-backward เข้ากับคำตอบของปัญหา 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" • ประยุกต์ใช้การแก้ปัญหา Soft Decoding กับการหา CG-islands ใน 1 ล้านนิวคลีโอไทด์แรกของโครโมโซม X ของมนุษย์ คำตอบนี้แตกต่างจากคำตอบที่ได้จากอัลกอริทึมวิเทอบิอย่างไร

ข้างต้นเป็นการคำนวณค่าความน่าจะเป็นแบบมีเงื่อนไข $\Pr(\pi_i = k|x)$ ที่ HMM จะผ่านโหนด (k,i) ในกราฟวิเทอบิโดยมีเงื่อนไขว่า HMM ส่งออกสายอักขระ x คำถามถัดไปคือแล้วค่าความน่าจะเป็นแบบมีเงื่อนไข $\Pr(\pi_i = l, \pi_{i+1} = k|x)$ ที่ HMM จะผ่านเส้นเชื่อมระหว่างโหนด (l,i) ไปยังโหนด (k,i+1) โดยมีเงื่อนไขว่า HMM ส่งออกสายอักขระ x เป็นเท่าไหร่ ถ้าใช้แนวคิดเดียวกับอัลกอริทึม forward-backward ในกรณีนี้เราสามารถแบ่งเส้นทางเป็นเส้นทางย่อยสีฟ้าที่รวมทุกเส้นทางที่ผ่านเส้นเชื่อม (l,i) -> (k, i+1) (เส้นหนาสีดำ) ที่เป็นคำถามจากโหนดต้นทางมายังเส้นเชื่อมนี้และเส้นทางย่อยสีแดงจากเส้นเชื่อมนี้ไปยังโหนดปลายทาง (รูปที่ 6.22)



รูปที่ 6.22 เส้นทางในกราฟวิเทอบิจากโหนดต้นทางไปยังโหนดปลายทางโดยผ่านเส้นเชื่อม (l,i) -> (k, i+1)
(ที่มา: รูปที่ 10.25 ของ [21])

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 ${\Pi^*}_{k,i}$ แสดงโหนดในกราฟวิเทอบิและมีค่าความน่าจะเป็นเท่ากับ $\Pr(\pi_i = k|x)$ (รูปที่ 6.23 บน) 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 $\Pr(\pi_i = l, \pi_{i+1} = k|x)$ (รูปที่ 6.23 ล่าง) เพื่อความกระชับเราสามารถใช้ Π ในการอ้างอิงถึงทั้งเมทริกซ์ Π^* และ Π^{**}

	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

	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 (บน) Π^* และ (ล่าง) Π^{**}จากปัญหาคาสิโน

การเรียนรู้ Baum-Welch

อัลกอริทึม Expectation Maximization ที่ถูกใช้ในการประมาณค่าพารามิเตอร์ เรียกว่า Baum-Welch learning มีการสลับการทำงานระหว่าง 2 ขั้นตอน โดยขั้นตอน E (E-step) จะทำการประมาณค่าโปรไฟล์ responsibility Π โดยใช้ค่าของพารามิเตอร์ปัจจุบันตามสมการต่อไปนี้

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

และในขั้นตอน M (M-step) จะมีการประมาณค่า Parameters ใหม่โดยใช้เมทริกซ์ responsibility Π ที่เป็นผลลัพธ์จากขั้นตอน E ตามสมการต่อไปนี้

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

ทั้งนี้เราทราบการประมาณค่า π ในขั้นตอน E ของอัลกอริทึม Expectation Maximization แล้ว แต่ยังไม่ได้ออกแบบการประมาณค่าพารามิเตอร์ในขั้นตอน M ทั้งนี้การทราบเส้นทางลำดับสถานะซ้อนเร้น π เราจะสามารถประมาณค่าพารามิเตอร์ที่ดีที่สุดสำหรับเส้นทาง 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 $T_{l,k}$ คือจำนวนครั้งของการเปลี่ยนจากสถานะซ้อนเร้น l ไปยังสถานะซ้อนเร้น k ในเส้นทาง π และ $E_k(b)$ คือจำนวนครั้งที่มีการส่งออกอักขระ b เมื่อใช้เส้นทาง π และอยู่ในสถานะซ้อนเร้น k

ฝึกหัด	ถ้าเราไม่ทราบเส้นทาง π เราจะสามารถปรับสมการข้างต้นให้สามารถประมาณค่าพารามิเตอร์ได้อย่างไร

ลองพิจารณาการคำนวณค่า $T_{l,k}$ และ $E_k(b)$ โดยทราบเส้นทาง π แต่มีการเปลี่ยนวิธีการคำนวณเล็กน้อย 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 $E_k(b)$ สามารถเขียนใหม่ได้เป็น

$$T_{l,k} = \sum_{i=1}^{n-1} T_{l,k}^{i} \qquad E_k(b) = \sum_{i=1}^{n} E_k^i(b)$$

ซึ่งในกรณีที่เราไม่ทราบเส้นทาง π เราสามารถจะแทนค่าตัวแปร $T_{l,k}$ และ $E_k(b)$ โดยใช้ตัวแปร $T_{l,k}^{i}$ และ $E_k^i(b)$ ตามลำดับ ซึ่งสามารถคำนวณได้จากความน่าจะเป็นแบบมีเงื่อนไขว่าเส้นทางแสดงสถานะซ้อนเร้น π ผ่านโหนดหรือเส้นเชื่อมที่กำหนดในกราฟวิเทอบิ

$$T_{l,k}^{i} = \Pr(\pi_i = l, \pi_{i+1} = k|x) \qquad E_k^i(b) = \Pr(\pi_i = k|x)$$
$$= \Pi_{l,k,i}^{**} \qquad = \Pi_{k,i}^{*} \; if \; x_i = b \; and \; 0 \; otherwise$$

ซึ่งจะเห็นว่าค่าความน่าจะเป็นเหล่านี้ได้มีการคำนวณมาแล้วในหัวข้อที่ผ่านมา ดังนั้นเราสามารถประมาณค่าพารามิเตอร์ใหม่ โดยใช้สมการต่อไปนี้ (หมายเหตุ: พบว่าการประมาณค่าโดยวิธีการนี้ส่วนใหญ่ให้ผลที่ดีกว่าการประมาณค่าพารามิเตอร์โดยการเรียนรู้วิเทอบิ)

$$transition_{l,k} = \sum_{i=1}^{n-1} \Pi_{l,k,i}^{**} \qquad emission_k(b) = \sum_{i=1}^{n} \Pi_{k,i}^{*}$$

หยุดคิด	เราควรทำการนอร์มัลไลซ์ค่าความน่าจะเป็น transition และ emission ในสมการข้างต้นหรือไม่ หรืออีกนัยยะหนึ่งคือในสมการข้างต้นอนุมานได้ว่าผลรวมของค่าความน่าจะเป็นของการเปลี่ยนจากสถานะ (transition) ต้องเป็น 1 หรือไม่

ฝึกหัด	ใช้ Baum-Welch ในการเรียนรู้ค่าพารามิเตอร์สำหรับ HMM ของ CG-islands และ HMM ของ profile-HMM ของเชื้อเอชไอวี เปรียบเทียบค่าของพารามิเตอร์เหล่านี้กับค่าพารามิเตอร์ที่ได้จากการเรียนรู้วิเทอบิ

บทส่งท้าย

ธรรมชาติในฐานะนักประกอบ

ส่วนของลำดับกรดอะมิโนในสายโปรตีนหนึ่งๆสะท้อนโครงสร้างสามมิติและฟังก์ชันการทำงานของโปรตีนนั้นๆ ตัวอย่างเช่น โดเมนซิงค์ฟิงเกอร์ (zinc finger) เป็นส่วนประกอบในโครงสร้างสามมิติของโปรตีนซิงค์ฟิงเกอร์ (รูปที่

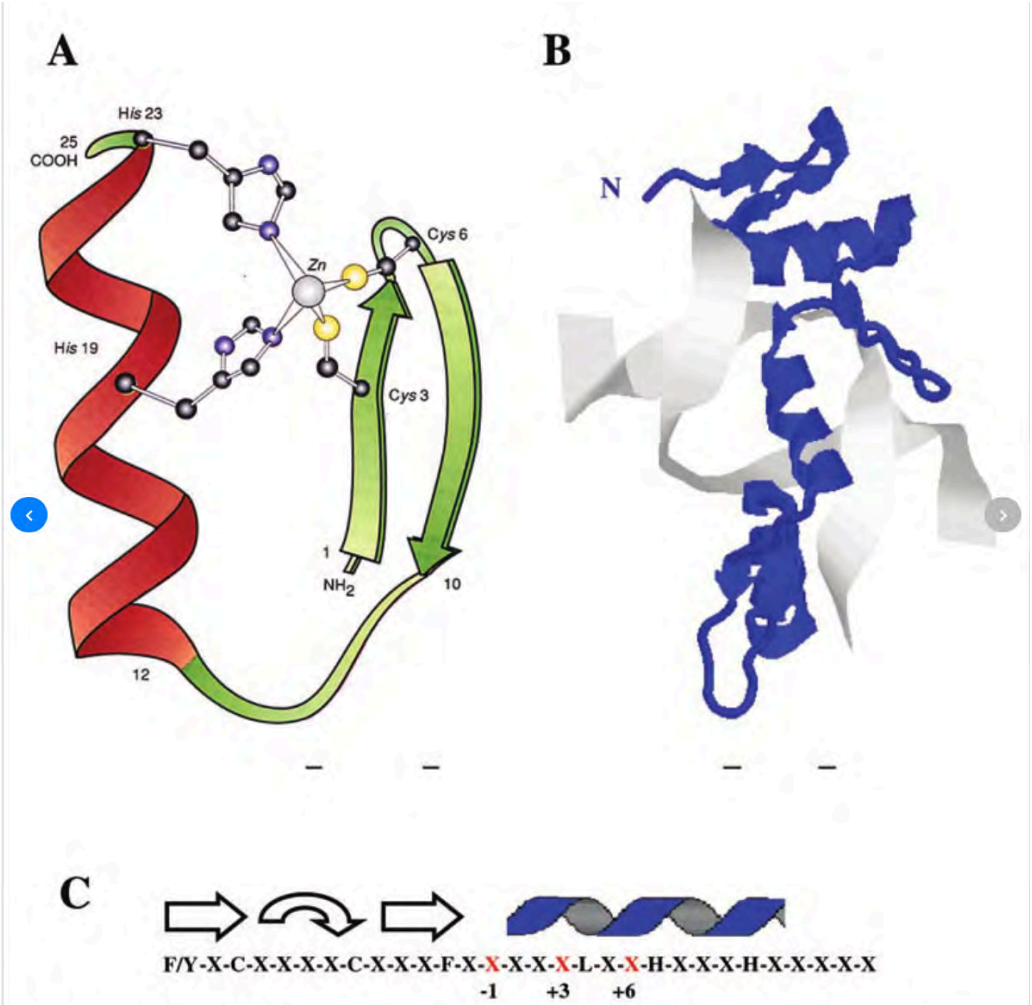
6.24) ด้วยการจัดการกรดอะมิโนซีสเทอีน (cysteine) 2 ตัวและฮีสทิดีน (histidine) อีก 2 ตัวที่อยู่ใกล้กันในโปรตีนซิงค์ฟิงเกอร์ ทำให้โปรตีนสามารถจับได้กับซิงค์ไอออน (zinc ion) และสามารถม้วนรอบตัวไอออนได้แน่นหนา ซิงค์ฟิงเกอร์เป็นส่วนของโปรตีนที่มีประโยชน์มากและเป็นส่วนโปรตีนในมนุษย์หลายพันโปรตีน นอกจากนี้โปรตีนซิงค์ฟิงเกอร์ยังสามารถจับกับไอออนอื่นทั้งที่เป็นเมทัล (metal) และนอนเมทัล (non-metal) ได้ด้วย

จากการทดลองในห้องปฏิบัติการมากกว่า 100,000 การทดลองเพื่อศึกษาโครงสร้างของโปรตีน พบว่าโปรตีนมากมายมีโครงสร้างหรือส่วนของโครงสร้างที่มีความคล้ายคลึงกันมาก โดยโปรตีนโดเมน (protein domain) จะเป็นส่วนของสายโปรตีนที่มักมีความอนุรักษ์ร่วมกันระหว่างโปรตีนและมีฟังก์ชันการทำงานจำเพาะและเป็นอิสระจากส่วนอื่นๆของสายโปรตีน ความยาวของโปรตีนโดเมนมีความหลากหลายแต่ความยาวโดยเฉลี่ยอยู่ที่ 100 กรดอะมิโน (โดเมนซิงค์ฟิงเกอร์มีความยาวโดยเฉลี่ยประมาณ 20-30 กรดอะมิโน) โปรตีนมากมายประกอบด้วยหลายโดเมนและแต่ละโปรตีนโดเมน (ที่มีความคล้ายคลึงกันมาก) ก็ปรากฏอยู่ในหลายๆโปรตีน

François Jacob ผู้ได้รับรางวัลโนเบลสาขาการแพทย์ ได้กล่าวไว้ในปีค.ศ. 1977 ว่า “ธรรมชาติเป็นเสมือนนักประกอบแต่ไม่ใช่นักประดิษฐ์” ทั้งนี้ธรรมชาติมีโปรตีนโดเมนเป็นหน่วยโครงสร้าง (building block) พื้นฐานและทำการสร้างโปรตีนที่ประกอบด้วยหลายโปรตีนโดเมน (multi-domain proteins) (ตัวอย่างในรูปที่ 6.25) โดยการนำโปรตีนโดเมนต่างๆมาเชื่อมต่อเข้าด้วยกันในลำดับที่แตกต่างกันไป ในการศึกษาที่ผ่านมาพบว่าโปรตีนโดเมนหลายชนิด ปรากฏอยู่ในโปรตีนที่ประกอบด้วยหลายโดเมนในมนุษย์ แต่โดเมนเหล่านี้กลับปรากฏเป็นโดเมนเดี่ยวในหลายๆโปรตีนในแบคทีเรีย ในธรรมชาติการเกิดโปรตีนที่ประกอบด้วยหลายโดเมนสามารถเกิดจากเรียงลำดับเบสใหม่ในจีโนม (genome rearrangement) ทำให้อาจเกิดยีนที่เกิดจากส่วนของยีนเดิมมากกว่า 1 ยีนมาต่อกัน การประกอบสองโดเมนเข้าเป็น 1 สายโปรตีนมักจะแสดงถึงขบวนการวิวัฒนาการที่เป็นประโยชน์ ตัวอย่างเช่น ถ้าทั้งสองโดเมนเป็นเอนไซม์ อาจช่วยให้เซลล์สามารถสนับสนุนการทำงานระหว่างสองเอนไซม์ได้ดีขึ้น เป็นต้น

เนื่องจากโปรตีนมักถูกสร้างขึ้นจากการประกอบหลายโปรตีนโดเมนที่มีโครงสร้างและฟังก์ชันการทำงานที่แตกต่างกันเข้าด้วยกัน นักชีววิทยามักวิเคราะห์โปรตีนโดเมนเดี่ยวๆแทนการศึกษาโปรตีนทั้งสายเพื่อศึกษาความสัมพันธ์ในเชิงวิวัฒนาการ ฐานข้อมูลพีแฟม (Pfam Database: http://pfam.xfam.org) มีมากกว่า 10,000 HMMs ที่ถูกสร้างจากการเปรียบเทียบความคล้ายคลึงกันระหว่างชุดของสายโปรตีน (multiple sequence alignments) ซึ่งสามารถใช้ในการวิเคราะห์ข้อมูลสายโปรตีนใหม่ว่าประกอบด้วยโปรตีนโดเมนอะไรบ้าง

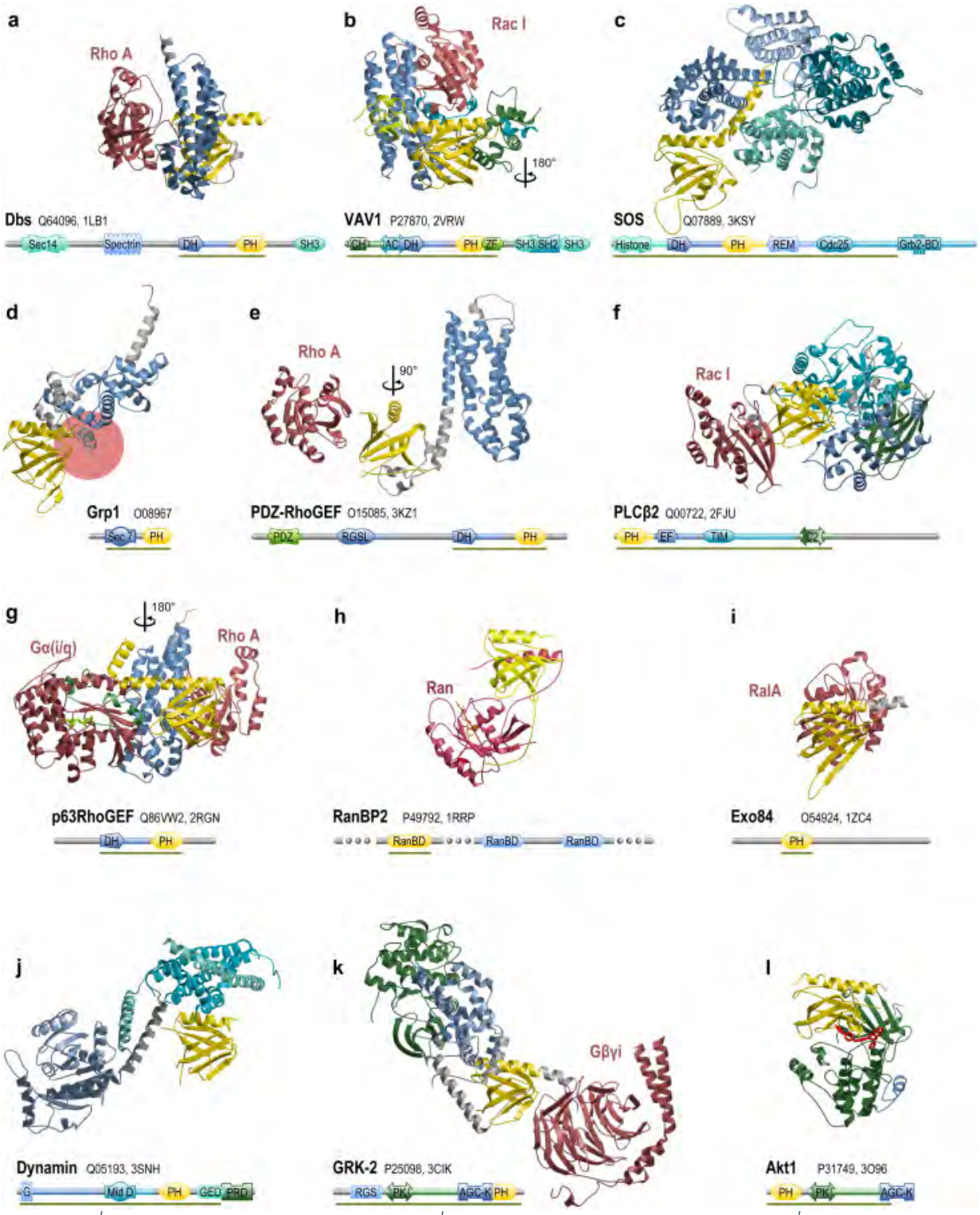
ฝึกหัด	ลองสำรวจข้อมูลของโปรตีนโดเมน Piwi (PF02171) ในฐานข้อมูล Pfam และดาวน์โหลดสายข้อมูลตั้งต้นหรือที่เรียกว่า seed sequences มาลองสร้าง HMM ของแฟมมิลี Piwi โดยใช้วิธีการที่ได้ศึกษาในบทเรียนนี้



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 C2H2 Zinc Finger motif

(ที่มา: รูปที่ 3 ของ [133])



รูปที่ 6.25 ตัวอย่างของโปรตีนโดเมน PH ที่พบว่าเป็นส่วนประกอบในหลายโปรตีนที่มีหลายโดเมน (ที่มา: รูปที่ 2 ของ [134])

การประยุกต์ใช้ HMMs ในโจทย์ทางชีวสารสนเทศอื่นๆ

โปรไฟล์ HMM สำหรับการเปรียบเทียบความคล้ายกันของสายดีเอ็นเอหรือโปรตีนหลายเส้น หรือ HMM สำหรับการหา CG-islands เป็นเพียงตัวอย่างเบื้องต้นของการประยุกต์ใช้ HMM ในการแก้โจทย์ในเชิงชีวสารสนเทศ ตัวอย่างอื่นๆ ของการประยุกต์ใช้ HMM เช่น การทำนายบริเวณที่เป็นยีนในจีโนม [135] การทำนายโครงสร้างสองมิติของโปรตีน [136-138] การทำนายตำแหน่งโอเมก้าในการศึกษา anchored proteins [139] การหาตำแหน่งที่เป็นโมทิฟ [140, 141] และหาอาร์เอ็นเอโมทิฟที่โปรตีนจะมาจับ [141, 142] การประยุกต์ใช้ infinite HMM เพื่อการวิเคราะห์ข้อมูลซิงเกิลโมเลกุล [143] การประยุกต์ใช้ sparely correlated HMM ในการเชื่อมโยงและศึกษาความสัมพันธ์ระหว่างบริเวณต่างๆในจีโนม [144] การประยุกต์ใช้ HMM ในการอนุมานการเกิดการแปรผันของนิวคลีโอไทด์เดี่ยว (Single Nucleotide Variants) ในจีโนม [145] การหาบริเวณที่เกิด Copy Number of Variations (CNVs) ในจีโนม [146, 147] การประยุกต์ใช้ HMM ในการประเมินความน่าเชื่อถือของผลของการประกอบร่างจีโนม [148] การประยุกต์ใช้ multivariate HMM ในการระบุสถานะของโครมาติน (Chromatin-state) ในบริเวณต่างๆของจีโนม [149] การประยุกต์ใช้ HMM ในการทำนายทรานสเมมเบรนโปรตีนในกลุ่มที่เป็นอัลฟาเฮลิกส์ [150] และกลุ่มที่เป็นเบต้าบาร์เรล [151] การประยุกต์ใช้ ensemble HMMs ในการจำแนกโปรตีนแฟมิลี [152] การประยุกต์ใช้ HMM ในการจำแนกการพับของโปรตีน [153] การประยุกต์ใช้ HMM ในการวิเคราะห์และตัดสินโฮโมโลจีของโปรตีนโดเมน (Protein Domain Homology) [154] การประยุกต์ใช้โปรไฟล์ HMM ของโปรตีนแฟมิลีในการสร้างต้นไม้ไฟโลจีนี (Phylogeny tree) [155] การประยุกต์ใช้ HMM ในการทำนายบริเวณที่เป็น MoRF (Molecular Recognition Features) ซึ่งอยู่ในสายของโปรตีนที่มีความผิดปกติในตัวเอง (Intrinsically Disordered Proteins: IDPs) [156] การประยุกต์ใช้ HMM เป็นวิธีการหนึ่งในการบริเวณที่มีฟังก์ชันของโปรตีนที่สามารถจับได้กับอาร์เอ็นเอ (RNA-binding proteins: RBPs) [157] เว็บเซิฟเวอร์ HMMER [158-160] เป็นเว็บไซต์ที่มีการพัฒนาอย่างต่อเนื่องเพื่อให้ผู้ใช้สามารถสืบค้นโปรตีนในฐานข้อมูลที่มีความคล้ายคลึงกันกับสายโปรตีนเข้าโดยใช้โปรไฟล์ HMM นอกจากนี้ยังมีการประยุกต์ใช้ Self-Organizing HMM map ในการจัดกลุ่มและแสดงผลการจัดกลุ่มของสายข้อมูล [161] การประยุกต์ใช้ HMM ในการศึกษาจีโนมิกส์ในระดับประชากร (population genomics) [162] การประยุกต์ใช้ HMM ในการจำลองการกระบวนการทางชีววิทยา [163] เป็นต้น ตัวอย่างอื่นๆ ในการประยุกต์ใช้ HMM ในโจทย์ทางชีววิทยาสามารถศึกษาเพิ่มเติมได้จากรีวิวเปเปอร์ เช่น [164, 165] เป็นต้น

แบบฝึกหัดบทที่ 6

จงเขียนโปรแกรมเพื่อแก้ปัญหาที่เกี่ยวข้องกับ 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/ba10j/)

บทที่ 7 การวิเคราะห์การแสดงออกของยีน (Gene expression analysis)

วัตถุประสงค์

- เพื่อให้นิสิตเห็นความสำคัญของการวัดการแสดงออกของยีน รวมทั้งความเกี่ยวข้องของการแสดงออกของยีนกับกระบวนการเซ็นทรัลดอกมา
- เพื่อให้นิสิตเห็นตัวอย่างของการประยุกต์ใช้การวัดการแสดงออกของยีนในการตอบโจทย์ทางชีววิทยา เช่น การศึกษากลุ่มของยีนที่มีผลต่อการเปลี่ยนเอทานอลเป็นน้ำตาลในยีสต์ เป็นต้น
- เพื่อให้นิสิตคุ้นเคยกับเทคโนโลยีที่เกี่ยวข้องกับการวัดการแสดงออกของยีนรวมทั้งลักษณะข้อมูลการแสดงออกของยีนที่มาจากเทคโนโลยีที่แตกต่างกัน
- เพื่อให้นิสิตคุ้นเคยกับแนวทางการวิเคราะห์การแสดงออกของยีน
- เพื่อให้นิสิตได้เห็นตัวอย่างงานวิจัยและผลงานวิจัยรวมทั้งตัวอย่างโปรแกรมที่ใช้ในการวิเคราะห์การแสดงออกยีน
- เพื่อให้นิสิตได้เห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทาย รวมทั้งงานวิจัยอื่นๆ ที่เกี่ยวข้อง

ผลลัพธ์ที่คาดหวัง

- นิสิตสามารถอธิบายความสำคัญของการวัดการแสดงออกของยีนรวมทั้งสามารถยกตัวอย่างการประยุกต์ใช้การวัดการแสดงออกของยีนเพื่อตอบโจทย์ทางชีววิทยา
- นิสิตสามารถยกตัวอย่างเทคโนโลยีที่ใช้ในการวัดการแสดงออกของยีน รวมทั้งสามารถอธิบายความแตกต่างระหว่างเทคโนโลยีได้
- นิสิตเข้าใจลักษณะข้อมูลการแสดงออกของยีนที่มาจากเทคโนโลยีที่แตกต่าง รวมทั้งสามารถอธิบายแนวทางในการวิเคราะห์การแสดงออกของยีนและอัลกอริทึมพื้นฐานที่เกี่ยวข้องได้
- นิสิตสามารถอธิบายการทำงานของอัลกอริทึมพื้นฐานที่ใช้ในการจัดกลุ่มการแสดงออกของยีนได้
- นิสิตสามารถเขียนโปรแกรมที่ใช้ในการจัดกลุ่มการแสดงออกของยีนแบบง่ายได้
- นิสิตสามารถยกตัวอย่างโปรแกรมที่ใช้ในการวิเคราะห์การแสดงออกของยีนรวมทั้งการจัดกลุ่มยีนที่มีลักษณะการแสดงออกร่วมกันได้

- นิสิตสามารถยกตัวอย่างความท้าทายที่มีอยู่และสามารถนำเสนอแนวทางในการพัฒนาวิธีการแก้ปัญหาเหล่านี้ได้ รวมทั้งสามารถประยุกต์องค์ความรู้จากบทเรียนเพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้

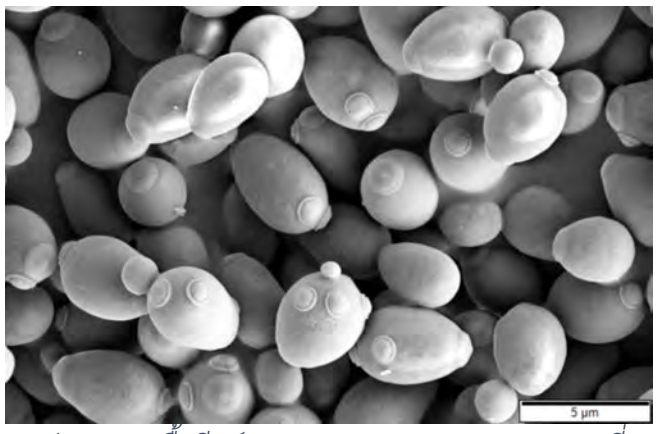
เนื้อหาโดยสรุป

การวัดการแสดงออกของยีนคือการวัดจำนวนอาร์เอ็นเอที่ถูกแปลรหัสมาจากดีเอ็นเอในขบวนการเซ็นทรัลดอกมา การวัดการแสดงออกของยีนมีความสำคัญเป็นอย่างมากในการศึกษาฟังก์ชันการทำงานของยีนในขบวนการต่างๆ ทางชีววิทยา ทั้งนี้สมมติฐานที่สำคัญอย่างหนึ่งคือยีนที่มีความเกี่ยวเนื่องกันมักมีการแสดงออกในรูปแบบเดียวกัน เทคโนโลยีที่สามารถวัดการแสดงออกของยีนจำนวนมากหรือของยีนทั้งจีโนมพร้อมๆกัน ประกอบด้วยเทคโนโลยีไมโครอะเรย์ (Microarray) และ อาร์เอ็นเอซีค (RNA-Seq) ซึ่งมีแนวทางในการวัดการแสดงออกและลักษณะของข้อมูลที่ได้จากการทดลองมีความแตกต่างกัน เนื่องจากข้อมูลการแสดงออกของยีนที่ได้จากการเทคโนโลยีทั้งสองกลุ่มมีความแตกต่างกันมาก การวิเคราะห์ข้อมูลเบื้องต้นจะมีความจำเพาะตามเทคโนโลยีที่ใช้ในการผลิตข้อมูล อย่างไรก็ตามเป้าหมายหลักของการวัดการแสดงออกของยีนคือการการจัดกลุ่มยีนที่มีรูปแบบการแสดงออกในรูปแบบเดียวกันซึ่งนำไปสู่การอนุมานหรือวิเคราะห์ฟังก์ชันการทำงานเพิ่มเติม เช่นการหา regulatory motif (บทที่ 4) ของยีนที่อยู่ในกลุ่มเดียวกัน ชุดของยีนที่มีการตอบสนองต่อยาในรูปแบบเดียวกัน หรือชุดของยีนที่แสดงออกร่วมกันในผู้ป่วยมะเร็งชนิดหนึ่งๆ ในระดับความรุนแรงหรือในสถานะต่างๆ เป็นต้น ในบทเรียนเรียนนี้จะอธิบายอัลกอริทึมพื้นฐานที่ใช้ในการจัดกลุ่มยีน (clustering algorithms) เช่น อัลกอริทึม K-Center และ K-means แนวคิดเรื่อง Soft clustering กระบวนการ Expectation Maximization (EM) และการจัดกลุ่มแบบเป็นลำดับชั้น (Hierarchical clustering) เป็นต้น รวมทั้งตัวอย่างโปรแกรมที่ใช้ในการวิเคราะห์การแสดงออกของยีน การประยุกต์ใช้องค์ความรู้เหล่านี้ในการแก้ปัญหาอื่นๆ และโจทย์วิจัยที่เกี่ยวข้อง

บทที่ 7 การวิเคราะห์การแสดงออกของยีน (Gene expression analysis)

ประวัติของการทำไวน์

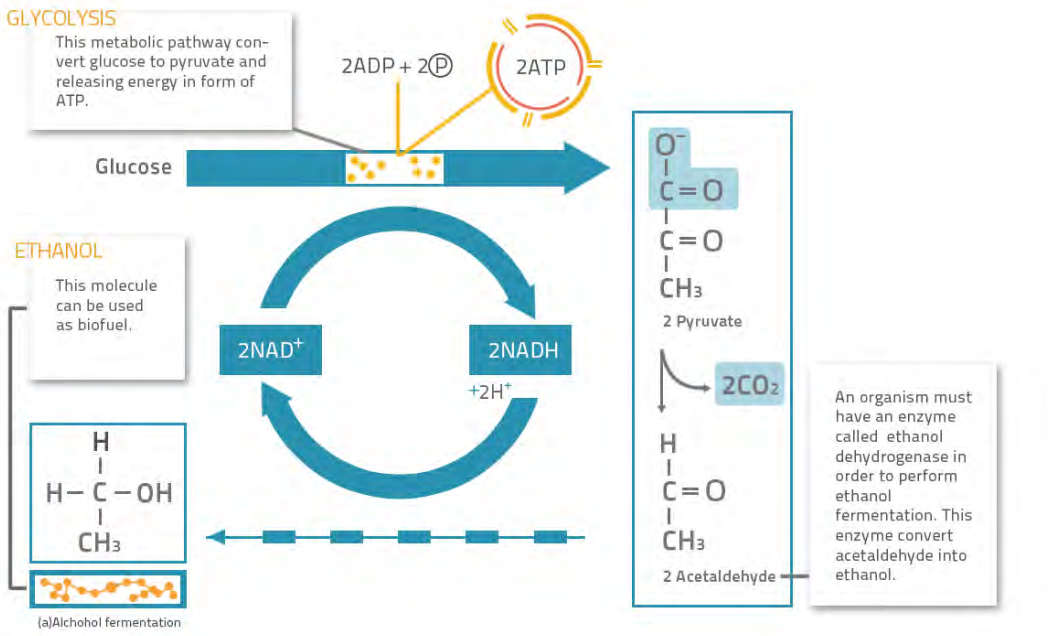
ยีสต์ (*Saccharomyces cerevisiae*) (รูปที่ 7.1) สามารถนำมาใช้ในการผลิตไวน์ได้เนื่องจากยีสต์สามารถเปลี่ยนกลูโคส (glucose) ที่อยู่ในผลไม้ให้เป็นเอทานอล (ethanol) ได้ (รูปที่ 7.2) คำถามคือถ้ายีสต์มักอยู่ที่ต้นองุ่นอยู่แล้วทำไมเวลาทำไวน์ ต้องนำองุ่นมาบดและบรรจุในถังไม้ที่มีการปิดถังอย่างแน่นหนา ยีสต์ใช้กลูโคสเป็นอาหาร เมื่อกลูโคสหมดยีสต์จะต้องมีการปรับกระบวนการภายในให้สามารถอยู่รอด ซึ่งกระบวนการนั้นคือการกลับด้านกระบวนการเมตาโบลิซึมที่เรียกว่า diauxic shift ซึ่งจะเกิดขึ้นเมื่อมีออกซิเจนเท่านั้น ถ้าไม่มีออกซิเจนยีสต์จะจำศีล (hibernate) และจะกลับมาทำงานอีกครั้งเมื่อมีออกซิเจนหรือกลูโคส อีกนัยยะหนึ่งคือถ้าผู้ผลิตไวน์ไม่ได้ปิดฝาถังบ่มไวน์ให้แน่นหนาเมื่อยีสต์ใช้กลูโคสหมดแล้วก็ยังสามารถใช้เอทานอลได้ผ่านกระบวนการ diauxic shift ทำให้เอทานอลถูกใช้ไปโดยยีสต์เองนั่นเอง



รูปที่ 7.1 ภาพถ่ายขยายเชื้อยีสต์ (Saccharomyces cerevisiae) ที่ 5 นาโนเมตร

(ที่มา: By Mogana Das Murtey and Patchamuthu Ramasamy - [1], CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=52254246)

กระบวนการ diauxic shift มีความซับซ้อนและมีผลกระทบต่อการแสดงออกของยีนจำนวนมาก รวมทั้งเกี่ยวข้องกับวิวัฒนาการในการเอาตัวรอดของเชื้อ *Saccharomyces cerevisiae* เนื่องจากเอทานอลที่ผลิตได้โดย *Saccharomyces cerevisiae* เหมือนอาวุธของมันเนื่องจากเอทานอลเป็นพิษกับเชื้อแบคทีเรียและยีสต์อื่นๆหลายชนิด นอกจากนี้ยีสต์ *Saccharomyces cerevisiae* ยังสามารถใช้เอทานอลเป็นแหล่งพลังงาน คำถามคือยีสต์ *Saccharomyces cerevisiae* พัฒนากระบวนการ diauxic shift ขึ้นมาได้อย่างไรและมียีนอะไรบ้างที่เกี่ยวข้อง



รูปที่ 7.2 กระบวนการการผลิตไวน์จากยีสต์โดยการเปลี่ยนกลูโคสที่อยู่ในผลไม้ให้เป็นเอทานอล
(ที่มา: https://theory.labster.com/ethanol-fermentation/)

การจัดกลุ่มของยีน

การวิเคราะห์การแสดงออกของยีน

ในปีค.ศ. 1997 Joseph DeRisi และคณะ [166] ออกแบบและทำการทดลองวัดการแสดงออกของยีนจำนวนมากพร้อมๆ กันในยีสต์ *S. cerevisiae* ที่เพาะเลี้ยงไว้ใน 7 ช่วงเวลาประกอบด้วย -6, -4, -2, 0, +2, +4, และ +6 ชั่วโมง โดยชั่วโมงที่ 0 คือช่วงเวลาที่เกิด diauxic shift โดยยีนที่ถูกวัดการแสดงออกมีประมาณ 6,400 ยีน ดังนั้นผลการทดลองจึงอยู่ในรูปแบบของเมทริกซ์ 6,400 x 7

หยุดคิด	เทคโนโลยีอะไรที่เราสามารถใช้ในการวัดการแสดงออกของยีนในตัวอย่างการทดลองข้างต้น

ในปีค.ศ. 1997 DeSiri และคณะใช้เทคโนโลยีไมโครอะเรย์ (microarrays) (รูปที่ 7.3) ซึ่งปัจจุบันมีการใช้งานน้อยลงเนื่องจากมีเทคโนโลยีการถอดรหัสอาร์เอ็นเอแบบ NGS (Next Generation Sequencing) ที่เรียกว่าอาร์เอ็นเอซีค (RNA-Seq) เข้ามาแทนที่ อย่างไรก็ตามระเบียบวิธีการวิเคราะห์ข้อมูลในเชิงอัลกอริทึมที่ DeSiri และคณะใช้ยังสามารถนำมาประยุกต์ใช้ในการจัดกลุ่มข้อมูลอื่นๆได้ ทั้งนี้ข้อมูลผลการทดลองต่างๆของ DeSiri และคณะสามารถเข้าถึงได้ที่ http://cmgm.stanford.edu/pbrown/explore/index.html

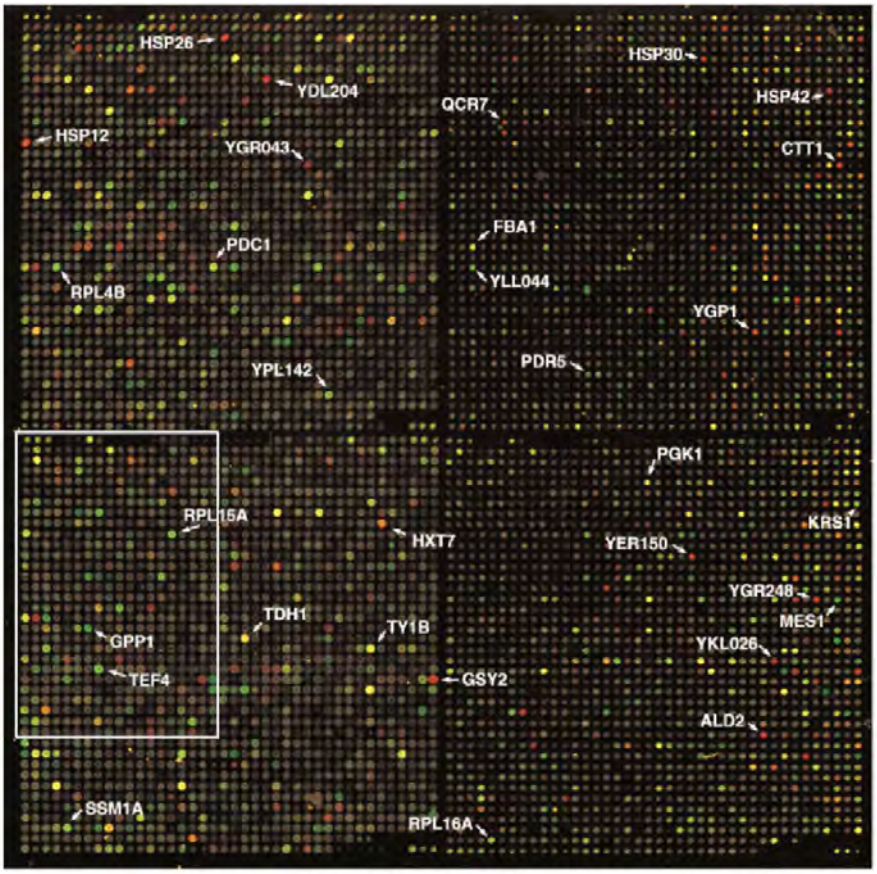


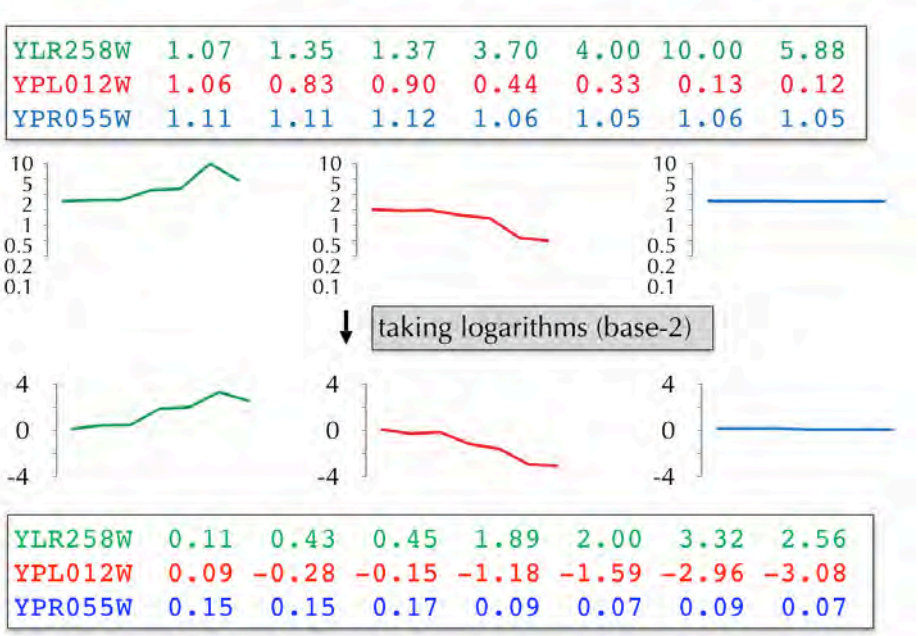
Fig. 1. Yeast genome microarray. The actual size of the microarray is 18 mm by 18 mm. The microarray was printed as described (*9*). This image was obtained with the same fluorescent scanning confocal microscope used to collect all the data we report (*49*). A fluorescently labeled cDNA probe was prepared 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 ไมโครอะเรย์ของจีโนมยีสต์จากการทดลองของ DeSiri et al.

(ที่มา: รูปที่ 1 ของ [166])

หยุดคิด	ในรูปที่ 7.4 แสดงรูปแบบการแสดงออกของยีนในยีสต์ *S. cerevisiae* จำนวน 3 ยีน ในช่วงเวลาที่แตกต่างกัน คำถามคือคิดว่ายีนไหนบ้างใน 3 ยีนนี้ที่น่าจะเกี่ยวข้องกับ diauxic shift

รูปแบบการแสดงออกของยีน YPR055W ในรูปที่ 7.4 ด้านบน มีค่าคงที่ตลอด 7 ช่วงเวลา ดังนั้นเราสามารถสรุปได้ว่ายีนนี้ไม่น่าจะเกี่ยวข้องกับ diauxic shift ในทางกลับกันการแสดงออกของยีน YLR258W มีการเปลี่ยนแปลงอย่างชัดเจนในช่วงเวลาที่ 0 ชั่วโมง ซึ่งนำไปสู่สมมติฐานว่ายีน YLR258W น่าจะเกี่ยวข้องกับ diauxic



รูปที่ 7.4 เวคเตอร์แสดงค่าการแสดงออกของยีน YLR258W, YPL012W, และ YPR055W โดยค่าด้านบนเป็นค่าที่ยังไม่ได้ใส่ลอการิทึม ส่วนค่าด้านล่างเป็นค่าที่ใส่ลอการิทึมฐานสองแล้ว

(ที่มา: รูปที่ 7.2 ของ [21])

shift ซึ่งถ้าไปตรวจสอบกับฐานข้อมูล Saccharomyces Genome Database (SGD) (https://www.yeastgenome.org) ก็จะพบว่ายีน YLR258W คือ glycogen synthase ซึ่งเป็นเอนไซม์ที่ควบคุมการผลิตไกลโคเจน (glycogen) ซึ่งก็คือกลูโคส โพลีแซคคาไรด์ (glucose polysaccharide) ซึ่งเป็นแหล่งเก็บกลูโคสในเซลล์ของยีสต์

ยีน	เวคเตอร์ค่าการแสดงออกของยีน						
YLR361C	0.14	0.03	-0.06	0.07	-0.01	-0.06	-0.01
YMR290C	0.12	-0.23	-0.24	-1.16	-1.40	-2.67	-3.00
YNR065C	-0.10	-0.14	-0.03	-0.06	-0.07	-0.14	-0.04
YGR043C	-0.43	-0.73	-0.06	-0.11	-0.16	3.47	2.64
YLR258W	0.11	0.43	0.45	1.89	2.00	3.32	2.56
YPL012W	0.09	-0.28	-0.15	-1.18	-1.59	-2.96	-3.08
YNL141W	-0.16	-0.04	-0.07	-1.26	-1.20	-2.82	-3.13
YJL028W	-0.28	-0.23	-0.19	-0.19	-0.32	-0.18	-0.18
YKL026C	-0.19	-0.15	0.03	0.27	0.54	3.64	2.74
YPR055W	0.15	0.15	0.17	0.09	0.07	0.09	0.07

รูปที่ 7.5 เมทริกซ์ขนาด 10x7 ซึ่งเป็นตัวอย่างเมทริกซ์ย่อยของเมทริกซ์ของ Desiri ขนาด 6,400x7 โดยค่าในเมทริกซ์ย่อยนี้ใส่ลอการิทึมฐาน 2 แล้ว

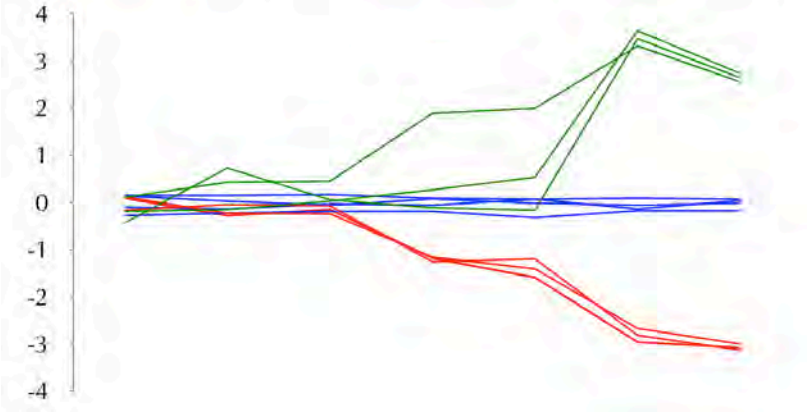
(ที่มา: รูปที่ 8.3 ของ [21])

ในทางปฏิบัตินักชีววิทยามักใช้ลอการิทึมในการปรับค่าการแสดงออกของยีน (รูปที่ 7.4 ล่าง) ซึ่งหลังปรับค่าแล้วค่าที่เป็นบวกหมายถึงการแสดงออกที่เพิ่มขึ้นในขณะที่ค่าที่เป็นลบแสดงถึงการแสดงออกที่ลดลง รูปที่ 7.5 แสดงเมทริกซ์ค่าการแสดงออกของยีนในยีสต์จำนวน 10 ยีนหลังจากที่มีการใส่ลอการิทึมแล้ว

การจัดกลุ่มของยีน

จากข้อมูลข้างต้นที่ได้จากการทดลองของ DeSiri เป้าหมายของเราคือการแบ่งกลุ่มของยีนให้ออกเป็น k กลุ่มตามรูปแบบการแสดงออกของยีน โดยแต่ละยีนต้องอยู่ในกลุ่มใดกลุ่มหนึ่งเท่านั้น ในเชิงปฏิบัติเราจะไม่ทราบจำนวนกลุ่มหรือค่า k ที่ควรจะเป็น ทั้งนี้นักชีววิทยามักลองจัดกลุ่มโดยใช้ค่า k ที่แตกต่าง และเลือกค่า k ที่ให้ผลของการจัดกลุ่มที่มีความหมายในเชิงชีววิทยา ในเนื้อหาต่อไปนี้เพื่อลดความซับซ้อนของคำอธิบาย ค่า k จะเป็นค่าที่ถูกกำหนดไว้ล่วงหน้า รูปที่ 7.6 แสดงการจัดกลุ่มของยีนในรูปที่ 7.5 โดยแบ่งออกเป็น 3 กลุ่มหลักๆ ประกอบด้วยกลุ่มที่มีการแสดงออกเพิ่ม ลด และไม่เปลี่ยนแปลงระหว่างเวลาที่ 0 ชั่วโมงที่เกิด diauxic shift

ยีน	เวคเตอร์ค่าการแสดงออกของยีน						
YLR361C	0.14	0.03	-0.06	0.07	-0.01	-0.06	-0.01
YMR290C	0.12	-0.23	-0.24	-1.16	-1.40	-2.67	-3.00
YNR065C	-0.10	-0.14	-0.03	-0.06	-0.07	-0.14	-0.04
YGR043C	-0.43	-0.73	-0.06	-0.11	-0.16	3.47	2.64
YLR258W	0.11	0.43	0.45	1.89	2.00	3.32	2.56
YPL012W	0.09	-0.28	-0.15	-1.18	-1.59	-2.96	-3.08
YNL141W	-0.16	-0.04	-0.07	-1.26	-1.20	-2.82	-3.13
YJL028W	-0.28	-0.23	-0.19	-0.19	-0.32	-0.18	-0.18
YKL026C	-0.19	-0.15	0.03	0.27	0.54	3.64	2.74
YPR055W	0.15	0.15	0.17	0.09	0.07	0.09	0.07



รูปที่ 7.6 ตัวอย่างของการจัดกลุ่มของยีนในรูปที่ 7.5 ออกเป็น 3 กลุ่มตามรูปแบบการแสดงออกของยีนที่แตกต่างกันเส้นสีเขียวมีการแสดงออกเพิ่มขึ้น สีแดงมีการแสดงออกลดลง และสีน้ำเงินไม่มีการเปลี่ยนแปลงการแสดงออก (ที่มา: รูปที่ 8.4 ของ [21])

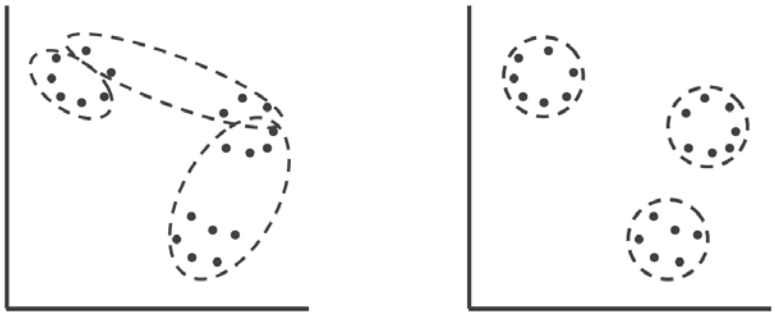
ถึงแม้ว่า diauxic shift จะเป็นกระบวนการสำคัญใน *S. cerevisiae* แต่ก็ไม่ได้เกี่ยวข้องกับฟังก์ชันหลักอื่นๆของยีสต์ ซึ่งสอดคล้องกับผลการจัดกลุ่มของยีนว่าในระหว่างที่เกิด diauxic shift ยีนส่วนใหญ่ไม่มีการเปลี่ยนแปลงระดับการแสดงออก และในการวิเคราะห์เพิ่มเติมยีนที่ไม่เกี่ยวข้องเหล่านี้จะถูกกำจัดออกไปจากเมทริกซ์ ทั้งนี้เพื่อเป็นการลดขนาดของเมทริกซ์นั่นเอง จากรูปที่ 7.6 ระดับการแสดงออกของยีนส่วนใหญ่ไม่มีการเปลี่ยนแปลงทั้งก่อนและหลังการเกิด diauxic shift (ยีนสีน้ำเงินในกราฟรูปที่ 7.6) ยีนเหล่านี้ยังมักมีค่าระดับการแสดงออกใกล้ศูนย์ในแต่ละช่วงเวลา และในการวิเคราะห์ถัดไปยีนเหล่านี้จะไม่ถูกนำมาพิจารณาต่อ

หลักเกณฑ์พื้นฐานในการจัดกลุ่มที่ดี

ในการจัดกลุ่มยีนที่มีรูปแบบการแสดงออกคล้ายคลึงกัน เราสามารถพิจารณาเวคเตอร์ของค่าระดับการแสดงออกของยีนหนึ่งๆ จำนวน m ค่า เป็นจุด 1 จุดในสเปซที่มีขนาด m มิติ และยีนที่มีค่าในเวคเตอร์ใกล้เคียงกันควรจะอยู่ใกล้ๆกันหรือเกาะกลุ่มกันภายในสเปซ m มิติ ในอุดมคติแต่ละคลัสเตอร์ (cluster) หรือกลุ่มของยีนควรจะมีลักษณะตรงตามเงื่อนไขต่อไปนี้ (และตามรูปที่ 7.7)

หลักเกณฑ์พื้นฐานในการจัดกลุ่มที่ดี : ทุกคู่ของจุดที่อยู่ในคลัสเตอร์เดียวกันควรจะมีความใกล้กันกว่าจุดที่อยู่ในคลัสเตอร์อื่นๆ

หลักเกณฑ์ข้างต้นนี้ถูกนำมารวมอยู่ในการวิเคราะห์การแสดงออกของยีนในส่วนที่จะต้องมีการจัดกลุ่มยีนที่มีลักษณะการแสดงออกไปในทิศทางเดียวกัน โดยจะมีการแบ่ง n จุด ที่อยู่ในสเปซ m มิติ ออกเป็น k กลุ่ม (คลัสเตอร์)



รูปที่ 7.7 (ซ้าย) การแบ่งจุด 20 จุดออกเป็น 3 กลุ่มโดยไม่เป็นไปตามเกณฑ์การจัดกลุ่มที่ดี (ขวา) ตัวอย่างการแบ่งกลุ่มอีกแบบที่เป็นไปตามเกณฑ์การจัดกลุ่มที่ดี

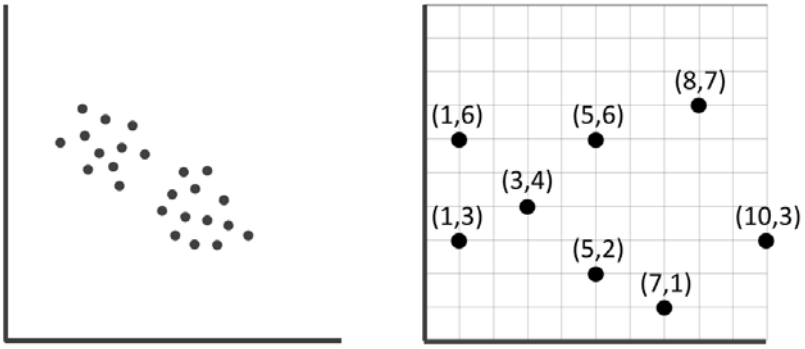
(ที่มา: รูปที่ 8.5 ของ [21])

นิยามปัญหาที่ 7.1 ปัญหาการจัดกลุ่มที่ดี

ทำการแบ่งชุดของจุดออกเป็นกลุ่มๆ	
ข้อมูลเข้า	ชุดของ n จุด ในสเปซ m มิติ และเลขจำนวนเต็ม k แสดงจำนวนกลุ่มที่ต้องการ
ผลลัพธ์	ชุดของจุดที่ถูกแบ่งออกเป็น k กลุ่มและเป็นไปตามเกณฑ์พื้นฐานในการจัดกลุ่มที่ดีข้างต้น

ฝึกหัด	สร้างจุด 10 ในสเปซ 2 มิติ จากคอลัมน์ที่ 4 และ 7 ของเมทริกซ์ในรูปที่ 7.7 คำถามเราจะสามารถแบ่ง 10 จุดนี้ออกเป็น 3 คลัสเตอร์ได้อย่างไร

ถ้าพิจารณาด้วยตา ชุดของจุดในรูปที่ 7.8 (ซ้าย) น่าจะถูกแบ่งออกเป็น 2 กลุ่ม อย่างไรก็ตาม 2 กลุ่มนี้จะไม่เป็นไปตามเกณฑ์การจัดกลุ่มที่ดี และในความเป็นจริงแล้วเราจะไม่สามารถแบ่งชุดของจุดในตัวอย่างนี้ออกเป็น 2 กลุ่มที่เป็นไปตามเกณฑ์การจัดกลุ่มที่ดีได้



รูปที่ 7.8 (ซ้าย) ชุดของจุดที่สามารถแบ่งด้วยตาได้ออกเป็น 2 กลุ่มอย่างชัดเจน อย่างไรก็ตามเราไม่สามารถแบ่งจุดชุดนี้ออกเป็น 2 กลุ่มเพื่อให้เป็นไปตามเกณฑ์การจัดกลุ่มที่ดีได้ (ขวา) ตัวอย่างจุด 8 จุดในสเปซ 2 มิติ (ที่มา: รูปที่ 8.6 ของ [21])

ฝึกหัด	ออกแบบอัลกอริทึมที่ใช้ในการตรวจสอบว่า เราสามารถแบ่งกลุ่มชุดของจุดที่เป็นข้อมูลเข้าออกเป็น k กลุ่มและเป็นไปตามเกณฑ์การจัดกลุ่มที่ดีได้หรือไม่ โดยอัลกอริทึมต้องทำงานในเวลาโพลีโนเมียล (polynomial time)

หยุดคิด	รูปที่ 7.8 ทางขวาแสดงข้อมูลจุด 8 จุดในสเปซ 2 มิติ เราจะสามารถแบ่ง 8 จุดนี้ออกเป็น 3 กลุ่มได้อย่างไรบ้าง และเราจะสามารถแปลงปัญหาการจัดกลุ่มที่ดี (Good Clustering Problem) ให้อยู่ในรูปแบบของปัญหาเชิงคำนวณที่ชัดเจนมากขึ้นได้อย่างไร

แปลงปัญหาการแบ่งกลุ่มข้อมูลเป็นปัญหาออปติไมเซชัน

จากปัญหาการแบ่งกลุ่มข้อมูลที่ดีข้างต้น ที่พยายามแบ่งข้อมูล n จุด (*Data*) ออกเป็น k กลุ่ม เราจะเปลี่ยนแนวทางการแก้ปัญหาเป็นการเลือกชุดของศูนย์กลาง (*Centers*) ของคลัสเตอร์จำนวน k จุด โดยต้องหา Centers ที่ทำให้ระยะทาง (distance) รวมระหว่าง Centers ใดๆ ไปยังจุดต่างๆ ใน *Data* มีค่าน้อยที่สุด คำถามที่สำคัญคำถาม

หนึ่งคือเรากำหนดฟังก์ชันที่ใช้ในการวัดระยะทางอย่างไร

ขั้นแรก เรากำหนดระยะทางยูคลิเดียน (Euclidian distance) ระหว่างจุด v = (v_1,...,v_m) และ w = (w_1,...w_m) ในสเปซ 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 มีข้อมูล 1 จุด เรียกว่า *DataPoint* ในสเปซ m มิติ และ *Centers* จำนวน k จุด เรากำหนดระยะทางระหว่าง *DataPoint* ไปยัง *Centers* แสดงโดย *d(DataPoint, Centers)* เป็นระยะทางยูคลิเดียนจาก *DataPoint* ไปยังจุดศูนย์กลาง (center) ที่ใกล้ที่สุด ดังนั้น

$$d(DataPoint, Centers) = \min_{all\ points\ x\ from\ Centers} d(DataPoint, x)$$

ถึงจุดนี้เราสามารถกำหนดระยะทางระหว่างทุกจุดใน *Data* กับ จุดศูนย์กลางใน *Centers* ระยะทางนี้แสดงโดย MAXDISTANCE(*Data, Centers)* ซึ่งเป็นค่าที่มากที่สุดของ *d(DataPoint, Centers)* ระหว่างทุกจุดข้อมูล *DataPoint*

$$MAXDISTANCE(Data, Centers) = \max_{all\ point\ DataPoint\ from\ Data} d(DataPoint, Centers)$$

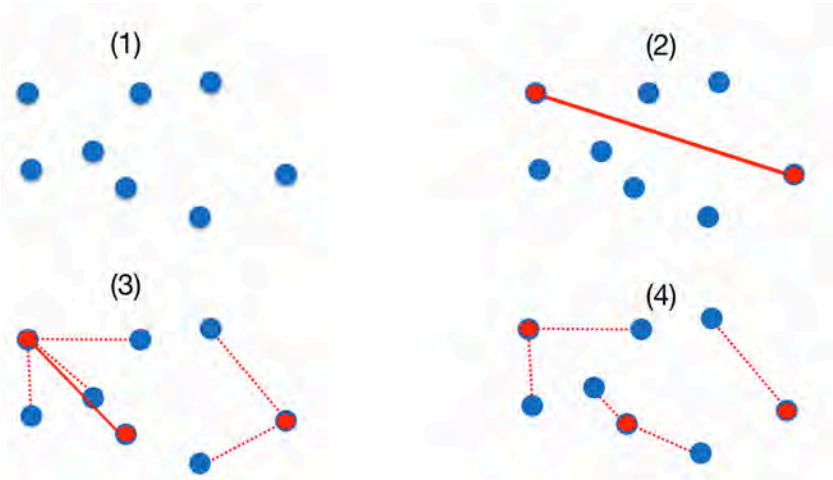
นิยามปัญหาที่ 7.2 ปัญหาการจัดกลุ่มข้อมูลแบบ k-Center

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

ถึงแม้ว่าปัญหาการจัดกลุ่ม k-Center จะดูง่าย ความซับซ้อนของการแก้ปัญหากลับอยู่ในระดับ NP-Hard เพื่อให้การแก้ปัญหามีความซับซ้อนน้อยลงจึงได้มีการเพิ่มฮิวริสติก (Heuristic) เข้ามาโดยเลือกจุดใดๆ จาก *Data* (แทนที่จะเลือกจากจุดในสเปซ m มิติ) โดยเลือกมาแบบสุ่มและเพิ่มจุดนั้นๆเข้าไปใน *Centers* และทำซ้ำโดยการเลือกจุดศูนย์กลางถัดไปจาก *Data* ที่มีระยะห่างที่สุดจากจุดศูนย์กลางทุกจุดที่ถูกเลือกมาก่อนหน้า ดังแสดงในรูปที่ 7.9

และสูโดโคดที่ 7.1



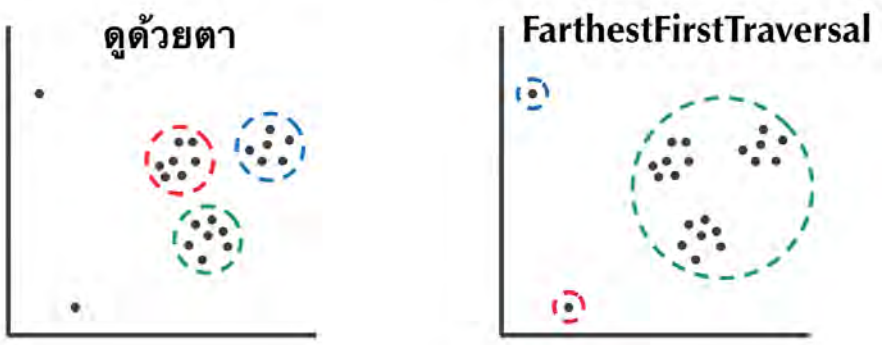
รูปที่ 7.9 แสดงผลการประยุกต์ใช้วิธี FarthestFirstTraversal () ในการจัดกลุ่มข้อมูล โดยจุดสีแดงในขั้นตอนที่ (2), (3) และ (4) เป็นจุดศูนย์กลางที่ถูกเลือกและเพิ่มเข้ามาใน Centers ในแต่ละรอบ

สูโดโคดที่ 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
```

วิธีการ FarthestFirstTraversal เร็วและผลของการจัดกลุ่มก็มีความใกล้เคียงกับผลลัพธ์ที่ดีที่สุดของการจัดกลุ่มแบบ k-Center อย่างไรก็ตามการจัดกลุ่มของยีนตามรูปแบบการแสดงออกจะไม่ใช้อัลกอริทึมนี้ ในกรณีของการจัดกลุ่มแบบ k-Center เราจะเลือกชุดของจุดศูนย์กลาง *Centers* ที่ทำให้ MAXDISTANCE(*Data, Centers*) มีค่าน้อยที่สุด ซึ่งค่า MAXDISTANCE ก็คือระยะทางที่มากที่สุดระหว่างจุดใดๆกับจุดศูนย์กลางที่ใกล้ที่สุด อย่างไรก็ตามนักชีววิทยามักสนใจความแตกต่างโดยรวมๆ (typical) มากกว่าการแตกต่างที่มากที่สุด เนื่องจากในบางกรณีความแตกต่างที่มากที่สุดอาจเป็นเพียงข้อมูลที่มีสัญญาณรบกวน รูปที่ 7.10 (ขวา) จากผลของ MAXDISTNACE() ไม่ว่าจุดใดจะถูกเลือกเป็นจุดศูนย์กลางแรก จุดที่เป็นสัญญาณรบกวนทางซ้ายมือ 2 จุด จะถูกเลือกเป็นจุดศูนย์กลางของกลุ่มที่ 2 และ 3 ที่มีสมาชิกจุดเดียวคือจุดศูนย์กลางเอง

หยุดคิด	จากปัญหาของ FarthestFirstTraversal ข้างต้น เราสามารถเปลี่ยนแปลงฟังก์ชันการให้คะแนนที่ใช้ MAXDISTANCE เป็นฟังก์ชันอื่นที่สามารถสะท้อนการพิจารณาข้อมูลความแตกต่างของสมาชิกในกลุ่มหนึ่งแบบรวมๆ จากมุมมองของนักชีววิทยาได้อย่างไร



รูปที่ 7.10 (ซ้าย) ชุดของข้อมูลที่เห็นได้ชัดเจนด้วยตาว่าสามารถแบ่งได้เป็น 3 กลุ่มและมีจุดข้อมูล 2 จุดที่เป็นสัญญาณรบกวน (ขวา) เนื่องจาก FarthestFirstTraversal ใช้ MAXDISTANCE ในการหา Centers (ที่มา: รูปที่ 8.9 ของ [21])

การจัดกลุ่มข้อมูลแบบ k-Means

Squared error distortion

เพื่อเป็นการลดข้อจำกัดของการใช้ MAXDISTANCE ได้มีการเสนอฟังก์ชันการให้คะแนนแบบใหม่โดยชุดของจุดใน *Data* ที่เป็นข้อมูลเข้า n จุด และชุดของจุดศูนย์กลาง k จุดใน *Centers* ค่า **squared error distortion** ของ *Data* และ *Centers* แสดงโดย *DISTORTION(Data, Centers)* คำนวณจากผลรวมเฉลี่ยของระยะทางระหว่างแต่ละ *DataPoint* ไปยังจุดศูนย์กลางที่ใกล้ที่สุดยกกำลังสอง ดังสมการต่อไปนี้

$$DISTORTION(Data, Centers) = \frac{1}{n} \sum_{all\ points\ DataPoint\ in\ Data} d(DataPoint, Centers)^2$$

ในขณะที่ MAXDISTANCE(*Data, Centers*) ใช้ระยะทางที่ยาวที่สุดจากจุดใดๆ ไปยังจุดศูนย์กลางที่ใกล้ที่สุด ในกรณีของ squared error distortion ใช้ระยะทางเฉลี่ยของทุกจุดไปยังจุดศูนย์กลางที่ใกล้ที่สุดของจุดเหล่านั้น

นิยามปัญหาที่ 7.3 ปัญหา Squared Error Distortion

คำนวณค่า squared error distortion จากชุดข้อมูล และ ชุดของจุดศูนย์กลาง	
ข้อมูลเข้า	ชุดของจุดข้อมูล *Data* และชุดของจุดศูนย์กลาง *Centers*
ผลลัพธ์	ค่า squared error distortion DISTORTION(*Data, Centers*)

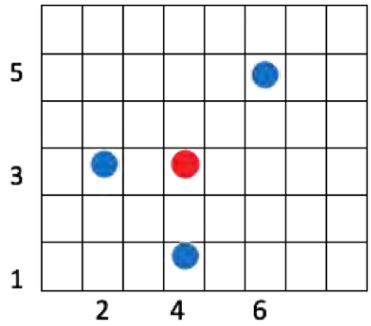
ค่า squared error distortion นำไปสู่การปรับปรุงการแก้ปัญหาการจัดกลุ่มข้อมูลแบบ k-Center

นิยามปัญหาที่ 7.4 ปัญหาการจัดกลุ่มข้อมูลแบบ k-Means

รับข้อมูลเข้าเป็นชุดของจุด หาจุดศูนย์กลาง k จุดที่ทำให้ค่า squared error distortion มีค่าน้อยที่สุด	
ข้อมูลเข้า	ชุดของจุดข้อมูล *Data* และค่าจำนวนเต็ม k
ผลลัพธ์	ชุดของ Centers จำนวน k จุด ที่ทำให้ค่า DISTORTION(*Data, Centers*) มีค่าน้อยที่สุดสำหรับทุกค่า k Centers ที่เป็นไปได้

การจัดกลุ่มข้อมูลแบบ k-Means และจุดศูนย์ถ่วง

ปัญหาการจัดกลุ่มข้อมูลแบบ k-Means เป็นปัญหา NP-Hard เมื่อ k > 1 ในกรณีที่ k = 1 อย่างไรก็ตามจะเท่ากับการหาจุดศูนย์กลาง x ที่ทำให้ค่า squared error distortion มีค่าน้อยที่สุด ถึงแม้ว่าการแบ่งชุดของข้อมูลโดยค่า k = 1 จะตรงไปตรงมา แต่ยังไม่ชัดเจนว่าเราจะหาจุดศูนย์กลางที่ทำให้ได้ค่า squared error distortion มีค่าน้อยที่สุดอย่างไร การพิจารณาวิธีการตอบคำถามนี้จะช่วยให้สามารถออกแบบวิธีการหาคำตอบในกรณีที่ k > 1 ด้วย



รูปที่ 7.11 จุดข้อมูล (สีน้ำเงิน) และจุดศูนย์ถ่วง (สีแดง) ที่คำนวณจากทั้ง 3 จุดข้อมูล

เรากำหนดจุดศูนย์ถ่วง (center of gravity) ของชุดข้อมูล *Data* ค่าในเวคเตอร์ลำดับที่ i ของจุดศูนย์ถ่วงสามารถคำนวณได้จากผลรวมเฉลี่ยของค่าลำดับที่ i ของทุกจุดที่อยู่ใน *Data* ตัวอย่างในรูปที่ 7.11 เราสามารถคำนวณค่าจุดศูนย์ถ่วงของจุด (2,3), (4,1) และ (6,6) มีค่าเท่ากับ

$$\left(\frac{2+4+6}{3}, \frac{3+1+5}{3}\right) = (4,3)$$

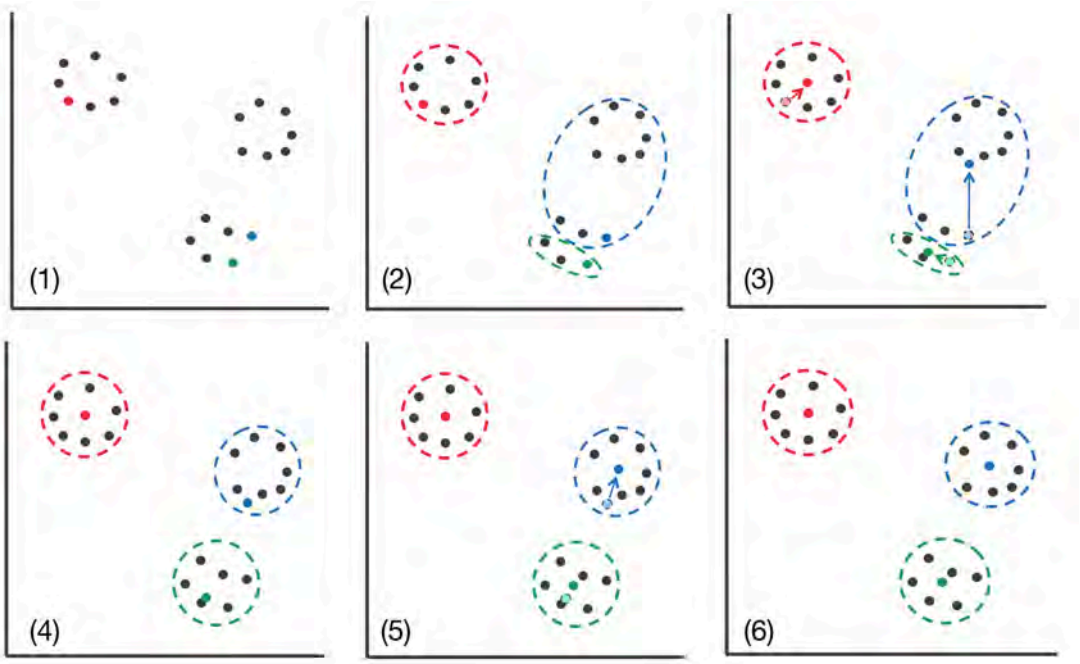
ทฤษฎีจุดศูนย์ถ่วง (Center of Gravity Theorem) : จุดศูนย์ถ่วงของชุดข้อมูลใน *Data* จะมีเพียงจุดเดียวซึ่งสามารถใช้ในการแก้ปัญหาการจัดกลุ่มแบบ k-Means ในกรณีที่ k = 1

อัลกอริทึม Lloyd

อัลกอริทึม Lloyd (รูปที่ 7.12) เป็นฮิวริสติสที่ถูกใช้ในการจัดกลุ่มข้อมูลแบบ k-Means ที่มีการใช้งานอย่างแพร่หลาย โดยในขั้นตอนแรกจะทำการเลือกจุดแบบสุ่มจาก *Data* จำนวน k จุดเพื่อเป็น *Centers* และทำการวน

ซ้ำ 2 ขั้นตอนต่อไปนี้

- **จากชุดของจุดศูนย์กลางไปยังชุดของคลัสเตอร์** จากชุดของจุดศูนย์กลางที่ถูกเลือกไว้ ทำการกำหนดกลุ่มหรือคลัสเตอร์ให้กับจุดข้อมูลอื่นๆ โดยจุดเหล่านี้จะไปเป็นสมาชิกของกลุ่มที่จุดศูนย์กลางของกลุ่มนั้นๆ มีระยะห่างจากจุดข้อมูลน้อยสุด
- **จากชุดของคลัสเตอร์ไปยังชุดของจุดศูนย์กลาง** หลังจากจุดข้อมูลต่างๆ ได้ถูกกำหนดให้ไปอยู่ในคลัสเตอร์ หนึ่งๆ แล้ว ทำการคำนวณจุดศูนย์ถ่วงใหม่สำหรับแต่ละคลัสเตอร์และใช้เป็นชุดของจุดศูนย์ถ่วงใหม่ในการกำหนดสมาชิกให้ในรอบถัดไป



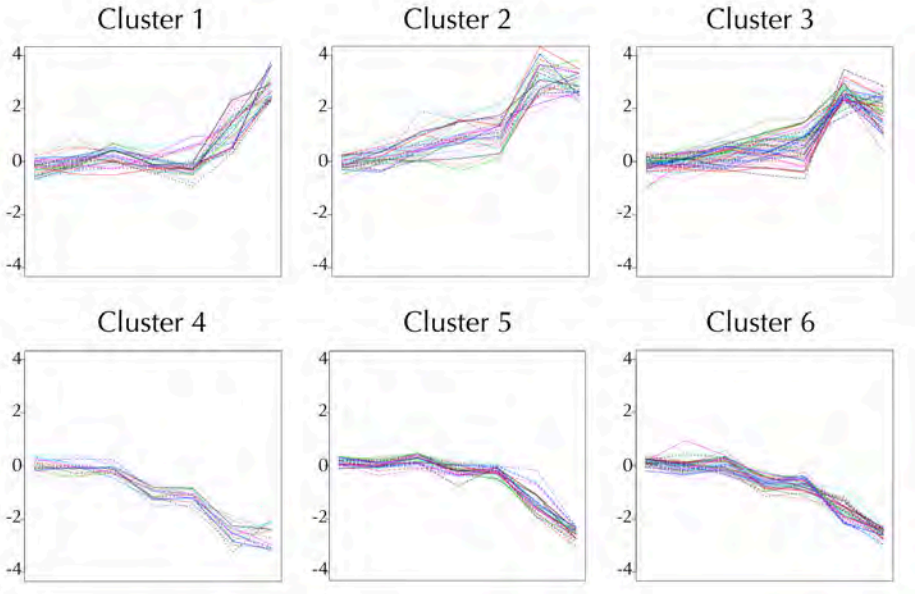
รูปที่ 7.12 การทำงานของอัลกอริทึม Lloyd ในแต่ละขั้นตอนโดย k = 3

(ที่มา: รูปที่ 8.12 ของ [21])

กลุ่มยีนตามรูปแบบการแสดงออกนำไปสู่ยีนที่เกี่ยวข้องกับ diauxic shift

เนื่องจากการเลือกค่า k ที่เหมาะสมและสอดคล้องกับองค์ความรู้ทางชีววิทยาสามารถเป็นปัญหาที่ท้าทายและอยู่นอกเหนือขอบเขตของบทเรียนนี้ รูปที่ 7.13 แสดงผลของการจัดกลุ่ม 230 ยีนของยีสต์ออกเป็น 6 กลุ่ม (k = 6) ซึ่งประกอบด้วย 37, 36, 58, 19, 36, และ 44 ยีนตามลำดับ โดยกราฟของแต่ละกลุ่มแสดงรูปแบบของระดับการแสดงออกของยีนที่แตกต่างกัน ที่เกี่ยวข้องกับ diauxic shift และเป็นจุดเริ่มต้นของคำถามทางชีววิทยาอื่นๆ เพิ่มเติม เช่น ชุดของยีนที่อยู่ในกลุ่มเดียวกันที่มีรูปแบบการแสดงคล้ายคลึงกันอาจถูกควบคุมโดยทรานสคริปชันแฟคเตอร์ (transcription factor: TF) เดียวกัน ซึ่งหมายถึงสายดีเอ็นเอในส่วนที่อยู่ก่อนหน้ายีนเหล่านี้น่าจะมี regulatory motif ที่มีรูปแบบเดียวกันหรือใกล้เคียงกัน (ตัวอย่างอัลกอริทึมพื้นฐานที่ใช้ในการการหา regulatory

motifs อยู่ในบทที่ 4) คำถามทางชีววิทยาอื่นๆ เช่น มีกระบวนการอะไรอยู่เบื้องหลังในการเพิ่มระดับการแสดงออกของยีนในคลัสเตอร์ที่ 1 หรือมีกระบวนการอะไรที่ถูกใช้ในการลดการแสดงออกของยีนในคลัสเตอร์ที่ 4 และการเปลี่ยนแปลงระดับการแสดงออกของยีนเหล่านี้มีความเกี่ยวข้องกับ diauxic shift อย่างไรบ้าง



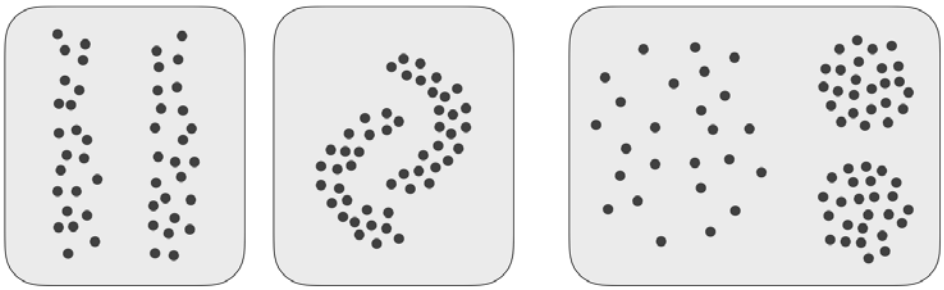
รูปที่ 7.13 ผลของการใช้อัลกอริทึม Lloyd ในการจัดกลุ่ม 230 ยีนของยีสต์ออกเป็น 6 กลุ่ม

(ที่มา: รูปที่ 8.14 ของ [21])

ข้อจำกัดของการจัดกลุ่มข้อมูลแบบ k-Means

หลังจากศึกษาและได้เห็นขั้นตอนการทำงานของอัลกอริทึม Lloyd แล้ว อาจดูเหมือนว่าการจัดกลุ่มข้อมูลเป็นเรื่องง่าย

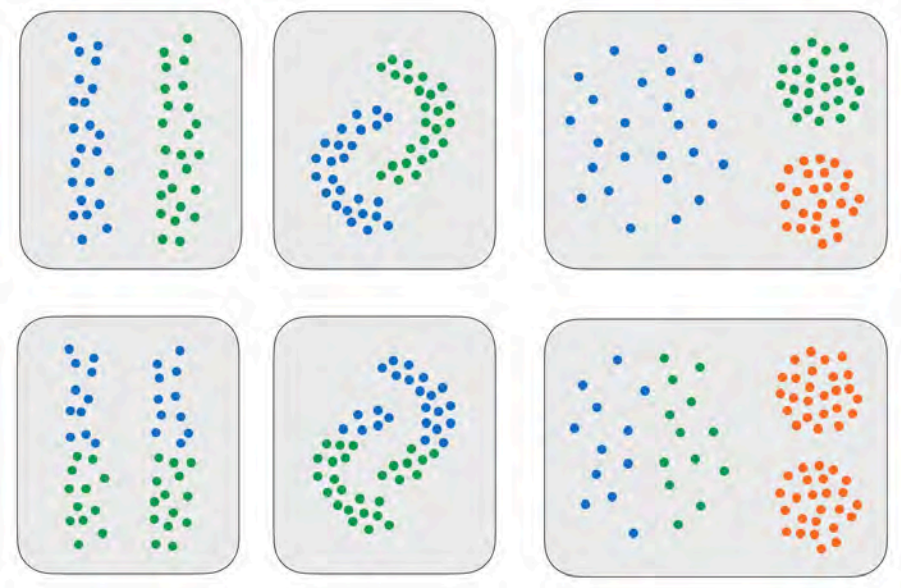
หยุดคิด	เราจะจัดกลุ่มของจุดในรูปที่ 7.14 อย่างไร



รูปที่ 7.14 ความท้าท้ายของปัญหาการจัดกลุ่มข้อมูล เมื่อ k = 2 สำหรับชุดข้อมูลในรูปซ้ายและรูปตรงกลาง และ k = 3 สำหรับชุดของข้อมูลรูปขวา

(ที่มา: รูปที่ 8.15 ของ [21])

จากชุดข้อมูลที่แสดงในรูปที่ 7.14 อัลกอริทึม Lloyd ไม่สามารถจัดกลุ่มข้อมูลได้ถูกต้องดังแสดงในรูปที่ 7.15 (ล่าง) โดยในกรณีที่การกระจายของจุดในคลัสเตอร์อยู่ในพื้นที่ที่เป็นสายที่ยาวออกไป รูปที่ 7.15 (ล่างซ้าย) คลัสเตอร์ที่การกระจายของจุดข้อมูลไม่ได้อยู่ในพื้นที่วงกลมหรือทรงกลม รูปที่ 7.15 (ล่างกลาง) และคลัสเตอร์ที่มีความหนาแน่นของจำนวนจุดที่แตกต่างกันออกไป รูปที่ 7.15 (ล่างขวา)



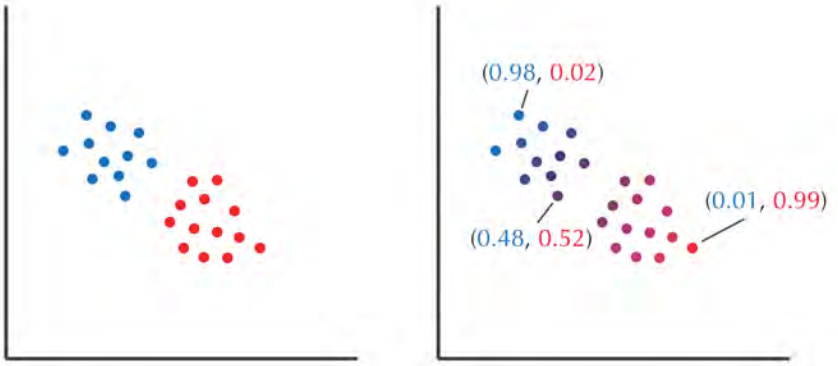
รูปที่ 7.15 (บน) ผลการจัดกลุ่มของจุดโดยใช้สายตา (ล่าง) ผลการจัดกลุ่มของจุดโดยใช้อัลกอริทึม Lloyd (ที่มา: รูปที่ 8.16 ของ [21])

ข้อจำกัดหนึ่งของการจัดกลุ่มแบบ k-Means ตามที่ได้อธิบายไปหัวข้อก่อนหน้า คือต้องมีกำหนดจุดหนึ่งๆ ไปยังกลุ่มหรือคลัสเตอร์ใดคลัสเตอร์หนึ่งเท่านั้น จุดหนึ่งจะอยู่มากกว่า 1 กลุ่มหรือ 1 คลัสเตอร์ไม่ได้ ซึ่งเงื่อนไขนี้เป็นเงื่อนไขของการจัดกลุ่มแบบ Hard clustering ซึ่งเงื่อนไขนี้จะไม่เหมาะสมกับจุดข้อมูลที่เป็น midpoints หรือจุดข้อมูลที่มีความใกล้เคียงกับกลุ่มข้อมูลมากกว่า 1 กลุ่ม วิธีการจัดการปัญหานี้จะเปลี่ยนวิธีการกำหนดกลุ่มให้กับจุดข้อมูลใดๆ โดยจุดข้อมูลสามารถถูกกำหนดให้อยู่ในกลุ่มข้อมูลได้มากกว่า 1 กลุ่ม โดยจะถูกกำกับด้วยค่าความน่าจะเป็นในการเป็นสมาชิกของกลุ่มนั้นๆ (รูปที่ 7.16)

การจัดกลุ่มข้อมูลแบบ Soft k-Means

การประยุกต์ใช้ Expectation Maximization ในการจัดกลุ่มข้อมูล

ในหัวข้อนี้แสดงการประยุกต์ใช้อัลกอริทึม Expectation Maximization (EM) ในการทำงานของอัลกอริทึม Lloyd เพื่อให้สามารถจัดกลุ่มข้อมูลแบบ Soft k-Means ได้ โดยอัลกอริทึมใหม่นี้เริ่มการทำงานโดยการสุ่มเลือกชุดของจุดศูนย์กลางจำนวน k จุดและทำการวนซ้ำ 2 ขั้นตอนต่อไปนี้



รูปที่ 7.16 (ซ้าย) ชุดของจุดจากรูปที่ 7.8 (ซ้าย) ที่ถูกแบ่งออกเป็น 2 กลุ่มโดยใช้อัลกอริทึม Lloyd (ขวา) แสดงผลของการจัดกลุ่มข้อมูลแบบ Soft โดยใช้ข้อมูลชุดเดียวกันและมี k = 2 เช่นกัน

(ที่มา: รูปที่ 8.17 ของ [21])

- **จากชุดของจุดศูนย์กลางไปยังชุดของซอฟต์คลัสเตอร์ (E-step):** จากชุดของจุดศูนย์กลางที่ถูกเลือกไว้ทำการกำหนดค่าความน่าจะเป็นที่จะเป็นสมาชิกของคลัสเตอร์หนึ่งๆให้กับจุดข้อมูลอื่นๆ โดยค่าความน่าจะเป็นที่มากกว่าหมายถึงจุดนั้นๆมีโอกาสอยู่ในกลุ่มหรือคลัสเตอร์นั้นมากกว่า
- **จากชุดของซอฟต์คลัสเตอร์ไปยังชุดของจุดศูนย์กลาง (M-step):** หลังจากจุดข้อมูลต่างๆ ได้ถูกกำหนดชุดค่าความน่าจะเป็นแล้ว ทำการคำนวณชุดของจุดศูนย์ถ่วงใหม่ และใช้เป็นชุดของจุดศูนย์ถ่วงในการกำหนดค่าความน่าจะเป็นในการเป็นสมาชิกของจุดข้อมูลต่างๆในรอบถัดไป

จากชุดของจุดศูนย์กลางไปยังการจัดกลุ่มแบบซอฟต์

เราเริ่มขั้นตอนการจัดกลุ่มจากชุดของจุดศูนย์กลางไปยังชุดของซอฟต์คลัสเตอร์ และที่ผ่านมาใช้ **“จุดศูนย์ถ่วง”** เป็นค่าของจุดศูนย์กลาง ถ้าคิดว่าจุดศูนย์กลางเหล่านี้ก็คือดวงดาว ในขณะที่จุดข้อมูลอื่นๆคือดาวบริวาร จุดที่อยู่ใกล้จุดศูนย์กลางหนึ่งๆ ก็จะมีแรงดึงดูดมากกว่า ถ้ากำหนดให้มีจุดศูนย์กลาง k จุด *Centers* = (x_1,...,x_k) และมีชุดของจุดข้อมูล n จุด *Data* = (*Data*$_1$,...,*Data*$_n$) เราสามารถที่สร้างเมทริกซ์ความรับผิดชอบ *HiddenMatrix* ขนาด k x n โดยที่ *HiddenMatrix*$_{i,j}$ เก็บค่าแรงดึงดูดระหว่างจุดศูนย์กลาง i กับจุดข้อมูล j โดยค่าแรงดึงดูดนี้คำนวณจากกฎแรงดึงดูดของนิวตัน (Newtonian inverse-square)

$$HiddenMatrix_{i,j} = \frac{\frac{1}{d(Data_j, x_i)^2}}{\sum_{all\ centers\ x_i} \frac{1}{d(Data_j, x_i)^2}}$$

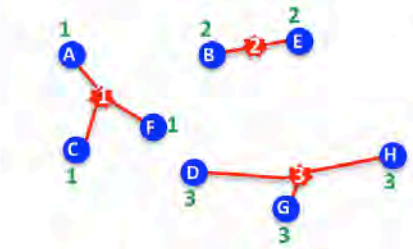
อย่างไรก็ตาม พาร์ทิชันฟังก์ชัน (partition function) จากฟิสิกส์สถิติต่อไปนี้ ใช้งานได้ดีกว่าในเชิงปฏิบัติ

$$HiddenMatrix_{i,j} = \frac{e^{-\beta \cdot d(Data_j, x_i)}}{\sum_{all\ centers\ x_i} e^{-\beta \cdot d(Data_j, x_i)}}$$

โดยในสมการนี้ e เป็นค่าฐานของลอการิทึมธรรมชาติ $(e \approx 2.718)$ และ β เป็นค่าพารามิเตอร์ที่สะท้อนความยืดหยุ่นในการกำหนดค่าความน่าจะเป็น โดยถูกเรียกว่า stiffness parameter รูปที่ 7.17 แสดงตัวอย่าง *HiddenMatrix* ของจุด 8 จุดที่ถูกแบ่งออกเป็น 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 ของจุด 8 จุดที่ถูกแบ่งออกเป็น 3 กลุ่ม

จากชุดของซอฟต์คลัสเตอร์ไปยังชุดของจุดศูนย์กลาง

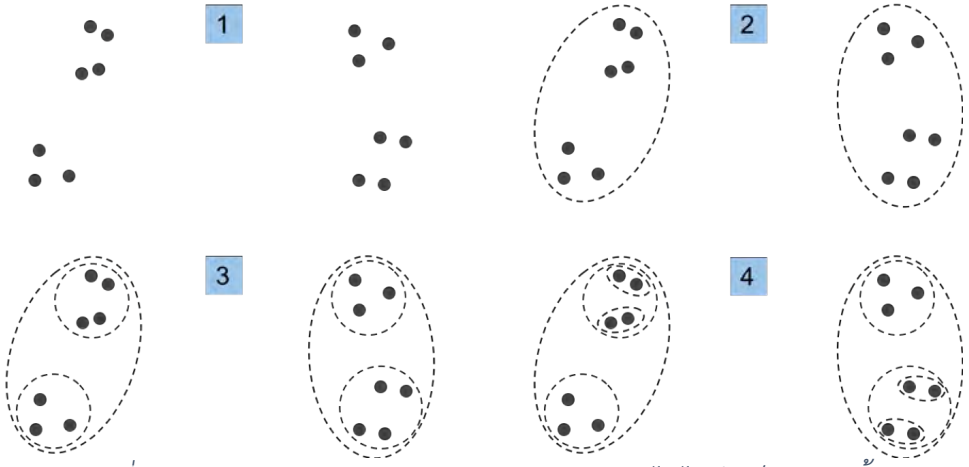
ในการจัดกลุ่มข้อมูลแบบ Soft k-Means ถ้าเรากำหนดให้ $HiddenMatrix_i$ แสดงบรรทัดที่ i ของ *HiddenMatrix* ดังนั้นเราสามารถอัพเดตค่าจุดศูนย์กลาง x_i โดยใช้สมการต่อไปนี้

$$x_{i,j} = \frac{HiddenMatrix_i \cdot Data^j}{HiddenMatrix_i \cdot \vec{1}}$$

โดย $Data^j$ เป็นเวคเตอร์ขนาด n มิติ ที่เก็บค่าโคออดิเนต (coordinate) ลำดับที่ j ของจุดข้อมูล n จุด ทั้งนี้จุดศูนย์กลาง x_i ที่มีการอัพเดตข้อมูลแล้วเรียกว่า **weighted center of gravity** ของจุดข้อมูลใน *Data*

การจัดกลุ่มข้อมูลแบบลำดับชั้น

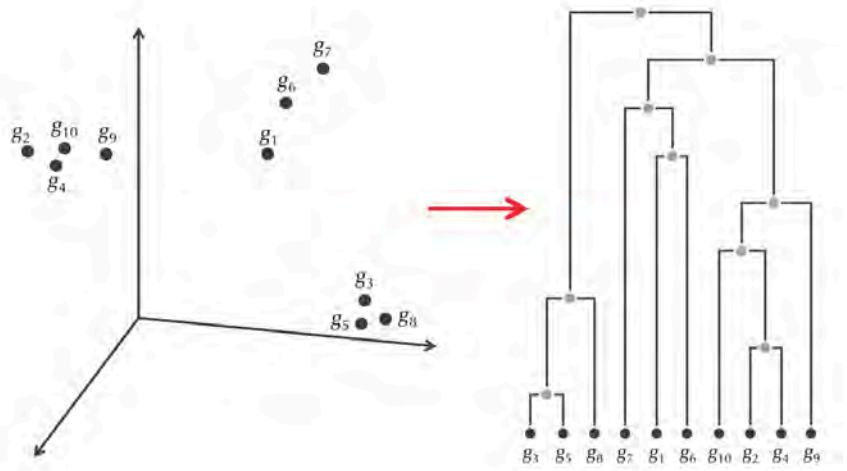
ในหัวข้อที่ผ่านมา สมมติฐานหลักในการจัดกลุ่มข้อมูลคือเราทราบจำนวนกลุ่มข้อมูล (ทราบค่า k) อย่างไรก็ตาม ในทางปฏิบัติ กลุ่มข้อมูลมักจะสามารถแบ่งแยกออกลงไปได้อีกเป็นลำดับชั้น เพื่อให้สามารถเห็นความสัมพันธ์ของกลุ่มข้อมูลย่อยเหล่านี้ดังแสดงในรูปที่ 7.18 โดยอัลกอริทึมในการจัดกลุ่มข้อมูลแบบลำดับชั้น (hierarchical clustering) นี้จะใช้เมทริกซ์ระยะทาง *D* ขนาด n x n ในการสร้างลำดับชั้นระยะห่างระหว่างจุดข้อมูล โดยสร้างจากจุดข้อมูลสองจุดที่มีความใกล้กันที่สุดก่อนและทำขึ้นมาเป็นลำดับชั้นจนถึงรูท (root) โดยผลการจัดกลุ่มข้อมูลแบบลำดับชั้นนี้จะอยู่ในรูปแบบของต้นไม้ (tree) ทั้งนี้โหนดใบในต้นไม้คือยีนส่วนโหนดภายในแสดงคลัสเตอร์หรือกลุ่มข้อมูล (รูปที่ 7.19)



รูปที่ 7.18 กลุ่มข้อมูลมักจะสามารถแบ่งแยกออกลงไปได้อีกเป็นลำดับชั้น

Distance matrix

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	0.7	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1	9.2	11.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	8.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.1	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	10.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	11.6	8.5	9.3	12.4	1.1	0.0



รูปที่ 7.19 (บน) ตัวอย่างเมทริกซ์ระยะทางที่สร้างจากระยะทางยูคลิด (Euclidian distance) (ล่างซ้าย) เวคเตอร์ระดับการแสดงออกของยีนที่แสดงด้วยจุดในสเปซ 3 มิติ (ล่างขวา) ต้นไม้ที่เป็นผลของการจัดกลุ่มข้อมูลแบบลำดับชั้นโดยใช้ข้อมูลเมทริกซ์ระยะทางด้านบน

(ที่มา: รูปที่ 8.22 ของ [21])

อัลกอริทึมในการจัดกลุ่มข้อมูลแบบลำดับชั้น

อัลกอริทึมในการจัดกลุ่มข้อมูลแบบลำดับชั้น (hierarchical clustering algorithm) พิจารณาข้อมูลเข้า n จุด เป็นข้อมูล n กลุ่ม จากนั้นจะทำการรวมข้อมูล 2 จุดที่มีระยะทางใกล้กันที่สุดเป็นกลุ่มใหม่ ทำการคำนวณหาตัวแทนข้อมูลของกลุ่มใหม่และวนซ้ำเพื่อทำการรวมสองจุดที่ใกล้กันที่สุดในรอบถัดๆไป วนซ้ำจนกระทั่งข้อมูลทั้ง n จุด รวมเป็นกลุ่มเดียวดังแสดงในสูโดโคดที่ 7.2 ทั้งนี้ในสูโดโคดนี้ยังไม่มีการกำหนดวิธีการคำนวณค่าระยะทางระหว่างกลุ่มใหม่ที่เกิดขึ้นกับกลุ่มเดิมทั้งหมดที่มีอยู่ $D(C_{new}, C)$ ในทางปฏิบัติแต่ละอัลกอริทึมอาจมีวิธีการคำนวณค่าระยะทางที่แตกต่างกันไปซึ่งอาจทำให้ได้ผลลัพธ์ของการจัดกลุ่มที่แตกต่างกันอย่างมาก

วิธีการพื้นฐานในการคำนวณค่าระยะทางวิธีการหนึ่งที่มีการใช้งานกันอย่างแพร่หลายคือการกำหนดค่าระยะทางระหว่างคลัสเตอร์ C_1 และ C_2 เท่ากับระยะทางที่สั้นที่สุดระหว่างทุกคู่ของสมาชิกระหว่างสองคลัสเตอร์ ดังสมการต่อไปนี้

$$D_{min}(C_1, C_2) = \min_{all\ points\ i\ in\ cluster\ C_1, all\ points\ j\ in\ cluster\ C_2} D_{i,j}$$

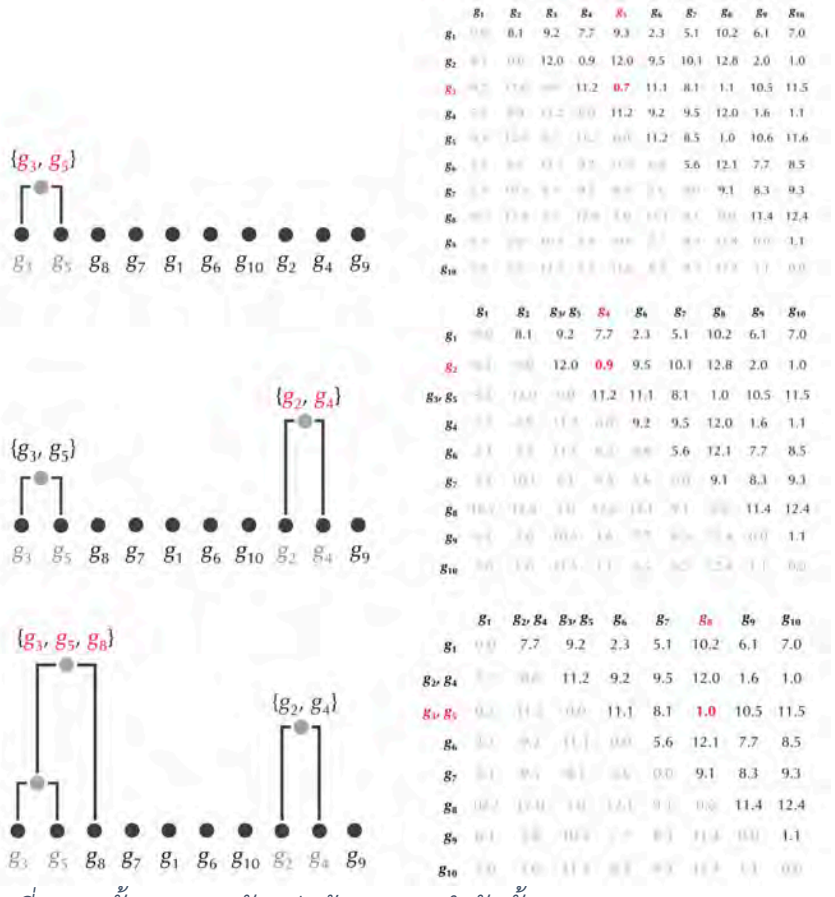
สูโดโคดที่ 7.2 HierarchicalClustering

```
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 (บน) แสดงเมทริกซ์ระยะทางจากรูปที่ 7.19 โดยตัวเลขสีแดงแสดงค่าระยะทางที่น้อยที่สุดระหว่างยีน g3 และ g5 รูปที่ 7.20 (กลาง) แสดงการรวม g3 และ g5 เข้าเป็นกลุ่มเดียวกัน อัพเดตค่าในเมทริกซ์ระยะทางหลังจากคำนวณ D_{min} ระหว่างกลุ่มใหม่กับกลุ่มเดิมทั้งหมด และรวมยีน g2 และ g4 เข้าเป็นกลุ่มเดียวกันเพราะมีระยะทางสั้นที่สุด รูปที่ 7.20 (ล่าง) แสดงการอัพเดตค่าในเมทริกซ์ระยะทางระหว่างกลุ่มใหม่กับทุกกลุ่มที่เหลือ รวมกลุ่ม g_3 และ g_5 กับกลุ่ม g_8 เข้าด้วยกัน และวนทำซ้ำจนกว่าทุกยีนจะอยู่ในกลุ่มเดียวกัน

สำหรับวิธีการ UPGMA (Unweighted Pair Group Method with Arithmetic Mean) ที่ใช้ในการสร้างต้นไม้วิวัฒนาการโดยใช้ข้อมูลเข้าเป็นเมทริกซ์ระยะทางเช่นเดียวกัน คำนวณค่าระยะทางระหว่างสองคลัสเตอร์โดยใช้สมการต่อไปนี้

$$D_{avg}(C_1, C_2) = \frac{\sum_{all\ points\ i\ in\ cluster\ C_1} \sum_{all\ points\ j\ in\ cluster\ C_2} D_{i,j}}{|C_1| \cdot |C_2|}$$



รูปที่ 7.20 ขั้นตอนการจัดกลุ่มข้อมูลแบบลำดับชั้น (Hierarchical clustering)

(ที่มา: รูปที่ 8.24 ของ [21])

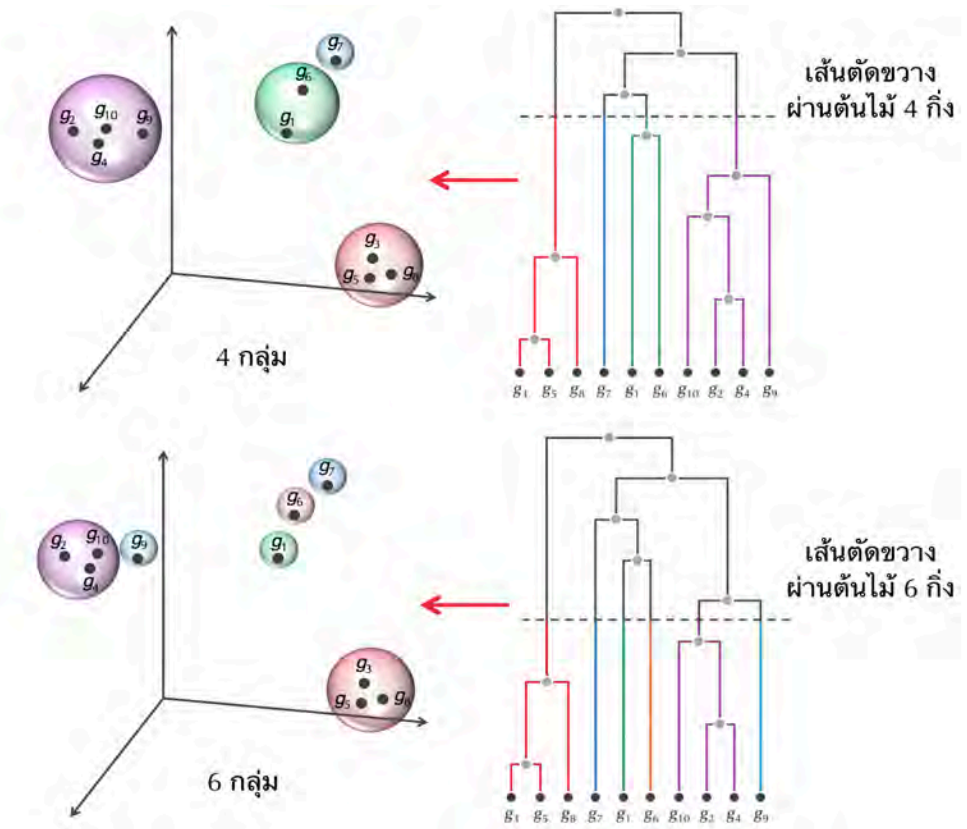
สำหรับรูปที่ 7.21 เส้นแนวขวางที่ตัดผ่านต้นไม้กลุ่มข้อมูล i กิ่งจะแบ่ง n ยีนในต้นไม้ออกเป็น i กลุ่ม

การวิเคราะห์ diauxic shift จากผลการจัดกลุ่มยีนแบบลำดับชั้น

รูปที่ 7.22 แสดงผลการจัดกลุ่มยีนโดยใช้วิธีการจัดกลุ่มแบบลำดับชั้น (Hierarchical clustering) ซึ่งถ้าเปรียบเทียบผลของการจัดกลุ่มโดยวิธีการนี้กับการจัดกลุ่มโดยอัลกอริทึม Lloyd ในรูปที่ 7.13 ก็จะพบว่ามีความแตกต่างกัน

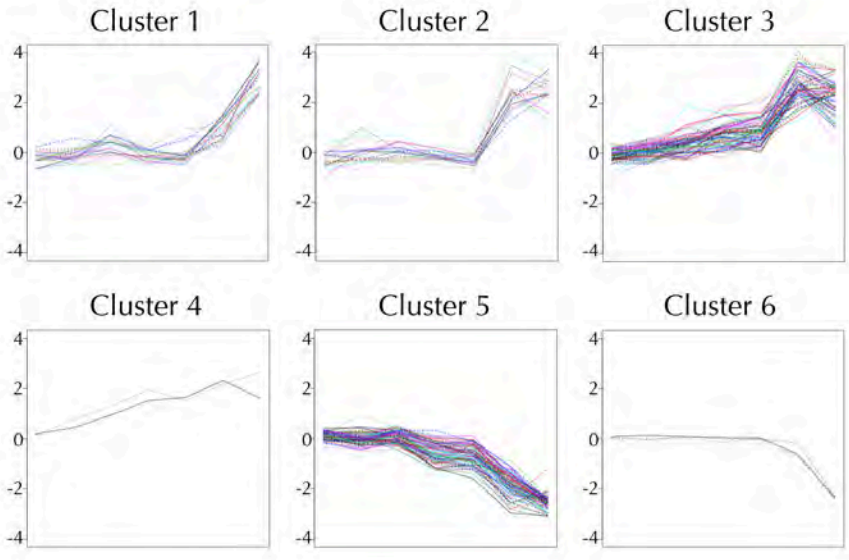
หยุดคิด	เราควรกังวลกับผลการจัดกลุ่มที่แตกต่างกันระหว่างการจัดกลุ่มแบบลำดับชั้นกับการจัดกลุ่มโดยใช้อัลกอริทึม Lloyd มากน้อยแค่ไหน อย่างไร

ฝึกหัด	ลองเขียนโคดเพื่อแสดงการจัดกลุ่มแบบลำดับชั้น (Hierarchical clustering) โดยใช้ D_{min} แทนที่จะเป็น D_{avg} แล้วลองประยุกต์ใช้โคดนี้เพื่อจัดกลุ่ม 230 ยีนในยีสต์ออกเป็น 6 กลุ่ม ผลของการจัดกลุ่มมีความแตกต่างจาก 6 คลัสเตอร์ในรูปที่ 7.22 อย่างไร



รูปที่ 7.21 (บน) แสดงการตัดต้นไม้ผ่าน 4 กิ่งซึ่งทำให้แบ่งกลุ่มของยีนออกเป็น 4 กลุ่ม (ล่าง) ตัดลึกลงมาผ่าน 6 กิ่งทำให้แบ่งกลุ่มของยีนออกเป็น 6 กลุ่ม

(ที่มา: ปรับเพิ่มเติมจากรูปที่ 8.23 ของ [21])



รูปที่ 7.22 ผลของการใช้การจัดกลุ่มแบบลำดับชั้นในการจัดกลุ่ม 230 ยีนของยีสต์ออกเป็น 6 กลุ่ม

(ที่มา: รูปที่ 8.25 ของ [21])

นักชีววิทยาไม่กังวลกับความจริงที่ว่าการใช้วิธีการจัดกลุ่มที่แตกต่างกันอาจให้ผลการจัดกลุ่มที่ไม่เหมือนกันเนื่องจากผลของการจัดกลุ่มนี้มักเป็นเพียงจุดเริ่มต้นของการศึกษาเพิ่มเติมในมิติต่างๆ ทางชีววิทยา โดยต้องมีการทดลองเพิ่มเติมในห้องปฏิบัติการเพื่อยืนยันผลของการจัดกลุ่มว่าชุดของยีนในแต่ละกลุ่มหรือบางกลุ่มที่เป็นที่สนใจจำเพาะนั้นมีความสัมพันธ์หรือเกี่ยวข้องกันในเชิงชีววิทยาจริงๆ ตัวอย่างเช่น แต่ละคลัสเตอร์ในรูปที่ 7.22 แสดงผลของการที่จัดกลุ่มแบบลำดับชั้นของ 230 ยีนในยีสต์ออกเป็น 6 กลุ่ม ซึ่งประกอบด้วย 22, 20, 87, 2, 95, และ 4 ยีนตามลำดับ ทั้งนี้สามารถวิเคราะห์เพิ่มเติมเพื่อหายีนกลุ่มย่อยภายในคลัสเตอร์ซึ่งอาจมีรูปแบบระดับการแสดงออกของยีนที่ชัดเจนมากขึ้นหรือเฉพาะกลุ่มมากขึ้น เช่นคลัสเตอร์ที่ 1 มี 7 ยีนที่มีการเปลี่ยน แปลงระดับการแสดงออกไม่มากนักใน 6 ช่วงเวลาแรกแต่กลับมีระดับการแสดงออกเพิ่มขึ้นชัดเจนในช่วงเวลา สุดท้ายของการทดลอง ซึ่งนักชีววิทยาค้นพบว่า 6 ใน 7 ยีนในกลุ่มนี้มีไบน์ดิงไซต์หรือโมทิฟที่เรียกว่า **carbon source response element (CSRE)** ซึ่งมีรูปแบบของลำดับเบสเป็น **CATTCATCCG** ในดีเอ็นเอส่วนหน้าของยีน (ที่เรียกว่า upstream region) และเมื่อทำการค้นหาโมทิฟนี้ในดีเอ็นเอส่วนหน้าของยีนของยีสต์ทั้งจีโนม พบว่ามีอีกเพียง 4 ยีนที่ส่วนของ upstream region มีโมทิฟหรือไบน์ดิงไซต์นี้เช่นกัน ซึ่งเป็นข้อสนับสนุนว่าเป็นความคิดที่ดีที่มีการจัดกลุ่มย่อยในคลัสเตอร์ที่ 1 โดยมี 6 ยีนนี้เป็นสมาชิกในกลุ่มย่อยนั้น

ประเด็นที่มีความสำคัญมากกว่า อย่างไรก็ตาม คือการพยายามเข้าใจว่าทำไม 6 ยีนนี้ถึงเกี่ยวข้องกัน ยีสต์ชอบใช้กลูโคสเป็นแหล่งพลังงานมากกว่าแหล่งพลังงานอื่น เช่น เอทานอล ดังนั้นในกรณีที่มีกลูโคสอยู่ ชุดของยีนที่รับผิดชอบต่อกระบวนการเมตาโบไลต์ของสารอื่น เช่น เอทานอล จะถูกปิดสวิตช์ หรือถูกกดไว้ไม่ให้แสดงออกมาเป็นเมสเซนเจอร์อาร์เอ็นเอเพื่อทำงาน นักวิจัยได้สรุปว่าโมทิฟ CSRE ด้วยวิธีการบางอย่าง ส่งสัญญาณให้ยีสต์ทราบถึงความมีอยู่ของกลูโคสและจะไป *เปิด* การทำงานของ 6 ยีนนี้ก็ต่อเมื่อยีสต์ไม่มีกลูโคสเป็นแหล่งอาหาร ดังนั้นทั้งโมทิฟ CSRE และ 6 ยีนเป็นส่วนประกอบที่สำคัญของกระบวนการ diauxic shift

ท้ายสุด ถึงจุดนี้หลายๆคนอาจคิดว่าได้ศึกษาวิธีการจัดกลุ่มข้อมูลครอบคลุมแล้ว อย่างไรก็ตามถ้ากลับไปพิจารณาลักษณะข้อมูลในรูปที่ 7.15 (บน) ถึงแม้ว่าการจัดการกลุ่มข้อมูลแบบลำดับชั้นโดยใช้ D_{min} จะสามารถจัดกลุ่มข้อมูลได้ถูกต้องสำหรับข้อมูลในรูปซ้าย และกลาง แต่ยังไม่มีอัลกอริทึมใดในบทเรียนนี้ที่สามารถจัดกลุ่มข้อมูลในรูปขวาได้ถูกต้อง การจัดกลุ่มข้อมูลในรูปที่ 7.15 (บน) อาจดูเป็นปัญหาที่ตรงไปตรงมาสำหรับมนุษย์ เพราะสายตามนุษย์มีความสามารถในการแยกกลุ่มข้อมูลออกเป็นรูปทรงได้ดี

การจัดกลุ่มผู้ป่วยโรคมะเร็ง

ตามที่ได้มีการกล่าวถึงในตอนต้นของบทเรียนนี้ การวัดระดับการแสดงออกของยีนถูกนำไปประยุกต์ใช้ในการตอบโจทย์ทางชีววิทยาที่หลากหลาย รวมทั้งการศึกษาเกี่ยวกับโรคมะเร็ง ในปีค.ศ. 1999 Uri Alon ได้ทำการวิเคราะห์ข้อมูลระดับการแสดงออกของ 2,000 ยีน จากเนื้อเยื่อมะเร็งลำไส้ 40 ตัวอย่าง และทำการเปรียบเทียบกับข้อมูลระดับการแสดงออกของยีนจากเนื้อเยื่อลำไส้ปกติ 20 ตัวอย่าง เราสามารถแสดงชุดข้อมูลของ Uri Alon ในรูปแบบ

เมทริกซ์การแสดงออกของยีนขนาด 2,000 x 60 โดยที่ 40 คอลัมน์แรกเป็นข้อมูลจากเนื้อเยื่อมะเร็งในขณะที่ 20 คอลัมน์หลังเป็นข้อมูลจากเนื้อเยื่อปกติ

สมมติว่าถ้าเราทำการวัดการแสดงออกของยีนของผู้ป่วยใหม่และเป็นนำมาเป็นข้อมูลคอลัมน์ที่ 61 เป้าหมายของเราคือการทำนายว่าผู้ป่วยใหม่นี้เป็นมะเร็งลำไส้หรือไม่ เนื่องจากเราทราบประเภทของเนื้อเยื่อ (มะเร็งและ ปกติ) อยู่แล้ว การจำแนกว่าตัวอย่างเนื้อเยื่อของผู้ปวยใหม่เป็นเนื้อเยื่อมะเร็งหรือเนื้อเยื่อปกติ ดูจะเป็นเรื่องง่าย ในความเป็นจริงแล้วข้อมูลผู้ป่วยแต่ละคนก็คือจุดหนึ่งจุดในสเปซ 2000 มิติ โดยเราสามารถคำนวณค่าจุดศูนย์ถ่วงของจุดที่เป็นเนื้อเยื่อมะเร็งและจุดศูนย์ถ่วงที่เป็นเนื้อเยื่อปกติได้ และเราสามารถตรวจสอบต่อไปได้ว่าจุดตัวอย่างใหม่ที่เข้ามานั้นอยู่ใกล้จุดศูนย์ถ่วงไหนมากกว่ากัน

อีกทางเลือกหนึ่งที่เป็นไปได้คือ เราสามารถทำการวิเคราะห์ข้อมูลเสมือนว่าไม่มีความรู้เรื่องกลุ่มของเนื้อเยื่อมาก่อน และทำใช้วิธีการในการจัดกลุ่มข้อมูลในเมทริกซ์ 2,000 x 61 โดยกำหนดค่า k = 2 ถ้ากลุ่มข้อมูลกลุ่มหนึ่งมีเนื้อเยื่อมะเร็งเป็นส่วนใหญ่ กลุ่มข้อมูลนี้อาจนำมาช่วยในการวินิจฉัยมะเร็งลำไส้ได้

ปัญหาท้าทาย	วิธีการข้างต้นที่ใช้ในการระบุเนื้อเยื่อของผู้ป่วยใหม่ว่าเป็นมะเร็งลำไส้หรือไม่นั้นตรงไปตรงมา อย่างไรก็ตามทั้งสองวิธีการยังมีความน่าเชื่อถือไม่เพียงพอในการวินิจฉัยผู้ป่วยใหม่ คำถามคือเพราะอะไร และถ้ามีข้อมูลเมทริกซ์การแสดงออกของยีนขนาด 2,000 x 60 ของ Alon และข้อมูลยีนของผู้ป่วยใหม่ ให้ลองนำเสนอวิธีการที่ใช้ในการประเมินว่าผู้ป่วยคนนี้มีโอกาสที่จะเป็นมะเร็งลำไส้หรือไม่

อาร์เอ็นเอซีค

การวัดการแสดงออกของยีนหนึ่งๆ คือการวัดปริมาณเมสเซนเจอร์อาร์เอ็นเอที่ถูกถอดรหัส (transcribed) มาจากยีนในระดับดีเอ็นเอที่เป็นส่วนหนึ่งของจีโนม การศึกษาการแสดงออกของยีนทั้งจีโนมเรียกว่าทรานสคริปโตมิกส์ (transcriptomics) และผลการแสดงออกของยีนทั้งจีโนมเรียกว่าทรานสคริปโตม (transcriptome) ปัจจุบันอาร์เอ็นเอซีค (RNA-Seq) เป็นเทคโนโลยีหลักที่ใช้การวัดปริมาณอาร์เอ็นเอที่แสดงออกจากทั้งยีนที่สามารถแปลรหัสไปเป็นโปรตีน (protein-coding genes) และยีนที่สามารถถอดรหัสเป็นอาร์เอ็นเอแต่อาร์เอ็นเอเหล่านั้นไม่แปลรหัสต่อไปเป็นโปรตีน (non-coding genes) ทั้งนี้ อาร์เอ็นเอซีคกลายเป็นเทคโนโลยีหลักที่ใช้ในการวัดการแสดงออกของยีนเนื่องจากเทคโนโลยีอาร์เอ็นเอซีคอยู่บนพื้นฐานเดียวกับเทคโนโลยี NGS (Next Generation Sequencing) คืออาร์เอ็นเอที่แสดงออกจะถูกแปลงให้เป็น cDNA (complementary DNA) จากนั้นถูกทำให้เป็นเส้นสั้นๆ (short reads) และทำการถอดรหัสจากเครื่อง NGS ในลักษณะเดียวกับการถอดรหัสจีโนม ซึ่งไม่ต้องมีการเตรียมอะเรย์ชิปและออกแบบโพรบ (probe) เพื่อให้สามารถจับกับ cDNA ของยีนแต่ละยีนที่สนใจวัดการแสดงออก นอกจากนี้ข้อมูลจากเอ็นเอซีคยังสามารถนำใช้ในการวิเคราะห์การเกิดสไปซ์ไซต์ (spliced sites) และ

ไอโซฟอร์ม (isoforms) แบบต่างๆที่เป็นไปได้ ที่ถูกทรานสไครป์มาจากยีนเดียวกัน วิธีการวิเคราะห์ข้อมูลเอ็นเอซีคพื้นฐานที่มีการอ้างอิงแพร่หลายได้มีการตีพิมพ์ใน [167] ทั้งนี้ในการจัดกลุ่มข้อมูลเอ็นเอซีคนั้นยังไม่มีหลักเกณฑ์แน่นอน Jaskowiak P.A. และคณะได้ทำการเปรียบเทียบวิธีการจัดกลุ่ม และฟังก์ชันวัดระยะทางที่เหมาะสมสำหรับการจัดกลุ่มไว้ใน [168]

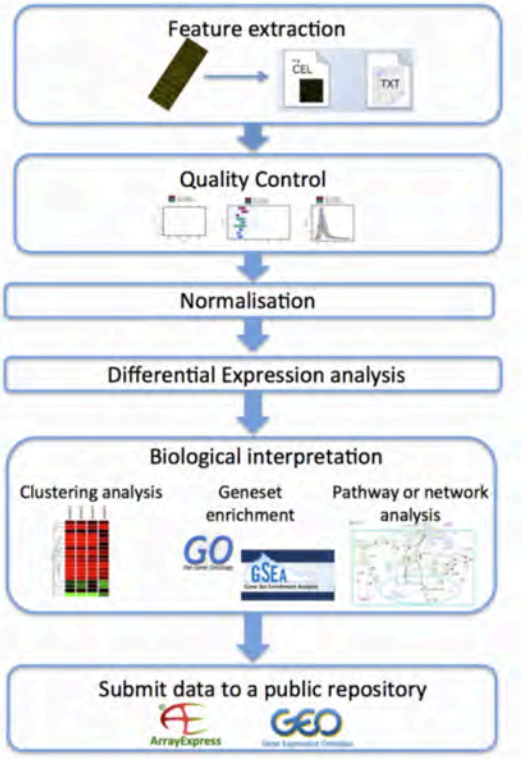
บทส่งท้าย

ปัญหาการจัดกลุ่มข้อมูลหรือการแบ่งกลุ่มข้อมูลโดยเครื่องคอมพิวเตอร์เป็นกลุ่มปัญหาที่สำคัญกลุ่มหนึ่งของการวิจัยและพัฒนาการเรียนรู้ของเครื่อง (Machine Learning: ML) ภายใต้หัวข้อที่เกี่ยวข้องกับการเรียนรู้ของเครื่องแบบไม่มีผู้สอน (Unsupervised Learning) การออกแบบและพัฒนาวิธีการจัดกลุ่มข้อมูลนอกเหนือจากที่มีการศึกษากันในบทเรียนนี้แล้ว ในเชิงวิจัยยังมีการนำเสนอวิธีการในการจัดกลุ่มข้อมูลอีกหลากหลายวิธีเช่น การนำ Nonnegative Matrix Factorization (NMF) และวิธีการอื่นที่เป็นส่วนขยายมาประยุกต์ใช้ในการจัดกลุ่มข้อมูลการแสดงออกของยีน [169] ข้อมูลการแสดงออกของยีนและข้อมูลไมโครไบโอม (microbiome) [170] ข้อมูลการเกิดปฏิสัมพันธ์ระหว่างอาร์เอ็นเอและโปรตีนที่มาจับผ่านข้อมูล CLIP-Seq โดยเป็นการจับกลุ่มข้อมูลแบบ Soft clustering [171] เป็นต้น นอกจากนี้ด้วยข้อมูลทางชีวสารสนเทศที่เกิดจากเทคโนโลยีใหม่ๆ ที่มีมิติของข้อมูลจำนวนมากมาย วิธีการจัดกลุ่มข้อมูลใหม่ๆ ยังเป็นที่ต้องการและยังเป็นปัญหาที่ท้าทาย เช่น การจัดกลุ่มข้อมูลอาร์เอ็นเอซีคที่มาจากเซลล์เดี่ยว (Single-cell RNA-Seq clustering) [172-176] และการจัดกลุ่มข้อมูลแมสไซโตเมทรี (Mass cytometry) ที่มาจากเซลล์เดี่ยว [177, 178] เป็นต้น ซึ่งมีตัวอย่างผลงานวิจัยและพัฒนาเช่น การประยุกต์ใช้ Nonnegative Matrix Factorization ในการทำ coupled clustering กับข้อมูลอาร์เอ็นเอซีคของเซลล์เดี่ยวและข้อมูล ATAC-Seq [179] การประยุกต์ใช้ Nonnegative Matrix Factorization ในการจัดกลุ่มข้อมูลอาร์เอ็นเอซีคในเซลล์เดี่ยว [180, 181] และที่นำเสนอเป็นไลบรารี่ภาษา R ชื่อ ccfindR (https://bioconductor.org/packages/devel/bioc/vignettes/ccfindR/inst/doc/ccfindR.pdf) เป็นต้น

ตัวอย่างโปรแกรมที่มีการใช้งานกันอย่างแพร่หลาย

ขั้นตอนพื้นฐานในการวิเคราะห์ข้อมูลไมโครอะเรย์ (รูปที่ 7.23) ประกอบด้วย (1) การสกัดคุณลักษณะ (Extract feature) (2) การตรวจสอบและควบคุมคุณภาพของข้อมูล (Quality control) (3) การปรับค่ามาตรฐานให้กับข้อมูลหรือการทำนอร์มัลไลเซชัน (Normalization) (4) การวิเคราะห์ระดับการแสดงออกของยีน (Differential expression analysis) (เป็นเนื้อหาหลักในบทเรียนนี้ โดยบทเรียนนี้เน้นวิธีการเชิงอัลกอริทึมในการจัดกลุ่มยีน) และ (5) การแปลความหมายหรือการตีความผลในเชิงชีววิทยา (ที่มา: ออนไลน์คอร์สที่ EMBL-EBI https://www.ebi.ac.uk/training/online/course/functional-genomics-ii-common-technologies-and-

data-analysis-methods/analysis-microarray-data) โดยขั้นตอนต่างๆ เหล่านี้มีตัวอย่างซอฟต์แวร์หรือไลบารี่ที่สามารถใช้งานได้แตกต่างกันไป



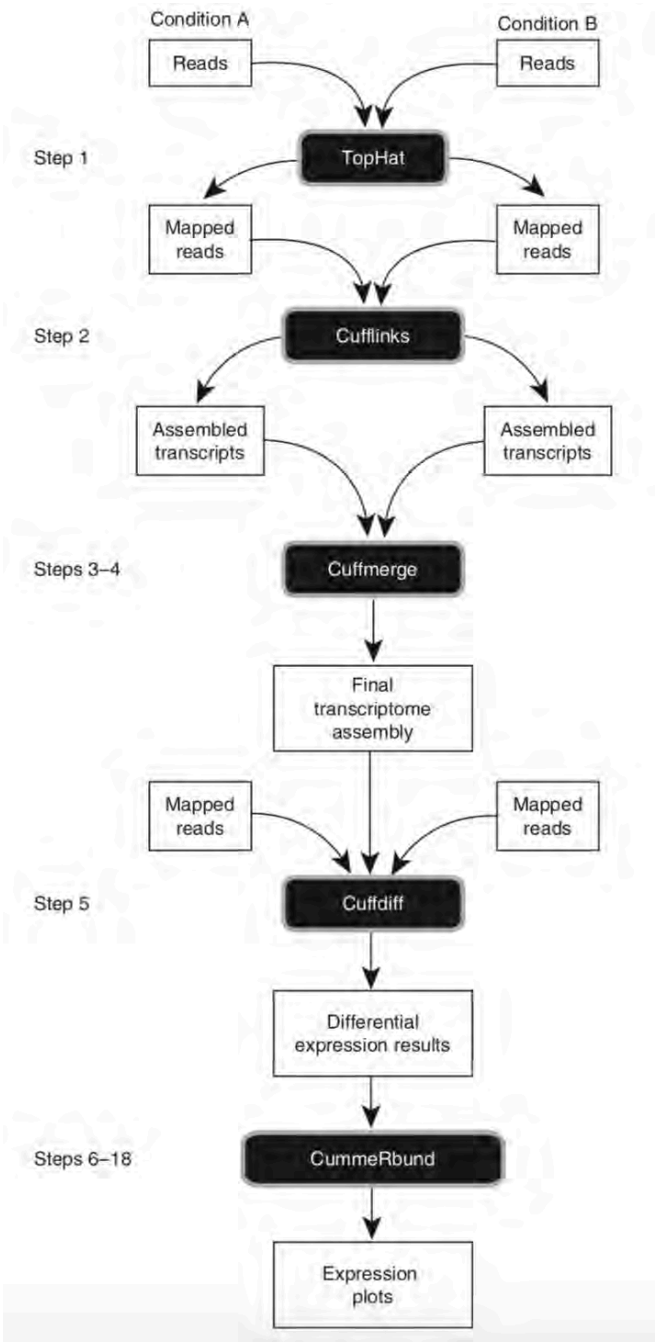
รูปที่ 7.23 ไปป์ไลน์มาตราฐานในการวิเคราะห์ข้อมูลไมโครอะเรย์

(ที่มา: รูปที่ 4 ของออนไลน์คอร์สที่ EMBL-EBI https://www.ebi.ac.uk/training/online/course/functional-genomics-ii-common-technologies-and-data-analysis-methods/analysis-microarray-data)

สำหรับการปรับค่าของข้อมูลให้เป็นมาตรฐานเดียวกัน (การทำนอร์มัลไลเซชัน) นั้นสามารถใช้เมธอด ชื่อ rma (Robust Multi-Array Average expression measure) [182-184] ในแพกเกจภาษา R ชื่อ oligo สำหรับการวิเคราะห์ระดับการแสดงออกของยีนที่แตกต่างกันสามารถใช้แพกเกจภาษา R ชื่อ limma [185] เป็นต้น ตัวอย่างซอฟต์แวร์และไลบรารีอื่นๆ ทั้งแบบที่เปิดให้ดาวน์โหลดโดยสาธารณะและแบบที่เป็นเชิงพาณิชย์สามารถศึกษาเพิ่มเติมได้จาก [186] เป็นต้น

จาก [167] โปรแกรมหลักที่ใช้ในการวิเคราะห์ข้อมูลอาร์เอ็นเอซีคประกอบด้วย TopHat [187] ซึ่งใช้ Bowtie [77] ในการเทียบ cDNA สายสั้นกับจีโนมอ้างอิง จากนั้น TopHat จะพยายามหาสไปซ์ไซต์ (splice sites) ที่เป็นไปได้ระหว่างเอ็กซอนที่อยู่ติดกัน สำหรับรีดที่ไม่สามารถแมพ (map) ได้โดย Bowtie จะถูกนำมาเทียบอีกครั้งโดย TopHat หลังจากหาสไปซ์ไซต์มาแล้วจะมีการใช้ซอฟต์แวร์แพกเกจชื่อ Cufflinks [188] ในการประกอบร่างทรานสคริปต์ (transcripts) (ซึ่งก็คือ cDNA ที่ถูกแปลงมาจากอาร์เอ็นเอและถูกถอดรหัสมาจากเครื่อง NGS นั่นเอง) และใช้ Cuffcompare ในการเปรียบเทียบทรานสคริปต์ที่ถูกประกอบร่างแล้วว่าตรงกันยีนไหนในจีโนม ใช้

Cuffmerge ในการรวมทรานสคริปต์ที่ประกอบร่างมาแล้วเข้ากันอีกที และใช้ Cuffdiff ในการหาชุดของยีนที่มีระดับการแสดงออกของอาร์เอ็นเอที่แตกต่างกัน รวมทั้งสามารถระบุสไปซ์ไซต์ที่แตกต่างและโปรโมเตอร์ที่ใช้ได้ แพกเกจ Cufflinks มีการปรับปรุงและพัฒนาอย่างต่อเนื่องโดยสามารถศึกษารายละเอียดเพิ่มเติมได้ที่ http://cole-trapnell-lab.github.io/cufflinks/ รูปที่ 7.24 แสดงขั้นตอนการวิเคราะห์ข้อมูลอาร์เอ็นเอซีคเพื่อเปรียบเทียบผลระหว่าง 2 เงื่อนไข [167]



รูปที่ 7.24 โปรโตคอลที่นำเสนอใน [162] เพื่อการวิเคราะห์ข้อมูลอาร์เอ็นเอซีคที่มาจาก 2 เงื่อนไข (ที่มา: รูปที่ 2 ของ [167])

แบบฝึกหัดบทที่ 7

1. ให้เขียนโปรแกรมเพื่อแก้ปัญหาที่เกี่ยวข้องกับการจัดกลุ่มข้อมูลโดยใช้โจทย์ที่โรซาลินด์ต่อไปนี้
   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. ศึกษาวิธีการวิเคราะห์ข้อมูลไมโครอะเรย์เพิ่มเติมจาก EMBL-EBI ออนไลน์คอร์สที่ https://www.ebi.ac.uk/training/online/course/functional-genomics-ii-common-technologies-and-data-analysis-methods/analysis-microarray-data
3. ศึกษาวิธีการพื้นฐานเกี่ยวกับอาร์เอ็นเอซีเควนซิ่งจาก EMBL-EBI ออนไลน์คอร์สที่ https://www.ebi.ac.uk/training/online/course/functional-genomics-ii-common-technologies-and-data-analysis-methods/rna-sequencing

บทที่ 8 การวิเคราะห์การแสดงออกของโปรตีน (Protein expression analysis)

วัตถุประสงค์

- โดยใช้เทคโนโลยีแมสสเปคโตรเมทรี รวมทั้งได้เห็นตัวอย่างการประยุกต์ใช้เทคโนโลยีแมสสเปคโตรเมทรีกับงานวิจัยทางชีววิทยาและชีวการแพทย์
- เพื่อให้นิสิตคุ้นเคยกับลักษณะข้อมูลของแมสสเปคโตรเมทรี อัลกอริทึมพื้นฐานที่ใช้ในการถอดรหัสลำดับกรดอะมิโนสายสั้นหรือเปปไทด์ รวมทั้งตัวอย่างวิธีการทางสถิติที่ใช้ในการประเมินความสำคัญของเปปไทด์ที่เป็นผลลัพธ์ของการสืบค้นฐานข้อมูลโปรตีโอมโดยใช้ข้อมูลสเปคตรัม
- เพื่อให้นิสิตได้เห็นตัวอย่างงานวิจัยและผลงานวิจัย ที่ใช้ในการอนุมานหรือการถอดรหัสลำดับกรดอะมิโนของสายเปปไทด์จากข้อมูลสเปคตรัม รวมทั้งในการระบุสายเปปไทด์โดยเทียบกับฐานข้อมูลโปรตีโอม
- เพื่อให้นิสิตได้เห็นแนวทางในการประยุกต์ใช้องค์ความรู้จากบทเรียนเพื่อตอบโจทย์ที่ยังเป็นปัญหาท้าทาย รวมทั้งงานวิจัยอื่นๆ ที่เกี่ยวข้อง

ผลลัพธ์ที่คาดหวัง

- นิสิตเข้าใจลักษณะการทำงานพื้นฐานของเครื่องแมสสเปคโตรเมทรี และการประยุกต์ใช้ในงานทางด้านโปรตีโอมิกส์เพื่อการวิจัยทางชีววิทยาและชีวการแพทย์
- นิสิตเข้าใจคุณลักษณะของข้อมูลแมสสเปคโตรเมทรี
- นิสิตสามารถอธิบายการทำงานของอัลกอริทึมหลักๆ ที่ใช้ในการถอดรหัสลำดับกรดอะมิโนในสายเปปไทด์จากข้อมูลสเปคตรัมของเครื่องแมสสเปคโตรเมทรี รวมทั้งสามารถแสดงการคำนวณการประเมินนัยยะสำคัญทางสถิติของชุดสายเปปไทด์ที่เป็นผลจากการสืบค้นฐานข้อมูลโปรตีโอมโดยใช้ลำดับกรดอะมิโนจากการถอดรหัส
- นิสิตสามารถเขียนโปรแกรมที่ใช้ในการถอดรหัสกรดอะมิโนอย่างง่ายได้
- นิสิตสามารถยกตัวอย่างโปรแกรมที่ใช้ในการถอดรหัสกรดอะมิโนจากข้อมูลสเปคตรัมที่มีการใช้งานกันอย่างแพร่หลายได้
- นิสิตสามารถยกตัวอย่างความท้าทายที่ยังมีอยู่และสามารถนำเสนอแนวทางในการพัฒนาวิธีการแก้ปัญหาเหล่านี้ได้ รวมทั้งสามารถประยุกต์องค์ความรู้จากบทเรียนเพื่อแก้ปัญหาอื่นๆ ที่เกี่ยวข้องได้

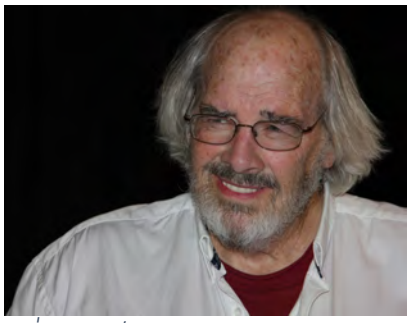
เนื้อหาโดยสรุป

งานวิจัยทางด้านโปรตีโอมิกส์เน้นการศึกษาโปรตีนจำนวนมากในเงื่อนไขหนึ่งๆและเนื้อเยื่อหนึ่งๆ โดยสาขาหนึ่งที่มีการวิจัยอย่างต่อเนื่องคือการประยุกต์ใช้เทคนิคแมสสเปคโตรเมทรีในการศึกษาการแสดงออกของโปรตีน โดยข้อมูลที่เป็นผลลัพธ์จากเครื่องแมสสเปคโตรมิเตอร์นั้นอยู่ในรูปแบบของสเปคตรัมที่แสดงชุดของค่าน้ำหนักของชิ้นส่วนย่อยต่างๆของโปรตีนเรียกว่าเปปไทด์ โดยงานทางด้านอัลกอริทึมมีเป้าหมายเพื่อถอดรหัสลำดับกรดอะมิโนที่ประกอบเป็นโปรตีนโดยใช้ข้อมูลสเปคตรัมเหล่านี้ หรืออีกนัยยะหนึ่งคือหาลำดับกรดอะมิโนของโปรตีนที่น่าจะทำให้เกิดข้อมูลสเปคตรัมเหล่านี้ การประยุกต์ใช้ทฤษฎีกราฟเป็นวิธีการพื้นฐานในการอนุมานลำดับกรดอะมิโนโดยแต่ละโหนดแสดงค่าน้ำหนักของเปปไทด์หนึ่งๆ และเส้นเชื่อมระหว่างโหนดแสดงค่าน้ำหนักที่แตกต่างกันระหว่าง 2 โหนดโดยค่าที่แตกต่างนี้จะต้องเท่ากับค่าน้ำหนักของกรดอะมิโนใดอะมิโนหนึ่ง โดยเส้นทางที่เชื่อมทุกโหนดเข้าด้วยกันนี้แสดงลำดับกรดออะมิโนที่อนุมานได้ เป็นต้น ลำดับกรดอะมิโนของโปรตีนที่อนุมานได้นี้มักมีจำนวนมาก วิธีการหนึ่งที่นำมาใช้คัดเลือกลำดับกรดอะมิโนที่มีความสำคัญคือการนำไปสืบค้นกับฐานข้อมูลโปรตีโอมที่มีอยู่ อย่างไรก็ตามสิ่งที่ต้องคำนึงถึงก็คือลำดับกรดอะมิโนเหล่านี้มีนัยยะสำคัญเชิงสถิติมากน้อยแค่ไหน ในบทเรียนนี้เราจะศึกษาหัวข้อเหล่านี้ รวมทั้งตัวอย่างโปรแกรมและงานวิจัยที่เกี่ยวข้องและมีการใช้งานอย่างแพร่หลาย และความท้าทายที่มีอยู่

บทที่ 8 การวิเคราะห์การแสดงออกของโปรตีน (Protein expression analysis)

เมื่อบรรพชีวินวิทยาพบกับการคำนวณ

แจ็ค ฮอนเนอร์ (Jack Horner) เกิดที่รัฐมอนทานาในปี ค.ศ.1946 และเติบโตที่นั่น เขาเป็นเด็กขี้อายและค่อนข้างเก็บตัว และมีพัฒนาการในการอ่านและการคำนวณทางคณิตศาสตร์ช้ากว่าเด็กคนอื่นในวัยเดียวกัน อย่างไรก็ตาม โครงงานในสมัยมัธยมของเขาเกี่ยวกับไดโนเสาร์เป็นหนึ่งในโครงงานที่ได้รับรางวัลสูงสุดในงานประกวดโครงงานวิทยาศาสตร์ในพื้นที่ และได้รับความสนใจจากศาสตราจารย์ที่มหาวิทยาลัยมอนทานา ซึ่งเป็นผู้ช่วยให้แจ็คได้รับคัดเลือกมาศึกษาต่อที่มหาวิทยาลัย



รูปที่ 8.1 แจ็ค ฮอนเนอร์ในปีค.ศ. 2015

(ที่มา: https://commons.wikimedia.org/wiki/File:2015JackHorner.jpg)

อย่างไรก็ตามฮอนเนอร์ทำเกรดได้ไม่ดีนัก และสอบไม่ผ่าน 5 ควอร์เตอร์ติดกันทำให้ฮอนเนอร์ต้องออกจากมหาวิทยาลัย หลายปีต่อมาจึงทราบว่าอาการของฮอนเนอร์นั้นเกิดจากโรคดิสเล็กเซีย (dyslexia) หรือโรคบกพร่องในการอ่านซึ่งเกิดจากความผิดปกติในพัฒนาการ ทำให้ไม่สามารถอ่านได้รู้ความถึงแม้ผู้ป่วยจะระดับสติปัญญาปกติหรือสูงกว่าปกติก็ตาม

หลังจากถูกเกณฑ์ไปเป็นทหารในสงครามเวียดนามและทำงานเป็นคนขับรถบรรทุก เขาได้งานใหม่เป็นเจ้าหน้าที่เทคนิคของพิพิธภัณฑ์ประวัติศาสตร์ธรรมชาติแห่งมหาวิทยาลัยพรินซ์ตัน (Princeton's Natural History Museum) ที่นี่เป็นที่ที่เขาได้รับชื่อเสียงและการยอมรับจากเพื่อนร่วมงานว่าเป็นนักวิจัยที่ความหลักแหลมอย่างมากและกลายมาเป็นนักบรรพชีวินวิทยา (paleontologist) ที่มีชื่อเสียงเป็นที่รู้จักในระดับโลก เป็นผู้สร้างแรงบันดาลใจให้กับตัวเอกตัวหนึ่งในนวนิยายที่มีชื่อเสียงเรื่องจูราสสิคพาร์ค (Jurassic Park) และเป็นที่ปรึกษาให้กับผู้กำกับภาพยนตร์ชื่อดังสตีเว่น สปีลเบิร์ก (Steven Spielberg) ในการสร้างภาพยนตร์เรื่องดังกล่าว

ฮอนเนอร์ประสบความสำเร็จอย่างมากถึงแม้ว่าว่าจะเป็นโรคดิสเล็กเซีย ส่วนหนึ่งเป็นเพราะงานทางบรรพชีวินวิทยาไม่ต้องการการคำนวณทางคณิตศาสตร์มากมาย อย่างไรก็ตามลูกศิษย์ของฮอนเนอร์แสดงให้เห็นว่างาน

ทางบรรพชีวินวิทยานั้นก็มีการใช้คณิตศาสตร์เช่นกัน ในปี ค.ศ. 2000 ฮอนเนอร์ค้นพบสุสานไดโนเสาร์สายพันธุ์ที่เขาสนใจอย่างมากในรัฐมอนทานาและขุดพบฟอสซิลกระดูกขาของ *Tyrannosaurus rex* หรือทีเร็กซ์ ที่มีอายุถึง 68 ล้านปี สามปีหลังจากนั้นฮอนเนอร์มอบส่วนของฟอสซิลให้กับลูกศิษย์ของเขาชื่อ Mary Schweitzer โดยทำการสลายกระดูกเพื่อศึกษาองค์ประกอบ แต่เนื่องจากแช่ใน demineralizing bath นานเกินไป จึงมีเพียงส่วนที่เป็นเนื้อเยื่อเส้นใยเหลือเท่านั้น เธอส่งส่วนที่เหลือเหล่านี้ไปให้กับ John Asara ซึ่งเป็นผู้เชี่ยวชาญเกี่ยวกับ Mass Spectrometry โดยหวังว่าจะสามารถตรวจพบเปปไทด์ของทีเร็กซ์ (*T. rex*) หรือส่วนของโปรตีนสายสั้นๆ ซึ่งอาจหลงเหลืออยู่ในกระดูก

ในปีค.ศ. 2007 หลังจากวิเคราะห์ข้อมูลสเปคตรัมหลายพันข้อมูล Asara และ Schweitzer ตีพิพม์ผลงานวิจัยในวารสาร *Science* [189] โดยแสดงถึงการค้นพบสายเปปไทด์ของทีเร็กซ์ที่มีความใกล้เคียงกับสายเปปไทด์ที่พบในไก่ (chicken) มาก ผลงานตีพิมพ์นี้เป็นผลงานแรกในเชิงอณูชีววิทยาที่สนับสนุนสมมติฐาน (ที่เป็นที่โต้เถียงกัน) ที่ว่าสัตว์ปีกมีวัฒนาการมาจากไดโนเสาร์

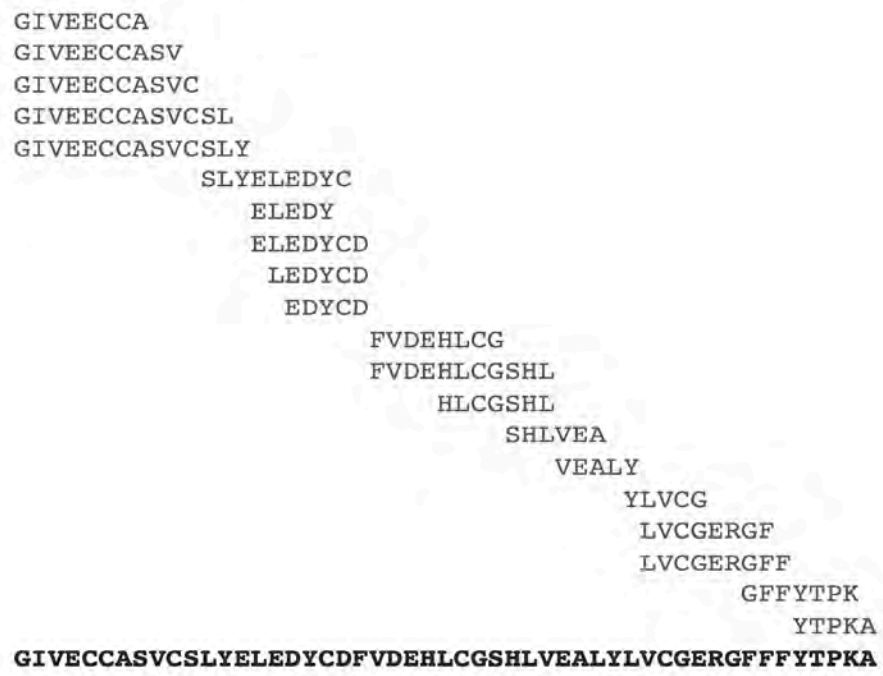
ความจริงที่ว่าโปรตีนสามารถคงอยู่มาเป็นเวลากว่าล้านๆปีเป็นเรื่องที่อัศจรรย์ที่นำไปสู่หัวข้อวิจัยต่างๆที่ยิ่งใหญ่เช่น นักบรรพชีวินวิทยา Hans Larsson เสนอว่าการศึกษาข้อมูลเชิงอณูชีววิทยาของไดโนเสาร์จะเป็นเส้นเชื่อมโยงงานทางบรรพชีวินวิทยาเข้ากับอณูชีววิทยาและวิทยาศาสตร์สมัยใหม่ ในขณะที่ *The Guardian* คาดการณ์ว่า ในวันหนึ่งข้างหน้านักวิทยาศาสตร์จะสามารถจำลองจูราสสิคพาร์คได้โดยการโคลนนิงไดโนเสาร์ ฮอนเนอร์เองได้ตีพิพม์หนังสือชื่อ *“How to Build a Dinosaur”* โดยมีรายละเอียดเกี่ยวกับแผนของเขาในการสร้างไดโนเสาร์จากการดัดแปลงพันธุกรรมของจีโนมไก่

อย่างไรก็ตามนักวิทยาศาสตร์ส่วนหนึ่งยังเคลือบแคลงกับสิ่งเหล่านี้ ในขณะที่การศึกษาในอดีตเกี่ยวกับไดโนเสาร์ไม่ต้องการการคำนวณมากนัก หลังการวิเคราะห์ข้อมูลเปปไทด์ของทีเร็กซ์โดย Asara ที่ใช้อัลกอริทึมที่อยู่บนพื้นฐานของสถิติที่ซับซ้อน ในปีค.ศ. 2008 ได้มีการตีพิมพ์ผลงานวิจัยในวารสาร *Science [190, 191]* ออกมาโต้แย้งว่า Asara และ Schweitzer ไม่สามารถพิสูจน์ได้ว่าสายเปปไทด์บางส่วนที่นำเสนอไว้ในปีค.ศ. 2007 นั้นเป็นเพียงผลข้างเคียงที่เกิดขึ้นจากวิธีการทางสถิติ คำถามคือเราจะทราบได้อย่างไรว่าฝ่ายไหนที่ถูกต้อง ในบทเรียนนี้เราจะพิจารณาว่าสายเปปไทด์ที่มีการโต้แย้งกันนั้นเป็นของทีเร็กซ์จริงหรือไม่ โดยการศึกษาอัลกอริทึมที่เกี่ยวข้องกับการวิเคราะห์ข้อมูลสเปคตรัม

มีโปรตีนอะไรบ้างอยู่ในตัวอย่างนี้

มีนักวิทยาศาสตร์เพียง 4 คนที่เคยได้รับรางวัลโนเบล 2 ครั้ง ท่านหนึ่งในนั้นคือ เฟรดเดอริก แซงเกอร์ (Frederick Sanger) ที่นำเสนอการประกอบร่างจีโนมแรกในปีค.ศ. 1977 โดยแซงเกอร์ได้รับรางวัลโนเบลครั้งแรกเมื่อ 20 ปีก่อนหน้าโดยนำเสนออินซูลินที่ประกอบด้วย 52 ลำดับกรดอะมิโน โดยอินซูลินเป็นโปรตีนที่จำเป็นต่อการดูดซึมกลูโคสในกระแสเลือด แซงเกอร์ทราบลำดับกรดอะมิโนของอินซูลินโดยใช้วิธีการที่คล้ายคลึงกับการถอดรหัสจีโนม

ในปัจจุบัน โดยแซงเกอร์ทำการแยกโมเลกุลของอินซูลินให้เป็นเปปไทด์ย่อยๆ ทำการถอดดูองค์ประกอบของกรดอะมิโนในสายเปปไทด์เหล่านั้น และต่อลำดับกรดอะมิโนสายสั้นเหล่านั้นเข้าด้วยกันดังแสดงในรูปที่ 8.2



รูปที่ 8.2 การประกอบร่างสายเปปไทด์ที่แฟรดเดอริกแซงเกอร์ใช้ในการหาลำดับกรดอะมิโนของอินซูลิน
(ที่มา: รูปที่ 11.1 ของ [21])

ในช่วงคริสต์ทศวรรษ 1950 การถอดลำดับกรดอะมิโนเป็นเรื่องยากมาก ในขณะที่การถอดลำดับเบสดีเอ็นเอเป็นไปไม่ได้เลย ในปัจจุบันการถอดรหัสดีเอ็นเอผ่านเทคโนโลยีอย่าง NGS มีการใช้งานกันอย่างกว้างขวาง ในขณะที่การถอดลำดับกรดอะมิโนในสายโปรตีนยังเป็นเรื่องที่ยาก ด้วยเหตุนี้โปรตีนส่วนใหญ่จะถูกค้นพบจากการถอดรหัสจีโนม และทำนายยีนทั้งจีโนมที่สามารถแปลรหัสต่อไปเป็นโปรตีน ซึ่งทำให้สามารถอนุมานโปรตีโอม (proteome) (ชุดของโปรตีนทั้งหมดในสิ่งมีชีวิตหนึ่งๆ) ได้

อย่างไรก็ตามเซลล์ที่แตกต่างกันและในเงื่อนไขที่แตกต่างกันมีชุดของโปรตีนที่แสดงออกแตกต่างกัน (เช่นเดียวกันกับการแสดงออกของอาร์เอ็นเอ) ตัวอย่างเช่นโปรตีนที่แสดงออกในเซลล์สมองจะเป็นกลุ่มที่เพิ่มจำนวนนิวโรเปไทด์ (neuropeptides) ในขณะที่เซลล์อื่นจะไม่มีการแสดงออกของโปรตีนกลุ่มนี้ การศึกษาโปรตีนที่แสดงออกในองค์รวมในเนื้อเยื่อและเงื่อนไขที่จำเพาะหนึ่งๆ เป็นสาขาหนึ่งในงานวิจัยเชิงโปรตีโอมิกส์ (proteo mics) และเป็นแนวทางหนึ่งที่สำคัญในการศึกษากระบวนการต่างๆ ทางชีววิทยา รวมทั้งการวินิจฉัยและรักษาโรค

ตัวอย่างเช่น การศึกษาไรโบโซมของไก่ ซึ่งไรโบโซมเป็นโมเลกุลที่มีความซับซ้อนประกอบด้วยหลายโปรตีน การทราบข้อมูลโปรตีโอมของไก่ไม่สามารถบอกได้มีโปรตีนอะไรบ้างที่ประกอบกันเป็นไรโบโซม ในทางกลับกันเราสามารถแยกไรโบโซมออกมาโดยทำการแตกโมเลกุลไรโบโซมให้เป็นส่วนย่อยๆ และตรวจสอบได้ว่ามีโปรตีนอะไร

บ้างประกอบอยู่ ในทางปฏิบัติการตรวจพบเปปไทด์ขนาด 10 ลำดับกรดอะมิโนที่ทราบว่าเป็นส่วนของโปรตีนไก่ที่ทราบลำดับสายข้อมูลมาก่อนหน้าก็เพียงพอแล้วในการยืนยันว่ามีโปรตีนนี้ประกอบอยู่ กระบวนการในการตรวจสอบว่ามีสายเปปไทด์จากโปรตีโอมที่ทราบข้อมูลอยู่ในตัวอย่างที่ทดสอบหรือไม่เรียกว่า **peptide identification** แต่เราจะทราบหรือสร้างโปรตีโอมของทีเร็กซ์มาได้อย่างไร

ถึงแม้ว่าการศึกษาส่วนใหญ่ในระดับโปรตีโอมิกส์ในปัจจุบันจะเน้นการทำ peptide identification เรายังไม่มีข้อมูลโปรตีโอมของสิ่งมีชีวิตมากมายรวมทั้งสิ่งมีชีวิตที่สูญพันธุ์ไปแล้วเช่นทีเร็กซ์ ในกรณีหลังนี้นักชีววิทยาจำเป็น ต้องใช้อาศัยการทดลองแบบ ***de novo* peptide sequencing** หรืออนุมานลำดับกรดอะมิโนของสายเปปไทด์โดยไม่มีข้อมูลโปรตีโอมซึ่งเป็นหัวข้อของบทเรียนนี้

การถอดรหัสข้อมูลจากสเปคตรัมที่มีลักษณะอุดมคติ

ถ้ามีสายเปปไทด์เดียวกันจำนวนมากในตัวอย่างทดสอบซึ่งมักประกอบด้วยหลายล้านเซลล์ เครื่องแมสสเปคโตรมิเตอร์จะทำการแตกสายเปปไทด์แต่ละเส้นเป็นสองส่วนที่สั้นลง ซึ่งเปปไทด์เดียวกันแต่ละเส้นอาจถูกแตกออกเป็นสองส่วนที่แตกต่างกันไปได้ ตัวอย่างเช่น เปปไทด์ REDCA แรกอาจแตกเป็น RE และ DCA ในขณะที่ เปปไทด์ REDCA ที่สองอาจแตกเป็น RED และ CA เป็นต้น ส่วนย่อยด้านหน้า RE และ RED เรียกว่าพรีฟิกซ์ (prefixes) ของ REDCA ในขณะที่ส่วนย่อยด้านหลัง DCA และ CA เรียกว่าซัฟฟิกซ์ของ REDCA ตามลำดับ รูปที่ 8.3 แสดงค่าน้ำหนักของกรดอะมิโนมาตรฐาน (เพื่อลดความซับซ้อนการอ้างถึงค่าน้ำหนักของกรดอะมิโนในเนื้อหาส่วนต่อๆไปจะเป็นเลขจำนวนเต็ม)

คำถามในเชิงอัลกอริทึมคำถามแรกคือเราจะอนุมานลำดับกรดอะมิโนจากชุดค่าน้ำหนักพรีฟิกซ์และซัฟฟิกซ์ได้อย่างไร ถ้ามีข้อมูลลำดับกรดอะมิโนของสายเปปไทด์ สเปคตรัมที่มีลักษณะอุดมคติของสายเปปไทด์นี้แสดงโดย *IDEALSPECTRUM(Peptide)* ซึ่งก็คือชุดค่าน้ำหนักของพรีฟิกซ์และซัฟฟิกซ์ที่เกิดขึ้นทั้งหมด ดังตัวอย่างในรูปที่ 8.4 (บน) โดยค่าข้อมูลในสเปคตรัมที่มีลักษณะอุดมคติอาจมีค่าซ้ำได้ เช่น IDEALSPECTRUM(GPG) = {0, 57, 57, 154, 154, 211} เป็นต้น เราสามารถพูดได้ว่าลำดับกรดอะมิโนของเปปไทด์อธิบายชุดของค่าตัวเลขสเปคตรัม ถ้า IDEALSPECTRUM(*Peptide*) = *Spectrum*

นิยามปัญหาที่ 8.1 ปัญหาการถอดรหัสข้อมูลจากสเปคตรัมที่มีลักษณะอุดมคติ

สร้างสายเปปไทด์จากชุดข้อมูลสเปคตรัมที่มีลักษณะอุดมคติ	
ข้อมูลเข้า	ชุดของค่าน้ำหนักทั้งพรีฟิกซ์และซัฟฟิกซ์ที่แสดงสเปคตรัม
ผลลัพธ์	ลำดับกรดอะมิโนของเปปไทด์ที่อธิบายสเปคตรัม

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 ค่าน้ำหนักของกรดอะมิโนมาตรฐาน

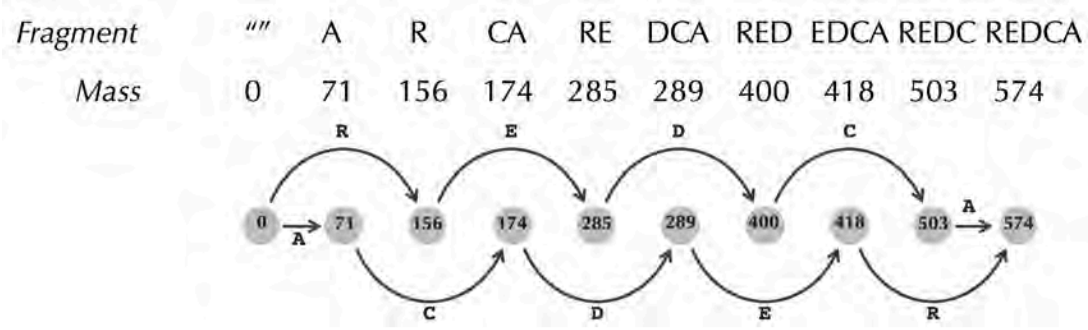
(ที่มา: https://commons.wikimedia.org/wiki/File:Amino_acid_fragment_ions.png)

จากกราฟแบบมีทิศทางในรูปที่ 8.4 (ล่าง) แสดงกราฟแบบมีทิศทาง (Directed Acyclic Graph: DAG) ผ่านฟังก์ชัน GRAPH(IDEALSPECTRUM(REDCA)) โดยค่าน้ำหนักแต่ละค่าเป็นโหนดและเส้นเชื่อมที่ชี้จากโหนด A ไปเป็นโหนด B บ่งชี้ว่าส่วนต่างของค่าน้ำหนักของสองโหนดมีค่าเท่ากับค่าน้ำหนักของกรดอะมิโนใดอะมิโนหนึ่งซึ่งก็จะกลายมาเป็นฉลากกำกับเส้นเชื่อม โดยลำดับกรดอะมิโนก็จะเรียงลำดับไปตามฉลากที่กำกับแต่ละเส้นเชื่อม จากรูปที่ 8.4 (ล่าง) เส้นทางด้านบนจากซ้ายไปขวาแสดงลำดับกรดอะมิโนที่อนุมานได้ส่วนเส้นทางด้านล่างของกราฟเป็นเส้นทางกลับด้านของเปปไทด์ (สูโดโคดที่ 8.1)

สูโดโคดที่ 8.1 DecodingIdealSpectrum

```
1  DecodingIdealSpectrum()
2      สร้าง GRAPH(Spectrum) แบบมีทิศทาง
3      หาเส้นทาง Path จากจุดตั้งต้น source ไปยังจุดสิ้นสุด sink ใน GRAPH(Spectrum)
4      ส่งกลับ สตริงแสดงลำดับกรดอะมิโนจากฉลากที่กำกับแต่ละเส้นเชื่อมใน Path
```

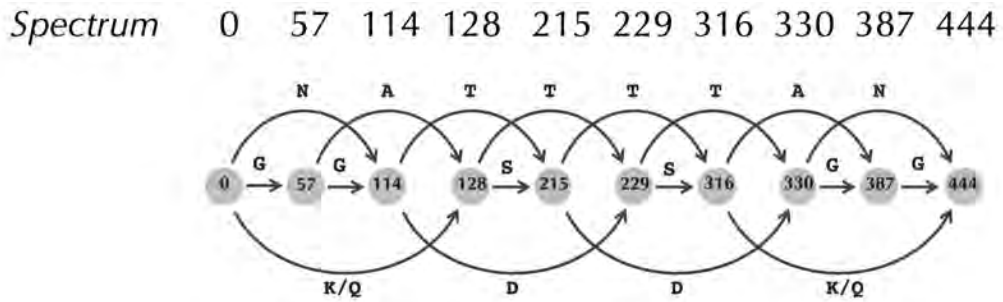
ฝึกหัด	ลองถอดรหัสสายเปปไทด์โดยใช้ข้อมูลสเปคตรัมที่มีลักษณะอุดมคติต่อไปนี้ {0, 57, 114, 128, 215, 229, 316, 330, 387, 444}



รูปที่ 8.4 (บน) ค่าน้ำหนักของพรีฟิกซ์และซัฟฟิกซ์ของ REDCA ซึ่งประกอบกันเป็น IDEALSPECTRUM(REDCA) = {0, 71, 156, 174, 285, 289, 400, 418, 503, 574} (ล่าง) กราฟแบบมีทิศทางโดยเส้นทางด้านบนจากซ้ายไปขวาแสดงลำดับกรดอะมิโนที่อนุมานได้

(ที่มา: รูปที่ 11.3 ของ [21])

หลังจากลองทำแบบฝึกหัดข้างต้นแล้ว อาจพบเส้นทางจากจุดตั้งต้นไปยังจุดปลายทางได้มากกว่า 1 เส้นทาง ซึ่งเส้นทางอื่นๆที่หาได้นั้นหลายเส้นทางไม่ได้แสดงลำดับกรดอะมิโนที่ถูกต้องของเปปไทด์ (เช่น เส้นทาง GGDTN ในรูปที่ 8.5) ดังนั้นจึงต้องมีการแก้ไขสูโดโคดข้างต้นเป็นสูโดโคดที่ 8.2



รูปที่ 8.5 GRAPH(Sepctrum) แบบมีทิศทางของสเปคตรัม {0, 57, 114, 128, 215, 229, 316, 330, 387, 444} โดยมีเพียง 8 ใน 32 เส้นทางจากจุดเริ่มต้นไปยังจุดสิ้นสุดที่สอดคล้องกับชุดของเปปไทด์ที่อธิบายสเปคตรัม

(ที่มา: รูปที่ 11.4 ของ [21])

สูโดโคดที่ 8.2 DecodingIdealSpectrum (ปรับปรุง)

```
DecodingIdealSpectrum()
    สร้าง GRAPH(Spectrum) แบบมีทิศทาง
    for แต่ละเส้นทาง Path จากจุดตั้งต้น source ไปยังจุดสิ้นสุด sink ใน GRAPH(Spectrum)
        Peptide <- ลำดับกรดอะมิโนจากฉลากที่กำกับแต่ละเส้นเชื่อมใน Path
        if IdealSpectrum(Peptide) เท่ากับ Spectrum
            ส่งกลับ Pedtide
```

ถึงแม้ว่าสูโดโคดที่ 8.2 นี้จะแก้ปัญหาการถอดรหัสข้อมูลจากสเปคตรัมที่มีลักษณะอุดมคติได้ การหาเส้นทางจากจุดเริ่มต้นไปยังจุดสิ้นสุดทุกเส้นทางที่เป็นไปได้อาจใช้เวลามาก

จากสเปคตรัมที่มีลักษณะอุดมคติไปเป็นสเปคตรัมที่วัดได้จริง

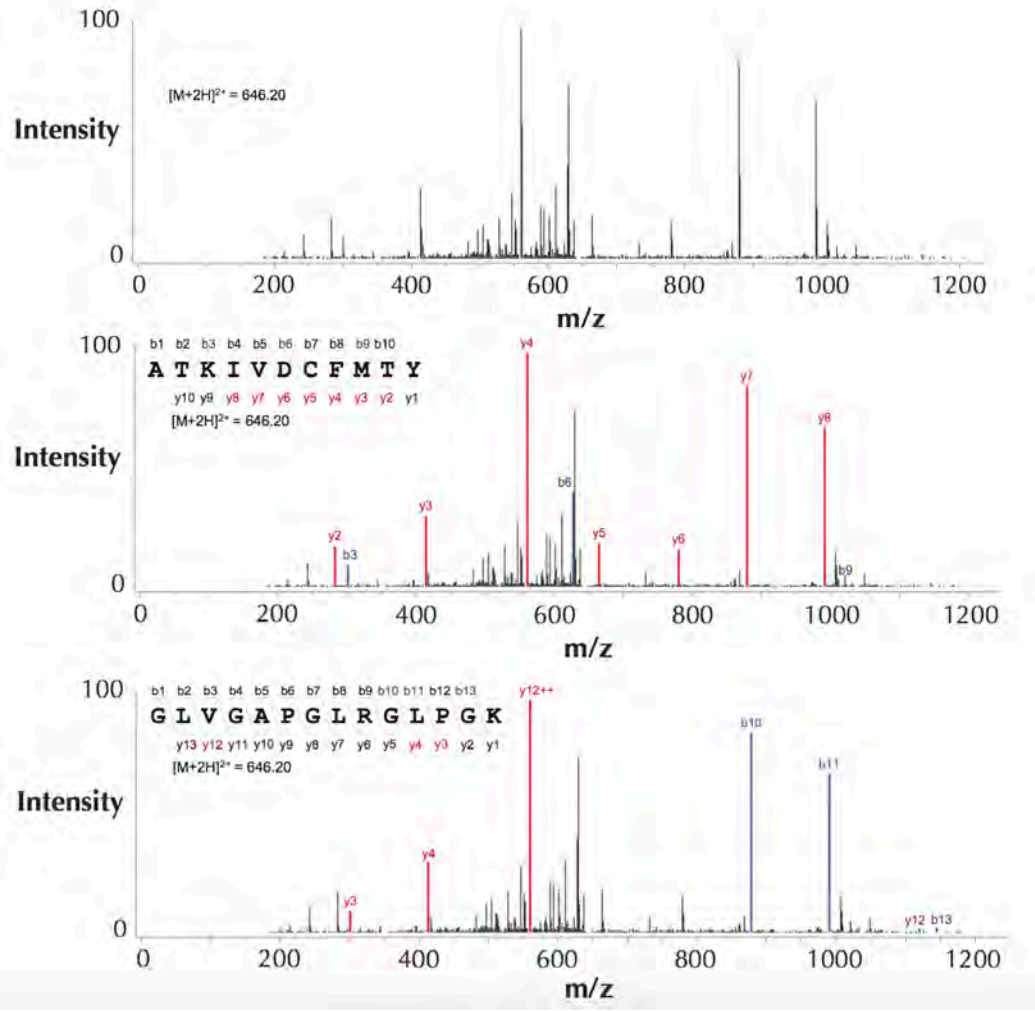
หลังจากเครื่องแมสสเปคโตรมิเตอร์ ทำการแยกสายเปปไทด์หนึ่งๆออกเป็นสองส่วนย่อยแล้ว สายเปปไทด์ย่อยแต่ละส่วนจะถูกไอออนไนซ์ กลายเป็นส่วนย่อยที่มีประจุ **fragment ions** ซึ่งเครื่องแมสสเปคโตรมิเตอร์จะทำการวัดค่า **mass-to-charge ratio (m/z)** ของแต่ละเปปไทด์ย่อยนี้รวมทั้งค่า **intensity** หรือจำนวน fragment ions ที่มีค่า m/z เดียวกัน (เปปไทด์หนึ่งๆ อาจมีโอกาสถูกแยกเป็นสองส่วนในรูปแบบใดรูปแบบหนึ่งมากกว่ารูปแบบอื่นๆ เนื่องจากบางพันธะในสายเปปไทด์อาจถูกแยกได้ง่ายกว่า) ผลที่ตามมาคือสเปคตรัมจะประกอบด้วยชุดของพีค (peaks) ในแผนภูมิโดยแต่ละพีคบนแกน x แสดงค่า mass-to-charge ratio ในขณะที่ความสูงของพีคแสดงค่า intensity (รูปที่ 8.6 บน)

เครื่องแมสสเปคโตรมิเตอร์สมัยใหม่ยังมีข้อจำกัดในช่วงของค่า mass-to-chare ratio ที่สามารถวัดได้ ดังนั้นจึงเป็นการยากที่จะวัดสายโปรตีนทั้งสายโดยใช้วิธีการแมสสเปคโตรเมทรี ผลที่ตามมาคือในการวิเคราะห์โปรตีน สายของโปรตีนจะถูกตัดออกเป็นเปปไทด์สั้นๆโดยใช้เอนไซม์กลุ่มโปรตีเอส (proteases) โดยโปรตีเอสที่มักถูกใช้อย่างแพร่หลายในการศึกษาโปรตีโอมิกส์และในการศึกษาเปปไทด์ของทีเร็กซ์ด้วย คือทริปซิน (trypsin) โดยทริปซินจะแตกสายโปรตีนหลังกรดอะมิโนอาร์จินีน (Arginine: R) และไลซีน (Lysine: K) และได้ผลเป็นเปปไทด์สายสั้นๆมีความยาวเฉลี่ยประมาณ 14 กรดอะมิโน

รูปที่ 8.6 แสดงตัวอย่างแมสสเปคตรัมของทีเร็กซ์ (บน) และตัวอย่างของการตีความ (รูปที่ 8.6 กลางและล่าง) ว่ามีเปปไทด์ ATKIVDCFMTY และ GLVGAPGLRGLPGK ตามลำดับ หลังจากที่เราได้ลำดับกรดอะมิโนมาแล้ว ในขั้นถัดไปก็สามารถเชื่อมโยงข้อมูลกลับไปที่พีคของสเปคตรัมในแผนภูมิที่แสดงพรีฟิกซ์และซัฟฟิกซ์แต่ละค่า ทั้งนี้เพื่อให้เป็นไปตามมาตรฐานของนิยามศัพท์ของแมสสเปคโตรเมทรี พีคที่ถูกระบุว่าเป็นพรีฟิกซ์ความยาว i จะถูกกำหนดฉลากเป็น b_i และพีคที่ถูกระบุว่าเป็นซัฟฟิกซ์ความยาว i จะถูกกำหนดฉลากเป็น y_i

ฝึกหัด	คิดว่าเปปไทด์ไหนในรูปที่ 8.6 น่าจะอธิบายสเปคตรัมของทีเร็กซ์ได้ดีกว่ากัน

การถอดรหัสลำดับกรดอะมิโนของเปปไทด์จากสเปคตรัมจริงมีความยากกว่าที่ผ่านมา ทั้งนี้ข้อมูลจากเครื่องแมสสเปคมักมีพีคที่เป็นสัญญาณรบกวน (noise) ซึ่งเป็นค่า m/z ที่ไม่ถูกต้อง และเนื่องจากบางพันธะอาจแตกได้ยาก ค่า m/z ที่แต่ละ intensity มีความแตกต่างกันไป และในสเปคตรัมอาจไม่เกิดบางพรีฟิกซ์หรือซัฟฟิกซ์เลย เช่น รูปที่ 8.6 (ล่าง) จะไม่มีพีคที่ติดฉลาก b_5 และ y_9 เป็นต้น



รูปที่ 8.6 (บน) ตัวอย่างสเปคตรัมของทีเร็กซ์ (กลาง) สเปคตรัมเดียวกันที่มีการระบุเปปไทด์ ATKIVDCFMTY (ล่าง) สเปคตรัมเดียวกันที่มีการระบุเปปไทด์ GLVGAPGLRGLPGK

(ที่มา: รูปที่ 11.5 ของ [21])

การถอดรหัสเปปไทด์

การให้คะแนนเปปไทด์เมื่อเทียบกับสเปคตรัม

ลองจินตนาการว่าเปปไทด์เส้นหนึ่งประกอบด้วยกรดอะมิโน 2 แบบ X และ Z ซึ่งมีค่าน้ำหนัก 4 และ 5 ตามลำดับ ถ้ามีเปปไทด์ XZZXX แล้วจะมีค่าน้ำหนักของพรีฟิกซ์เป็น 4, 9, 14, 18, 22 ในขณะที่ค่าน้ำหนักของซัฟฟิกซ์จะเป็น 22, 18, 13, 8 , 4 และลองพิจารณาเพิ่มเติมค่าเวคเตอร์ของสเปคตรัมต่อไปนี้

(0, 0, 0, 3, 8, 7, 2, 1, 100, 0 ,1, 4, 3, 500, 2, 1, 3, 9, 1, 2, 2, 0)

ที่มาจากเปปไทด์ข้างต้น โดยตำแหน่งที่ i ของเวคเตอร์เป็นค่า intensity ที่วัดได้ของน้ำหนัก i ในตัวอย่างนี้พรีฟิกซ์ของ XZZXX มีค่า intensity เป็น 3, 100, 500, 9, และ 0 ในขณะที่ซัฟฟิกซ์มีค่า intensity เป็น 0, 8, 0, 2, และ

1 เป้าหมายของเราคือการหาวิธีการให้คะแนนเปปไทด์ที่สอดคล้องกับข้อมูลจากสเปคตรัม โดยหวังว่าเปปไทด์ที่สอดคล้องกับสเปคตรัมจะมีค่าคะแนนสูงสุด

หยุดคิด	เราจะให้คะแนนเปปไทด์หนึ่งๆโดยเทียบกับสเปคตรัมได้อย่างไร

แนวทางแรกที่เป็นไปได้คือการนับผลรวมของค่า intensity เรียกว่า **intensity count** ของทุกพีคที่เป็นค่าน้ำหนักพรีฟิกซ์หรือซัฟฟิสโดยในตัวอย่างข้างต้นค่าผลรวมของ intensity ของเปปไทด์ XZZXX มีค่าเท่ากับ 3+100+500+9+0+0+8+0+2+1 อย่างไรก็ตามวิธีนี้มีข้อจำกัดเนื่องจากในข้อมูลจริงค่า intensity ของพีคต่างๆ มีความแตกต่างกันมาก ซึ่งทำให้พีคบางพีคที่มีค่า intensity สูงมาก (ถึงแม้อาจจะเป็นสัญญาณรบกวน) มีผลกระทบต่อคะแนนโดยรวมมาก ในขณะที่พีคที่ถูกต้องแต่มีค่า intensity น้อยก็จะมีน้ำหนักน้อยเกินไป

เพื่อเป็นการแก้ไขข้อจำกัดข้างต้นวิธีการถัดมาเรียกว่า **shared peaks count** วิธีการนี้นับจำนวนของพีคที่มีค่า intensity สูงกว่าค่า threshold ค่าหนึ่ง สมมติว่าค่า threshold คือ 5 จากตัวอย่างข้างต้น จะได้ค่า shared peaks count = 4 โดยมาจาก 3 พีค 100, 500, และ 9 ของพรีฟิกซ์และ 1 พีคที่มี intensity 8 ของซัฟฟิกซ์ จากตัวอย่างในรูปที่ 8.6 (กลาง) มีจำนวน shared peaks count เท่ากับ 10 ในขณะที่รูปที่ 8.6 (ล่าง) มีจำนวน shared peaks count เท่ากับ 6 ถึงแม้ว่า shared peaks count จะใช้งานได้ดีกว่า intensity count ข้างต้น อย่างไรก็ตามก็ยังให้ผลไกลจากอุดมคติมาก แนวทางที่ดีกว่าควรจะเป็นแนวทางที่สามารถใช้ intensity แต่พีคที่มีค่า intensity สูงไม่ควรเป็นตัวชี้นำผลของความถูกต้องของเปปไทด์มากกว่าพีคที่ถูกต้องอื่นๆที่มีค่า intensity น้อยกว่า เพื่อให้ได้วิธีการที่ดีกว่า เราได้ทำการแปลงเปปไทด์และสเปคตรัมให้อยู่ในรูปแบบของเวคเตอร์ และกำหนดฟังก์ชันการให้คะแนนเป็นของการทำของ dot product ของทั้งสองเวคเตอร์

ในขั้นแรกจากสายสตริงที่แสดงลำดับกรดอะมิโน *Peptide* = $a_1,...a_n$ ความยาว n อะมิโน เราจะแสดงค่าน้ำหนักของพรีฟิกซ์ต่างๆโดยใช้ **binary peptide vector** $\overrightarrow{Peptide}$ โดยตำแหน่งที่มีค่าน้ำหนักพรีฟิกซ์ แสดงด้วย MASS(*Peptide*) (ดังตัวอย่างต่อไปนี้) จะมีค่าเป็น 1 และตำแหน่งที่เหลือทั้งหมดจะเป็น 0

MASS(a1), MASS(a1a2), ..., MASS(a1,a2,...,an)

ในกรณีของเปปไทด์ตัวอย่าง XZZXX ข้างต้น ชุดค่าน้ำหนักพรีฟิกซ์ประกอบด้วย 4, 9, 14, 18, 22 ซึ่งสอดคล้องกับเปปไทด์เวคเตอร์ (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 ปัญหาการแปลงเปปไทด์เป็นเปปไทด์เวคเตอร์

แปลงสายเปปไทด์ให้เป็นเปปไทด์เวคเตอร์	
ข้อมูลเข้า	สายสตริงของลำดับกรดอะมิโน *Peptide*
ผลลัพธ์	เปปไทด์เวคเตอร์ $\overrightarrow{Peptide}$ ของ *Peptide*

เนื่องจากเปปไทด์เวคเตอร์เป็นตัวแทนของเปปไทด์ ดังนั้นในส่วนต่อไปนี้คำว่าเปปไทด์เวคเตอร์และเปปไทด์สามารถใช้แทนกันได้

นิยามปัญหาที่ 8.3 ปัญหาการแปลงเปปไทด์เวคเตอร์ไปเป็นเปปไทด์

แปลงเปปไทด์เวคเตอร์ให้เป็นสายเปปไทด์	
ข้อมูลเข้า	ไบนารีเปปไทด์เวคเตอร์ *P*
ผลลัพธ์	เปปไทด์ที่มีเปปไทด์เวคเตอร์เท่ากับ *P* (ถ้ามีเปปไทด์นั้นอยู่จริง)

ซัฟฟิกซ์เปปไทด์หายไปไหน

ในความเป็นจริงแล้วซัฟฟิกซ์เปปไทด์ไม่ได้หายไปไหน เนื่องจากแต่ละพีคที่แสดงค่าน้ำหนักนั้นอาจเป็นค่าน้ำหนักของพรีฟิกซ์หรือซัฟฟิกซ์ก็ได้ จากความไม่แน่นอนนี้ผู้เชี่ยวชาญการใช้งานเครื่องแมสสเปคโตรเมทรีจะทำการแปลง *Spectrum* ให้อยู่ในรูปแบบเวคเตอร์ $\overrightarrow{Spectrum}$ ที่มีการรวมข้อมูลเกี่ยวกับค่า intensity ของแต่ละพีคและพีคฝาแฝด (twin peak: พีคที่แสดงค่า MASS(*Peptide*)-s) ของมันเข้าเป็นค่าเดี่ยวเรียกว่า **แอมพลิจูด** (amplitude) ทั้งนี้ค่าแอมพลิจูดที่เป็นลบมักแสดงถึงตำแหน่งในสเปคตรัมที่ไม่มีพีคหรือเป็นพีคที่มีค่า intensity ต่ำ ทั้งนี้ค่าแอมพลิจูดที่ค่าน้ำหนัก i สะท้อนโอกาส (*likelihood)* ที่สายเปปไทด์ (ที่ไม่ทราบลำดับกรดอะมิโน) จะทำให้เกิดสเปคตรัมมีค่าน้ำหนักพรีฟิกซ์เท่ากับ i

หลังจากสายเปปไทด์ *Peptide* ได้ถูกแปลงไปเป็นเปปไทด์เวคเตอร์ $\overrightarrow{Peptide} = (p_1,\ldots,p_m)$ และสเปคตรัม *Spectrum* ได้ถูกแปลงไปเป็นสเปคตรัมเวคเตอร์ $\overrightarrow{Spectrum} = (s_1,\ldots,s_m)$ ที่มีจำนวนตำแหน่งเท่ากัน เราสามารถกำหนดฟังก์ชันการให้คะแนนสายเปปไทด์ 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($\overrightarrow{Peptide}$, $\overrightarrow{Spectrum}$) ก็คือผลรวมของค่าแอมพลิจูดหรือ ***amplitude count*** ที่สามารถเป็นตัวแทนของแต่ละพีคของสายเปปไทด์นั่นเอง อย่างไรก็ตามในกรณีนี้จะไม่พบปัญหาเหมือนในกรณี intensity count เนื่องจากเรามีการแปลงค่า intensities เป็นค่าแอมพลิจูด ซึ่งพีคที่มีค่า intensity สูงจะไม่ชี้นำการให้คะแนนของสายเปปไทด์มากเกินไป

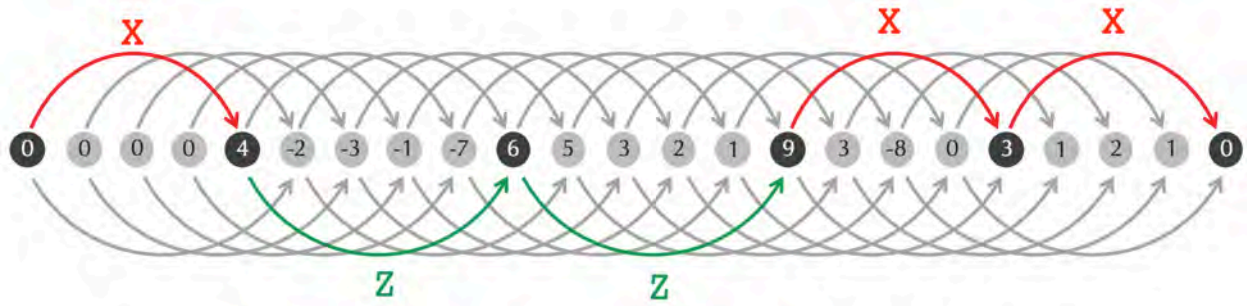
ในส่วนที่เหลือของบทเรียนนี้เราจะใช้สเปคตรัมเวคเตอร์แทนสเปคตรัม โดยถ้ามีข้อมูลเข้าเป็นสเปคตรัมเวคเตอร์ $\overrightarrow{Spectrum}$ เป้าหมายของเราคือหาเปปไทด์ *Peptide* ที่ให้ค่าคะแนน SCORE($\overrightarrow{Peptide}$, $\overrightarrow{Spectrum}$) สูงสุด

นิยามปัญหาที่ 8.4 ปัญหาการถอดรหัสลำดับกรดอะมิโนของเปปไทด์

จากสเปคตรัมเวคเตอร์ที่เป็นข้อมูลเข้า หาเปปไทด์ที่ให้ค่าคะแนนมากที่สุดเมื่อเทียบกับสเปคตรัมเวคเตอร์	
ข้อมูลเข้า	สเปคตรัมเวคเตอร์ $\overrightarrow{Spectrum}$
ผลลัพธ์	ลำดับกรดอะมิโนในเปปไทด์ *Peptide* ที่ทำให้ค่าคะแนน SCORE($\overrightarrow{Peptide}$, $\overrightarrow{Spectrum}$) สูงสุดจากลำดับกรดอะมิโนที่เป็นไปได้ทั้งหมด

อัลกอริทึมเพื่อถอดรหัสลำดับกรดอะมิโนของเปปไทด์

ถ้าข้อมูลเข้าเป็นสเปคตรัมเวคเตอร์ $\overrightarrow{Spectrum} = (s_1, \ldots, s_m)$ เราต้องการสร้างกราฟแบบทิศทางที่ประกอบด้วย m+1 โหนด โดยแต่ละโหนดมีฉลากแสดงค่าน้ำหนักจาก 0 (จุดเริ่มต้น) ถึง m (จุดปลายทาง) และเพิ่มเส้นเชื่อมโหนดจาก i ไปโหนด j ถ้า j – i มีค่าเท่ากับค่าน้ำหนักของกรดอะมิโน รูปที่ 8.7 แสดงกราฟแบบมีทิศทางแสดงการเชื่อมต่อโหนดในสเปคตรัมเวคเตอร์ที่มีจำนวนจุดทั้งหมด 22 จุดและมีกรดอะมิโนสองตัว X และ Z ที่มีค่าน้ำหนัก 4 และ 5 ตามลำดับ เส้นทางจาก 0 ถึง m แสดงเปปไทด์ XZZXX ที่มีค่าเปปไทด์เวคเตอร์เป็น (0, 0, 0, **1**, 0, 0, 0, 0, **1**, 0, 0, 0, 0, **1**, 0, 0, 0, **1,** 0, 0, 0, **1**) โดยมีค่าคะแนนเท่ากับ 0 + 4 + 6 + 9 + 3 + 0



รูปที่ 8.7 กราฟแบบมีทิศทางแสดงการเชื่อมต่อโหนดในสเปคตรัมเวคเตอร์ที่มีจำนวนจุดทั้งหมด 22 จุดและมีกรดอะมิโนสองตัว X และ Z ที่มีค่าน้ำหนัก 4 และ 5 ตามลำดับ

(ที่มา: รูปที่ 11.9 ของ [21])

หยุดคิด	กราฟแบบมีทิศทางในรูปที่ 8.7 นี้ มีความแตกต่างจากกราฟในรูปที่ 8.4 อย่างไร

ทุกเส้นทางจากจุดเริ่มต้นไปยังจุดสิ้นสุดเป็นตัวแทนของลำดับกรดอะมิโนของเปปไทด์และน้ำหนักรวมของทุกโหนดในเส้นทางหนึ่งๆมีค่าเท่ากับ SCORE($\overrightarrow{Peptide}$, $\overrightarrow{Spectrum}$) ดังนั้นความซับซ้อนของปัญหาการถอดรหัสลำดับกรดอะมิโนกลายมาเป็นปัญหาการหาเส้นทางที่มีค่าน้ำหนักรวมมากที่สุดจากจุดเริ่มต้นถึงจุดปลายทาง

เมื่อใช้วิธีการข้างต้นซึ่งมีข้อมูลเข้าเป็นสเปคตรัมเวคเตอร์ของทีเร็กซ์ เราพบเปปไทด์ ATKIVDCFMTY ด้วยคะแนนเป็น 96 (รูปที่ 8.6 กลาง) อย่างไรก็ตาม Asara ได้เสนอเปปไทด์อีกเส้น GLVGAPGLRGLPGK ที่มีคะแนน

ในวิธีการข้างต้นเป็น -19 ที่เราเรียกว่าเป็นทีเร็กซ์เปปไทด์ (รูปที่ 8.6 ล่าง) โดยเปปไทด์ที่เสนอโดย Asara นี้มีคะแนนรวมต่ำกว่า ATKIVDCFMTY มาก และในความเป็นจริงแล้วมีเปปไทด์หลายพันล้านเปปไททด์ได้คะแนนมากกว่าทีเร็กซ์เปปไทด์

หยุดคิด	ในบทที่ 5 เราได้ศึกษาอัลกอริทึมที่ใช้หาเส้นทางที่มีน้ำหนักรวมมากสุด เราสามารถดัดแปลงอัลกอริทึมนั้นเพื่อนำมาใช้ในการหาเส้นทางที่มีค่าน้ำหนักรวมมากที่สุดจากจุดเริ่มต้นถึงจุดปลายทางในกราฟแบบมีทิศทางข้างต้นได้อย่างไร

หยุดคิด	ลองหาคำตอบดูว่าทำไม Asara ถึงเสนอทีเร็กซ์เปปไทด์ แทนที่จะเป็นเปปไทด์ ATKIVDCFMTY ที่มีคะแนนสูงกว่ามาก

การระบุเปปไทด์

ปัญหาการระบุเปปไทด์

ถึงแม้ว่าจะมีความพยายามสร้างฟังก์ชันการให้คะแนนที่มีความสอดคล้องกับความถูกต้องของเปปไทด์ โดยเปปไทด์ที่ทำให้เกิดสเปคตรัมควรเป็นเปปไทด์ที่มีคะแนนสูงสุด แต่ฟังก์ชันการให้คะแนนที่มีอยู่ก็ยังให้ผลที่ไม่ดีพอ อย่างไรก็ตามถึงแม้ว่าเปปไทด์ที่ถูกต้องมักไม่ได้มีคะแนนสูงสุดเมื่อเทียบกับเปปไทด์ทั้งหมด แต่เปปไทด์ที่ถูกต้องนี้ก็มักมีคะแนนสูงสุดเมื่อถูกเปรียบเทียบกับเฉพาะเปปไทด์ที่เป็นส่วนประกอบของโปรตีโอมในสิ่งมีชีวิตหนึ่งๆ ผลที่ตามมาคือเราสามารถที่จะเปลี่ยนจากการถอดรหัสเปปไทด์ผ่านชุดของค่าสเปคตรัมไปเป็นการระบุเปปไทด์โดยเทียบกับข้อมูลเปปไทด์ที่ปรากฏในโปรตีโอมของสิ่งมีชีวิต ทั้งนี้ให้พิจารณาว่าโปรตีโอมเกิดจากการนำเปปไทด์ทั้งหมดที่พบในสิ่งมีชีวิตนั้นๆ มาต่อกันเป็นสายสตริงเส้นยาว

นิยามปัญหาที่ 8.5 ปัญหาการระบุเปปไทด์

หาเปปไทด์จากโปรตีโอมโดยเป็นเปปไทด์ที่ให้ค่าคะแนนสูงสุดเมื่อนำไปเทียบกับสเปคตรัม	
ข้อมูลเข้า	สเปคตรัมเวคเตอร์ $\overrightarrow{Spectrum}$ และลำดับกรดอะมิโนโปรตีโอม *Proteome*
ผลลัพธ์	ลำดับกรดอะมิโนในเปปไทด์ *Peptide* ที่ทำให้ค่าคะแนน SCORE($\overrightarrow{Peptide}$, $\overrightarrow{Spectrum}$) สูงสุด และเป็นลำดับกรดอะมิโนที่อยู่ในโปรตีโอม

หยุดคิด	ในทางปฏิบัติข้อมูลเข้าของการระบุเปปไทด์จะเป็นชุดของสายโปรตีนมากกว่าที่เป็นเป็นโปรตีโอมยาวสายเดียว อะไรที่อาจจะเป็นหลุ่มพรางเมื่อเราใช้ข้อมูลเข้าเป็นโปรตีโอมที่เกิดจากนำโปรตีนสายต่างๆมาเรียงต่อกัน

การระบุเปปไทด์ในโปรตีโอมของทีเร็กซ์

ถึงจุดนี้อาจมีข้อสงสัยว่าทำไมเราถึงมาพิจารณาปัญหาการระบุเปปไทด์ ในเมื่อเราไม่มีข้อมูลโปรตีโอมของทีเร็กซ์ ซึ่งก็หมายความว่าเราไม่น่าจะสามารถระบุเปปไทด์ของทีเร็กซ์โดยใช้โปรตีโอมได้ อย่างไรก็ตามประมาณ 90% ของโปรตีนของกระดูกสัตว์เป็นคอลลาเจน (collagen) ดังนั้นกระดูกไดโนเสาร์ก็น่าจะประกอบด้วยคอลลาเจนเช่นกัน และก็มีความเป็นไปได้น้อยที่โปรตีนอื่นๆจะคงสภาพในฟอสซิลมากว่าหลายล้านปี Asara ให้เหตุผลว่าโปรตีนใดก็ตามที่พบในฟอสซิลของทีเร็กซ์น่าจะมีความคล้ายคลึงกับคอลลาเจนที่พบในสิ่งมีชีวิตในปัจจุบัน

เพื่อเป็นการตรวจสอบสมมติฐานนี้ Asara ทำการเปรียบเทียบสเปคตรัมของทีเร็กซ์กับโปรตีนทั้งหมดที่อยู่ในฐานข้อมูลโปรตีนยูนิพรอต (UniProt) ซึ่งมีข้อมูลของสายโปรตีนจากสิ่งมีชีวิตหลายร้อยสปีชีส์ และมีข้อมูลอย่างน้อย 200 ล้านกรดอะมิโน นอกจากนี้ Asara ยังมีการเพิ่มข้อมูลสายโปรตีนคอลลาเจนของสิ่งมีชีวิตในปัจจุบันที่มีการแปรผันเข้าไปเป็นส่วนหนึ่งของฐานข้อมูลด้วย เพื่อเป็นการจำลองความแตกต่างที่เป็นไปได้ระหว่างคอลลาเจนเหล่านี้และคอลลาเจนของทีเร็กซ์ด้วย (ขอเรียกฐานข้อมูลนี้ว่า UniProt+) ซึ่งพบว่าเปปไทด์ส่วนใหญ่ในฐานข้อมูลที่แมชกับสเปคตรัมด้วยคะแนนสูงๆนั้นเป็นคอลลาเจนของไก่ (chicken) ซึ่งสนับสนุนสมมติฐานว่าสัตว์ปีกมีวิวัฒนาการมาจากไดโนเสาร์ ในความเป็นจริงแล้วทีเร็กซ์เปปไทด์ต่างจากเปปไทด์คอลลาเจนของไก่เพียง 1 กรดอะมิโน อย่างไรก็ตามยังมีคำถามว่าเราจะตรวจสอบได้อย่างไรว่าการตีความเกี่ยวกับทีเร็กซ์เปปไทด์นี้เป็นการตีความที่ถูกต้องเกี่ยวกับข้อมูลสเปคตรัมของไดโนเสาร์

การระบุเปปไทด์กับทฤษฎีลิงพิมพ์ดีด

ถ้ากำหนดให้ PSM (Peptide-Spectrum Match) เป็นปัญหาที่ต่อยอดมาจากปัญหาที่ 8.5 โดยมีการเปลี่ยนข้อมูลเข้าจากสเปคตรัมเวคเตอร์เดียวไปเป็นชุดของสเปคตรัมเวคเตอร์ และมีตัวแปรเพิ่มอีกหนึ่งตัวคือค่า threshold สิ่งที่เป็นผลลัพธ์จะประกอบด้วยชุดของเปปไทด์ที่พบในโปรตีโอมโดยที่มีค่าคะแนนมากกว่าค่า threshold ทั้งนี้ผลของการใช้ PSM เพื่อระบุเปปไทด์จากเวคเตอร์สเปคตรัมก็ยังได้ผลว่าทีเร็กซ์เปปไทด์เป็นเปปไทด์ที่มีค่าคะแนนสูงสุดเมื่อเทียบกับเปปไทด์ที่ได้มาจากสเปคตรัมเวคเตอร์อื่นๆที่ผ่านค่า threshold เช่นกัน เนื่องจากยังมีเปปไทด์อีกหลายพันล้านเส้นที่ไม่ได้อยู่ในฐานข้อมูล UniProt+ นี้ที่มีค่าคะแนนของการถอดรหัสเปปไทด์สูงกว่าทีเร็กซ์เปปไทด์มาก คำถามคือฐานข้อมูล UniProt+ ของ Asara นั้นไม่สมบูรณ์ ไม่เพียงพอ และสเปคตรัมของไดโนเสาร์ก็เป็นผลมาจากเปปไทด์อื่นใช่หรือไม่ หรือ PSM ทำงานไม่ถูกต้อง คำตอบคือมีความเป็นไปได้ว่าจะมีเปปไทด์จำนวนมากมายที่น้ำหนักเท่ากันกับทีเร็กซ์เปปไทด์แต่ให้คะแนนมากกว่าตอนถอดรหัส ดังนั้นไม่มีปัญหากับการทำงานของ PSM แต่สิ่งที่ต้องพิจารณาเพิ่มเติมคือชุดของเปปไทด์ที่เป็นผลลัพธ์จาก PSM นั้นแต่ละเปปไทด์นี้มีความสำคัญทางสถิติมากน้อยเพียงใด

False discovery rate

ในการประมาณจำนวนของเปปไทด์ที่ไม่มีความสำคัญทางสถิติจากชุดของเปปไทด์ที่เป็นผลลัพธ์ของ PSM เรา

สร้างโปรตีโอมหลอกเรียกว่าโปรตีโอมดีคอย (decoy proteome) โดยแต่ละลำดับกรดอะมิโนในโปรตีโอมดีคอยนี้จะมาจากการสุ่มเลือกจากกรดอะมิโนที่เป็นไปได้ 20 ตัว ด้วยความน่าจะเป็นที่เท่ากัน (1/20) โดยจะสร้างโปรตีโอมดีคอยให้มีความยาวเท่ากับโปรตีโอมจริง จากนั้นเมื่อลองใช้ PSM เพื่อหาชุดของเปปไทด์ที่พบในโปรตีโอมดีคอยและมีค่าคะแนนเกินค่า threshold ซึ่งจำนวนของเปปไทด์ที่เป็นผลลัพธ์นี้สามารถนำไปคำนวณค่า false discovery rate (FDR) ได้ตามสมการต่อไปนี้

$$\frac{|PSM_{threshold}(DecoyProteome, SpectralVectors)|}{|PSM_{threshold}(Proteome, SpectralVectors)|}$$

ตัวอย่างเช่น ในการใช้ PSM เพื่อค้นหาเปปไทด์ในโปรตีโอมแล้วได้ผลลัพธ์เป็น 100 เปปไทด์ในขณะที่ถ้าใช้ PSM เพื่อค้นหาเปปไทด์ในโปรตีโอมดีคอยแล้วได้ผลลัพธ์เป็น 5 เปปไทด์ เราสามารถสรุปได้ว่าเปปไทด์ที่หาได้จาก PSSM 95% น่าจะเป็นเปปไทด์ที่ถูกต้อง ในทางกลับกันถ้าผลของการรัน PSM กับโปรตีโอมดีคอยแล้วได้ 100 เปปไทด์เป็นผลลัพธ์ ค่า FDR จะเข้าใกล้ 1 ซึ่งหมายความว่าเราจะตัดสินใจได้ยากว่าเปปไทด์ที่เป็นผลลัพธ์แต่ละเส้นนั้นมีความสำคัญหรือมีความหมายทางชีววิทยาหรือไม่

หยุดคิด	ถ้าค่า FDR มีค่ามากสำหรับค่า threshold หนึ่งๆ เรายังสามารถหาเปปไทด์ที่เป็นผลลัพธ์จาก PSM ที่มีความหมายทางชีววิทยาได้หรือไม่

ถึงแม้ว่า FDR จะมีค่าสูงเราไม่ควรสรุปว่าข้อมูลสเปคตรัมที่มีนั้นไม่มีประโยชน์ หรือไม่ควรสรุปว่าเรากำลังใช้ฐานข้อมูลที่ไม่ถูกต้องหรือไม่ เพราะในความเป็นจริงแล้วค่า FDR ที่สูงนั้นอาจเป็นผลมาจากการเลือกค่า threshold ที่ไม่ถูกต้องก็ได้ เนื่องจากค่า FDR อาจมีความแปรผันมากโดยขึ้นอยู่ค่า threshold นี้

สำหรับข้อมูลสเปคตรัมของทีเร็กซ์เราพบ 27 เปปไทด์เมื่อใช้โปรตีโอม UniProt+ และพบเพียง 1 เปปไทด์ในโปรตีโอมดีคอยเมื่อใช้ค่า threshold = 100 โดย FDR = 3.7% อย่างไรก็ตามเรายังไม่สามารถสรุปได้ว่าทั้ง 26 เปปไทด์ที่เหลือนั้นเป็นเปปไทด์ของไดโนเสาร์ทั้งหมด เนื่องจากหลายๆเปปไทด์ที่ค้นพบนี้เกิดจากการปนเปื้อนที่พบเป็นปกติในกระบวนการทางห้องปฏิบัติการ เป้าหมายถัดไปของเราคือเราจะสามารถประเมินได้อย่างไรว่าเปปไทด์ที่เหลืออยู่นี้เป็นเปปไทด์ของไดโนเสาร์จริงๆ

ในขณะที่ FDR ช่วยประเมินคุณภาพโดยรวมของชุดเปปไทด์ที่เป็นผลลัพธ์จากการรัน PSM โดยใช้สเปคตรัมของทีเร็กซ์ คำถามคือเปปไทด์แต่ละเส้นนี้มีความสำคัญเชิงสถิติมากน้อยแค่ไหน

ลิงกับเครื่องพิมพ์ดีด

ทฤษฎี infinite monkey กล่าวว่าถ้าปล่อยให้ลิงใช้เครื่องพิมพ์ดีดพิมพ์อะไรไปเรื่อยๆ เราจะพบว่าคำที่พิมพ์ออกมานั้นจะมีบางคำที่เป็นคำศัพท์ที่ถูกต้อง ถ้ามีชุดของคำศัพท์ *Dictionary* เรากำหนด E(*Dictionary, n*) เป็น

จำนวนคำศัพท์ใน *Dictionary* ที่คาดหวังว่าจะพบในสายสตริงความยาว n อักขระที่สร้างมาแบบสุ่มโดยแต่ละอักขระมีโอกาสที่จะถูกสุ่มเลือกมาเท่ากัน และกำหนดให้ *EnglishDictionary* แสดงชุดของคำศัพท์ทั้งหมดในภาษาอังกฤษ ถ้าปรากฏว่าหลังจากพิมพ์อักขระมา n อักขระ จำนวนคำศัพท์ภาษาอังกฤษที่ลิงพิมพ์มีจำนวนมากกว่าค่า E(*Dictionary, n*) ชัดเจน เราก็อาจจะสรุปว่าลิงสะกดคำศัพท์ได้

นิยามปัญหาที่ 8.6 ปัญหาลิงและเครื่องพิมพ์ดีด

ประมาณจำนวนคำศัพท์ในพจนานุกรมที่คาดหวังว่าจะพบในสายข้อมูล n อักขระที่สร้างมาแบบสุ่ม	
ข้อมูลเข้า	ชุดของคำศัพท์ *Dictionary* และเลขจำนวนเต็ม n
ผลลัพธ์	E(*Dictionary*, *n*)

หยุดคิด	ลิงกับพิมพ์ดีดเกี่ยวข้องกับแมสสเปคโตรเมทรีอย่างไร

นัยยะสำคัญทางสถิติของ PSM

ถ้าเปลี่ยนจากการปล่อยให้ลิงพิมพ์อักขระไปเรื่อยๆ เป็นการใช้อัลกอริทึมในการสร้างชุดของเปปไทด์ที่มีคะแนนเกินค่า threshold เมื่อเทียบกับเวคเตอร์สเปคตรัม $\overrightarrow{Spectrum}$ และเราเรียกชุดของเปปไทด์นี้ว่าเป็นพจนานุกรมสเปคตรัม (spectral dictionary) ซึ่งแสดงโดย

$$DICTIONARY_{threshold}(\overrightarrow{Spectrum})$$

และสำหรับพจนานุกรมสเปคตรัมของไดโนเสาร์หรือ *DinosaurPSM* จะถูกแสดง โดย

$$DICTIONARY\_{19}(\overrightarrow{DinosaurSpectrum})$$

แทนการตรวจสอบว่าคำศัพท์ที่ลิงพิมพ์ออกมามีคำใดบ้างที่อยู่ในพจนานุกรม เราจะตรวจสอบว่าเปปไทด์ในพจนานุกรมสเปคตรัมปรากฏในโปรตีโอมหรือไม่ ถ้าพบเราจะต้องตัดสินใจว่าเปปไทด์เหล่านี้มีความหมายทางชีววิทยาหรือไม่ หรือเป็นเพียงผลพวงของค่าสถิติที่ใช้ เพื่อให้ตัดสินใจได้เราต้องพิจารณา

$$E(DICTIONARY_{threshold}(\overrightarrow{Spectrum}), n)$$

เป็นจำนวนเปปไทด์ในโปรตีโอมดีคอยความยาว n อักขระ ที่พบใน $DICTIONARY_{threshold}(\overrightarrow{Spectrum})$ ถ้าค่านี้มากกว่า 1 ก็ไม่น่าแปลกใจที่จะพบเปปไทด์ที่มีคะแนนค่า threshold เมื่อเทียบกับสเปคตรัมเวคเตอร์ ดังนั้นจึงต้องมีการกำหนดปัญหาการทดสอบความสำคัญทางสถิติให้ชัดเจนมากขึ้น

เพื่อหาจำนวนเปปไทด์นี้ เราเริ่มจากพจนานุกรมสเปคตรัมที่มีเปปไทด์เพียงเส้นเดียวซึ่งเราจะนำไปตรวจสอบว่าพบเปปไทด์สายนี้ในโปรตีโอมดีคอยที่มีความยาว n อะมิโนหรือไม่ โดยค่าความน่าจะเป็นที่เปปไทด์สายนี้ถูกพบในโปรตีโอมดีคอยที่ตำแหน่งจำเพาะหนึ่งๆมีค่าเท่ากับ $\frac{1}{20^{|Peptide|}}$ ดังนั้นจำนวนครั้งที่เปปไทด์นี้สามารถถูกพบในโปรตีโอมดีคอยมีค่าเท่ากับ

$$\frac{n-|Peptide|+1}{20^{|Peptide|}} \approx n \cdot \frac{1}{20^{|Peptide|}}$$

ถ้าสมมติว่าเรามีชุดของเปปไทด์อยู่ใน *Dictionary* โดยแต่ละเปปไทด์อาจมีความยาวแตกต่างกันไป ถ้าใช้การประมาณค่าข้างต้น ค่าประมาณจำนวนครั้งที่สามารถพบเปปไทด์เหล่านี้ทั้งในโปรตีโอมดีคอยและโปรตีโอมปกติคำนวณได้จากสมการต่อไปนี้

$$E(Dictionary, n) \approx n \cdot \left(\sum_{each\ peptide\ Peptide\ in\ Dictionary} \frac{1}{20^{|Peptide|}} \right)$$

โดยเราจะอ้างอิงถึงพจน์ผลรวมภายในวงเล็บว่าเป็นค่าความน่าจะเป็นของ *Dictionary* แสดงโดย Pr(*Dictionary*) ซึ่งทำให้เราสามารถเขียนสมการข้างต้นได้ใหม่ดังต่อไปนี้

$$E(Dictionary, n) \approx n \cdot \Pr(Dictionary)$$

สมการนี้แสดงการวิเคราะห์ค่าความสำคัญทางสถิติของเปปไทด์ในรูปแบบการคำนวณค่าความน่าจะเป็น จากสมการนี้ถ้าเราต้องการทดสอบความสำคัญเชิงสถิติของ *DinosaurPSM* ขั้นแรกสร้างพจนานุกรม $DICTIONARY(DinosaurPSM)$ และคำนวณค่า

$$n \cdot \Pr(DICTIONARY(DinosaurPSM))$$

โดยที่ n คือความยาวของลำดับกรดอะมิโนที่เกิดจากการนำสายโปรตีนทั้งหมดที่อยู่ในฐานข้อมูล UniProt+ มาต่อกัน ถ้าค่าที่คำนวณได้มีค่าน้อย เช่น 0.001 เราก็จะสามารถยืนยันได้ว่า *DinosaurPeptide* เป็นเปปไทด์ของทีเร็กซ์จริงไม่ใช่ผลพวงที่เกิดจากการคำนวณทางสถิติ อย่างไรก็ตาม $DICTIONARY(DinosaurPSM)$ มีจำนวนเปปไทด์มากกว่า 200 พันล้านเปปไทด์ซึ่งจะใช้เวลานานมากในการคำนวณ คำถามคือเราสามารถจะคำนวณค่าความน่าจะเป็นของพจนานุกรมนี้โดยที่ไม่ต้องสร้างเปปไทด์ออกมาทั้งหมดได้หรือไม่

พจนานุกรมสเปคตรัม

ในขั้นแรกเราจะคำนวณจำนวนเปปไทด์ในพจนานุกรมสเปคตรัม ซึ่งเมื่อทราบจำนวนแล้วเราอาจจะได้แนวทางในการคำนวณค่าความน่าจะเป็นของพจนานุกรมสเปคตรัม

นิยามปัญหาที่ 8.7 ปัญหาการหาขนาดของพจนานุกรมสเปคตรัม

หาขนาดของหรือจำนวนเปปไทด์ของพจนานุกรมสเปคตรัมเมื่อมีข้อมูลเข้าเป็นเวคเตอร์สเปคตรัม และค่า threshold	
ข้อมูลเข้า	เวคเตอร์สเปคตรัม $\overrightarrow{Spectrum}$ และค่าจำนวนเต็ม threshold
ผลลัพธ์	จำนวนของเปปไทด์ใน $DICTIONARY_{threshold}(\overrightarrow{Spectrum})$

เราใช้ dynamic programming ในการคำนวณหาขนาดของพจนานุกรมสเปคตรัม โดยถ้ามีข้อมูลเข้าเป็นเวคเตอร์สเปคตรัม $\overrightarrow{Spectrum} = (s_1, \dots, s_m)$ และกำหนด i-prefix (สำหรับ i ที่มีค่าระหว่าง 1 ถึง m) โดย $\overrightarrow{Spectrum_i} = (s_1, \dots, s_i)$ และมีตัวแปรใหม่ $\mathrm{SIZE(i,t)}$ เป็นจำนวนของเปปไทด์ *Peptides* ที่มีน้ำหนัก i และทำให้ $\mathrm{SCORE}(Peptide, \overrightarrow{Spectrum})$ มีค่าเท่ากับ t ตัวอย่างเช่น ในเวคเตอร์สเปคตรัม $\overrightarrow{Spectrum} = (4, -3, -2, 3, 3, -4, 5, -3, -1, -1,\ 3, 4, 1, 3)$ ที่มีจำนวน 14 ค่า และอักขระที่แสดงประเภทของกรดอะมิโนเพียง 2 กรดคือ X และ Z ซึ่งมีน้ำหนัก 4 และ 5 ตามลำดับ และมีเพียง 3 เปปไทด์ที่มีน้ำหนักเท่ากับ 13 คือ XXZ, XZX, และ ZXX โดยที่ 2 เปปไทด์แรกจะมีค่าคะแนนเป็น 1 เมื่อเทียบกับ $\overrightarrow{Spectrum_{13}}$ ในขณะที่เปปไทด์ที่สามจะมีค่าคะแนนเท่ากับ 3 ดังนั้น $\mathrm{SIZE}(13,1) = 2$ และ $\mathrm{SIZE}(13,3) = 1$ และ $\mathrm{SIZE}(13, t) = 0$ สำหรับค่า t อื่นๆทั้งหมด

หัวใจการทำงานเพื่อให้ได้จำนวนเปปไทด์นี้สามารถแก้ปัญหาด้วยความสัมพันธ์แบบเวียนเกิดสำหรับการคำนวณค่า $\mathrm{SIZE(i,t)}$ โดยชุดของเปปไทด์ที่เป็นสมาชิกของ $\mathrm{SIZE(i,t)}$ นี้สามารถถูกแบ่งออกเป็น 20 กลุ่มย่อยขึ้นอยู่กับกรดอะมิโนตัวสุดท้าย และเปปไทด์ที่ลงท้ายด้วยกรดอะมิโน *a* จำเพาะหนึ่งๆนี้ จะเป็นเปปไทด์ที่สั้นลง 1 กรดอะมิโนและมีน้ำหนักเท่ากับ i - |a| (โดย |a| คือค่าน้ำหนักของกรดอะมิโน *a*) และมีค่าคะแนนเท่ากับ t – s_i ถ้าเรานำกรดอะมิโน *a* ออกจากเปปไทด์ และสามารถแสดงความสัมพันธ์แบบเวียนเกิดในสมการต่อไปนี้

$$\mathrm{SIZE}(i,t) = \sum_{all\ amino\ acids\ a} \mathrm{SIZE}(i - |a|, t - s_i)$$

และเนื่องจากมีเปปไทด์ที่ “ว่าง” คือมีความยาวเป็น 0 อะมิโน ในกรณีนี้ค่า $\mathrm{SIZE}(0,0) = 1$ และมีการกำหนดค่า $\mathrm{SIZE(i,t)} = 0$ สำหรับค่า i ที่เป็นลบ โดยการใช้สมการเวียนเกิดข้างต้นเราสามารถคำนวณหาขนาดของพจนานุกรมสเปคตรัมของ $\overrightarrow{Spectrum} = (s_1, \dots, s_m)$ โดยมีค่าเท่ากับ

$$\left|\mathrm{DICTIONARY}_{threshold}(\overrightarrow{Spectrum})\right| = \sum_{t \geq threshold} \mathrm{SIZE(m,t)}$$

ทั้งนี้สมการ *ค่าความน่าจะเป็น* ของพจนานุกรมคือ

$$\Pr(Dictionary) = \sum_{each\ peptide\ Peptide\ in\ the\ Dictionary} \frac{1}{20^{|Peptide|}}$$

มีความคล้ายคลึงกับสมการที่ใช้หา *ขนาด* ของพจนานุกรม

$$|Dictionary| = \sum_{each\ peptide\ Peptide\ in\ Dictionary} 1$$

จากความคล้ายคลึงกันนี้เราสามารถอนุมานสมการเวียนเกิดในการหาค่าความน่าจะเป็นของพจนานุกรมโดยใช้ตัวแปรที่คล้ายคลึงกับที่ใช้ในกรณีของการหาขนาดของพจนานุกรม

กำหนด $\Pr(i,t)$ ให้เป็นผลรวมของค่าความน่าจะเป็นของเปปไทด์ทั้งหมดที่มีน้ำหนักเท่ากับ i สำหรับ $\text{SCORE}(\overrightarrow{Peptide}, \overrightarrow{Spectrum})$ ที่มีค่าเท่ากับ t โดยที่ชุดของเปปไทด์นี้สามารถแบ่งออกเป็น 20 กลุ่มย่อย ขึ้นอยู่กับว่าลงท้ายด้วยกรดอะมิโนใด โดยแต่ละเปปไทด์ *Peptide* ลงท้ายด้วยกรดอะมิโน *a* จะถูกแสดงโดยเปปไทด์ $Peptide_a$ และถ้าเราเอา *a* เปปไทด์ $Peptide_a$ ก็จะมีน้ำหนักเท่ากับ i - |*a*| และมีคะแนนเท่ากับ t - s_i เนื่องจากค่าความน่าจะเป็นของ *Peptide* มีค่าน้อยกว่าค่าความน่าจะเป็นของ $Peptide_a$ 20 เท่า หรืออีกนัยยะหนึ่งคือ *Peptide* มีส่วนสนันสนุนค่า Pr(i, t) น้อยกว่าที่ $Peptide_a$ สนันสนุนค่า Pr(i-|*a*|, t- s_i) 20 เท่า ดังนั้นค่า Pr(i, t) สามารถคำนวณได้จากสมการต่อไปนี้

$$\Pr(i,t) = \sum_{all\ amino\ acids\ a} \frac{1}{20} \cdot \Pr\left(i - |a|, t - s_i\right)$$

ซึ่งแตกต่างจากสมการเวียนเกิดในการคำนวณ SIZE(i, t) เฉพาะการมีสัมประสิทธิ์ 1/20 มาเป็นตัวคูณเพิ่ม ดังนั้นค่าความน่าจะเป็นของพจนานุกรมสามารถคำนวณได้จากสมการต่อไปนี้

$$\Pr\left(\text{DICTIONARY}_{threshold}\overrightarrow{Spectrum}\right) = \sum_{t \geq threshold} \Pr\left(m,t\right)$$

สำหรับ DICTIONARY(*DinosaurPSM*) ประกอบด้วย 219,136,251,374 เปปไทด์ และมีค่าความน่าจะเป็นเท่ากับ 0.00018 ถึงจุดนี้เราก็พร้อมที่จะทดสอบความสำคัญเชิงสถิติของ *DinosuarPSM* ที่พบในฐานข้อมูล UniProt+ ที่มีความยาว n เท่ากับ 194,613,142 อะมิโนจากจำนวนโปรตีนทั้งหมด 546,799 เส้น โดยเป้าหมายของเราคือการคำนวณค่า $n \cdot \Pr(\text{DICTIONARY}(DinosaurPSM))$ ซึ่งก็คือค่าประมาณของจำนวนเปปไทด์จาก $\text{DICTIONARY}(DinosaurPSM)$ ที่ถูกคาดหวังว่าจะพบในโปรตีโอมดีคอยความยาว n กรดอะมิโน เนื่องจาก $\Pr(\text{DICTIONARY}(DinosaurPSM))$ = 0.00018 และได้ค่าจำนวนเปปไทด์ที่คาดหวังว่าจะพบเท่ากับ 35,311 เปปไทด์จากการคำนวณต่อไปนี้

$$n \cdot \Pr\left(\text{DICTIONARY}(DinosaurPSM)\right) = 35{,}311$$

ดังนั้นเราคาดหวังที่จะพบมากกว่าหมื่นเปปไทด์ที่มีค่าคะแนนอย่างน้อยเท่ากับคะแนนของ *DinosaurPeptide* (เมื่อนำไปเทียบกับ $\overrightarrow{DinosaurSpectrum}$) ในฐานข้อมูลดีคอย ซึ่งก็ไม่เป็นเรื่องน่าแปลกใจถ้าจะพบ

DinosuarPSM ในขณะที่ทำการค้นหาในฐานข้อมูล UniProt+ ซึ่งทำให้สรุปได้ว่า *DinosaurPeptide* เป็นเพียงผลข้างเคียงจากการคำนวณทางสถิติ มากกว่าที่จะเป็นเปปไทด์ของทีเร็กซ์จริงๆ คำถามคือแล้วเปปไทด์อื่นที่ถูกรายงานว่าเป็นเปปไทด์ของทีเร็กซ์นั้นเป็นเปปไทด์ของทีเร็กซ์จริงไหม

เปปไทด์ของทีเร็กซ์ โปรตีนปนเปื้อนหรือขุมสมบัติล้านปี

ปริศนาฮีโมโกลบิน

หลังจากได้รับคำวิพากษ์วิจารณ์เกี่ยวกับค่าสถิติที่ใช้เบื้องหลังในการสรุปผลเกี่ยวกับเปปไทด์ของทีเร็กซ์ Asara ยอมรับปัญหาบางส่วนที่เกิดจากการวิเคราะห์ข้อมูลของเขาและมีการถอน *DinosaurPeptide* จากการตีความว่าเป็นของเปปไทด์ที่แสดง *DinosaurSpectrum* และทำการปรับบางเปปไทด์ที่เคยเสนอไว้ว่าเป็นเปปไทด์ของทีเร็กซ์ออก รวมทั้งเปิดข้อมูลทั้ง 31,372 สเปคตรัมที่ได้จากฟอสซิลของทีเร็กซ์ หลังจากนั้นนักวิทยาศาสตร์คนอื่นๆ ได้ทำการวิเคราะห์ข้อมูลใหม่สำหรับทุกสเปคตรัมและยืนยันว่าถึงแม้บางเปปไทด์ที่ถูกรายงานโดย Asara อาจเป็นที่สงสัยแต่ก็มีชุดเปปไทด์ที่มีความสำคัญจริงในเชิงสถิติ รูปที่ 8.8 แสดงเปปไทด์ 7 เส้นที่อาจเป็นตัวแทนเปปไทด์คอลลาเจนของทีเร็กซ์ (P1-P7) ที่ถูกรายงานโดย Asara รวมทั้งเปปไทด์ฮีโมโกลบิน (P8) โดยคอลัมน์สุดท้ายแสดงค่าความน่าจะเป็นของพจนานุกรม PSM ที่มีเปปไทด์เหล่านี้เป็นสมาชิก อักษรสีแดงแสดงกรดอะมิโนที่แตกต่างไปจากเปปไทด์ในฐานข้อมูล UniProt และกรดอะมิโน P_{oh} แสดง hydroxyproline ซึ่งเป็นรูปแบบหนึ่งของโพลีน (proline) ที่มีการถูกดัดแปลงและพบโดยทั่วไปในคอลลาเจน

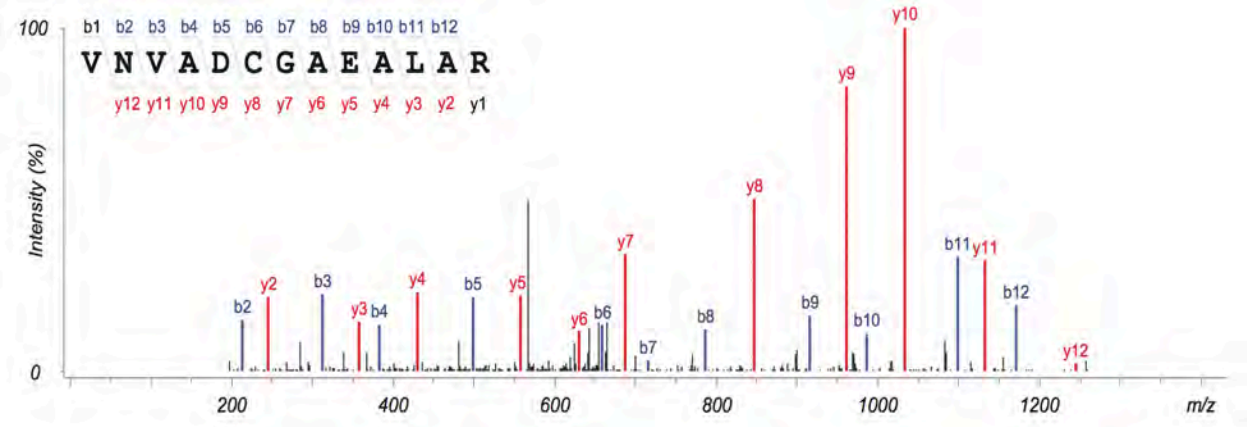
ชุดสเปคตรัมที่ Asara เปิดเผยออกมาทั้งหมดนี้ก่อให้เกิดคำถามเพิ่มเติมมากกว่าตอบคำถาม โดยจากชุดของสเปคตรัมเหล่านี้ Matthew Fitzgibbon และ Martin McIntosh พบสเปคตรัมเพิ่มเติม (รูปที่ 8.9) ซึ่งตรงกับเปปไทด์ฮีโมโกลบินของนกกระจอกเทศ ดังนั้นจึงมีการเพิ่มเปปไทดของทีเร็กซ์ไปในบรรทัดที่ 8 (P8) ของรูปที่ 8.8 โดยฮีโมโกลบิน PSM นี้ไม่ได้ถูกรายงานไว้โดย Asara ทั้งที่ในความเป็นจริงแล้วเปปไทด์เส้นนี้มีความสำคัญเชิงสถิติมากกว่าเปปไทด์ทุกเส้นที่ Asara รายงานไว้ว่าเป็นเปปไทด์คอลลาเจนของทีเร็กซ์ และจะมีความน่าตกใจมากกว่าถ้าเปปไทด์ฮีโมโกลบินนี้เป็นของทีเร็กซ์จริงๆ เนื่องจากฮีโมโกลบินมีความอนุรักษ์ (conserved) ระหว่างสิ่งมีชีวิตต่างๆ น้อยกว่าคอลลาเจนมาก ตัวอย่างเช่น เบต้าเชนฮีโมโกลบินของมนุษย์มีความยาว 146 กรดอะมิโน และมีความยาวแตกต่างจากหนู (mouse) จิงโจ้ (kangaroo) และไก่ (chicken) เท่ากับ 27, 38, และ 45 กรดอะมิโน ตามลำดับ นอกจากนี้เปปไทด์ฮีโมโกลบินที่สมบูรณ์ก็ไม่เคยถูกพบในฟอสซิลที่อายุน้อยกว่ามากๆหรือฟอสซิลที่ถูกขุดพบโดยทั่วไป ฟอสซิลเหล่านี้มักถูกพบในถ้ำแถวยุโรปโดยจะถูกใช้เป็นแหล่งฟอส เฟตเพื่อผลิตดินปืนระหว่างสงครามโลกครั้งที่ 1 เพราะว่า Asara ได้ทำการวิเคราะห์ตัวอย่างนกกระจอกเทศหลายตัวอย่าง ก่อนมาวิเคราะห์ตัวอย่างทีเร็กซ์ Fitzgibbon และ McIntosh จึงตั้งข้อสงสัยว่าเปปไทด์ฮีโมโกลบินที่พบนั้นอาจเป็นผลของการเกิดการปนเปื้อนของตัวอย่างในลักษณะที่เรียกว่า carryover หรือการระบุเปปไทด์ของการทดลองครั้งที่ผ่านมาที่ยังตกค้างอยู่ในเครื่องแมสสเปคโตเมทรี และการปนเปื้อนนี้เป็นสิ่งที่เกิดขึ้นได้ในห้องปฏิบัติการทางโปรตีโอมิกส์ในที่ต่างๆ ผู้เชี่ยวชาญทางด้านแมสสเปคโตรเทรีจะไม่แปลกใจเลยถ้าพบว่าในตัวอย่างที่ทำการวิเคราะห์ข้อมูล

อยู่นั้นมีเคราตินของมนุษย์ปนอยู่ด้วย ทั้งนี้เพราะอากาศในห้องต่างๆมักมีชิ้นส่วนผิวหนังมนุษย์ขนาดเล็กมากๆ เป็นหลายล้านชิ้นส่วนเป็นส่วนประกอบ

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 แสดงเปปไทด์ 7 เส้นที่อาจเป็นตัวแทนเปปไทด์คอลลาเจนของทีเร็กซ์ (P1-P7) ที่ถูกรายงานโดย Asara รวมทั้งเปปไทด์ฮีโมโกลบิน (P8) ที่ไม่ได้ถูกรายงาน

(ที่มา: รูปที่ 11.13 ของ [21])



รูปที่ 8.9 สเปคตรัมของทีเร็กซ์คุณภาพสูงที่ตรงกับเปปไทด์ฮีโมโกลบินของนกกระจอกเทศ VNVADCGGAEAIAR โดยทั้งพรีฟิกซ์และซัฟฟิกซ์ที่เป็นไปได้เกือบทั้งหมดถูกแสดงโดยพีคที่มีค่า intensity สูง และการทำระบุเปปไทด์จากสเปคตรัมโดยตรงก็ให้ผลเป็นเปปไทด์เดียวกัน

(ที่มา: รูปที่ 11.14 ของ [21])

ถ้าเปปไทด์ฮีโมโกลบินเป็น carryover ก็หมายความว่าตัวอย่างทีเร็กซ์นี้เกิดการปนเปื้อนและหมายถึงเปปไทด์ต่างๆที่มีการรายงานว่าเป็นของทีเร็กซ์ก็จะไม่มีความหมาย อย่างไรก็ตาม Asara สามารถแสดงได้ว่าไม่มีการปนเปื้อนในการทดลองของเขาและฮีโมโกลบินของนกกระจอกเทศจะต้องเป็นหนึ่งในเปปไทด์ของทีเร็กซ์ ซึ่งทำให้ขยายจำนวนกลุ่มของโปรตีนที่สามารถพบได้ในฟอสซิลโบราณที่มีอายุมากกว่า 68 ล้านปี นอกเหนือจากคอลลาเจน อย่างไรก็ตามถ้าฟอสซิลทีเร็กซ์ของฮอนเนอร์เป็นขุมสมบัติของโปรตีนโบราณและเราเชื่อว่าเปปไทด์ฮีโมโกล

บินมาจากทีเร็กซ์จริง ทำไมเราถึงควรต้องจำกัดการสืบค้นเปปไทด์ในฐานข้อมูลไปที่กลุ่มเปปไทด์คอลลาเจนและเปปไทด์คอลลาเจนที่มีความแปรผัน ทำไมเราถึงไม่สืบค้นกับโปรตีนทั้งหมดของสัตว์มีกระดูกสันหลัง และในการสืบค้นเราก็ยังสามารถใช้เงื่อนไขเดียวกันกับที่ Asara ใช้ได้ เช่น เปปไทด์ที่สืบค้นกับเปปไทด์ในฐานข้อมูลแตกต่างกันได้อย่างมาก 1 กรดอะมิโน เป็นต้น ซึ่งถ้าเราใช้เงื่อนไขนี้ เราก็จะได้เปปไทด์เพิ่มเติมจากทั้งนกกระจอกเทศ ไก่ หนู (mouse) และมนุษย์ ซึ่งเปปไทด์เหล่านี้ทำให้การสรุปผลของ Asara ว่าสัตว์ปีกและไดโนเสาร์มีความเกี่ยวเนื่องกันในเชิงอณูชีววิทยานั้นมีน้ำหนักน้อยลงไปอีก ในทางกลับกันถ้าเราจะทำการยกเลิกเปปไทด์เพิ่มเติมเหล่านี้ทั้งหมดโดยบอกว่าเป็นผลข้างเคียงของค่าทางสถิติ เราก็อาจจำเป็นต้องยกเลิกเปปไทด์ของทีเร็กซ์ในรูปที่ 8.8 ด้วย

ข้อโต้เถียงเกี่ยวกับดีเอ็นเอของไดโนเสาร์

ถึงแม้ว่าผลงานตีพิมพ์เกี่ยวกับเปปไทด์ของไดโนเสาร์ยังคงเป็นที่โต้แย้งกันอยู่ ในความเป็นจริงแล้วผลงานนี้ไม่ใช่ผลงานวิจัยแรกที่รายงานเกี่ยวกับสารพันธุกรรมของไดโนเสาร์ ในปีค.ศ. 1994 Scott Woodward ประกาศว่าได้ทำการถอดรหัสดีเอ็นเอของกระดูกไดโนเสาร์ที่มีอายุ 80 ล้านปี คำวิพากษ์วิจารณ์ที่ร้อนแรงที่สุดของผลงานวิจัยนี้คือ เชื่อหรือไม่เชื่อ (Believe it or not) – Mark Schweitzer เป็นคนพิสูจน์ว่าผลงานของ Woodward เป็นเพียงรหัสพันธุกรรมของมนุษย์ที่ปนเปื้อน

บทส่งท้าย

การเปลี่ยนแปลงสายเปปไทด์หลังการแปลรหัส

อัลกอริทึม PSM สามารถใช้ในการค้นหาสายเปปไทด์ที่มีอยู่ในฐานข้อมูลเท่านั้น ซึ่งฐานข้อมูลมักมีเฉพาะข้อมูลสายเปปไทด์ปกติโดยไม่รวมสายเปปไทด์ที่แปรผันไป

หยุดคิด	เราจะสามารถปรับแก้อัลกอริทึม PSM ให้สามารถค้นหาเปปไทด์ที่มีการแปรผันได้หรือไม่

ในการค้นหาเปปไทด์ในฐานข้อมูลโดยเปปไทดนั้นอาจมีความแปรผันไปมากสุด n กรดอะมิโน วิธีการหนึ่งคือ เราจะสร้างสายเปปไทด์ทุกรูปแบบที่เป็นไปได้ที่มีการแปรผันไปอย่างมาก n กรดอะมิโน และนำเปปไทด์เหล่านี้เข้าไปรวมเป็นส่วนหนึ่งของโปรตีโอมตั้งต้น และทำการรันอัลกอริทึม PSM กับโปรตีโอมใหม่นี้ อย่างไรก็ตามแนวทางนี้มักใช้ไม่ได้ในเชิงปฏิบัติ เนื่องจากโปรตีโอมใหม่นี้จะมีชุดของเปปไทด์จำนวนมากๆ แม้แต่ในกรณีอนุญาตให้มีการแปรผันได้เพียง 1 กรดอะมิโน

หยุดคิด	เปปไทด์ที่ต้องสร้างขึ้นใหม่มีจำนวนทั้งสิ้นกี่เส้น ถ้าสายเปปไทด์ยาว L กรดอะมิโน และมีการแปรผันได้อย่างมาก n กรดอะมิโน

นอกจากจะต้องค้นหาเปปไทด์ที่มีความแปรผันข้างต้นจากฐานข้อมูลโปรตีโอมแล้ว เรายังต้องค้นหาเปปไทด์ที่ถูกเปลี่ยนแปลงไปหลังการแปลรหัส (post-translational modification) ซึ่งกรดอะมิโนบางตำแหน่งจะถูกเปลี่ยนแปลงไปหลังกระบวนการแปลรหัสจากเมสเซนเจอร์อาร์เอ็นเอมาเป็นโปรตีน ซึ่งในความเป็นจริงแล้วโปรตีนแทบทั้งหมดจะถูกเปลี่ยนแปลงหลังการแปลรหัส โดยลักษณะของการเปลี่ยนแปลงที่ถูกค้นพบและรายงานมีจำนวนหลายร้อยประเภท ตัวอย่างเช่น ปฏิกิริยาเอนไซม์ของหลายๆโปรตีนถูกควบคุมโดยการเพิ่มหรือการลดกลุ่มฟอสเฟตที่ตำแหน่งกรดอะมิโนจำเพาะ โดยขบวนการนี้เรียกว่า ฟอสฟอรีเลชัน (phosphorylation) สามารถทำปฏิกิริยากลับด้านได้ (reversible) โดยโปรตีนกลุ่มไคเนส (protein kinases) ทำหน้าที่เพิ่มกลุ่มฟอสเฟตในขณะที่โปรตีนกลุ่มฟอสฟาเตส (protein phosphatases) ทำหน้าที่ดึงกลุ่มฟอสเฟตออก จากรูปที่ 8.8 จะสังเกตได้ว่าเปปไทด์ของทีเร็กซ์แทบทั้งหมดกรดอะมิโนโปรลีน (proline น้ำหนัก 97) จะถูกเปลี่ยนแปลงไปเป็นไฮดรอกซีโปรลีน (hydroxyproline น้ำหนัก 113) โดยไฮดรอกซีโปรลีนนี้เป็นองค์ประกอบหลักของคอลลาเจน โดยมีความสำคัญในการเพิ่มความเสถียรให้กับคอลลาเจน

ตัวอย่างของการเปลี่ยนแปลงโปรตีนหลังการแปลรหัสที่พบไม่มากนักแต่มีความสำคัญมากเช่นกัน เช่น diphthamide ซึ่งเป็นผลจากการเปลี่ยนแปลงฮีสทีดีน (histidine) โดยพบเพียงในโปรตีน protein synthesis elongation factor-2 แต่โปรตีนนี้ถูกพบในยูแคริโอต (eukaryotes) ทั้งหมด นักวิจัยพบว่า diphthamide นี้เป็นเป้าหมายของหลายท็อกซิน (toxin) หรือสารเป็นพิษที่ผลิตโดยแบคทีเรียก่อโรคหลายชนิด และนำไปสู่คำถามว่าทำไมสิ่งมีชีวิตในกลุ่มยูแคริโอตยังรักษาลักษณะการเปลี่ยนแปลงนี้ไว้ ไม่สูญหายไปในขบวนการวิวัฒนาการ เพราะลักษณะการเปลี่ยนแปลงนี้ทำให้ยูแคริโอตมีความเสี่ยงในการเกิดโรคจากแบคทีเรียก่อโรค ดังนั้นสมมติฐานคือลักษณะการเปลี่ยนแปลงนี้น่าจะมีความสำคัญเพียงแต่เรายังไม่ทราบฟังก์ชันการทำงาน

ตัวอย่างโปรแกรมที่มีการใช้งานกันอย่างแพร่หลาย

เทคโนโลยีแมสสเปคโตรเมทรีมีการประยุกต์ใช้ในงานวิจัยในมิติต่างๆอย่างกว้างขวาง เช่น การศึกษาเกี่ยวกับโรคมะเร็ง [192, 193] การศึกษาเมตาโบลิซึมของมะเร็ง [194] การหาโปรตีนที่เป็นตัวชี้วัดทางชีวภาพของการเป็นโรคมะเร็งตับอ่อน [195] โรคเอสแอลอี [196] การศึกษาแปรผันของโปรตีนกลุ่ม SPOP ในผู้ป่วยโรคมะเร็ง [197] การศึกษาการมีปฏิสัมพันธ์ระหว่างโปรตีน [198] การวิเคราะห์โครงสร้างของโปรตีน [199] การศึกษาเกี่ยวกับการทำลายดีเอ็นเอและความเกี่ยวข้องกับยูบิควิทิเลชันของโปรตีน (protein ubiquitylation) [200] เป็นต้น

โดยโปรแกรมที่มีการใช้กันอย่างแพร่หลายโปรแกรมหนึ่งคือโปรแกรม Mascot ซึ่งถูกพัฒนาและตีพิมพ์ผลงานวิจัยครั้งแรกในปี ค.ศ.1999 [201] โดยเป็นโปรแกรมที่ใช้สืบค้นฐานข้อมูลของโปรตีนโดยใช้ข้อมูลแมสสเปคตรัม และมีการออกแบบการคำนวณค่าคะแนนที่มีความเสถียรมากขึ้นในปีค.ศ. 2008 [202] โปรแกรม MaxQuant เป็นอีกโปรแกรมที่มีการใช้งานอย่างแพร่หลายโดย MaxQuant ถูกตีพิมพ์ครั้งแรกในปีค.ศ. 2008 [203] โดยเป็นการรวมชุดของอัลกอริทึมที่ใช้ในการวิเคราะห์ข้อมูลแมสสเปคโตรเมทรีที่มีความละเอียดสูง โดยมีการใช้การวิเคราะห์สหสัมพันธ์และทฤษฎีกราฟเป็นเครื่องมือในการตรวจจับพีค ชุดของไอโซโทป (isotope

clusters) และคู่ของเปปไทด์ที่มีการการติดสลากด้วยกรดอะมิโนหรือไซแลค (SILAC: Stable Isotope Labelling by/with Amino acids in Cell culture) ในรูปแบบของออบเจ็ค 3 มิติของค่า m/z ค่า elution time และค่า signal intensity ตามลำดับ ทั้งนี้ MaxQuant เวอร์ชันแรกนี้มีการใช้ Mascot ในการหาชุดของเปปไทด์ที่อาจเป็นตัวทำให้เกิดสเปคตรัมหนึ่งๆ โปรแกรม MaxQuant [204] ที่ตีพิมพ์ปีค.ศ. 2016 ได้สนับสนุนแพลตฟอร์มแมสสเปคโตรเทรีที่หลากหลายมากขึ้นรวมทั้งมีการนำ Andromeda [205] เข้ามาแทน Mascot ซึ่งเป็นโปรแกรมในเชิงพาณิชย์ นอกจาก Mascot และ Andromeda แล้วยังมีโปรแกรม SEQUEST [206] ที่มีการใช้งานอย่างแพร่หลายและมีการพัฒนาเป็นเครื่องมือแรกๆและตีพิมพ์ครั้งแรกในปีค.ศ.1994 และ Tide [207] เป็นโปรแกรมที่เน้นการเพิ่มความเร็วของ SEQUEST โปรแกรม MSFragger [208] เป็นอีกโปรแกรมที่ใช้ในสืบค้นฐานข้อมูลโปรตีนโดยใช้ข้อมูลสเปคตรัม

นอกจากงานวิจัยข้างต้นที่เน้นอัลกอริทึมและวิธีการวิเคราะห์ข้อมูล ผลงานตีพิมพ์อีกลักษณะหนึ่งคือการนำเสนอไปป์ไลน์หรือเวิร์คโฟล์ที่ใช้ในการวิเคราะห์ข้อมูลแมสเปคโตรเมทรี เช่น [209, 210] เป็นต้น Tao Chen และคณะ [211] ได้รีวิวเว็บรีซอร์สต่างๆที่สนับสนุนงานวิจัยที่เกี่ยวข้องกับแมสสเปคโตรเมทรีสำหรับงานทางด้านโปรตีโอมิกส์ รายละเอียดเพิ่มเติมเกี่ยวกับเทคโนโลยีที่เกี่ยวข้องกับแมสสเปคโตรเมทรีและการประยุกต์ใช้สามารถศึกษาเพิ่มเติมได้จาก [212-214] และวิธีการเชิงคำนวณต่างๆที่เกี่ยวข้องกับการวิเคราะห์ข้อมูลแมสสเปคโตรเมทรีสำหรับงานวิจัยเชิงโปรตีโอมิกส์สามารถศึกษาเพิ่มเติมได้จากบทในหนังสือเขียนโดย Li, S และคณะ [215] Tommi Välikangas และคณะได้ทำการเปรียบเทียบเวิร์คโฟล์ต่างๆที่ใช้ในการวิเคราะห์ข้อมูลแมสสเปคโตรเมทรีของโปรตีโอมิกส์ใน [216] เป็นต้น

แบบฝึกหัดบทที่ 8

ให้เขียนโปรแกรมเพื่อแก้ปัญหาที่เกี่ยวข้องกับการวิเคราะห์ข้อมูลแมสสเปคโตรเมทรีโดยใช้โจทย์ที่โรซาลินด์ต่อไปนี้

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. Miller, J.R., S. Koren, and G. Sutton, *Assembly algorithms for next-generation sequencing data.* Genomics, 2010. 95(6): p. 315-27.
18. 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.
19. 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.
20. 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.
21. Compeau, P. and P. Pevzner, *Bioinformatics algorithms : an active learning approach*. 2nd Edition. ed. 2015, La Jolla, CA: Active Learning Publishers. volumes.
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. 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).
26. 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.
27. Thomas Shafee, R.L., *Eukaryotic and prokaryotic gene structure.* WikiJournal of Medicine, 2017. 4(1): p. 2.
28. Crick, F.H., *On protein synthesis.* Symp Soc Exp Biol, 1958. 12: p. 138-63.
29. Crick, F., *Central dogma of molecular biology.* Nature, 1970. 227(5258): p. 561-3.
30. Horgan, R.P. and L.C. Kenny, *'Omic' technologies: genomics, transcriptomics, proteomics and metabolomics.* The Obstetrician & Gynaecologist, 2011. 13(3): p. 189-195.

31. 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.
32. Ewing, B. and P. Green, *Base-calling of automated sequencer traces using phred. II. Error probabilities.* Genome Res, 1998. 8(3): p. 186-94.
33. Mouse Genome Sequencing, C., et al., *Initial sequencing and comparative analysis of the mouse genome.* Nature, 2002. 420(6915): p. 520-62.
34. Gibbs, R.A., et al., *Genome sequence of the Brown Norway rat yields insights into mammalian evolution.* Nature, 2004. 428(6982): p. 493-521.
35. Ostrander, E.A. and R.K. Wayne, *The canine genome.* Genome Res, 2005. 15(12): p. 1706-16.
36. Chimpanzee, S. and C. Analysis, *Initial sequence of the chimpanzee genome and comparison with the human genome.* Nature, 2005. 437(7055): p. 69-87.
37. Rhesus Macaque Genome, S., et al., *Evolutionary and biomedical insights from the rhesus macaque genome.* Science, 2007. 316(5822): p. 222-34.
38. Wade, C.M., et al., *Genome sequence, comparative analysis, and population genetics of the domestic horse.* Science, 2009. 326(5954): p. 865-7.
39. Samollow, P.B., *The opossum genome: insights and opportunities from an alternative mammal.* Genome Res, 2008. 18(8): p. 1199-215.
40. Zimin, A.V., et al., *A whole-genome assembly of the domestic cow, Bos taurus.* Genome Biol, 2009. 10(4): p. R42.
41. Li, R., et al., *The sequence and de novo assembly of the giant panda genome.* Nature, 2010. 463(7279): p. 311-7.
42. 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.
43. Goff, S.A., et al., *A draft sequence of the rice genome (Oryza sativa L. ssp. japonica).* Science, 2002. 296(5565): p. 92-100.
44. Rahman, A.Y., et al., *Draft genome sequence of the rubber tree Hevea brasiliensis.* BMC Genomics, 2013. 14: p. 75.
45. Tang, C., et al., *The rubber tree genome reveals new insights into rubber production and species adaptation.* Nat Plants, 2016. 2(6): p. 16073.
46. 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.
47. Teh, B.T., et al., *The draft genome of tropical fruit durian (Durio zibethinus).* Nat Genet, 2017. 49(11): p. 1633-1641.

48. Marx, V., *The DNA of a nation.* Nature, 2015. 524(7566): p. 503-5.
49. Head, S.R., et al., *Library construction for next-generation sequencing: overviews and challenges.* Biotechniques, 2014. 56(2): p. 61-4, 66, 68, passim.
50. 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.
51. 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.
52. 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.
53. Boetzer, M., et al., *Scaffolding pre-assembled contigs using SSPACE.* Bioinformatics, 2011. 27(4): p. 578-9.
54. Luo, R., et al., *SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler.* Gigascience, 2012. 1(1): p. 18.
55. Luo, R., et al., *Erratum: SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler.* Gigascience, 2015. 4: p. 30.
56. Butler, J., et al., *ALLPATHS: de novo assembly of whole-genome shotgun microreads.* Genome Res, 2008. 18(5): p. 810-20.
57. 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.
58. Simpson, J.T., et al., *ABySS: a parallel assembler for short read sequence data.* Genome Res, 2009. 19(6): p. 1117-23.
59. Nagarajan, N. and M. Pop, *Sequence assembly demystified.* Nat Rev Genet, 2013. 14(3): p. 157-67.
60. Warr, A., et al., *Exome Sequencing: Current and Future Perspectives.* G3 (Bethesda), 2015. 5(8): p. 1543-50.
61. 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.
62. Reinert, K., et al., *Alignment of Next-Generation Sequencing Reads.* Annu Rev Genomics Hum Genet, 2015. 16: p. 133-51.
63. Handelsman, J., *Metagenomics: application of genomics to uncultured microorganisms.* Microbiol Mol Biol Rev, 2004. 68(4): p. 669-85.

64. Thomas, T., J. Gilbert, and F. Meyer, *Metagenomics - a guide from sampling to data analysis.* Microb Inform Exp, 2012. 2(1): p. 3.

65. Schnable, P.S., et al., *The B73 maize genome: complexity, diversity, and dynamics.* Science, 2009. 326(5956): p. 1112-5.

66. Venter, J.C., et al., *The sequence of the human genome.* Science, 2001. 291(5507): p. 1304-51.

67. 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.

68. Seo, J.S., et al., *De novo assembly and phasing of a Korean human genome.* Nature, 2016. 538(7624): p. 243-247.

69. Roach, J.C., et al., *Analysis of genetic inheritance in a family quartet by whole-genome sequencing.* Science, 2010. 328(5978): p. 636-9.

70. Ferragina, P. and B.B. Mishra, *Algorithms in Stringomics (I): Pattern-Matching against "Stringomes".* bioRxiv, 2014.

71. 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.

72. 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.

73. Schneeberger, K., et al., *Simultaneous alignment of short reads against multiple genomes.* Genome Biol, 2009. 10(9): p. R98.

74. Dilthey, A., et al., *Improved genome inference in the MHC using a population reference graph.* Nat Genet, 2015. 47(6): p. 682-8.

75. Li, H. and R. Durbin, *Fast and accurate short read alignment with Burrows-Wheeler transform.* Bioinformatics, 2009. 25(14): p. 1754-60.

76. Li, H. and R. Durbin, *Fast and accurate long-read alignment with Burrows-Wheeler transform.* Bioinformatics, 2010. 26(5): p. 589-95.

77. Langmead, B., et al., *Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.* Genome Biol, 2009. 10(3): p. R25.

78. Langmead, B. and S.L. Salzberg, *Fast gapped-read alignment with Bowtie 2.* Nat Methods, 2012. 9(4): p. 357-9.

79. 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.

80. Li, R., et al., *SOAP2: an improved ultrafast tool for short read alignment.* Bioinformatics, 2009. 25(15): p. 1966-7.
81. Liu, C.M., et al., *SOAP3: ultra-fast GPU-based parallel alignment tool for short reads.* Bioinformatics, 2012. 28(6): p. 878-9.
82. Fonseca, N.A., et al., *Tools for mapping high-throughput sequencing data.* Bioinformatics, 2012. 28(24): p. 3169-77.
83. 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.
84. Li, H., et al., *The Sequence Alignment/Map format and SAMtools.* Bioinformatics, 2009. 25(16): p. 2078-9.
85. Konopka, R.J. and S. Benzer, *Clock mutants of Drosophila melanogaster.* Proc Natl Acad Sci U S A, 1971. 68(9): p. 2112-6.
86. 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.
87. Bailey, T.L., et al., *MEME SUITE: tools for motif discovery and searching.* Nucleic Acids Res, 2009. 37(Web Server issue): p. W202-8.
88. Bailey, T.L., et al., *The MEME Suite.* Nucleic Acids Res, 2015. 43(W1): p. W39-49.
89. Das, M.K. and H.K. Dai, *A survey of DNA motif finding algorithms.* BMC Bioinformatics, 2007. 8 **Suppl** 7: p. S21.
90. Jayaram, N., D. Usvyat, and R.M. AC, *Evaluating tools for transcription factor binding site prediction.* BMC Bioinformatics, 2016.
91. 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.
92. 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.
93. 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.
94. Khan, A. and A. Mathelier, *JASPAR RESTful API: accessing JASPAR data from any programming language.* Bioinformatics, 2017.
95. 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.

96. 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.

97. 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.

98. 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.

99. 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.

100. Weirauch, M.T., et al., *Determination and inference of eukaryotic transcription factor sequence specificity.* Cell, 2014. 158(6): p. 1431-1443.

101. 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.

102. 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.

103. Heinemeyer, T., et al., *Databases on transcriptional regulation: TRANSFAC, TRRD and COMPEL.* Nucleic Acids Res, 1998. 26(1): p. 362-7.

104. Wingender, E., et al., *TRANSFAC: an integrated system for gene expression regulation.* Nucleic Acids Res, 2000. 28(1): p. 316-9.

105. Wingender, E., et al., *TRANSFAC: a database on transcription factors and their DNA binding sites.* Nucleic Acids Res, 1996. 24(1): p. 238-41.

106. 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.

107. 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.

108. 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.

109. Munch, R., et al., *PRODORIC: prokaryotic database of gene regulation.* Nucleic Acids Res, 2003. 31(1): p. 266-9.

110. Stormo, G.D., *DNA motif databases and their uses.* Curr. Protoc. Bioinform., 2015. 51(2.15.1-2.15.6).

111. 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.

112. 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.
113. 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.
114. Crooks, G.E., et al., *WebLogo: a sequence logo generator.* Genome Res, 2004. 14(6): p. 1188-90.
115. 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.
116. 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.
117. 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.
118. 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.
119. 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.
120. Xiong, J., *Essential Bioinformatics*. 2006, Cambridge: Cambridge University Press.
121. Smith, T.F. and M.S. Waterman, *Identification of common molecular subsequences.* J Mol Biol, 1981. 147(1): p. 195-7.
122. 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.
123. 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.
124. 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.
125. Edgar, R.C., *MUSCLE: multiple sequence alignment with high accuracy and high throughput.* Nucleic Acids Res, 2004. 32(5): p. 1792-7.
126. Sievers, F. and D.G. Higgins, *Clustal Omega for making accurate alignments of many protein sequences.* Protein Sci, 2018. 27(1): p. 135-145.
127. Sievers, F. and D.G. Higgins, *Clustal Omega, accurate alignment of very large numbers of sequences.* Methods Mol Biol, 2014. 1079: p. 105-16.

128. 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.

129. 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.

130. Henikoff, J.G. and S. Henikoff, *Blocks database and its applications.* Methods Enzymol, 1996. 266: p. 88-105.

131. Clamp, M., et al., *The Jalview Java alignment editor.* Bioinformatics, 2004. 20(3): p. 426-7.

132. Waterhouse, A.M., et al., *Jalview Version 2--a multiple sequence alignment editor and analysis workbench.* Bioinformatics, 2009. 25(9): p. 1189-91.

133. Ganss, B. and A. Jheon, *Zinc finger transcription factors in skeletal development.* Crit Rev Oral Biol Med, 2004. 15(5): p. 282-97.

134. 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.

135. 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.

136. Bystroff, C. and A. Krogh, *Hidden Markov Models for prediction of protein features.* Methods Mol Biol, 2008. 413: p. 173-98.

137. Won, K.J., et al., *An evolutionary method for learning HMM structure: prediction of protein secondary structure.* BMC Bioinformatics, 2007. 8: p. 357.

138. 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.

139. Pierleoni, A., P.L. Martelli, and R. Casadio, *PredGPI: a GPI-anchor predictor.* BMC Bioinformatics, 2008. 9: p. 392.

140. Wu, J. and J. Xie, *Hidden Markov model and its applications in motif findings.* Methods Mol Biol, 2010. 620: p. 405-16.

141. 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.

142. Wang, T., et al., *Finding RNA-Protein Interaction Sites Using HMMs.* Methods Mol Biol, 2017. 1552: p. 177-184.

143. Sgouralis, I. and S. Presse, *An Introduction to Infinite HMMs for Single-Molecule Data Analysis.* Biophys J, 2017. 112(10): p. 2021-2029.

144. Choi, H., et al., *Sparsely correlated hidden Markov models with application to genome-wide location studies.* Bioinformatics, 2013. 29(5): p. 533-41.

145. Bian, J. and X. Zhou, *Hidden Markov Models in Bioinformatics: SNV Inference from Next Generation Sequence.* Methods Mol Biol, 2017. 1552: p. 123-133.
146. 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.
147. 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.
148. Abante, J., et al., *HiMMe: using genetic patterns as a proxy for genome assembly reliability assessment.* BMC Genomics, 2017. 18(1): p. 694.
149. Ernst, J. and M. Kellis, *Chromatin-state discovery and genome annotation with ChromHMM.* Nat Protoc, 2017. 12(12): p. 2478-2492.
150. 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.
151. 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.
152. Nguyen, N.P., et al., *HIPPI: highly accurate protein family classification with ensembles of HMMs.* BMC Genomics, 2016. 17(Suppl 10): p. 765.
153. Lampros, C., et al., *HMMs in Protein Fold Classification.* Methods Mol Biol, 2017. 1552: p. 13-27.
154. Jablonowski, K., *Hidden Markov Models for Protein Domain Homology Identification and Analysis.* Methods Mol Biol, 2017. 1555: p. 47-58.
155. Huo, L., et al., *pHMM-tree: phylogeny of profile hidden Markov models.* Bioinformatics, 2017. 33(7): p. 1093-1095.
156. Sharma, R., et al., *Predicting MoRFs in protein sequences using HMM profiles.* BMC Bioinformatics, 2016. 17(Suppl 19): p. 504.
157. 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.
158. Finn, R.D., et al., *HMMER web server:* 2015 *update.* Nucleic Acids Res, 2015. 43(W1): p. W30-8.
159. 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.

160. 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.

161. 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.

162. Dutheil, J.Y., *Hidden Markov Models in Population Genomics.* Methods Mol Biol, 2017. 1552: p. 149-164.

163. 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.

164. Vogl, C. and A. Futschik, *Hidden Markov models in biology.* Methods Mol Biol, 2010. 609: p. 241-53.

165. Vijayabaskar, M.S., *Introduction to Hidden Markov Models and Its Applications in Biology*. Methods Mol Biol, 2017. 1552: p. 1-12.

166. 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.

167. 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.

168. 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.

169. Zhu, R., et al., *A Robust Manifold Graph Regularized Nonnegative Matrix Factorization Algorithm for Cancer Gene Clustering.* Molecules, 2017. 22(12).

170. Ma, Y., et al., *Hessian regularization based symmetric nonnegative matrix factorization for clustering gene expression and microbiome data.* Methods, 2016. 111: p. 80-84.

171. 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.

172. Menon, V., *Clustering single cells: a review of approaches on high-and low-depth single-cell RNA-seq data.* Brief Funct Genomics, 2018.

173. Menon, V., *Clustering single cells: a review of approaches on high-and low-depth single-cell RNA-seq data.* Brief Funct Genomics, 2017.

174. 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.

175. Zurauskiene, J. and C. Yau, *pcaReduce: hierarchical clustering of single cell transcriptional profiles.* BMC Bioinformatics, 2016. 17: p. 140.

176. 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.

177. Newell, E.W. and Y. Cheng, *Mass cytometry: blessed with the curse of dimensionality.* Nat Immunol, 2016. 17(8): p. 890-5.

178. Irish, J.M. and D.B. Doxie, *High-dimensional single-cell cancer biology.* Curr Top Microbiol Immunol, 2014. 377: p. 1-21.

179. 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.

180. Zhu, X., et al., *Detecting heterogeneity in single-cell RNA-Seq data by non-negative matrix factorization.* PeerJ, 2017. 5: p. e2888.

181. Mukherjee, S., et al., *Scalable preprocessing for sparse scRNA-seq data exploiting prior knowledge.* Bioinformatics, 2018. 34(13): p. i124-i132.

182. Irizarry, R.A., et al., *Exploration, normalization, and summaries of high density oligonucleotide array probe level data.* Biostatistics, 2003. 4(2): p. 249-64.

183. 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.

184. Irizarry, R.A., et al., *Summaries of Affymetrix GeneChip probe level data.* Nucleic Acids Res, 2003. 31(4): p. e15.

185. Ritchie, M.E., et al., *limma powers differential expression analyses for RNA-sequencing and microarray studies.* Nucleic Acids Res, 2015. 43(7): p. e47.

186. 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.

187. Trapnell, C., L. Pachter, and S.L. Salzberg, *TopHat: discovering splice junctions with RNA-Seq.* Bioinformatics, 2009. 25(9): p. 1105-11.

188. 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.

189. Asara, J.M., et al., *Protein sequences from mastodon and Tyrannosaurus rex revealed by mass spectrometry.* Science, 2007. 316(5822): p. 280-5.

190. 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.

191. 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.

192. Cho, W.C., *Mass spectrometry-based proteomics in cancer research.* Expert Rev Proteomics, 2017. 14(9): p. 725-727.

193. 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.

194. 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.

195. 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.

196. 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.

197. 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.

198. 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.

199. Artigues, A., et al., *Protein Structural Analysis via Mass Spectrometry-Based Proteomics.* Adv Exp Med Biol, 2016. 919: p. 397-431.

200. 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.

201. Perkins, D.N., et al., *Probability-based protein identification by searching sequence databases using mass spectrometry data.* Electrophoresis, 1999. 20(18): p. 3551-3567.

202. 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.

203. 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.

204. 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.

205. Cox, J., et al., *Andromeda: a peptide search engine integrated into the MaxQuant environment.* J Proteome Res, 2011. 10(4): p. 1794-805.

206. 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.

207. 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.

208. Kong, A.T., et al., *MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics.* Nat Methods, 2017. 14(5): p. 513-520.

209. 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.

210. 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.

211. Chen, T., et al., *Web resources for mass spectrometry-based proteomics.* Genomics Proteomics Bioinformatics, 2015. 13(1): p. 36-9.

212. Aebersold, R. and M. Mann, *Mass spectrometry-based proteomics.* Nature, 2003. 422(6928): p. 198-207.

213. Domon, B. and R. Aebersold, *Mass spectrometry and protein analysis.* Science, 2006. 312(5771): p. 212-7.

214. 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.

215. Li, S. and H. Tang, *Computational Methods in Mass Spectrometry-Based Proteomics.* Adv Exp Med Biol, 2016. 939: p. 63-89.

216. Valikangas, T., T. Suomi, and L.L. Elo, *A comprehensive evaluation of popular proteomics software workflows for label-free proteome quantification and imputation.* Brief Bioinform, 2017.

ดัชนี