ใบรับรองวิทยานิพนธ์

บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง ระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนมใหม่แบบอัตโนมัติ
โดย นายพัทธพล เปยานนท์

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

_____ คณบดีบัณฑิตวิทยาลัย
(อาจารย์ ดร.มงคล หวังสถิตย์วงษ์)

21 พฤษภาคม 2550

คณะกรรมการสอบวิทยานิพนธ์

_____ ประธานกรรมการ
(รองศาสตราจารย์ ดร.วรา วราวิทย์)

_____ กรรมการ
(ดร.ศิษเฎศ ทองสิมา)

_____ กรรมการ
(ดร.สิทธิรักษ์ รอยตระกูล)

_____ กรรมการ
(รองศาสตราจารย์ ดร.ณชล ไชยรัตนะ)

ระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนมใหม่แบบอัตโนมัติ

นายพัทธพล เปยานนท์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2549

ชื่อ : นายพัทธพล เปยานนท์
ชื่อวิทยานิพนธ์ : ระเบียบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนมใหม่แบบอัตโนมัติ
สาขาวิชา : วิศวกรรมไฟฟ้า
สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ที่ปรึกษาวิทยานิพนธ์ : รองศาสตราจารย์ ดร.วรา วราวิทย์
ดร.ศิษเฎศ ทองสิมา
ปีการศึกษา : 2549

บทคัดย่อ

จากความก้าวหน้าของเทคโนโลยี ด้านระบบชีวสารสนเทศตลอดจนเทคนิคการหาลำดับเบสใหม่ๆ ทำให้เราวิเคราะห์หน้าที่การทำงานของยีนในจีโนม (Genome) โปรคารีโอตและยูคารีโอตได้เป็นจำนวนมาก และด้วยเหตุนี้ทำให้การระบุตำแหน่งและกำหนดหน้าที่ของยีนด้วยวิธีเดิมเป็นไปได้ยาก เนื่องจากจำนวนของลำดับเบสมีมากเกินไป วิทยานิพนธ์นี้นำเสนอระบบสารสนเทศในการคำนวณเพื่อรองรับการทำนายยีน การระบุตำแหน่งและกำหนดหน้าที่ยีนแบบอัตโนมัติซึ่งถูกตรวจสอบความถูกต้องของผลลัพธ์โดยนักชีววิทยาและวางกรอบหลักให้สามารถปรับได้ตามความเหมาะสม ประกอบด้วย 1) การเชื่อมต่อลำดับเบส (Fragment Assembly), 2) การตีความกรอบเปิดการอ่าน (Open Reading Frame Identification), 3) การระบุตำแหน่งและกำหนดหน้าที่ของกรอบเปิดการอ่านโดยหาความเหมือนในฐานข้อมูลต่างๆ (ORF Similarity Matching), ขั้นสุดท้ายคือ 4) การตรวจสอบความถูกต้องโดยนักชีววิทยา (Human Verification) ทั้ง 4 ขั้นตอนจะถูกส่งผ่านข้อมูลจากขั้นตอนหนึ่งสู่อีกขั้นตอนหนึ่งในฐานข้อมูลแล้วทำงานร่วมกันตามลำดับผ่าน TurboGears ข้อมูลจะถูกเก็บอยู่ในรูปแบบเอกสาร XML ทำให้แต่ละขั้นตอนทำงานร่วมกันโดยการส่งผ่านข้อมูลได้อย่างถูกต้อง

(วิทยานิพนธ์มีจำนวนทั้งสิ้น 64 หน้า)

คำสำคัญ : ชีวสารสนเทศ, จีโนม, การระบุตำแหน่งและกำหนดหน้าที่, การลำดับเบส

วรา วราวิทย์

อาจารย์ที่ปรึกษาวิทยานิพนธ์

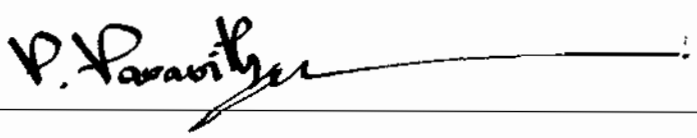
Name : Mr. Pattapol Payanon
Thesis Title : An Automatic Annotation Workflow System for *de novo* Genome Sequencing
Major Field : Electrical Engineering
King Mongkut's Institute of Technology North Bangkok
Thesis Advisors : Associate Professor Dr.Vara Varavithya
Dr.Sissades Tongsima
Academic Year : 2006

Abstract

In this work, we present an automated fragment assembly (sequence assembly) and gene predicting workflow for both prokaryotes and eukaryotes. The database created by software further annotated and validated by biologist. This workflow is composed of four major stages: 1) fragments assembly, 2) open reading frame (ORF) identification, 3) ORF similarity matching and finally 4) human verification. All stations work closely together by passing data from one to the next station sequentially. This framework records intermediate data from each stations in a database for storing genome of generic organisms. The XML standard is formulated as communication message between stations to ensure the compatibility of output from one station as input to the next station. In view of bioinformatics developers, XML reduces the number of specific parsers required to convert output from each program in such stations.

(Total 64 pages)

Keywords : Bioinformatics, Genome, Annotation, Assembly



Advisor

กิตติกรรมประกาศ

การดำเนินการจัดทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงอย่างสมบูรณ์ด้วยความอนุเคราะห์อย่างดียิ่งจาก รองศาสตราจารย์ ดร.วรา วราวิทย์ และ ดร.ศิษฎศ ทองสิมา ที่ได้กรุณาให้คำแนะนำในการแก้ไขปัญหาต่างๆ ตลอดจนให้ความสะดวกในด้านตำราเอกสารที่ใช้ประกอบการจัดทำวิทยานิพนธ์ ผู้จัดทำจึงขอกราบของพระคุณเป็นอย่างยิ่งไว้ ณ ที่นี้

ขอขอบคุณเพื่อนร่วมงานพระจอม คุณสุภาดา เหล่าสุขสถิตย์ และคุณกมล จุติวัฒนนุกูลสำหรับโครงร่าง LaTeX คุณอิศเรศ สมณะ ที่เอื้อเฟื้อช่วยเหลือระบบเครือข่าย และเพื่อนร่วมงานที่ BIOTEC ที่อำนวยความสะดวกในด้านอุปกรณ์สำหรับการทำงาน และขอขอบคุณอาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้าและเจ้าหน้าที่ของภาควิชาทุกท่าน ที่อำนวยความสะดวกในด้านต่างๆ จนทำให้วิทยานิพนธ์ ฉบับนี้สำเร็จลงได้ด้วยดี

พัทธพล เปยานนท์

สารบัญ

สารบัญตาราง

สารบัญภาพ

บทที่ 1
บทนำ

จากความก้าวหน้าของเทคโนโลยี ด้านระบบชีวสารสนเทศตลอดจนเทคนิคการลำดับเบส ทำให้วิเคราะห์หน้าที่การทำงานของยีนในจีโนม (Genome) โปรคาริโอตและยูคาริโอตได้เป็นจำนวนมาก [1] และด้วยเหตุนี้เองที่ทำให้การระบุตำแหน่งและกำหนดหน้าที่ของยีนด้วยวิธีเดิมเป็นไปได้ยาก เนื่องจากจำนวนของลำดับเบสมีมากเกินไป วิทยานิพนธ์นี้นำเสนอระบบการทำนายยีน การระบุตำแหน่งและกำหนดหน้าที่ของยีนแบบอัตโนมัติซึ่งผลลัพธ์ถูกตรวจสอบโดยนักชีววิทยากับวางกรอบหลักให้สามารถปรับได้ตามความเหมาะสม ประกอบด้วย 1) การเชื่อมต่อลำดับเบส , 2) การตีความกรอบเปิดการอ่าน, 3) การระบุตำแหน่งกรอบเปิดการอ่านโดยหาความเหมือนในฐานข้อมูลต่าง, สุดท้ายคือ 4) การตรวจสอบความถูกต้องโดยนักชีววิทยา ทั้ง 4 ขั้นตอนจะถูกส่งผ่านข้อมูลจากขั้นตอนหนึ่งสู่อีกขั้นตอนหนึ่งในฐานข้อมูลแล้วทำงานร่วมกันตามลำดับ ข้อมูลจะถูกเก็บอยู่ในรูปแบบเอกสาร XML ทำให้แต่ละขั้นตอนทำงานร่วมกันโดยการส่งผ่านข้อมูลได้อย่างถูกต้อง ในแง่ของนักพัฒนาชีวสารสนเทศแล้ว XML ยังช่วยลดตัวตีความ (Parsers) วิทยานิพนธ์นี้ยังช่วยลดเวลาการทำงาน การเขียนโปรแกรมทำให้นักชีววิทยาทำงานในส่วนของตนเองได้โดยมีการรองรับอย่างเป็นระบบ

ในสมัยก่อนการศึกษาการสืบทอดทางพันธุกรรมสามารถทำได้เพียงเฝ้าสังเกตุเท่านั้น จนแนวคิดของ Mendel พิสูจน์ว่าสามารถนำการทดลองทางวิทยาศาสตร์มาศึกษา การสืบทอดทางพันธุกรรมได้ Genetics หรือพันธุศาสตร์คือชื่อสำหรับการศึกษาเรื่องการสืบทอดทางพันธุกรรม การสืบทอดทางพันธุกรรม คือกระบวนการที่สืบทอดลักษณะจากบรรพบุรุษ สู่ลูกหลานซึ่งเกิดในทุกๆ สิ่งมีชีวิตรวมถึงมนุษยชาติ พวกเราคือชิ้นส่วนจิ๊กซอว์ ที่ประกอบขึ้นมาจากบรรพบุรุษของนี่เอง หัวใจหลักของการศึกษาพันธุศาสตร์ ก็คือการศึกษาสิ่งที่ควบคุมการสืบทอดทางพันธุกรรม ซึ่งประกอบจากปัจจัยจำนวนมาก ที่เรียกว่า Gene หรือยีน (โดยยีนเป็นสิ่งที่มีอิทธิพล สำหรับใช้ควบคุมการสืบทอดลักษณะทางพันธุกรรม)

ในช่วงก่อนทศวรรษที่ 1930s นั้นนักพันธุศาสตร์มุ่งความสนใจที่ ยีนทำอย่างไรในการสืบทอดจากบรรพบุรุษมาสู่ลูกหลานซึ่งเกิดในช่วงของการสืบพันธุ์ และยีนทำได้อย่างไรในการควบคุมตัวแปรลักษณะการแสดงออกอย่างเช่น ความสูง สีของผม เหล่านี้เป็นต้น ซึ่งช่วงการเปลี่ยนแปลงความสนใจอยู่ในช่วงทศวรรษที่ 1930s เมื่อได้ตระหนักว่าแท้ที่จริงแล้วยีนควบคุมลักษณะทางกายภาพ ด้วยแนวความคิดนี้นี่เองที่นำไปสู่การศึกษายีนในระดับของโมเลกุล จนเกิดสาขาใหม่ซึ่งมีชื่อเรียกกันว่า ชีววิทยาโมเลกุลหรือ Molecular Biology โดยมีจุดประสงค์อย่างแรกนั่นก็คือทำการศึกษายีน เพื่อระบุถึงหน้าที่ธรรมชาติทางเคมีของยีน ซึ่งแนวคิดนี้ก่อให้เกิดแนวคิดใหม่ๆ ตามมาและในไม่ช้านักชีววิทยาก็ถือว่า แต่ละยีนหรือในยีนแต่ละตัวก็คือ หน่วยพื้นฐานของการสืบทอดทางพันธุกรรมหรือ Unit of

Inheritance และเริ่มมองยีนเป็นหน่วยข้อมูลทางชีววิทยาหรือ Unit of Biological Information ในตลอดช่วงระยะเวลา 40 ปี นับตั้งแต่ในช่วงทศวรรษที่ 1930s ทั้งนักพันธุศาสตร์และนักชีววิทยาโมเลกุลซึ่งถือกำเนิดอันเนื่องมาจากแนวคิดใหม่ในการศึกษาเรื่องของยีน พยายามที่จะทำความเข้าใจว่ายีนนั้นทำอย่างไรถึงสามารถที่จะบันทึกข้อมูลทางชีววิทยาได้ หรือยีนทำหน้าที่บันทึกพิมพ์เขียวของข้อมูลที่จำเป็นต่อการสร้างสิ่งมีชีวิตได้อย่างไร และข้อมูลที่บันทึกเหล่านั้นทำให้เซลล์ดำรงชีวิตอยู่ได้อย่างไร

โครงการจีโนมเป็นความพยายามทางวิทยาศาสตร์มีวัตถุประสงค์เพื่อที่จะทราบหน้าที่การทำงานทั้งหมดในจีโนมของสิ่งมีชีวิต อาทิเช่น มนุษย์และสัตว์ชนิดต่างๆ รวมถึงแบคทีเรีย ไวรัส โดยอาศัยดีเอ็นเอในโครโมโซมทำการทดลอง โครงการจีโนมก็จะสร้างแผนที่โครโมโซมในการนำเสนอจีโนมที่ทดลองเสร็จสมบูรณ์ ตัวอย่างโครงการจีโนมที่เป็นที่รู้จักคือโครงการจีโนมมนุษย์ ที่เริ่มในปี 1990 และเสร็จสมบูรณ์ในปี 2003 ทำให้พบว่าในร่างกายมนุษย์มียีนประมาณ 20,000-25,000 ยีน หน้าที่ต่อมาก็คือการศึกษาหาความเกี่ยวข้องของเบสจำนวน 3,000 ล้านในโครโมโซมต่อไป นอกจากนี้ยังมีโครงการจีโนมอื่นๆ อีก เช่นในหนู แมลงหวี่ (*D. melanogaster*) หนอนตัวกลม (*C. elegans*) ยีสต์ (*S. cerevisiae*) แบคทีเรีย (*E. coli*) ข้าว (*Oryza sativa*) และอื่นๆ โครงการจีโนมเหล่านี้มีความสำคัญอย่างมาก การศึกษานี้สามารถนำไปสู่ การตรวจหาโรคทางพันธุกรรม สเตมเซลล์ พัฒนาพันธุ์พืชและสัตว์ พัฒนายาจากโปรตีน สร้างวัคซีนที่กินได้ ยีนบำบัด [2] เป็นต้น

ในวิทยานิพนธ์ฉบับนี้จะทำการศึกษาวิธีการหาข้อมูลทางชีววิทยา, ศึกษาการจัดระเบียบของยีนบนโมเลกุลดีเอ็นเอ, ศึกษาการแสดงออกของยีน, ศึกษาการล่าตับเบส และศึกษาการระบุตำแหน่งของยีนบนโมเลกุลดีเอ็นเอ เพื่อที่จะมีความเข้าใจในการพัฒนาระบบการระบุตำแหน่งยีนแบบอัตโนมัติให้ถือกำเนิดขึ้น [3]

เป็นที่ทราบกันดีว่ายีนเกิดจากดีเอ็นเอ (DNA) และดีเอ็นเอมีโครงสร้างเป็นเกลียวคู่ (Double Helix) เกิดจากสายโพลีนิวคลีโอไทด์ 2 สายเรียงตัวขนานและกลับทิศทางกัน (Antiparallel) มีน้ำตาลหมู่ฟอสเฟตอยู่รอบนอกของโมเลกุลเบสซึ่งมีโครงสร้างเป็นวงแหวนจะจับกันด้วยพันธะไฮโดรเจน A จับคู่ T และ C จับคู่ G ตอนนี้ทราบแล้วว่ายีนคืออะไรและมันสืบทอดข้อมูลทางชีววิทยาได้อย่างไร ต่อไปนี้จะพิจารณาข้อมูลอยู่ในยีน อ่านและใช้โดยเซลล์ได้อย่างไร จริงๆ แล้วยีนคือชิ้นส่วนของโมเลกุลดีเอ็นเอโดยมีขนาดตั้งแต่ 75 เบสหรืออาจมากกว่า 2,300,000 เบส

ข้อมูลทางชีววิทยาที่อยู่ในยีนขึ้นกับ การเรียงลำดับนิวคลีโอไทด์ ข้อมูลเหล่านี้คือสาระสำคัญสำหรับสังเคราะห์ RNA ซึ่งอาจจะถูกแปลรหัสเป็นโมเลกุลโปรตีน กระบวนการสังเคราะห์นี้เกิดขึ้นได้เพราะยีนมีการแสดงออก ซึ่งจะพูดต่อไป

ยีนคือชิ้นส่วนของดีเอ็นเอที่ไม่ต่อเนื่อง แต่ละชิ้นส่วนถูกแยกจากกันโดยบริเวณ Intergenic และยีนถูกจัดเรียงแตกต่างกันออกไปตามแต่ชนิดของสิ่งมีชีวิต ยกตัวอย่างเช่น ในไวรัส ยีนอยู่ติดกันและบริเวณ Intergenic ก็แทบจะไม่มีหรือถ้ามีก็มีไม่มาก ในทางกลับกันในสิ่งมีชีวิตชั้นสูง ยีนอยู่อย่างกระจัดกระจายส่วนมากแล้วบริเวณ Intergenic จะกว้าง ในมนุษย์มียีนอยู่ 30 เปอร์เซ็นต์ของดีเอ็นเอทั้งหมดในเซลล์

นอกจากนี้ ความสามารถของยีนที่ทำหน้าที่ส่งผ่านข้อมูลแทบจะปราศจากข้อจำกัด เนื่องจาก

ถึงแม้ว่ายีนมีความยาวเพียง 150 คู่เบสซึ่งถือว่ายีนนี้ค่อนข้างจะสั้นแต่ในความเป็นจริงแล้วมีความเป็นไปได้ที่จะเป็นลำดับเบสได้ถึง 4^{150} ที่แตกต่างกันออกไป ทำให้ยีนทั้งหมดไม่จำเป็นต้องมีความหมายทางชีววิทยาเพราะว่ามีกฎที่แน่นอนที่ระบุว่ายีนนี้มีความหมาย ถึงกระนั้นก็ดีอัตราส่วนความเป็นไปได้ที่มากถึง 4^{150} ที่จะเก็บข้อมูลทางชีววิทยาแต่ก็ไม่ยากที่จะทราบถึงธรรมชาติ ความไพศาลของความหลากหลายทางชีววิทยา

ในเซลล์ของสิ่งมีชีวิตชั้นสูง เช่น ยีนในมนุษย์ทั้งหมดถูกส่งผ่านโดยโครโมโซม (Chromosomes) ในแต่ละโครโมโซมมีดีเอ็นเอบรรจุยีนเป็นร้อยเป็นพันยีน ในยีนของสิ่งมีชีวิตชั้นต่ำก็เป็นเหมือนกันเพียงแต่มีน้อยกว่า ถึงแม้ว่าในแบคทีเรียจะไม่ซับซ้อนเท่าของมนุษย์และโดยส่วนมากแบคทีเรียมีเพียงหนึ่งโครโมโซม และยีนเป็นพันในหนึ่งโมเลกุลมีลักษณะการจัดเรียงแบบเป็นกลุ่ม, แบบไม่ต่อเนื่อง

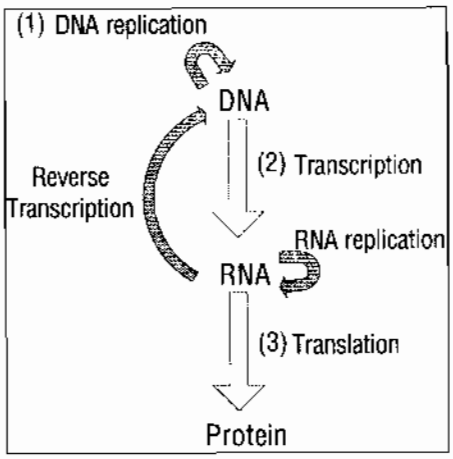
ยีนโดยส่วนมากถูกสุ่มวางตลอดความยาวโมเลกุลดีเอ็นเอห่างกันไม่มากก็น้อย ในบางกรณียีนที่เกี่ยวข้องกันก็ถูกจัดกลุ่ม (Group) เป็นคลัสเตอร์ของยีน (Gene Cluster) และบางยีนในคลัสเตอร์เองก็ไม่เกี่ยวข้องกับยีนอื่นๆ ในคลัสเตอร์ และยังไม่มีเหตุผลที่ชัดเจนรวมถึงข้อดีที่ยีนถูกจัดเรียงอย่างนี้ บ่อยครั้งคลัสเตอร์เกิดจากยีนที่เกี่ยวข้องกัน สองตัวอย่างคือ Operon และ Multigene Family

นักวิจัยพบว่าข้อมูลทางชีววิทยาที่ถูกส่งผ่านโดยยีนประกอบด้วย Exon กับ Intron คือส่วนที่เก็บข้อมูลทางชีววิทยาและส่วนที่ไม่เก็บข้อมูลทางชีววิทยาตามลำดับ ยีนไม่ต่อเนื่องนี้มักพบในสิ่งมีชีวิตชั้นสูงและในไวรัสหลายๆ ชนิดและในปัจจุบันเชื่อว่าในแบคทีเรียยังคงไม่มียีนที่ไม่ต่อเนื่อง ในสิ่งมีชีวิตชั้นสูงบางทียีนอาจจะไม่มี Intron เลยหรือว่ามีเป็นร้อยเป็นพัน โดยส่วนมาก Intron จะยาวมากกว่า Exon และทำไมยีนถึงถูกแบ่งออกเป็น Intron กับ Exon ก็เป็นหัวข้อโต้เถียงกันอย่างรุนแรง มีทฤษฎีว่าเอาไว้ว่า แต่ละ Exon ของยีนที่ไม่ต่อเนื่องมีองค์ประกอบย่อยของข้อมูลทางชีววิทยาต่างๆ ที่ยีนมีหน้าที่ส่งต่อ ถึงแม้ว่าองค์ประกอบย่อย จะมีข้อมูลไม่เพียงพอในการสร้างโปรตีนที่สมบูรณ์ได้ แต่ข้อมูลเหล่านี้ก็สามารถระบุส่วนที่สามารถใช้สำหรับการทำงานของโปรตีน

สิ่งมีชีวิตทุกชนิดมีการถ่ายทอดข้อมูลทางพันธุกรรม (Genetic Information) ที่บ่งบอกลักษณะเฉพาะของสิ่งมีชีวิตนั้นๆ ไปสู่ลูกหลานเพื่อการคงอยู่ของเผ่าพันธุ์ สิ่งมีชีวิตส่วนใหญ่ใช้ดีเอ็นเอ (DNA: Deoxyribonucleic Acid) เป็นสารพันธุกรรม มีเพียงไวรัสบางชนิดใช้อาร์เอ็นเอ (RNA: Ribonucleic Acid) เป็นสารพันธุกรรม ข้อมูลทางพันธุกรรมกำหนดโดยลำดับเบสในโมเลกุลของ DNA หรือ RNA ซึ่งลำดับเบสจะเป็นตัวกำหนดลำดับของกรดอะมิโนในโมเลกุลของโปรตีนต่อไป โดยโปรตีนหลากหลายชนิดที่สิ่งมีชีวิตสังเคราะห์ขึ้นมา ต่างก็มีบทบาทเกี่ยวข้องกับการแสดงลักษณะเฉพาะ และการดำรงชีวิตของสิ่งมีชีวิตชนิดนั้นๆ วิถีการเปลี่ยนถ่ายข้อมูลพันธุกรรมจาก DNA ไปยัง RNA และไปสิ้นสุดที่โปรตีน เรียกว่า "Central Dogma" ผู้นำเสนอชื่อ The Central Dogma [4] เป็นครั้งแรกคือฟรานซิส คริ๊ก (Francis Crick)

เมื่อปี 1985 ในการบรรยายชื่อ "On Protein Synthesis" โดยคริ๊กได้ตั้งสมมุติฐานว่าข้อมูลทางชีววิทยาที่อยู่ในดีเอ็นเอของยีนถูกเปลี่ยนเป็น RNA ก่อนถึงเป็นโปรตีน ซึ่งในตอนนี้เป็นที่ยอมรับเป็นรูปแบบพื้นฐานของการสังเคราะห์โปรตีน คริ๊กยังกล่าวว่าข้อมูลนี้มีทิศทางเดียว (Unidirectional) ไม่สามารถย้อนกลับได้ กล่าวคือโปรตีนไม่สามารถสังเคราะห์ RNA ได้โดยตรงและ RNA ไม่สามารถ

สังเคราะห์ DNA ได้โดยตรง แต่อาจจะพบว่า Central Dogma [4] ไม่จริงในไวรัสบางชนิดที่สามารถย้อน RNA เป็น DNA ได้ Central Dogma ประกอบด้วยกระบวนการที่สำคัญ 3 กระบวนการดังในภาพที่ 1-1 คือ (1) การถ่ายแบบ DNA (DNA Replication) เพื่อสังเคราะห์ DNA ชุดใหม่ (2) การถอดรหัส (Transcription) เพื่อเปลี่ยนถ่ายข้อมูลพันธุกรรมจาก DNA มาอยู่ในรูปของ RNA และ (3) การแปลรหัส (Translation) เพื่อสังเคราะห์โปรตีนโดยแปลลำดับเบสในโมเลกุลของ mRNA เป็นลำดับของกรดอะมิโนในโมเลกุลของโปรตีน

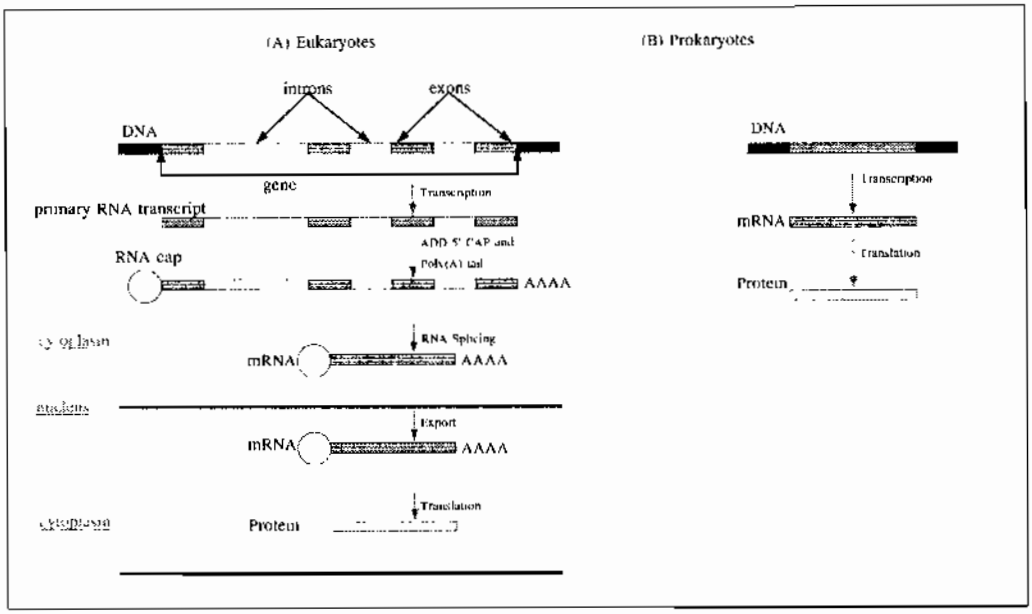


ภาพที่ 1-1 วิถีการเปลี่ยนถ่ายข้อมูลพันธุกรรม หรือ Central Dogma

จากที่ทราบข้างต้นยีน (Gene) คือ หน่วยพื้นฐานของข้อมูลพันธุกรรม ยีนตามความหมายทางด้านชีวเคมี คือ ส่วนของ DNA (หรือ RNA ในบางกรณี) ที่เป็นรหัสข้อมูลพันธุกรรมสำหรับสร้างสารชีวโมเลกุล ซึ่งมีบทบาทและหน้าที่สำคัญในสิ่งมีชีวิต สารชีวโมเลกุลที่เป็นผลิตผลของยีน (Gene Product) ส่วนใหญ่จะหมายถึง โปรตีนชนิดต่างๆ ที่เซลล์สร้างขึ้น นอกจากนี้ผลิตผลของยีนยังรวมถึงโมเลกุลของ RNA บางชนิด เช่น Ribosomal RNA และ Transfer RNA ซึ่งเป็นผลิตผลจากการถอดรหัสของยีน (Transcription) แล้วสามารถทำหน้าที่ในเซลล์ได้โดยตรงในรูปของ RNA โดยไม่จำเป็นต้องมีการแปลรหัส (Translation) เพื่อสร้างเป็นโปรตีน

การแสดงออกของยีน (Gene Expression) หมายถึง การเปลี่ยนถ่ายข้อมูลพันธุกรรมที่เก็บไว้ในรูปของ DNA (หรือ RNA ในสิ่งมีชีวิตบางชนิด) เพื่อสร้างเป็นสารชีวโมเลกุลที่สามารถทำหน้าที่และมีบทบาทต่อการดำรงชีวิตของสิ่งมีชีวิต กระบวนการสำคัญที่เกี่ยวข้องกับการแสดงออกของยีน ประกอบด้วย การถอดรหัส, การตัด-แต่ง RNA (RNA Splicing) และ การแปลรหัส (Translation)

การถอดรหัสเป็นกระบวนการ เปลี่ยนถ่ายข้อมูลพันธุกรรมในรูปของ DNA เกลียวคู่ไปอยู่ในรูปของ RNA สายเดี่ยวซึ่งมีลำดับเบสเป็นสายองค์ประกอบ (Complementary) กับสาย DNA ที่ใช้เป็นแม่แบบโดย RNA สังเคราะห์จากกระบวนการถอดรหัสมีอยู่ 3 ชนิดใหญ่ๆ คือ 1) Messenger RNA (mRNA) เป็น RNA ซึ่งถอดรหัสมาจากยีนสำหรับสังเคราะห์เป็นโปรตีนจำเพาะชนิดต่างๆ โดย RNA ชนิดนี้ทำหน้าที่เป็นตัวกลางในการเปลี่ยนถ่ายข้อมูลพันธุกรรมจากลำดับนิวคลีโอไทด์ใน DNA ไปเป็นลำดับของกรดอะมิโนในโปรตีน 2) Transfer RNA (tRNA) เป็น RNA ที่มีลักษณะเฉพาะ



ภาพที่ 1-2 โครงสร้างยีนของ Eukaryotes และ Prokaryotes

และทำหน้าที่เสมือนเป็นเครื่องมือสำหรับอ่านข้อมูลพันธุกรรมหรือลำดับเบสบนสาย mRNA แล้วนำกรดอะมิโนที่เหมาะสมมาต่อกันเป็นสายโพลีเพปไทด์ 3) Ribosomal RNA (rRNA) เป็น RNA ซึ่งถอดรหัสมาจากยีนสำหรับสังเคราะห์ rRNA โดย RNA ชนิดนี้จะรวมอยู่กับ Ribosomal Protein ประกอบกันเป็น "ไรโบโซม (Ribosome)" ซึ่งมีบทบาทสำคัญในการสังเคราะห์โปรตีนของสิ่งมีชีวิต

เป็นขั้นตอนหลังกระบวนการถอดรหัสหรือ Post - Transcriptional RNA Processing โมเลกุลของ mRNA ที่ได้หลังจากเสร็จสิ้นกระบวนการถอดรหัส เรียกว่า "Primary Transcript" ซึ่งในสิ่งมีชีวิตโปรคาริโอตไม่จำเป็นต้องมีการตัด-แต่งก็สามารถทำหน้าที่เป็นแม่แบบสำหรับการสังเคราะห์โปรตีนได้ทันที โดยพบว่ากระบวนการแปลรหัส หรือการสังเคราะห์โปรตีนของโปรคาริโอตนั้น มักจะเกิดขึ้นก่อนที่การถอดรหัสจะเสร็จสิ้นอย่างสมบูรณ์ แต่ในยูคาริโอต Primary Transcript ที่ได้จะต้องผ่านกรรมวิธีการตัด-แต่งก่อนถึงจะสามารถใช้เป็นแม่แบบสำหรับการสังเคราะห์โปรตีนได้

RNA Splicing เนื่องจาก Primary Transcript ของยูคาริโอต ประกอบด้วยส่วนที่เรียกว่า "Intron" ซึ่งเป็นส่วนที่ไม่ได้บรรจุข้อมูลพันธุกรรมสำหรับแปลรหัสเป็นโปรตีน และแทรกอยู่ระหว่างส่วนที่เรียกว่า "Exon" ซึ่งเป็นส่วนที่บรรจุข้อมูลพันธุกรรมสำหรับแปลรหัสเป็นโปรตีนโดยส่วน Intron นี้ คั่นอยู่ระหว่างส่วน Exon จึงเรียกอีกชื่อหนึ่งว่า "Intervening Sequence" นอกจากนี้ยีนส่วนใหญ่ของยูคาริโอตจะมี Intron อยู่ ยกเว้นยีนของโปรตีนฮิสโทน ปัจจุบันพบว่ายีนบางชนิดในโปรคาริโอตมี Intron อยู่เช่นกันแต่พบได้น้อยมาก ดังนั้นก่อนที่จะมีการแปลรหัส (Translation) จึงต้องมีการตัดเอาส่วน Intron ออกไปก่อน แล้วค่อยนำเอาเฉพาะส่วน Exon มาต่อกันเพื่อใช้เป็นต้นแบบสำหรับการแปลรหัสต่อไป กระบวนการในการตัดเอาส่วน Intron ออก และนำเอาเฉพาะส่วน Exon มาต่อกันนี้เรียกว่า "RNA Splicing" (ภาพที่ 1-2)

RNA Splicing ใช้อาร์เอนเอทั้งสามชนิดมาร่วมกันแปลรหัสพันธุกรรม ซึ่งเรียงลำดับเบสอยู่

บน mRNA มาเป็นโปรตีน การเรียงลำดับกรดอะมิโนบนสาย Polypeptide การสร้างพันธะ Peptide ระหว่างกรดอะมิโนนี้ หากทำในหลอดทดลองแล้ว ก็จะเป็นเพียงปฏิกิริยาเคมีง่ายๆ แต่การสังเคราะห์โปรตีนในเซลล์กลับเป็นกระบวนการที่มีความสลับซับซ้อน ต้องใช้เอนไซม์และโปรตีนหลายชนิดร่วมกันทำงาน ที่เป็นเช่นนี้เพราะเซลล์สร้างพันธะ Peptide ระหว่างกรดอะมิโนที่จำเพาะเจาะจง ภายใต้การกำหนดโดยยีนผ่านทางการเรียงลำดับเบสบน mRNA

mRNA ของโปรคาริโอต 1 สาย มักประกอบด้วย Coding Sequence สำหรับแปลรหัสเป็นพอลิเพปไทด์ ได้หลายชนิด mRNA แบบนี้ เรียกว่า "Polycistronic mRNA" ส่วน mRNA ยูคาริโอตจะประกอบด้วยบริเวณที่เป็นยีนสำหรับการแปลรหัสเป็นพอลิเพปไทด์ได้เพียงชนิดเดียว (Monocistromic mRNA)

การลำดับเบส (Genome Assembly) เมื่อหลายสิบปีก่อนหายีนในจีโนมเพียงตัวเดียวใช้เวลาเป็นเดือน แต่ด้วย "เทคนิค Shotgun" (Shotgun Technique) ทำให้นักวิทยาศาสตร์สามารถทำงานอย่างอภิมหาโครงการนี้เสร็จภายในเวลาแค่ 4 เดือนแค่นั้น วิธีการก็คือแยกชิ้นส่วนดีเอ็นเอหรือยีนจากสิ่งมีชีวิตที่ต้องการศึกษามาใส่ในเซลล์ของสิ่งมีชีวิตอีกชนิดหนึ่งที่ทำหน้าที่เป็นผู้รับให้เป็นชิ้นๆ ด้วยการโคลนดีเอ็นเอเป็นชิ้นเล็กๆ เพื่อเพิ่มจำนวนดีเอ็นเอให้มากพอจนได้ห้องสมุดดีเอ็นเอซึ่งก็คือ DNA Library ยกตัวอย่างเช่น จีโนมของคนมีขนาดประมาณ $3x10^8$ ถ้านำมาตัดด้วยเอมไซม์ซึ่งมีบริเวณจดจำ 6 คู่เบส จะได้ดีเอ็นเอที่มีขนาดประมาณ 4 กิโลเบส ดังนั้นจะได้ชิ้นดีเอ็นเอทั้งหมดประมาณ $7.5x10^5$ ชิ้น ($3x10^8 \div 4$) ดังนั้นต้องมีการถ่ายฝากลงในเซลล์ผู้รับอย่างน้อย $7.5x10^5$ โคลน

วิธีทำห้องสมุดจีโนม (Genomic Library) จึงนิยมใช้ชิ้นดีเอ็นเอขนาดใหญ่ที่มีการเหลื่อมกันเพื่อลดจำนวนโคลน โดยใช้เวคเตอร์ที่สามารถสอดใส่ชิ้นดีเอ็นเอขนาดใหญ่ขึ้นกว่าเดิมเช่น Bacterial Artificial Chromosome (BAC) และ P1 Artificial Chromosome (PAC) เป็นต้น เมื่อโคลนเสร็จแล้วก็นำเข้าเครื่องหารหัสดีเอ็นเออัตโนมัติ ขั้นต่อไปก็ต้องอาศัยคอมพิวเตอร์ในการเปรียบเทียบรหัส จากแต่ละชิ้นเข้าด้วยกัน ว่ารหัสจากโคลนใดควรจะต่อกับโคลนใด คือดูว่าโคลนใดเหลื่อมกัน (Overlap) จนกระทั่งได้รหัสรวมตลอดความยาวของโคลนใหญ่นั้น

ปัญหาในขั้นตอนนี้ก็คือต้องการสมรรถนะภาพการคำนวณสูง นั่นก็เพราะว่าวิธีการก็คือแยกชิ้นส่วนดีเอ็นเอแล้วประกอบขึ้นมาใหม่ เปรียบเสมือนการต่อภาพต่อปริศนา ยิ่งจีโนมในสิ่งมีชีวิตชั้นสูง ยกตัวอย่างเช่นมนุษย์มีมวลจีโนมประมาณ 3000 พันล้านเบสหรือ 3000 พันล้านชิ้นมาต่อกันจึงต้องใช้คอมพิวเตอร์สมรรถนะสูงจำนวนมากมาใช้ในการวิเคราะห์ เพื่อดูว่าข้อมูลชิ้นใดน่าจะต่อกับข้อมูลชิ้นใด นอกจากปัญหานี้แล้ว ปัญหาความยากในการคำนวณก็เป็นอีกสิ่งหนึ่ง นั่นก็เพราะว่าในจีโนมของสิ่งมีชีวิตชั้นสูงมีมวลขนาดใหญ่มาก และมีบริเวณซ้ำ (Repeat) ซึ่งบางที่อาจจะยาวถึง 1000 เบสหรือนิวคลีโอไทด์ และบางครั้งเกิดขึ้นอยู่ทั่วไปบริเวณต่างๆ โดยเฉพาะอย่างยิ่งจีโนมที่มีขนาดใหญ่อย่างเช่นพืชหรือสัตว์ เป็นต้น

(Genome Annotation) ซึ่งคล้ายกับการสร้างแผนที่พันธุกรรม แต่เป็นการใส่ข้อมูลทางชีววิทยาอย่างเช่น Genotype หรือ Phenotype และกระบวนการ Transcription, Translation และ Genetic Regulation โดยข้อมูลเหล่านี้จะแอกข้อมูลพันธุกรรมบนโครโมโซม ซึ่งได้จากระบบการระบุตำแหน่ง

ตารางที่ 1-1 ประวัติความเป็นมาและเครื่องมือที่เกี่ยวข้องกับ Genome Assembly

โปรแกรม	ที่มาของโปรแกรม
AMOS (A Modular, Open-Source assembler)	ให้บริการฟรี S. Salzberg, M. Popand และ A. Delcher จาก สถาบันวิจัย เกี่ยวกับจีโน ม หรือ The Institute for Genomic Research เป็นผู้คิดค้น
Celera Assembler	ให้บริการฟรีเช่นกันสามารถดาวน์โหลดโปรแกรมได้ที่ http://sourceforge.net/projects/wgs-assembler เริ่มพัฒนาโดย G. Myers, G. Sutton, A. Delcher และคนอื่นๆ ที่ Celera Genomics
Arachne	เป็นวิทยานิพนธ์ระดับปริญญาเอกของ S. Batzoglou ตอนนี้อยู่มหาวิทยาลัย Stanford
PCAP [11] หรือ CAP3 [12]	เริ่มพัฒนาในปี 1999 โดย Huang และ Madan
Phrap	พัฒนาโดยห้องทดลอง PHIL GREEN มหาวิทยาลัย Washington

ยีน (Genome Annotation System) ยีนเป็นที่สนใจเพราะยีนคือหน่วยพื้นฐานของการสืบทอดทางพันธุกรรม ระบบการระบุตำแหน่งยีนนี่เองที่ช่วยให้ทราบข้อมูลต่างๆ ที่กล่าวในข้างต้น ในปัจจุบันมีระบบการระบุตำแหน่งยีนเช่น Mage [5], BaSys [6], RiceGAAS [7] และ Ensembl [8] รวมถึงโปรแกรมอย่างเช่น Ergo [9] และ Artemis [10] ทำการระบุตำแหน่งยีนในจีโนมให้ทราบ

เนื่องจากในปัจจุบันยังไม่มีมาตรฐานกลางของระบบการระบุตำแหน่งยีนที่ดีควรจะเป็นอย่างไร ทำให้มีการพัฒนาอย่างหลากหลาย อย่างเช่น ระบบระบุตำแหน่งยีนในจีโนมข้าวของ RiceGAAS [7] ได้พัฒนาเครื่องมือโดยเฉพาะ คือ RiceHMM [13] ทำนายยีนโดเมนบนจีโนมที่ขึ้นกับแบบจำลองเชิงความน่าจะเป็นซึ่งอิงกับฐานข้อมูล ESTs (Expressed Sequence Tags) ของข้าว แล้วนำ HMM (Hidden Markov Model) มาทำนายหายีนที่ได้จากการคำนวณครั้งแรก แต่ไม่ใช่ว่ามีเครื่องมือที่จะระบุตำแหน่งยีนแบบเฉพาะเจาะจงอย่างเช่น RiceGAAS [7] ทำให้โครงการพัฒนาระบบการระบุตำแหน่งยีนแบบอัตโนมัติจึงถือกำเนิดขึ้นด้วยความร่วมมือระหว่าง ศูนย์พันธุวิศวกรรมและเทคโนโลยีชีวภาพแห่งชาติ National Center for Genetic Engineering and Biotechnology: BIOTEC มีวัตถุประสงค์เพื่อที่จะสร้างระบบที่ยืดหยุ่นและมีประสิทธิภาพ โดยวิทยานิพนธ์ฉบับนี้เสนอ MVC (Model-View-Control) ซึ่งสามารถทำงานได้อย่างสอดคล้องกับวัตถุประสงค์ที่ตั้งไว้

กระบวนการศึกษาวิจัย "จีโนม" นี้มี 3 ขั้นตอนหลัก ได้แก่ 1) หาการเรียงลำดับของ "เบส" บนเส้นดีเอ็นเอ เรียกตามภาษาเทคนิคว่า "ดีเอ็นเอ ซีเควนซิง (DNA Sequencing)" ซึ่งใช้เครื่องหาลำดับเบสอัตโนมัติสำหรับงานนี้ เช่น ABI, Pyrosequencing เป็นต้น 2) การประกอบเข้าด้วยกันใหม่ (Assembly) จนกระทั่งได้ลำดับการเรียงตัวของ "เบส" บนเส้นดีเอ็นเอทั้งหมดดังแสดงในตารางที่ 1-1 3) ระบุตำแหน่งของยีนพร้อมทั้งทำหมายเหตุประกอบยีนในจีโนม (Annotation)

ประโยชน์ที่ได้รับจากวิทยานิพนธ์ฉบับนี้ประกอบไปด้วย

1. พัฒนาระบบการระบุตำแหน่งยืนแบบอัตโนมัติเพื่อทำงานแบบภายใน
2. เป็นแม่แบบการศึกษาการประยุกต์ใช้ MVC กับระบบการระบุตำแหน่งยืน
3. พัฒนาฐานข้อมูลเพื่อรองรับการทำงานในระบบขนาดใหญ่
4. ทำการศึกษาเพื่อวางกระบวนการทำงานที่เหมาะสมสำหรับแต่ละสิ่งมีชีวิต
5. ทราบข้อดีข้อเสีย หรือจุดแข็งจุดอ่อนของการใช้เครื่องในแต่ละแบบ
6. นำความรู้ได้ไปพัฒนาและนำเครื่องมือไปใช้แบบบูรณาการ

ในบทต่อไปของวิทยานิพนธ์ฉบับนี้กล่าวถึงการถอดรหัสยีนในจีโนมแบบอัตโนมัติ คุณลักษณะการทำงานของการถอดรหัส รวมถึงการพัฒนาวิธีการถอดรหัสยีนในจีโนมที่ผ่านมา บทที่ 3 อธิบายถึงขั้นตอนวิธีการระบุตำแหน่งและกำหนดหน้าที่ของยีนแบบอัตโนมัติที่เป็นระบบ คุณลักษณะการระบุตำแหน่งและกำหนดหน้าที่ของยีนแบบอัตโนมัติที่เป็นระบบ รวมถึงการพัฒนาวิธีการระบุตำแหน่งและกำหนดหน้าที่ของยีนแบบที่ผ่านมา บทที่ 4 กล่าวถึงระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนแบบอัตโนมัติ การรวมขั้นตอนการทำงานต่างๆ เข้าด้วยกัน สถาปัตยกรรม รวมถึงการพัฒนาระบบการทำหมายเหตุอัตโนมัติที่ผ่านมาโดยละเอียด และแสดงผลลัพธ์ที่ได้จากการทำงานของระบบการทำหมายเหตุอัตโนมัติ ด้วยการถอดรหัสยีนในจีโนม ในบทที่ 2 กับ การระบุตำแหน่งและกำหนดหน้าที่ของยีนแบบอัตโนมัติที่เป็นระบบ ในบทที่ 3 และสรุปผลจากการปรับปรุงขั้นตอนวิธีสำหรับระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนแบบอัตโนมัติ ในบทที่ 5

บทที่ 2
การถอดรหัสยีนในจีโนมแบบอัตโนมัติ

ในบทนี้จะกล่าวถึงเทคนิคการค้นหายีนในจีโนมที่มีอยู่แล้ว อาทิเช่น การค้นหายีนภายนอก, การค้นหายีนภายใน, การค้นหายีนโดยการศึกษาเชิงเปรียบเทียบ ตลอดจนถึงการค้นหายีนแบบผสมที่นำเทคนิคต่างๆ มาทำงานร่วมกัน เป็นที่ทราบว่าหัวข้อวิจัยในสารสนเทศมีหลากหลาย วิทยานิพนธ์นี้เกี่ยวข้องในหลายหัวข้อ อาทิเช่น การจัดตำแหน่งยีน Sequence Alignment, การค้นหายีน Gene Finding, การลำดับเบส Genome Assembly, การศึกษาเชิงเปรียบเทียบ Comparative Genomic มาตรฐานที่ใช้ก็หลากหลาย อาจจะใช้กันบนระบบปฏิบัติการวินโดว์หรือไม่ก็ยูนิกซ์ นักวิทยาศาสตร์อาจจะใช้โปรแกรมในเว็บที่ให้บริการสาธารณะหรือโปรแกรมที่อนุญาตให้ทำงานบนเครื่องส่วนตัว ซึ่งก็สะดวกถ้านักวิทยาศาสตร์ใช้โปรแกรมเดียวสามารถทำงานวิเคราะห์ข้อมูลได้อย่างสมบูรณ์ แต่ในความเป็นจริง งานด้าน ชีววิทยาโมเลกุลตลอดจนเครื่องมือชีวสารสนเทศ ไม่สามารถรองรับความต้องการที่ว่าได้ ทำให้เกิดระบบการถอดรหัสยีนในจีโนมขึ้น ดังที่ได้ยกตัวอย่างแล้ว และวิธีการถอดรหัสก็มีหลากหลาย

แต่ก่อนถ้ากล่าวถึงการค้นหายีน ก็อยู่ในสาขาชีววิทยาเชิงคำนวณที่สนใจระเบียบวิธีคำนวณ หาหน้าที่การทำงานทางชีววิทยาของดีเอ็นเอที่แยกออกมาจากจีโนมในเซลล์ทั้งหมด เช่น ยีนที่เป็นโปรตีน รวมถึงส่วนที่เกี่ยวข้องกับยีนเช่น RNA, Regulatory Regions (Promotor, Enhancer, Acceptor, Donor) เป็นต้น ในตอนแรกการค้นหายีนต้องอาศัยการทดลองในห้องปฏิบัติการเพื่อวิเคราะห์ข้อมูลสร้างแผนที่ทางพันธุกรรม (Genetic Map) แต่ด้วยเทคโนโลยีในปัจจุบัน ปัญหาการค้นหายีนเปลี่ยนจากปัญหาการทดลองเป็นปัญหาทางการคำนวณเป็นหลัก การทดสอบหน้าที่การทำงานของลำดับเบสที่สนใจ ก็ควรแยกออกจากการทดสอบหน้าที่การทำงานของยีนหรือผลิตผลจากยีน นั่นก็เพราะว่าการทดสอบส่วนหลังยังคงต้องนำข้อมูลจากห้องปฏิบัติการมาพิสูจน์

หลังจากได้ลำดับเบสบนโครโมโซมแล้ว ขั้นตอนต่อไปคือการระบุส่วนต่างๆ ในจีโนมซึ่งขั้นตอนนี้ อาศัยโปรแกรมการค้นหายีน (Gene Finding) โดยหน้าที่ของโปรแกรมชนิดนี้ก็คือ พยายามระบุให้ได้ว่ารหัสที่ต้องการหาคืออะไร ซึ่งอาจได้จากการป้อนข้อมูลหรือให้โปรแกรมทำงานแบบอัตโนมัติเพื่อที่จะทำการค้นหายีนว่าอยู่ในบริเวณใด นอกจากนั้นยีนที่โปรแกรมหามาได้มีความถูกต้องมากน้อยเท่าไหร่ ขึ้นกับคะแนนที่คำนวณโดยโปรแกรมหลังจากทำนายโครงสร้างยีนเสร็จแล้ว วิธีการคำนวณก็มีหลายรูปแบบอาจจะเป็น Hidden Markov Model (HMM) เป็นต้น

ปัญหาโดยส่วนมากที่พบในการค้นหายีน จากบทที่แล้วคือเรื่องการจัดเรียงตัวของยีนบนโมเลกุลดีเอ็นเอ โดยเฉพาะยีนจาก Eukaryotes ที่ไม่อยู่ติดกันหรือ Contigous และไม่อยู่ต่อเนื่องกันหรือ Continuous ทำให้การค้นหาสัญญาณหรือ Signal ในดีเอ็นเอมีรูปแบบที่ไม่แน่นอน และยังยากต่อ

การชี้เฉพาะลงไป ในกรณียีน Prokaryotes ปัญหาจะแตกต่างกันโดยยีนจะอยู่ติดกันถ้าไม่ก็จะมีบริเวณคั่นเล็กน้อย และบริเวณอินทรอนก็ไม่มี แต่ยีนจะเหลื่อมกันบ่อย ทำให้ยากแก่การทำนายจุดเริ่มต้นการแปลรหัส (Translation Initiation Sites) และเป็นไปไม่ได้ที่จะชี้เฉพาะสัญญาณที่เกี่ยวข้องกับกรรมวิธีของเซลล์แปลงดีเอ็นเอเป็นโปรตีนได้อย่างชัดเจน ทำให้การชี้เฉพาะสัญญาณการทำงานต่างๆ ในยีนด้วยโมเลกุลในดีเอ็นเอ อย่างเดียวไม่เพียงพอต้องอาศัยวิธีการคำนวณอื่นๆ เข้ามาเกี่ยวข้องด้วยเพื่อเพิ่มความเป็นไปได้ในการทำนายโครงสร้างยีนในดีเอ็นเอ ซึ่งในวิทยานิพนธ์นี้ได้แบ่งหัวข้อวิทยานิพนธ์ที่เกี่ยวข้องกับการค้นหายีนดังนี้

2.1 การค้นหายีนจากภายนอก (Extrinsic Approach)

วิธีการค้นหายีนแบบนี้ใช้การเปรียบเทียบความเหมือนหรือความเหมือนกันระหว่างลำดับเบสที่ต้องการทดสอบกับโมเลกุลดีเอ็นเอในจีโนมที่ทำการระบุตำแหน่ง (Annotated) มาก่อนหน้านี้จนทราบหน้าที่การทำงานต่างๆ ซึ่งจะประกอบด้วย ลำดับเบส, ลำดับกรดอะมิโน และลำดับ mRNA กับข้อมูลในฐานข้อมูลสาธารณะต่างๆ เช่น ศูนย์ข้อมูลเทคโนโลยีชีวภาพแห่งชาติ The National Center for Biotechnology Information (NCBI), ฐานข้อมูลดีเอ็นเอแห่งชาติญี่ปุ่น DDBJ (DNA Data Bank of Japan) โปรแกรมพื้นฐานที่ใช้วิธีเปรียบเทียบการค้นหาแบบนี้มีทั้งแบบ เปรียบเทียบความเหมือนทั้งเส้น (Global Comparison), เปรียบเทียบความเหมือนเฉพาะส่วน (Local Comparison)

2.1.1 Homology and Similarity ความเหมือนกัน (Similarity) ของลำดับของสายดีเอ็นเอหรือโปรตีนสองสาย มีสาเหตุที่เป็นไปได้ 2 ประการคือ

2.1.1.1 ดีเอ็นเอหรือโปรตีนของสิ่งมีชีวิตทั้งสองชนิดมีต้นกำเนิดมาจากแหล่งเดียวกัน จากนั้นเมื่อเวลาผ่านไปดีเอ็นเอหรือโปรตีนแต่ละสายเกิดการเปลี่ยนแปลงในระหว่างที่สิ่งมีชีวิตแต่ละชนิดมีวิวัฒนาการ (Divergent Evolution) ซึ่งการเปลี่ยนแปลงที่เกิดขึ้นในลำดับสายดีเอ็นเอหรือโปรตีนมีไม่มากนัก เมื่อทำการวิเคราะห์จะยังพบว่าดีเอ็นเอหรือโปรตีนทั้งสองสายมีความเหมือนกัน

2.1.1.2 บริเวณของดีเอ็นเอหรือโปรตีนที่เหมือนกัน มีหน้าที่ทางชีววิทยาเหมือนกัน วิวัฒนาการจึงทำให้ลำดับในสิ่งมีชีวิตแต่ละชนิดมีความเหมือนกัน (Convergent Evolution)

ในการพิจารณาว่าลำดับดีเอ็นเอหรือโปรตีน 2 สายเป็น Homologous (มี Homology) หรือไม่ นอกจากความเหมือนกันของลำดับทั้งสองสายแล้ว จะต้องพิจารณาว่าลำดับทั้งสองสายมีต้นกำเนิดมาจากแหล่งเดียวกันหรือไม่ ลำดับที่มีความเหมือนกันอาจไม่ Homologous กัน ลำดับสายสั้นๆ อาจมีความเหมือนกันโดยบังเอิญ ตัวอย่างเช่น ถ้าเบสของสายดีเอ็นเอมีการกระจายตัวอย่างอิสระไม่ขึ้นแก่กัน โอกาสที่จะพบชิ้นดีเอ็นเอขนาด 10 คู่เบสที่มีลำดับเบสที่จำเพาะมีค่าความน่าจะเป็นเท่ากับ 1 ใน 4^{10} หรือประมาณ 10^{-6} ซึ่งนับเป็นค่าที่สูงเพียงพอที่จะพบลำดับเบสดังกล่าวในจีโนมของแบคทีเรียส่วนใหญ่ (ซึ่งมีขนาดประมาณ 10^6 คู่เบสขึ้นไป) ในขณะที่โอกาสที่จะพบชิ้นดีเอ็นเอขนาด 1000 คู่เบสที่มีลำดับเบสที่จำเพาะมีค่าความน่าจะเป็นเท่ากับ 1 ใน 4^{1000} หรือประมาณ 10^{-600} ซึ่งเป็นค่าที่ต่ำมาก ดังนั้นถ้าดีเอ็นเอขนาด 1000 คู่เบสสองสายมีลำดับเบสที่เหมือนกัน หรือเหมือนกัน มีโอกาสสูงมากๆ ที่ดีเอ็นเอสองสายนี้มีต้นกำเนิดมาจากแหล่งเดียวกัน วิธีการที่ใช้ในการหาความเหมือนกัน

ระหว่างลำดับสองสายมีหลายวิธีเช่น วิธี Dot Matrix, Dynamic Programming และ Hidden Markov Model เป็นต้น

2.1.2 Dynamic Programming วิธีการตรวจสอบความเหมือนของลำดับเบสหรือกรดอะมิโนอีกวิธีหนึ่ง คือ นำลำดับที่ต้องการเปรียบเทียบมาเรียงขนานคู่กัน (Align) โดยพยายามจัดให้คล้ายกันมากที่สุดเท่าที่จะเป็นไปได้ ในการนี้บางครั้งอาจต้องยอมให้ลำดับเบสที่ไม่ตรงกันอยู่คู่กันบ้าง หรือยอมให้เกิดช่องว่างขึ้นบ้าง เพื่อให้ส่วนที่คล้ายกันได้อยู่คู่กันมากที่สุด เมื่อเปรียบเทียบลำดับเบส 2 สาย ผลที่เป็นไปได้ในแต่ละตำแหน่งคือ 1) Match คือมีเบสหรือกรดอะมิโนเหมือนกัน 2) เบสหรือกรดอะมิโนที่ตำแหน่งนั้นต่างกัน (Mismatch) และ 3) เกิดช่องว่าง (Gap) ในลำดับสายหนึ่ง ณ ตำแหน่งนั้น การเปรียบเทียบลำดับ 2 สายจะมี Alignment รูปแบบที่เป็นไปได้เป็นจำนวนมาก ต้องทำการเปรียบเทียบใน Alignment ทุกรูปแบบที่เป็นไปได้ แล้วคัดเลือก Alignment รูปแบบที่ดีที่สุด โดยทั่วไปจะมีปัญหาคือ จะตัดสินอย่างไรว่ารูปแบบการเรียงแบบใดที่มีลักษณะใกล้เคียงกันมากที่สุด การจะตัดสินใจได้ก็จะต้องมีการให้คะแนนผลการเรียงทั้งสามชนิดที่กล่าวข้างต้น แล้วรวมเข้าด้วยกันตลอดทั้งเส้นหรือบริเวณที่สนใจ แล้วจึงใช้ผลรวมที่ได้ช่วยในการตัดสินใจ เช่น อาจให้คะแนนสำหรับเบสที่ Match มีค่าเป็นบวก คะแนนสำหรับเบสที่มี Mismatched มีค่าเป็นศูนย์หรือลบ และคะแนนค่าติดลบมากขึ้นสำหรับ Gap (Gap Penalty) ดังนั้น Alignment แต่ละรูปแบบจะมีคะแนนความคล้าย (Similarity Score) ที่แตกต่างกัน Alignment รูปแบบที่มีคะแนนสูงสุดจะถือว่าเป็น Alignment ที่ดีที่สุด อย่างไรก็ตาม การคัดเลือก Alignment ที่ดีที่สุด ด้วยการหาคะแนนของการ Alignment ไปเรื่อยๆ ทีละคู่นั้น ทำได้ในกรณีที่สายดีเอ็นเอมีขนาดค่อนข้างสั้นเท่านั้น ตัวอย่างเช่น หากต้องการเปรียบเทียบดีเอ็นเอที่มีขนาด 1000 คู่เบส รูปแบบ Alignment ที่เป็นไปได้มีประมาณ 10^{600} แบบ ซึ่งมากกว่าจำนวนอะตอมทั้งหมดในเอกภพเสียอีก การคิดคำนวณคะแนนของรูปแบบของ Alignment ที่เป็นไปได้ทั้งหมดโดยการไล่ทำไปเรื่อยๆ จึงใช้เวลานานมากเกินไป ดังนั้นจึงจำเป็นจะต้องใช้เครื่องมือและวิธีการทางคอมพิวเตอร์ช่วย Dynamic Programming Algorithm ซึ่งถูก Needleman กับ Wunsch นำมาใช้ เป็นวิธีการทางคอมพิวเตอร์ที่สามารถช่วยหารูปแบบ Alignment ที่ดีที่สุด โดยไม่ต้องคิดคำนวณ Similarity Score ของ Alignment ทุกแบบ โดยที่สามารถพิสูจน์ได้ว่า วิธีนี้ให้ผล Alignment ที่ดีที่สุด ถูกต้องเสมอ

2.1.3 การเปรียบเทียบสายลำดับสองเส้นของ Smith และ Waterman ด้วยระเบียบวิธีแบบ Dynamic Programming ซึ่งถูกคิดค้นโดย Needleman และ Wunsch จะใช้สำหรับ Global Alignment ซึ่งเป็นการเปรียบเทียบลำดับสองสายตลอดทั้งสาย แต่เนื่องจากดีเอ็นเอและโปรตีนอาจมี Homology เพียงส่วนใดส่วนหนึ่งภายในโมเลกุล ดังนั้นในเวลาต่อมา Smith และ Waterman ได้คิดค้น Local Alignment Algorithm ซึ่งเป็นวิธีการเปรียบเทียบลำดับเบสของดีเอ็นเอ 2 สายหรือลำดับกรดอะมิโนของโพลีเปปไทด์ 2 สาย โดยหาความเหมือนในส่วนใดส่วนหนึ่งในสายชีวโมเลกุลดังกล่าว โปรแกรมที่ทำ Dynamic Programming เป็นโปรแกรมที่ใช้การคำนวณสูงมาก ในกรณีที่ต้องการเปรียบเทียบสายลำดับที่สนใจกับสายลำดับอื่นทั้งหมดที่มีอยู่ในฐานข้อมูล คอมพิวเตอร์ต้องใช้เวลานานมากกว่าจะสามารถเปรียบเทียบได้หมด ซึ่งทำให้ไม่ทันการ ในการเปรียบเทียบสายลำดับที่สนใจกับสายลำดับอื่น

ตารางที่ 2-1 โปรแกรม BLAST โดย P, N และ T คือ Protein, Nucleotide และ Translated

โปรแกรม	คำอธิบายและวิธีการนำไปใช้	การเปรียบเทียบ
BLASTN	ใช้เพื่อเปรียบเทียบลำดับเบสบนดีเอ็นเอที่ต้องการศึกษา (Nucleotide Query Sequence) กับลำดับเบสของดีเอ็นเอที่มีอยู่ในฐานข้อมูล	Nucleotide กับ Nucleotide Database
BLASTP	ใช้เพื่อเปรียบเทียบ (Protein Query Sequence) กับลำดับเกรดอะมิโนของโปรตีน(หรือโพลีเปปไทด์)ที่มีอยู่ในฐานข้อมูล	Protein กับ Protein Database
BLASTX	ใช้เพื่อหาโปรตีนที่น่าจะเป็นไปได้จากการแปลงรหัส (Translation) ดีเอ็นเอที่ทำการศึกษา โดยโปรแกรมจะทำการแปลงรหัสจากลำดับเบสบนดีเอ็นเอที่ต้องการศึกษา ซึ่งจะได้ทั้งหมด 6 Reading Frames (จากสายดีเอ็นเอเอง 3 Reading Frames และจาก Complementary Strand อีก 3 Reading Frames) แล้วจึงนำข้อมูลลำดับเกรดอะมิโนทั้ง 6 Frames นี้ไปเปรียบเทียบกับโปรตีนที่มีอยู่ในฐานข้อมูล	Translated กับ Protein Database
TBLASTN	ใช้เพื่อเปรียบเทียบลำดับกรดอะมิโนบนโปรตีนที่ต้องการศึกษากับลำดับกรดอะมิโนทั้ง 6 Reading Frames ที่ได้จากการแปลงรหัสดีเอ็นเอที่มีอยู่ในฐานข้อมูล	Protein กับ Translated Database
TBLASTX	ในกรณีนี้ทั้งดีเอ็นเอที่ต้องการศึกษาและดีเอ็นเอที่อยู่ในฐานข้อมูลจะถูกแปลงรหัสเป็นโปรตีนก่อน ทำให้ได้ข้อมูลของลำดับกรดอะมิโนจากดีเอ็นเอที่ต้องการศึกษา 6 Reading Frames และข้อมูลกรดอะมิโนจากดีเอ็นเอที่มีอยู่ในฐานข้อมูลอีก 6 Reading Frames แล้วจึงนำมาเปรียบเทียบกัน	Translated กับ Translated Database

ทั้งหมดที่มีอยู่ในฐานข้อมูล จึงได้มีผู้ที่คิดค้นและพัฒนาโปรแกรมคอมพิวเตอร์เพื่อใช้ในการเปรียบเทียบหาความคล้ายกันของสายลำดับให้เร็วขึ้น โดยยอมให้มีความผิดพลาดเกิดขึ้นได้บ้าง (ต่างจากโปรแกรม Dynamic Programming ที่ให้ผลถูกต้องแน่นอน)

BLAST เป็นโปรแกรมยอดนิยมใช้หาความเหมือน (Similarity) ของลำดับเบสบนดีเอ็นเอหรือลำดับเกรดอะมิโนบนโปรตีนที่สนใจศึกษากับลำดับต่างๆ ที่อยู่ในฐานข้อมูล BLAST เป็นกลุ่มของโปรแกรมที่ทำงานและแสดงผลได้ค่อนข้างรวดเร็ว ประกอบด้วย 5 โปรแกรมดังตารางที่ 2-1 สามารถทำได้หลายทางเช่น ไปยังเว็บไซต์ของ NCBI ก็เสามารถโหลดโปรแกรม BLAST มาใช้

2.2 การค้นหายีนจากภายใน (Intrinsic Approach)

วิธีนี้ต่างจากวิธีแรกที่ไม่ได้นำไปเปรียบเทียบกับดีเอ็นเอหรือโครโมโซม แต่วิธีอย่าง *Ab Initio* หรือ Intrinsic ใช้สำหรับค้นหาสัญญาณหรือลักษณะใดๆ ก็ตามที่เกี่ยวข้องกับการสร้างโปรตีน สำหรับโปรคารีโอต (Prokaryotes) และยูคาริโอต (Eukaryotes) ทั้งนี้ขึ้นกับความแตกต่างของสิ่งมีชีวิตทั้งสอง

โครโมโซมของโปรคาริโอตมีขนาดเล็กประมาณ (5-10,000,000) คู่เบสและบริเวณที่เป็นยีนก็มีประมาณ (90%) นอกจากนั้นในโปรคาริโอตก็ไม่มีอินทรอน (Intron) สัญญาณที่บ่งบอกต่างๆ ก็ง่ายต่อการระบุ เช่น กรดอะมิโนเริ่มต้นคือ (ATG) ส่วนกรดอะมิโนสุดท้ายคือ (TAG/TGA/TAA) ซึ่งก็แตกต่างกันไปตามสายพันธุ์ของโปรคาริโอต ถึงกระนั้นก็ตามยีนในโปรคาริโอตส่วนมากมักจะเหลื่อมกันทำให้ยากต่อการระบุจุดเริ่มต้น

อย่างไรก็ตามสำหรับยูคาริโอตวิธี *Ab Initio* จะซับซ้อนกว่าเพราะว่ายีนถูกแบ่งบริเวณกว้างที่เรียกว่าอินเตอร์เจนนิค (Intergenic) ยีนยังอยู่อย่างกระจัดกระจายไม่อยู่ต่อๆ กัน ยีนยังแบ่งออกเป็นอินทรอน (Intron) กับเอ็กซ์ซอน (Exon) โดยบริเวณอินทรอนจะถูกกำจัดออกไปในขั้นตอน Splicing ของกระบวนการสังเคราะห์โปรตีน การกำจัดบริเวณอินทรอนนี่เองที่ทำให้ยากมากกว่าโปรคาริโอตเพราะความซับซ้อนและยากในการระบุชี้ชัด ยกตัวอย่างเช่น Cpg Islands กับ Poly-A

จากข้างต้นสามารถหายีนได้ด้วยวิธีคร่าวๆ 2 วิธีได้แก่วิธีหาบริเวณที่เป็นยีนได้ (วิธีนี้สามารถหาบริเวณที่เป็นยีนไม่ได้เช่นกัน) กับหาบริเวณที่ยีนมีการแสดงออก

2.2.1 วิธีหาบริเวณที่เป็นยีน (Search by Content)

2.2.1.1 Base Composition Bias ดีเอ็นเอบริเวณที่ไม่มีรหัสสำหรับการสร้างโปรตีน (Non-Coding Region) นั้น มีผู้คาดว่าลำดับเบสแต่ละชนิดที่เป็นองค์ประกอบของดีเอ็นเอบริเวณนี้มีการกระจายตัวอย่างอิสระ แต่สำหรับดีเอ็นเอบริเวณที่มีรหัสสำหรับการสร้างโปรตีน (Coding Region) เบส 2 ตำแหน่งแรกในแต่ละ Codon จะถูกกำหนดโดยชนิดของกรดอะมิโนที่ Codon นั้นมีรหัสอยู่ โดยทั่วไปแล้ว สัดส่วนของเบสกวานีนและไซโตซีนในตำแหน่งที่ 1 และ 2 ของ Codon ทั้งหมดใน Coding Region มักจะเป็นประมาณ 50% ส่วนชนิดของเบสในตำแหน่งที่ 3 จะแปรเปลี่ยนได้ตามการเลือกใช้ Codon ของสิ่งมีชีวิตแต่ละชนิด เนื่องจาก Codon สำหรับกรดอะมิโนส่วนมากจะมี Degeneracy ตัวอย่างเช่น ถ้าสิ่งมีชีวิตหนึ่งมีสัดส่วนเบสกวานีนและไซโตซีนรวมประมาณ 70% เช่นในกรณีของแบคทีเรียกลุ่มแอคติโนไมซีส (Actinomycetes) เบสในตำแหน่งที่ 3 ของ Codon น่าจะเป็นกวานีนและไซโตซีนเกือบทั้งหมด ลักษณะเช่นนี้เป็นสิ่งหนึ่งซึ่งบ่งชี้ได้ว่าดีเอ็นเอในบริเวณดังกล่าวมีรหัสสำหรับการสร้างโปรตีน

โปรแกรมทางคอมพิวเตอร์สำหรับการวิเคราะห์หา Coding Region โดยวิธีนี้คำนวณหาปริมาณเบสที่เป็นองค์ประกอบเป็นส่วนๆ (Window) ซึ่งมีความยาวตามที่กำหนด แล้วจะเลื่อนไปคำนวณหาปริมาณเบสในส่วนถัดไป โปรแกรมลักษณะดังกล่าวสามารถทำการวิเคราะห์ได้โดยไม่ต้องการข้อมูลอื่น แต่โปรแกรมเหล่านี้ไม่สามารถบอกได้ว่า Coding Region เริ่มต้นและสิ้นสุดที่ตำแหน่งใด และไม่สามารถบอก Reading Frame ได้ นอกจากนี้ความยาวที่ถูกวิเคราะห์ในแต่ละส่วนมีขนาดค่อนข้างใหญ่ประมาณ 100 - 200 คู่เบส ซึ่งไม่เหมาะสำหรับการวิเคราะห์หา Coding Region ในบริเวณที่เป็น Exon ช่วงสั้นๆ ตัวอย่างของโปรแกรมที่ทำการวิเคราะห์โดยวิธีนี้ได้แก่ โปรแกรม Testcode ซึ่งอยู่ในซอฟต์แวร์ GCG package

2.2.1.2 Codon Usage Patterns Codon ที่มีรหัสสำหรับกรดอะมิโนชนิดหนึ่งๆ มักมีมากกว่า 1 Codon และมักมี tRNA สำหรับกรดอะมิโนชนิดหนึ่งๆ มากกว่า 1 แบบ ซึ่งแต่ละแบบมี

รหัสที่สอดคล้องกับลำดับเบสของแต่ละ Codon มีรายงานว่า tRNA แต่ละแบบสำหรับกรดอะมิโนชนิดเดียวกันมีปริมาณต่างกัน ยีนที่มีการแสดงออกในระดับสูงจะมี Codon ที่สอดคล้องกับตัวถอดรหัส (Anti-Codon) บน tRNA ชนิดที่มีอยู่ปริมาณมากภายในเซลล์ รูปแบบของการใช้ Codon ที่ใช้บ่อยเหล่านี้เพื่อประกอบเป็นยีน เป็นลักษณะที่เรียกว่า "Codon Preference" โดยคำนี้จะมีความหมายต่างจาก "Codon Usage" ซึ่งจะหมายถึงความถี่ที่ Codon แต่ละตัวถูกใช้เพื่อเป็นองค์ประกอบของลำดับเบสของยีนแต่ละยีน

แนวคิดหนึ่งซึ่งสามารถใช้วิเคราะห์หา Coding Region นั้น ทำโดยการคำนวณหา Codon Usage ของ ORF ที่ต้องการวิเคราะห์ ถ้า Codon Usage ที่คำนวณได้สอดคล้องกับ Codon Preference สำหรับสิ่งมีชีวิตนั้น ก็น่าจะเป็นไปได้ว่าดีเอ็นเอบริเวณที่ทำการวิเคราะห์เป็น Coding Region แต่ในทางปฏิบัติอาจจะไม่ทราบ Codon Preference ของสิ่งมีชีวิตแต่ละชนิด แนวทางหนึ่งที่ถูกนำมาช่วยคือใช้ Codon Usage ของยีนที่มีการแสดงออกในระดับสูงแทน Codon Preference ซึ่งยีนเหล่านี้ได้แก่ ยีนสำหรับ Ribosomal Protein, ยีน Histone, ยีน Actin, และยีน Tubulin เป็นต้น

มีเงื่อนไขที่จะต้องพิจารณาสำหรับวิธีนี้คือ วิธีดังกล่าวอาจใช้วิเคราะห์หา Coding Region ได้ดีสำหรับยีนที่มีการแสดงออกในระดับสูง แต่อาจไม่เหมาะสมหรือให้ผลที่ไม่แม่นยำสำหรับยีนที่มีการแสดงออกในระดับที่ต่ำมาก ทั้งนี้เนื่องจากยีนเหล่านี้อาจมีรูปแบบการใช้ Codon (Codon Usage) แตกต่างจาก Codon Preference นอกจากนี้วิธีนี้อาจไม่เหมาะสมสำหรับสิ่งมีชีวิตที่เป็น Multicellular ทั้งนี้เนื่องจากสิ่งมีชีวิตประเภทนี้อาจมี Codon Preference มากกว่า 1 แบบที่แตกต่างกันในเนื้อเยื่อส่วนต่างๆ หรือในระยะเวลาที่ต่างกันในระหว่างการเจริญเติบโต (Different Stages of Development สามารถใช้โปรแกรมสำหรับการคำนวณหา Codon Usage ของ ORF ตามเว็บไซต์ที่ให้บริการ

2.2.1.3 Hidden Markov Model (HMM) โปรแกรมที่ทำงานโดยอาศัยหลักการของ HMM เพิ่งเริ่มใช้กันแพร่หลายในระยะไม่กี่ปีที่ผ่านมา การอธิบายหลักการของ HMM ต้องใช้เนื้อที่มาก จึงจะกล่าวถึงหลักการเพียงสังเขปดังนี้ Markov Chain หมายถึงลำดับของเหตุการณ์ที่เกิดขึ้นโดยที่โอกาส (Probability) ของการเกิดเหตุการณ์ต่างๆ ณ จุดใดๆ ในลำดับ ขึ้นกับเหตุการณ์ที่เกิดขึ้นก่อนหน้านั้นเท่านั้น

การวิเคราะห์ลำดับเบสหรือกรดอะมิโนสามารถทำได้โดยการสมมติว่า ลำดับเบสหรือกรดอะมิโนก็เป็น Markov Chain นั่นคือ โอกาสที่ตำแหน่งใดตำแหน่งหนึ่ง (ตำแหน่ง n) ในลำดับเบสจะเป็น A, T, G หรือ C ขึ้นอยู่กับว่าเบสที่อยู่ด้านซ้าย (ตำแหน่ง n-1) เป็นอะไรเท่านั้น หรือ โอกาสที่จะพบกรดอะมิโนอะไร ณ ตำแหน่งใดตำแหน่งหนึ่งในโปรตีนขึ้นอยู่กับว่ากรดอะมิโนที่อยู่ด้านซ้ายเป็นอะไร นักชีววิทยาอาจโต้แย้งว่าไม่น่าจริง อย่างไรก็ดี แนวคิดดังกล่าวเป็นประโยชน์อย่างมากในการสร้าง Algorithm และโปรแกรมคอมพิวเตอร์ในการทำนายเรื่องต่างๆ เช่น ลำดับเบสนี้เป็น Coding Sequence หรือไม่ หรือเป็น CpG Island หรืออื่นๆ ที่ทำได้ทั้งนี้เป็นเพราะในธรรมชาตินั้น ลำดับที่มีหน้าที่ทางชีววิทยาแตกต่างกันก็จะมีโอกาสเช่นที่กล่าวถึงข้างต้นแตกต่างกันไปด้วย

หลักการ HMM ไม่เพียงแต่จะสามารถทำนายคุณสมบัติของลำดับที่กำหนดให้เท่านั้น แต่ยังสามารถทำนายว่า ส่วนใดของลำดับมีคุณสมบัติอย่างไรด้วย เช่นสามารถบอกได้ว่า ส่วนใดของลำดับ

กรดอะมิโนน่าจะเป็น α-Helix ส่วนใดน่าจะเป็น β-Pleated Sheet โดยที่ผู้ใช้ไม่ต้องช่วย จึงเป็นวิธีการที่ได้รับความนิยมมากสิ่งที่เป็นปัญหาคือ จะต้องบอกโปรแกรมว่า โอกาสที่ให้มองหาอยู่นั้นคือเท่าไร วิธีที่สะดวกที่สุดในการบอกก็คือ "สอน"โปรแกรมว่า ลำดับเบสกลุ่มนี้เป็น Coding Sequences กลุ่มนั้นเป็น Non-Coding Sequences แล้วให้โปรแกรมไปคำนวณโอกาสเอาเอง ลำดับที่ใช้สอนโปรแกรมเรียกว่า Training Set ยิ่งมีลำดับใน Training Set จำนวนมากยิ่งดี ยิ่งใกล้เคียงกับสิ่งที่สนใจมากยิ่งดี เมื่อโปรแกรมได้รับการสอนดีแล้วก็จะสามารถไปทำงานได้ดังใจ โปรแกรมที่ใช้ HMM จึงเป็นโปรแกรมในกลุ่มที่เรียกว่า Neural Network

ในทางปฏิบัติ ปัญหาที่สำคัญที่สุดของการใช้ HMM คือไม่มี Training Set ที่ดี ยกตัวอย่าง เช่น ในการทำโครงการจีโนมปลาปักเป้าซึ่งเป็นปลาตัวแรกที่ถูกหาลำดับเบสเสร็จ การทำนาย Coding Sequences ก็ต้องอาศัยลำดับเบสของยีนสัตว์อื่นๆ เช่น คน ซึ่งจะทำให้การทำนายผิดพลาดไปได้พอควร Web Sites ที่ให้บริการโปรแกรมที่ใช้ค้นหายีน เช่น HMMgene เวอร์ชั่น 1.1 ซึ่งใช้ทำนาย Coding Regions ของสัตว์มีกระดูกสันหลังและ *C. elegans* ตามเว็บไซต์ที่ให้บริการหรือ Genemark.hmm ซึ่งมี Version ที่ใช้ได้กับทั้ง Prokaryotes และ Eukaryotes เป็นต้น

2.2.2 วิธีหาการแสดงออกของยีน (Search by Signal) บริเวณที่เกี่ยวข้องกับการควบคุมการแสดงออกของยีน (Regulatory Element) มักจะเป็นลำดับเบสสายสั้นๆ ที่มีลำดับแตกต่างกันได้บ้าง วิธีการหาลำดับเบสบริเวณดังกล่าวสามารถทำโดยการหาลำดับเบสที่ตรงหรือใกล้เคียงกับ Regulatory Element ที่ต้องการ ประสิทธิภาพของวิธีหาบริเวณที่เกี่ยวข้องกับการควบคุมการแสดงออกของยีนขึ้นอยู่กับว่า จะสามารถกำหนดลักษณะของลำดับเบสของ Regulatory Element ให้ดีชัดเจนได้เพียงใด วิธีกำหนดลักษณะของ Regulatory Element มีดังนี้

2.2.2.1 การใช้ลำดับเบสที่เป็น Consensus วิธีนี้เป็นวิธีที่ง่ายที่สุด อย่างเช่น บริเวณโปรโมเตอร์ที่ตำแหน่ง -10 ของยีนในแบคทีเรีย *E. coli* มีลำดับเบสที่เป็น Consensus คือ TATAAT ดังนั้นในการหาบริเวณโปรโมเตอร์ของยีนในแบคทีเรีย *E. coli* โปรแกรมคอมพิวเตอร์จะทำการหาบริเวณที่มีลำดับเบส TATAAT หรือลำดับเบสที่ใกล้เคียงกับ TATAAT มากที่สุด และสามารถใช้โปรแกรมคอมพิวเตอร์ทำนายบริเวณตัด-แต่งด้วย (Splice Sites Prediction) เช่น SPLICEVIEW และ SplicePredictor เป็นต้น

2.2.2.2 Positional Weight Matrices (PWMs) บอกความน่าจะเป็นของการปรากฏเบส ที่ตำแหน่งใดตำแหน่งหนึ่ง (ใช้วิธี Multiple Alignment กับลำดับเบสที่มีเกี่ยวข้องกัน) อาจกล่าวได้ว่า PWMs เป็น Markov Model อันดับศูนย์ ดังนั้น PWMs สามารถใช้เครือข่าย Neural เข้าช่วยให้การทำนายดียิ่งขึ้น และรวมถึงวิธีซึ่งนำ Markov Model มาใช้เช่นวิธี Weight Array Model (WAM) ซึ่งโปรแกรม Genscan ใช้ WAM สำหรับจำลองการตัด-แต่งบริเวณ Acceptor ส่วน Maximal Dependence Decomposition (MDD) เป็นอีกวิธีสำหรับจำลองการตัด-แต่งบริเวณ Donor โปรแกรมดังต่อไปนี้ใช้สำหรับจำลองบริเวณการตัด-แต่ง ดังแสดงในตารางที่ 2-2

โปรแกรมการหาบริเวณการตัด-แต่ง นั้นมีวัตถุประสงค์เพื่อที่จะแสดงบริเวณตัด-แต่งที่เป็นไปได้ทั้งหมด แล้วสร้างโครงสร้างยีนต่างๆ เท่าที่ทำได้ จุดประสงค์ของโปรแกรมการหาบริเวณการตัด-

ตารางที่ 2-2 โปรแกรมจำลองบริเวณการตัด-แต่ง (Splice Sites Prediction)

โปรแกรม	สิ่งมีชีวิต	วิธีการ
GeneSplicer	*Arabidopsis*, Human	HMM + MDD
NETPLANTGENE	*Arabidopsis*	NN
NETGENE2	Human, *C. elegans*, *Arabidopsis*	NN + HMM
SPLICEVIEW	Eukaryotes	Score with Consensus
SPLICEPREDICTOR	*Arabidopsis*, *maize*	(i) Score with Consensus (ii) Local Composition Linear Discriminant Analysis

แต่ง พยายามทำนายกรอบของ Exon ที่ถูกต้องจะได้นำไปทำนายโครงสร้างยีน ท้ายสุดการนำโปรแกรมแบบ HMM ร่วมกับโปรแกรมต่างๆ ช่วยให้การจำลองสัญญาณต่าง ๆ เช่น Poly(A) หรือ 3'-UTRs (Untranslated Regions), Promotor และ Translation Initiation Codon ได้ดียิ่งขึ้น

2.3 การค้นหายีนโดยการศึกษาเชิงเปรียบเทียบ (Comparative Genomics)

Comparative Genomics คือการค้นหายีนโดยศึกษาจีโนมในเชิงเปรียบเทียบ โดยเฉพาะอย่างยิ่งศึกษาความสัมพันธ์ระหว่างจีโนมในสายพันธุ์หรือสายเลือดที่แตกต่างกัน ยกตัวอย่างเช่น นำข้อมูลโครงสร้างและหน้าที่ของยีนจากแบคทีเรีย ยีสต์ หนู และแมลงหวี่มาเปรียบเทียบกับรหัสพันธุกรรมจากจีโนมของมนุษย์ เพื่อหาหน้าที่ของยีนและโปรตีน โดยใช้สมมุติฐานว่ายีนเหล่านี้มีความสัมพันธ์ในช่วงวิวัฒนาการ (Evlutionaly) ทำให้บริเวณที่เป็นยีนมีอัตราการกลายพันธุ์ (Mutate) น้อยกว่าบริเวณที่ไม่เป็นยีน ใช้สายพันธุ์ที่ใกล้เคียงกันมาทำการศึกษาจีโนมในเชิงเปรียบเทียบเพื่อหาบริเวณที่มีการสงวน (Conservation) นอกจากนี้การศึกษาจีโนมในเชิงเปรียบเทียบมีความสำคัญอย่างมากในการศึกษากระบวนการวิวัฒนาการ เช่นในยีสต์ เป็นต้น

ตัวอย่างหนึ่งของการศึกษาเชิงเปรียบเทียบคือการศึกษาจีโนมของข้าว [14] ข้าวเป็นพืชต้นแบบสำหรับใช้ค้นหายีน ในกลุ่มธัญพืชด้วยกัน ข้าวมีขนาดของจีโนมประมาณ 400 ล้านเบส ซึ่งคาดว่าจะมียีนที่แสดงออกต่างที่และต่างเวลา อยู่ประมาณ 30,000 ชนิด ในขณะที่จีโนมของธัญพืชอื่นๆ มีขนาดใหญ่โตมาก เช่น ข้าวโพดมีประมาณ 7 เท่าของข้าว และข้าวสาลี 40 เท่าของข้าว ดังนั้นโอกาสที่มีจีโนมของพืชเหล่านี้จึงเป็นไปได้น้อย เพราะต้องลงทุนสูงพอๆ กับโครงการจีโนมมนุษย์ แต่เนื่องจากธัญพืชเหล่านี้มี วิวัฒนาการร่วมกับข้าวมา ดังนั้นจึงมีการกล่าวถึงความเป็นไปได้ที่จะใช้ จีโนมข้าวสำหรับอ้างอิง (Reference Genome) หลักฐานชิ้นแรกที่มาสนับสนุนแนวความคิด "Reference Genome" คือ การศึกษา Comparative Mapping

การทำแผนที่เพื่อศึกษาจีโนมเชิงเปรียบเทียบระหว่าง Species โดยทำ DNA Marker ชุดเดียวกัน ระหว่าง ข้าว ข้าวโพด ข้าวฟ่าง ข้าวสาลี ข้าวบาเลย์ และข้าวไรย์ พบว่าบริเวณชิ้นส่วนของโครโมโซมในตำแหน่งที่ทำเครื่องหมายดีเอ็นเอ ยังคงมีการวางลำดับเหมือนเดิม เชื่อกันว่าบริเวณชิ้นส่วนดังกล่าว

เป็น "เขตอนุรักษ์" (Syntenic Region คือส่วนของโครโมโซมจาก สิ่งมีชีวิตต่างสกุลกัน ที่มีการอนุรักษ์ลำดับการเรียงตัวของยีนเอาไว้) ที่ได้รับการถ่ายทอดมาจากจีโนมบรรพบุรุษ ในแต่ละเขตอนุรักษ์ พบว่ายีนยังวางตำแหน่งในลำดับเดียวกัน (Conservation of Gene Order) ปรากฏการณ์ที่น่าทึ่งนี้เกิดขึ้นโดยไม่ขึ้นกับจำนวนโครโมโซมและจำนวนชุดของโครโมโซมของ Species ต่างๆ ที่สนใจ องค์ความรู้ดังกล่าวได้จุดประกายความเป็นไปได้ที่จะเปรียบเทียบยีนในกลุ่มธัญพืชเพื่อศึกษาวิวัฒนาการของจีโนมในพืชตระกูลหญ้าอย่างละเอียดและลงแนวความคิดสรุปว่าธัญญพืชมีระบบจีโนมเป็นหนึ่งเดียว

2.4 การค้นหายีนแบบผสม

การค้นหายีนแบบผสมคือการรวมกันระหว่าง การค้นหายีนจากภายนอก, การค้นหายีนภายใน และ การค้นหายีนโดยการศึกษาเชิงเปรียบเทียบ จากความก้าวหน้าของเทคโนโลยีในยุคปัจจุบัน ทำให้ทราบว่าการค้นหายีนด้วยวิธีหนึ่งวิธีใดเพียงอย่างเดียวไม่เพียงพอ ยกตัวอย่างเช่น การค้นหายีนจากภายนอก ส่วนมากต้องอาศัยฐานข้อมูลสำหรับสืบค้น ดังนั้นฐานข้อมูลที่ใช้ต้องมีคุณภาพเพราะต้องใช้ฐานข้อมูล เปรียบเทียบกับจีโนมที่ต้องการค้นหา นอกจากคุณภาพแล้วยังต้องครอบคลุมการค้นหายีนด้วย หรือแม้ต่อให้ฐานข้อมูลมีทั้งคุณภาพและครอบคลุมเพียงพอ อย่างเช่นโปรแกรม BLAST ก็มักจะพลาดหรือหาไม่เจอถ้ายีนนั้นมีขนาดเล็กมากๆ โดยเฉพาะอย่างยิ่งกับเยูคาริโอตที่ Exon มีขนาดเล็ก สำหรับการค้นหาจากภายในที่อาศัยการคำนวณการใช้โคดอน (Codon Usage) หรือ HMM ก็ตามอาศัยหลักการทำนายโครงสร้างของยีน แล้วทำนายว่ายีนควรจะเป็นอะไรด้วยความน่าจะเป็นมากน้อยแค่ไหน สามารถหายีนได้ด้วยวิธีนี้ แต่อย่างไรก็ดีมักจะหาได้เกิน โดยวิธีค้นหายีนจากภายนอกมักหาได้ขาด ทำให้เกิดแนวคิดรวมทั้งสองวิธีนี้เข้าด้วย เพื่อชดเชยซึ่งกันและกัน นอกจากนี้วิธีการค้นหายีนโดยการศึกษาเชิงเปรียบเทียบทำให้สามารถสร้างแผนภูมิต้นไม้ของความสัมพันธ์ระหว่างสิ่งมีชีวิตได้ (Phylogenetic Tree) เป็นประโยชน์ต่อการออกแบบการทดลอง เช่น การออกแบบ PCR Primer ที่สามารถเพิ่มจำนวนดีเอ็นเอของลำดับเบสทั้งหมดได้

พอจะกล่าวได้ว่าการค้นหายีนจากภายใน คือการหายีนโดยทำการวิเคราะห์สายลำดับดีเอ็นเอเพียงเส้นเดียว ส่วนการศึกษาจากภายนอกและการค้นหายีนโดยการศึกษาเชิงเปรียบเทียบ คือการวิเคราะห์สายลำดับดีเอ็นเอมากกว่าหนึ่งเส้น ทั้งนี้ทั้งนั้นโปรแกรมทำนายโครงสร้างยีนส่วนใหญ่จะทำการรวมข้อมูล Homology อยู่ด้วย ยกตัวอย่างเช่นโปรแกรม GSA (Genome Structure Assembly) เกิดจากการรวมกันระหว่าง AAT และ Genscan ซึ่งเมื่อนำ 2 โปรแกรมนี้มาพิสูจน์แล้วใช้รวมกันผลลัพธ์ที่ได้ดีกว่าใช้โปรแกรมใดโปรแกรมเพียงอย่างเดียว อีกตัวอย่างหนึ่งคือ GenomeScan โดยใช้โปรแกรม Genscan รวมกับโปรแกรม BLAST โดยเปรียบเทียบความเหมือนระหว่างโปรตีนที่ได้จากโปรแกรม BLASTX กับ BLASTP ยีนที่ถูกทำนายโครงสร้างจากโปรแกรม GenomeScan จะมีความเป็นไปได้มากกว่าเมื่อใช้โปรแกรม Genscan หรือ BLASTX เพียงอย่างหนึ่งอย่างใด

อีกตัวอย่างหนึ่งก็คือ FGENESH นำโปรตีนที่เหมือนกันหรือนำ cDNA (Complementary DNA เป็นสำเนาของลำดับเบส mRnA ที่ผ่านกระบวนการถอดรหัส) มาเพื่อปรับปรุงการทำนายยีนโดยได้ทำการทดสอบกับจีโนมข้าว และเปรียบเทียบกับโปรแกรม Genscan กับ HMMGene นอกจากนี้

แนวคิดก็ได้ใช้ใน EuGène ด้วยการนำ NetGene2, SplicePredictor สำหรับการทำนายบริเวณตัดแต่ง, NetStart สำหรับการทำนายจุดเริ่มต้นการแปล (Translation Initiation Prediction), IMM (Interpolated Markov Model) ร่วมกับข้อมูลโปรตีน, ESTs (Expressed Sequence Tag) และ cDNA

เนื่องจากการทำหมายเหตุจีโนม ประกอบด้วยการค้นหายีนและการทำหมายเหตุ ในบทนี้จึงได้อธิบายวิธีการค้นหายีนในแบบต่างๆ โดยคร่าวและ อธิบายถึงข้อดีข้อเสียของการค้นหายีน ในแต่ละวิธีซึ่งก็แตกต่างกันออกไป ตามลักษณะของสิ่งมีชีวิต การจะจำแนกยีนออกจากสิ่งที่สนใจ นั้นต้องใช้ความพยายามมากกว่าหนึ่งโปรแกรมเป็นต้นไป เพื่อให้ผลลัพธ์เป็นที่ยอมรับ ดังนั้นเองการศึกษาในบทนี้จึงมีความสำคัญอย่างมาก นอกจากการทำหมายเหตุประกอบเพียงอย่างเดียว แล้วในบทต่อไปจะกล่าวถึง วิธีการ, สถาปัตยกรรม และระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนในจีโนมที่ผ่านๆ มาและนำไปสู่ระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนในจีโนมแบบอัตโนมัติที่จะนำเสนอในบทที่ 4 ต่อไป

บทที่ 3
ระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนม

ในบทที่ 3 ของวิทยานิพนธ์ฉบับนี้กล่าวถึง ระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนม อันประกอบไปด้วย ระบบเครือข่าย ระบบฐานข้อมูล ระบบสารสนเทศ ซึ่งในอดีตจนถึงปัจจุบันว่ามีปัจจัยอะไรบ้างถึงทำให้หลายโครงการจีโนมประสบความสำเร็จเป็นอย่างมาก ในบทนี้จะกล่าวถึงระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนมโดยละเอียด ส่วนระบบอื่นๆ ที่มีส่วนเกี่ยวข้องจะอธิบายโดยสังเขป ถึงแนวทางประยุกต์นำระบบเหล่านี้ทำงานร่วมกับ ระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนม

3.1 ปัญหาการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนม

3.1.1 ปัญหาเนื่องจากจำนวนของลำดับเบสมีมากเกินไปจากภาพที่ 3-1 แสดงปริมาณข้อมูลใน GenBank ที่เพิ่มขึ้นจากปี 1982 ถึง 2007 จาก 7 แสนเบสเป็น 76 พันล้านเบสในเดือนเมษายน ปี 2007 ทำให้การทำการระบุตำแหน่งในแบบเก่า ต้องเปลี่ยนเป็นทำการระบุตำแหน่งแบบอัตโนมัติ ในแบบที่หลายเว็บไซต์นำมาใช้กันอย่างแพร่หลาย

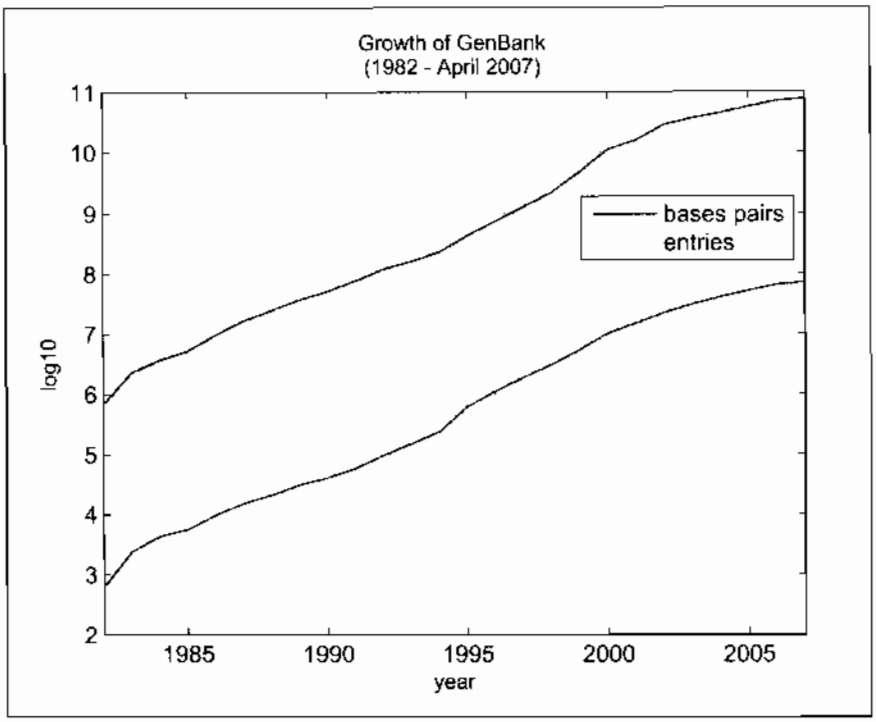
3.1.2 ปัญหาในเชิงคำนวณอันเกี่ยวเนื่องกับปัญหาความยากง่ายในแต่ละสิ่งมีชีวิตที่แตกต่างกัน ยกตัวอย่างเช่นปัญหาความแตกต่างระหว่างโปรคาริโอตกับยูคาริโอต ปัญหาอันเกิดจากยีนไม่อยู่ติดกันหรือไม่ก็อยู่อย่างกระจัดกระจายอันมักพบได้บ่อยในยูคาริโอต เป็นต้น

3.1.3 ปัญหาสมรรถนะภาพการคำนวณ เกิดจากความต้องการของ ในการศึกษาสิ่งมีชีวิตในระดับจีโนม ซึ่งขนาดของจีโนมก็แปรผันตรงกับความต้องการสมรรถนะภาพการคำนวณที่สูงตาม

3.1.4 ปัญหาอันเกิดจากผู้ใช้เอง ปัญหานี้เกิดจากความผิดพลาดโดยไม่ได้ตั้งใจซึ่งอาจจะเกิดขึ้นได้เสมอโดยไม่รู้ตัว

3.1.5 ปัญหาการปฏิสัมพันธ์ระหว่างโปรแกรมในระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนม เพราะว่าโปรแกรมไม่ได้ถูกออกแบบมาเพื่อทำงานแบบปฏิสัมพันธ์กันตั้งแต่แรก ทำให้การทำงานระหว่างโปรแกรมไม่ค่อยราบรื่น มักเจอปัญหาเข้ากันไม่ค่อยได้บ่อยครั้ง

จากปัญหาที่ได้กล่าวข้างต้น เป็นเพียงปัญหาในหลายๆ ปัญหาที่ไม่ได้กล่าวถึง แต่ก็เพียงพอต่อการศึกษาเพื่อนำไปพัฒนาต่อไปอย่างเพียงพอ ส่วนปัญหาเชิงคำนวณสามารถแก้ด้วยการใช้โปรแกรมให้เหมาะสมกับสิ่งที่ต้องการค้นหา เช่น บริเวณตัด-แต่ง, บริเวณ Promotor, Acceptor, Donor หรือบริเวณเริ่มต้นการถอดรหัสให้เหมาะสม แต่จากปัญหาการปฏิสัมพันธ์ระหว่างโปรแกรม ทำให้ถึงแม้การแก้ปัญหาเชิงคำนวณได้ แต่ปัญหาระหว่างโปรแกรมก็ยังอยู่ ปัญหาการปฏิสัมพันธ์ระหว่างโปรแกรมก็คือปัญหาที่โปรแกรมไม่เข้าใจกัน เพราะข้อมูลจากโปรแกรมหนึ่ง ไปเป็นข้อมูลส่งเข้าอีกโปรแกรมมี



ภาพที่ 3-1 ข้อมูลใน GenBank ตั้งแต่ปี 1982 ถึงเดือนเมษายน ปี 2007

ปัญหา เนื่องจากไม่ได้ถูกออกแบบให้ใช้ร่วมกันตั้งแต่แรก การแก้ปัญหานี้ทำโดยสร้างตัวปรับเปลี่ยน (Adaptor) ขึ้นมาทำการแปลงผลลัพธ์จากโปรแกรม ให้อยู่ในรูปแบบที่โปรแกรมอื่น สามารถอ่านเข้าใจได้ โดยผ่านตัวปรับเปลี่ยนก่อนนำไปใช้งาน เพื่อมั่นใจได้ว่าข้อมูลถูกส่งอย่างถูกต้อง ไม่ขาดหายไประหว่างการส่งข้อมูล ให้อยู่ในรูปแบบเอกสาร XML (Extensible Markup Language) ซึ่งการพัฒนากันอย่างแพร่หลายเป็นภาษา XML เองหรือภาษาอื่นๆ ก็นำ XML เข้าไปเป็น Extension ทำให้ภาษานั้นๆ มีความยืดหยุ่นในตัวมันเองมากยิ่งขึ้น

ปัจจุบันมีโปรแกรมและระบบทำการระบุตำแหน่งของยีนมากมาย อาทิเช่น Ergo [9] , Pedant-Pro, Artemis [10] , GenDB[15] , Mantee (TIGR, unpublished), MaGe [5], BaSys [6], RiceGAAS [7], AMIGene [16] และ Ensembl เหล่านี้มีวิธีการค้นหายีนแตกต่างกันและระบบที่ใช้ก็ไม่เหมือนกัน เช่น ใช้กับโปรคาริโอตเท่านั้นก็มี MaGe (A Microbial Genome Annotation System), BaSys (Bacterial Annotation System) หรือ RiceGAAS (Rice Genome Automated Annotation System) ใช้ระบุตำแหน่งยีนของข้าว ดังเห็นได้จากตัวอย่างที่ยกมานี้ ระบบการระบุตำแหน่งจะสนใจเฉพาะเจาะจงในสิ่งมีชีวิตใดชีวิตหนึ่งทำให้ เป็นข้อจำกัดอย่างหนึ่ง

3.2 กลวิธีในระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนม: การค้นหายีน

กลวิธีที่ใช้ต้องสอดคล้องกับการแก้ปัญหาในเชิงคำนวณ ยังต้องสามารถรับมือกับจำนวนข้อมูลจำนวนมหาศาลที่รอการระบุตำแหน่งของยีน ส่วนวิธีที่ใช้อย่างกว้างขวางในการระบุตำแหน่งของยีนบนจีโนม [17] ประกอบด้วย (i) Homology หรือ การค้นหายีนจากภายนอก [18] (ii) การค้นหายีน

ด้วยการทำนายยีน [1], [3] ด้วยวิธีแรกสามารถค้นหายีนจากภายนอก จากยีนหรือโปรตีนในฐานข้อมูลต่างๆ ได้อย่างมากก็ 50% (ถึงแม้ว่าเปอร์เซ็นต์นี้จะขึ้นกับความสำเร็จของโครงการถอดรหัสพันธุกรรมสิ่งมีชีวิตต่างๆ ยิ่งถอดรหัสได้มากเปอร์เซ็นต์ก็มากตาม) ส่วนยีนที่ยังไม่มีการตีความมาก่อน ก็ต้องใช้วิธีทำนายยีนหรือโปรตีน ส่วนที่เหลือด้วยการทำนายยีนที่อย่างรวดเร็ว, แม่นยำและน่าเชื่อถือ [19]

ในปัจจุบันมีโปรแกรมการค้นหายีนหลายตัวโดย Wentian Li [20] ได้ลิงค์โปรแกรมการค้นหายีนตามเว็บต่างๆ นอกจากนี้ยังมีการวิจารณ์ถึงจุดแข็งจุดอ่อนของแต่ละโปรแกรมสำหรับโปรคาริโอตและยูคาริโอตโดย Claverie [21], Guigó [22] , Haussler [23] และ Burge กับ Karlin [24] ปัจจุบันโปรแกรมโดยส่วนมากมีแนวโน้ม รวมเอาวิธีการค้นหายีนมากกว่า 1 วิธีมาใช้ร่วมกัน เช่นเดียวกับระบบการระบุตำแหน่งยีนบนจีโนม ซึ่งรวบรวมโปรแกรมต่างๆ ทำการคำนวณและเปรียบเทียบข้อมูล เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด ดังนั้นกลวิธีต่างๆ จึงถูกนำมาใช้ร่วมกัน ต่อไปนี้จึงขอยกตัวอย่างกรณีศึกษาของระบบค้นหายีนที่ผ่านมา ดังต่อไปนี้

3.3 กรณีศึกษาระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนในโปรคาริโอต

3.3.1 กรณีศึกษาระบบการระบุตำแหน่งและกำหนดหน้าที่ BaSys ให้บริการผ่านเว็บและระบุตำแหน่งของยีนบนจีโนมแบคทีเรียแบบอัตโนมัติ BaSys รับข้อมูลในรูปของดีเอ็นเอและสามารถรับรายละเอียดข้อมูลต่างๆ เกี่ยวกับเอกลักษณ์ของยีน และแสดงผลลัพธ์เป็นรูปภาพหรือข้อความ อย่างครอบคลุม ใน BaSys ประกอบด้วยโปรแกรมมากถึง 30 โปรแกรม ระบุตำแหน่งในกว่า 60 หัวข้อยกตัวอย่างเช่น ชื่อยีนและชื่อโปรตีน, หน้าที่ความหมายของยีนที่หามาได้, หากลุ่มหน้าที่การทำงานของยีนที่ Orthologus (ได้จากเปรียบเทียบ Proteome ทั้งหมดและรวมถึง Orthologs และ Paralogs), ขนาดน้ำหนักของโมเลกุล, ค่าการตกตะกอนของโปรตีน (Isoelectric Point), โครงสร้างของ Operon, ตำแหน่ง Subcellular Localization, ข้อมูล Signal Peptides, ตำแหน่ง Transmembrane Region, โครงสร้างทุติยภูมิของยีน (Secondary Structure), โครงสร้าง 3 มิติ, ข้อมูลกลไกการขับเคลื่อนทางชีววิทยา (Pathway) เป็นต้น

BaSys ประกอบด้วย 3 ส่วน

3.3.1.1 เว็บอินเตอร์เฟสเพื่อส่งข้อมูลดีเอ็นเอสู่ระบบ นอกจากนั้นยังสามารถตั้งเวลาระบุตำแหน่งและ ทำรายงานหรือเฝ้าดูความเคลื่อนไหวในการระบุตำแหน่งของยีน ทำเสร็จแล้วไหนสามารถส่งข้อมูลโครโมโซมแบคทีเรียให้ BaSys ซึ่งข้อมูลจะอยู่ในรูป FASTA พร้อมทั้งระบุว่าโครโมโซมเป็นวงกลมหรือเป็นเส้นตรงด้วย โดยคร่าว BaSys ใช้โปรแกรม Glimmer [25] สำหรับการทำนายยีนของแบคทีเรีย ที่มีเปอร์เซ็นต์ GC น้อยกว่า 60% ถ้า GC มากกว่านี้ควรใช้ด้วยความระมัดระวัง นอกจากโปรแกรม Glimmer แล้ว BaSys ยังใช้โปรแกรมอื่นอีกเช่น Critica เป็นต้น ในกรณีที่ทราบตำแหน่งยีนจากโปรแกรมอื่น (เช่น BLAST) ก็สามารถใช้ร่วมโดยการใช้ TAB คั่น สำหรับไฟล์นามสกุล ".ffn" แบบ FASTA โดยไฟล์นามสกุล ".ffn" ของ NCBI สามารถดาวน์โหลดข้อมูลของจีโนมแบคทีเรียได้เว็บ NCBI นอกจากข้อมูลยีนบนจีโนมแบคทีเรีย รวมถึงข้อมูลนิวคลีโอไทด์ในบริเวณต่างๆ ทำให้ผู้ใช้ BaSys สามารถนำข้อมูลเหล่านี้ไปหายีนในโครโมโซมแบคทีเรียได้อย่างถูกต้อง ซึ่งรายละเอียดที่

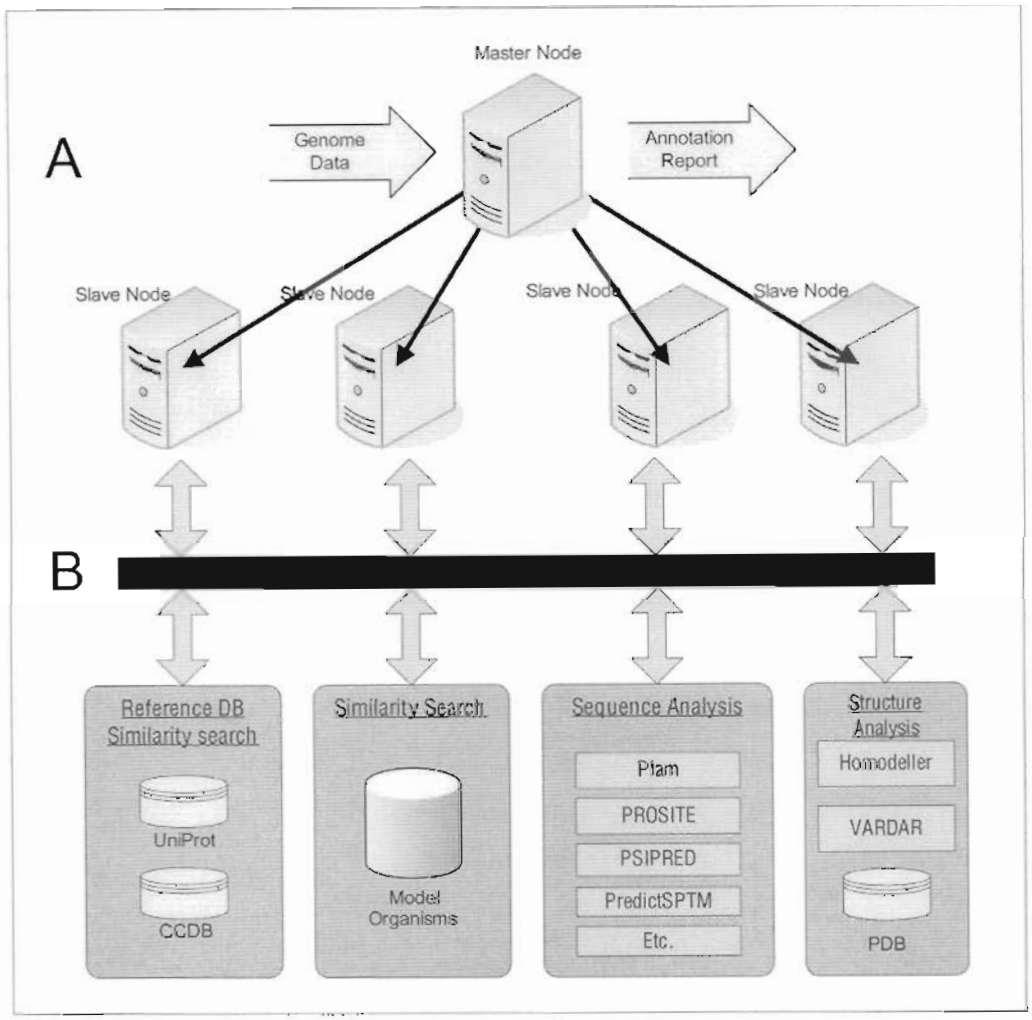
ต้องการและข้อมูลเพิ่มเติมจัดไว้ให้ในหน้าเว็บอินเตอร์เฟสสำหรับส่งข้อมูลให้ BaSys พร้อมเอกสารในการใช้เบื้องต้นก็สามารถเข้าไปดูได้จากเว็บที่ให้บริการ

การที่ BaSys ต้องระบุตำแหน่งสำหรับแบคทีเรียทั้งจีโนม ทำให้ BaSys ต้องใช้การคำนวณที่สูงและใช้ระยะเวลานาน จึงได้ผลลัพธ์ที่ต้องการ (ยังไม่รวมถึงการระบุตำแหน่งซ้ำเพื่อมั่นใจว่าข้อมูลถูกต้อง) โดยปกติจีโนมแบคทีเรียประมาณ 5 Mb หรือ 3,000 ยีน บนเครื่อง 32-bit Intel/Linux PC พร้อมระบุตำแหน่งใช้เวลาประมาณ 24 ชั่วโมง เพื่อที่จะให้ผู้ใช้สามารถทำงานได้หลายคนและทำงานได้อย่างราบรื่น BaSys ใช้ระบบ Cluster กระจายงานดังภาพที่ 3-2 โหนดแม่ (Master Node) ทำหน้าที่เป็นเครื่องให้บริการหลัก (Host for Web Server) ทำการคำนวณโหลดของโหนดลูก นอกจากนี้จัดลำดับงานและตั้งเวลาการทำงานให้ BaSys และแต่ละโหนดลูก (Slave Node) ของโหนดแม่จะลงเครื่องมือสำหรับระบุตำแหน่งประกอบต่างๆ เนื่องจากโหนดแม่ทำการคำนวณโหลดของโหนดลูก โหนดแม่จึงสามารถตัดสินใจส่งงานให้ทำงานที่โหนดลูกได้ตามโหลดที่เหมาะสม นอกจากนี้โหนดแม่ยังสามารถ หยุดการทำงาน เริ่มการทำงาน ลบการทำงานหรือแม้แต่รายงานผลต่างๆ ที่โหนดลูกได้ด้วยโปรแกรมจัดการงานอย่างเช่น SGE (Sun Grid Engine)

3.3.1.2 เครื่องมือในการระบุตำแหน่งทั้งหมด สำหรับวิเคราะห์ข้อมูลโครโมโซมเพื่อสร้างข้อมูลระบุตำแหน่ง ดังแสดงกระบวนการทำงานทั้งหมดในภาพที่ 3-2 โดยวิธีระบุตำแหน่งของ BaSys เป็นการรวม 1) การเปรียบเทียบฐานข้อมูล กับ 2) การวิเคราะห์ลำดับเบสในภาพที่ 3-2 ตอนแรกเริ่มทำการเปรียบเทียบฐานข้อมูลด้วยโปรแกรม BLAST ฐานข้อมูลที่ใช้ได้กับ UniProt และ CyberCell ซึ่งเป็นฐานข้อมูลโมเลกุลของแบคทีเรีย *E. coli* โดยทั้ง 2 ฐานข้อมูลนี้ประกอบด้วยข้อมูลโปรตีนตลอดจนรายละเอียดหน้าที่การทำงานของโปรตีน, Metabolic Role, ข้อมูลเอ็นไซม์ นอกจากนี้เงื่อนไขการใช้โปรแกรม BLAST เช่นค่า Threshold จะขึ้นกับชนิดข้อมูลที่ต้องการค้นหา เช่น ข้อมูล Transmembrane ต้องการค่า E-value น้อยกว่า $1x10^{-10}$

นอกจากผลลัพธ์ที่ได้จาก BLAST ฐานข้อมูล UniProt กับ CyberCell แล้ว BaSys ยังพยายามรวบรวมข้อมูลจากฐานข้อมูลต่างๆ ได้แก่ ฐานข้อมูลโปรตีนของ *C. elegans, H. sapiens, S. cerevisiae* และ *D. melanogaster* ฐานข้อมูลโปรตีน NR ของแบคทีเรีย, ฐานข้อมูล PDB ของโครงสร้างทางชีววิทยา 3 มิติในระดับไมโครโมเลกุล และฐานข้อมูล COG เพื่อศึกษาวิวัฒนาการกับการทำงาน จากกลุ่มยีน Orthologs ซึ่งพบได้ในแบคทีเรียที่สืบทอดมาจากบรรพบุรุษเดียวกัน ยังมีการวิเคราะห์สายลำดับเบสเพื่อวิเคราะห์สกุลโปรตีน (Protein Family) ยังฐานข้อมูล Pfam, นอกจากนี้ยังวิเคราะห์รูปแบบต่างๆ ของสายลำดับเบสจาก PROSITE และทำการวิเคราะห์โดเมนของ Signal Peptide และ Transmembrane ด้วยโปรแกรม PredictSPTM กับทำนายโครงสร้างทุติยภูมิ ด้วยโปรแกรม PSIPRED

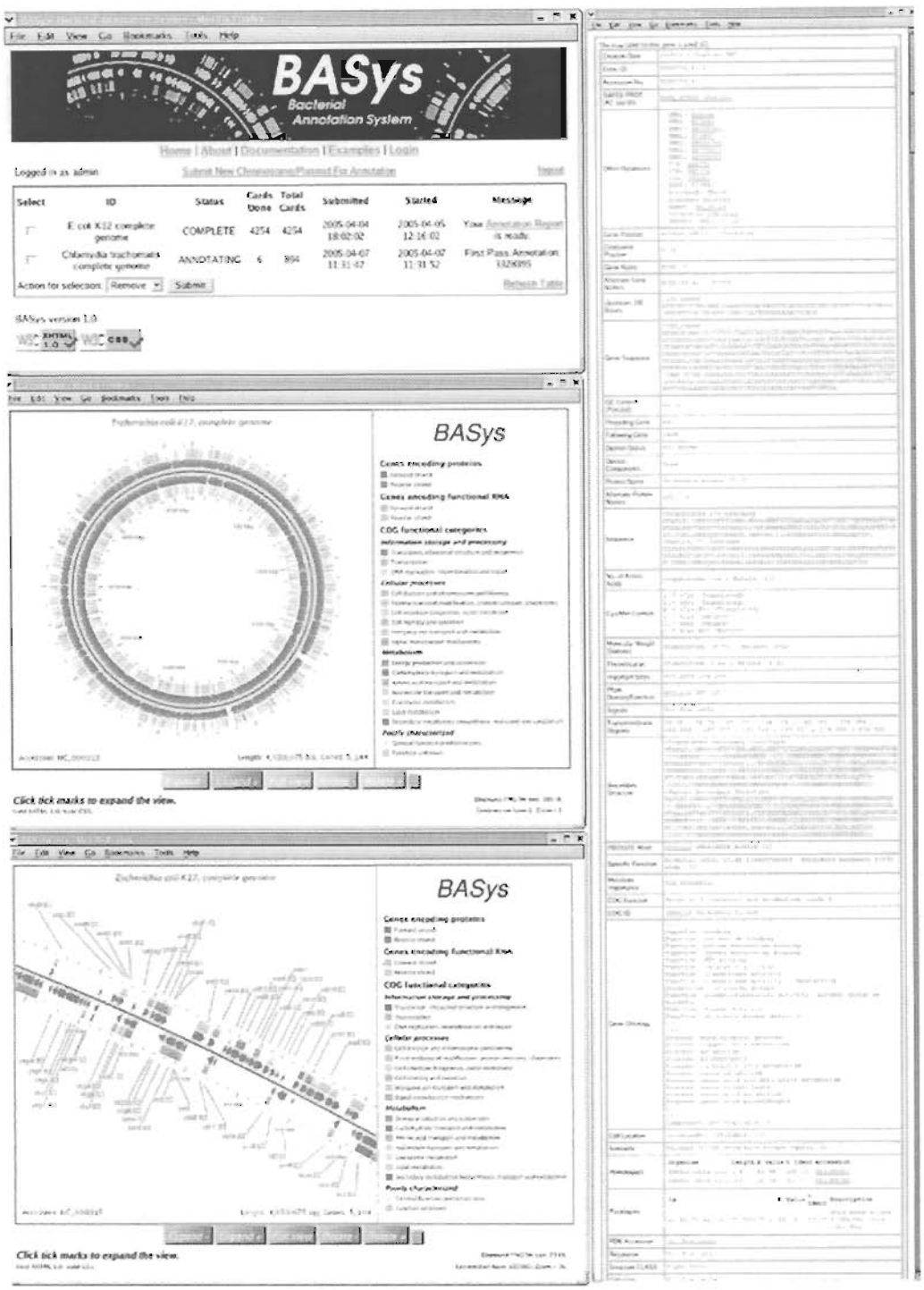
ถ้าสายลำดับเบสมีความเหมือนมากพอกับสายลำดับเบสใน PDB แล้ว BaSys อาจจะใช้โปรแกรม HOMODELLER ทำนายโครงสร้างโปรตีน 3 มิติ และวิเคราะห์โครงสร้างโปรตีนและเปปไทด์โดยใช้โปรแกรม VADAR (Volume, Area, Dihedral Angle Reporter) ตามลำดับ นอกจากนี้ข้อมูลต่างๆ เช่น น้ำหนักของโมเลกุล, ค่าการตกตะกอนของโปรตีน (Isoelectric Point) และโครงสร้างของ Operon จะถูกคำนวณโดยตรงจากโปรตีนในโครโมโซม



ภาพที่ 3-2 รูปแสดงสถาปัตยกรรมของ BaSys

BaSys เป็นระบบระบุตำแหน่งในกว่า 60 หัวข้อโดยรวบรวมมาจากหลายข้อมูลและพยายามระบุตำแหน่งยีนให้ได้มากที่สุด แต่อย่างไรก็ดีความผิดพลาดก็อาจเกิดขึ้นได้ ในกรณีของระบบการระบุตำแหน่งและกำหนดหน้าของยีนทำงานแบบอัตโนมัติทั้งหมด ทำให้ผลลัพธ์ไม่ผ่านการตรวจสอบจากผู้ใช้ ด้วยเหตุนี้ BaSys ได้แก้ปัญหานี้โดยจัดเตรียมข้อมูลให้มากที่สุด ซึ่งข้อมูลจะบอกถึงคุณภาพการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนมอย่างคร่าวๆ ยกตัวอย่างเช่น การระบุตำแหน่งของยีนด้วยโปรแกรม BLAST ประกอบด้วย เวอร์ชั่นของฐานข้อมูล, ฐานข้อมูลที่ใช้, และคุณภาพที่ใช้บอกฐานข้อมูล โดย BaSys จะเก็บข้อมูลเหล่านี้แยกต่างหากเพื่อง่ายแก่การตรวจสอบ

3.3.1.3 ระบบจะทำการรวบรวมข้อมูลเพื่อแสดงผล นำไปสร้างเป็นรูปภาพ ในรูปแบบ HTML และข้อมูลที่ผู้ใช้สามารถเลื่อนผลการระบุตำแหน่งเข้าไปดูรายละเอียดเพิ่มเติม โดยสามารถเลือกเข้าไปดูผ่าน ชื่อของยีนที่ปรากฏบนรูปภาพ ในรูปภาพที่ 3-3 แสดงผลลัพธ์ที่ BaSys จัดแสดงให้ ส่วนรูปภาพของจีโนมถูกสร้างขึ้นจากโปรแกรม CGVIEW โดย BaSys ส่งผลที่ได้จากการระบุตำแหน่งในแบบ XML ให้ CGVIEW แล้วโปรแกรมจะทำการเชื่อมโยงรายละเอียดข้อมูลของยีนและข้อมูล



ภาพที่ 3-3 เว็บอินเตอร์เฟส Basys แสดงผลลัพธ์ด้วยภาพและตัวอักษรของแบคทีเรีย *E.coli*

COG ผ่านรูปแบบไฟล์ ".png" โปรแกรม CGVIEW มีวิธีสร้างรูปภาพในหลายระดับความละเอียด และยังคงแสดงผลในรูปแบบวงกลมในทุกระดับของความละเอียด ซึ่งด้วยวิธีนี้ทำให้การเลือกซูมเข้าไปทำให้สามารถเลือกได้เอง นอกจากนี้ในแต่ละยีนเมื่อเลือกเข้าไปจะแสดงผลในรูปแบบตารางหรือถ้ายีนนั้นมีรายละเอียดเพิ่มเติมที่เว็บอื่นก็สามารถที่จะเลือกไปอ่านที่เว็บนั้นได้ เช่น ผลลัพธ์จากการค้นหา

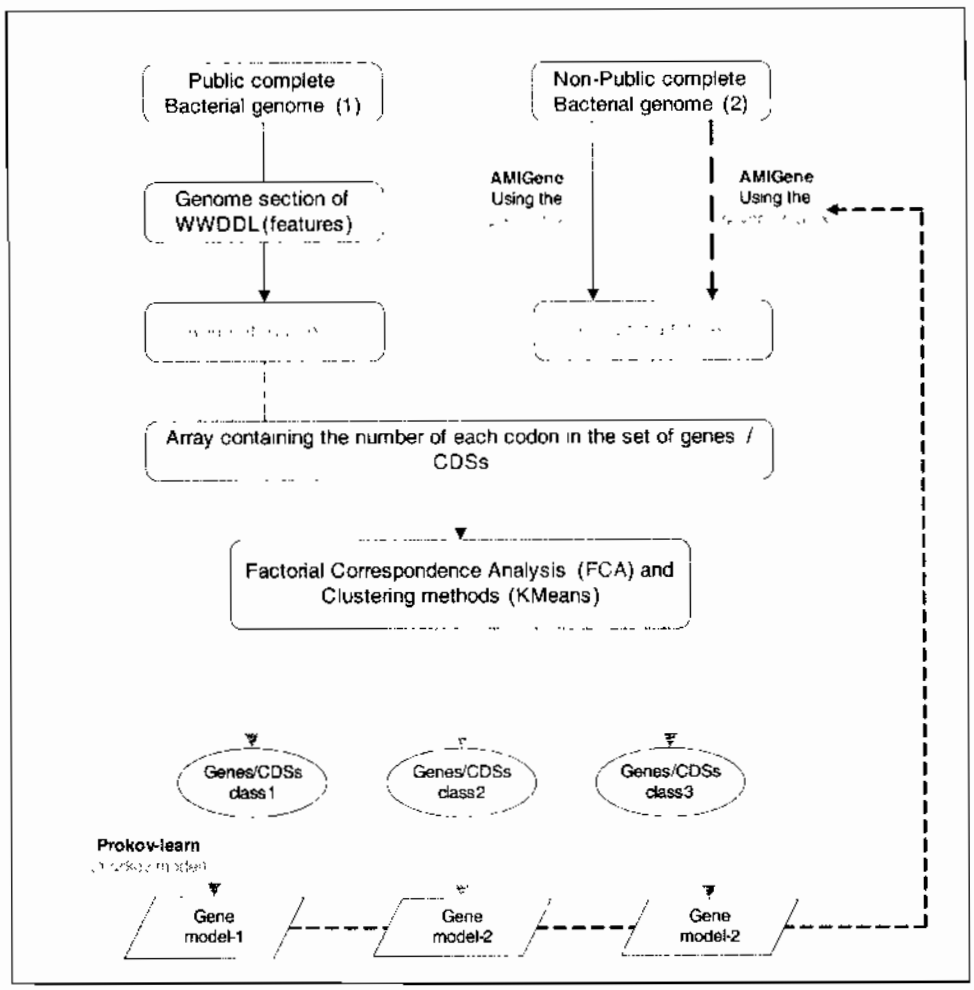
ความเหมือนจากฐานข้อมูลจะบอกรหัสให้ทราบถึงแหล่งที่มา อาทิเช่น [S] หมายถึง 100% สายลำดับเบสที่ตรงกับ SwissProt, [H] หมายถึงคล้ายคลึงกับข้อมูลใน SwissProt, [C] หมายถึงคล้ายคลึงกับข้อมูลใน CCDB ในยีนยังแสดงข้อมูลที่บอกรายละเอียดต่างๆ ในการระบุตำแหน่งและกำหนดหน้าที่ของยีนที่บอกถึงคุณภาพของการระบุตำแหน่ง นอกจากนี้วิธีการค้นหาข้อมูลของยีนของ BaSys เร็ว เพราะเป็นการค้นหาที่ฝั่งเครื่องแม่ข่าย และยังจัดบริการ BLAST สำหรับการค้นหาด้วย

BaSys เป็นเว็บให้บริการระบุตำแหน่งที่ประกอบไปด้วยกว่า 30 โปรแกรม และระบุตำแหน่งข้อมูลได้กว่า 60 รูปแบบ และยังแสดงผลลัพธ์ด้วยรูปภาพที่สามารถเลื่อนเข้า เลื่อนออกได้ พร้อมปรับรายละเอียดรูปภาพโดยอัตโนมัติและเชื่อมโยงข้อมูลของยีนผ่าน รูปภาพที่ระบุถึงยีนที่มีอยู่ในรูปจีโนมของแบคทีเรียที่ระบุตำแหน่ง ข้อดีของเว็บ BaSys คือมีรายละเอียดที่ครอบคลุมแต่ BaSys ยังไม่สามารถรับมือกับจีโนมที่ไม่สมบูรณ์หรือ จีโนมที่นำมาทำเพียงบางส่วนและกรอบเปิดการอ่านที่ซับซ้อน นอกจากนี้ BaSys ไม่สามารถรับมือกับลำดับเบสที่ลำดับเบสผิดพลาดในขั้นตอนการลำดับเบสขึ้นมาใหม่ ทำให้อาจจะนำไปสู่การระบุตำแหน่งที่ผิดพลาดได้

3.3.2 กรณีศึกษาระบบการระบุตำแหน่งและกำหนดหน้าที่ MaGe ระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนมสำหรับจุลชีพ ซึ่งใช้ระบบฐานข้อมูลสัมพันธ์ของข้อมูลแบคทีเรียเป็นหลัก และสามารถแสดงผลลัพธ์ผ่านเว็บ ลักษณะการทำงานของ MaGe ประกอบด้วย

3.3.2.1 การผสมผสานข้อมูล ระบุตำแหน่งและกำหนดหน้าที่ยีนบนจีโนมแบคทีเรีย กับข้อมูลซึ่งถูกปรับปรุงโดยกระบวนการทำซ้ำจากการใช้แบบจำลองยีน ที่มีความแม่นยำ ขั้นแรกเริ่มจากไฟล์ FASTA ซึ่งผ่านการประกอบขึ้นมาใหม่และผ่านระเบียบวิธีที่เหมาะสมกับแบคทีเรีย จนได้เป็นไฟล์ FASTA ที่น่าจะอยู่ติดกัน Contig(s) นำมาเริ่มระบุตำแหน่งโดยสร้างแบบจำลองยีนที่คำนวณการใช้โคดอนเป็นหลัก [16] โดยผสานแบบจำลองบริเวณที่เป็นโปรตีนกับบริเวณที่ไม่ใช่โปรตีน ด้วยระเบียบวิธีของเบย์ (Bayesian Algorithm) และวิเคราะห์ดีเอ็นเอแบบเฉพาะส่วนด้วย Sliding Window สำหรับแบคทีเรียบางชนิดมีการใช้โคดอนที่แตกต่างกันมากกว่า 1 แบบในตัวมันเอง ทำให้โปรแกรม GeneMark [26] ตรวจหายีนไม่เจอในการใช้โคดอนบางแบบ ทำให้ต้องใช้เทคนิคการวิเคราะห์ความสอดคล้องระหว่างสองตัวแปรหรือมากกว่าพร้อมกัน (Factorial Correspondence Analysis: FCA) เพื่อที่จะระบุแนวโน้มส่วนใหญ่ภายในชุดของข้อมูล เช่น ยีนที่จะถูกระบุตำแหน่ง หรือ บริเวณที่ถูกทำนาย ดังแสดงกลวิธีในภาพที่ 3-4

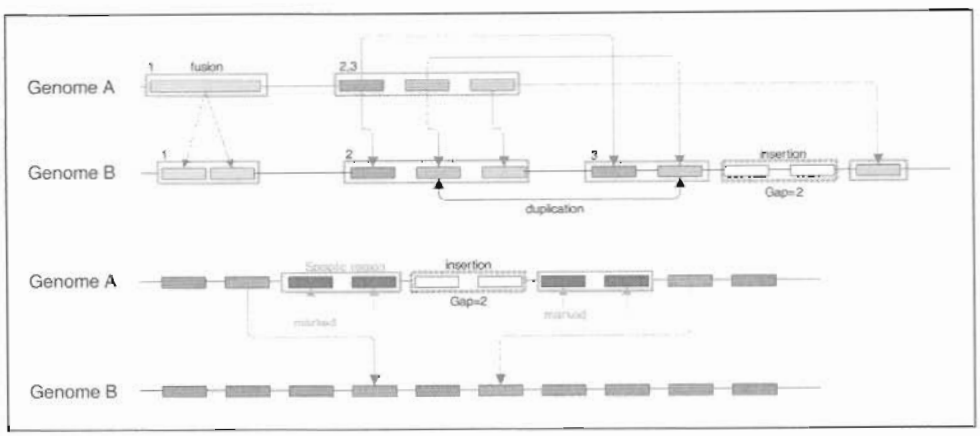
3.3.2.2 การรวมผลลัพธ์จากโปรแกรมสำหรับค้นหา "บริบทของยีน" (Gene Context) ด้วยวิธีการค้นหา บริเวณเขตสงวนการสังเคราะห์ (Conservative of Synthesis Regions) และทำการสร้าง Metabolic Pathway โดยแนวคิดการหายีนด้วยวิธีนี้ ใช้กับโปรตีนที่ยังไม่ทราบหน้าที่ๆ ชัดเจน แต่ทราบหน้าที่ของโปรตีนการศึกษาเชิงเปรียบจากจีโนมอื่นๆ ที่มีบรรพบุรุษเดียวกัน ดังนั้น MaGe จึงหาคู่ของโปรตีนที่ Orthologous (มี Orthologs) สามารถตรวจว่าโปรตีนนั้นมี Orthologs หรือไม่โดยการเปรียบเทียบ Ortholog I บนจีโนม A กับ Ortholog II บนจีโนม B โดย 1) มาทำการหาความคล้ายคลึง (Similarity Search) จากฐานข้อมูล Proteome B โดยใช้ I เป็นโจทย์ และได้ II เป็นผลลัพธ์ที่ดีที่สุด และ 2) I จะเป็นผลลัพธ์ที่ดีที่สุดเมื่อใช้ II เป็นโจทย์ของ Proteome B ผลลัพธ์ที่ได้เรียกว่า



ภาพที่ 3-4 วิธีคำนวณยีนโมเดลของ AMIGene

Bi-Directional Best Hit หรือ BBH

MaGe นำเสนอวิธีการค้นหายีนด้วยวิธีการศึกษาเชิงเปรียบเทียบที่แตกต่างจากวิธีอื่นๆ โดยนำเสนอยีนที่ Homologous มากกว่า 1 ยีนโดยนำลำดับโปรตีนที่ต้องการศึกษาไป BLAST (ทำ Pairwise Comparison) บนจีโนมอื่นๆ แล้วเอาเฉพาะลำดับโปรตีนที่ Orthologous กันระหว่าง 2 จีโนม ลำดับต่อไปจะหากลุ่มของยีน (Gene Cluster) ที่ถูกอนุรักษ์ เช่น กลุ่มของยีนในกลุ่มของการสังเคราะห์ในหลายๆ จีโนมแบคทีเรีย โดยวิธีนี้เรียกว่า Syntonizer ด้วยวิธีนี้ทำให้สามารถหายีนที่มีความสัมพันธ์ได้หลากหลายประเภท เช่น ความสัมพันธ์แบบ Paralogy หรือ ยีนฟิวชั่น (Gene Fusion) ได้ง่าย ดังภาพที่ 3-5 จากรูป (A) คือการตรวจจับกลุ่มของยีนที่ถูกสงวนไว้หรือ Synteny Group ด้วยโปรแกรม Syntonizer ซึ่งใช้สำหรับหายีนที่มีความสอดคล้องได้หลายรูปแบบ (ลูกศรสีแดงคือผลลัพธ์ที่ได้จากโปรแกรม BLASTP) เพื่อที่จะตรวจจับยีนที่มีความเหมือนกันหรือยีนที่ทำสำเนา และยีนฟิวชั่นหรือยีนฟิสชั่น นอกจากนี้การแทรกช่องหรือแก๊ป หรือการลบก็ใช้ได้ในโมเดลนี้ แก๊ปคือยีนที่ต่อเนื่องกันซึ่งไม่เกี่ยวข้องกับกลุ่มของที่ถูกสงวนไว้ หมายเลข 1 คือยีนฟิวชั่นที่เกิดขึ้นบนจีโนม A หมายเลข 2 คือยีนที่เปรียบเทียบระหว่างจีโนมแล้วพบว่า กลุ่มของยีนที่ถูกสงวนไว้มีการเรียงตัวเป็นลำดับที่เหมือน



ภาพที่ 3-5 การแสดงกลุ่มของยีนที่ถูกสงวนไว้หรือ Synteny Group

กันทุกประการ หมายเลข 3 คือผลลัพธ์ของการทำสำเนาที่เกิดขึ้นบนจีโนม B; จากรูป (B) แสดงตัวอย่างบริเวณจำเพาะที่มีสีม่วงบนจีโนม A เนื่องจากไม่สามารถเปรียบเทียบเจอยีนที่เป็น Ortholog บนจีโนม B เพราะว่าการขาดความสัมพันธ์ที่สอดคล้อง และได้ทำการแทรกยีน 2 แก๊ปให้จีโนม A ด้วยวิธีเชิงเปรียบเทียบอย่างนี้จะคำนวณกลุ่มต่างๆ ที่เกี่ยวข้องกับการสังเคราะห์ทุกรูปแบบ ดังที่ได้กล่าวมาก่อนหน้านี้ บนฐานข้อมูลอย่าง NCBI ผลลัพธ์ที่ได้จากการเปรียบเทียบ จะถูกนำมาพิจารณา

3.3.2.3 การทำงานแบบโปรเจคมีผู้ใช้ช่วยกันทำงานระบุตำแหน่งและกำหนดหน้าที่ของยีน และผู้ร่วมงานสามารถแก้ไขผลลัพธ์โดยอัตโนมัติ

นอกจากนี้ MaGe ยังใช้โปรแกรม RBSfinder, tRNAscan-SE, Petrin เพื่อที่จะระบุว่ายีนที่หาเริ่มถอดรหัสบริเวณไหน (เนื่องจากสำหรับแบคทีเรียยีนมักจะเหลื่อมกัน ทำให้ระบุจุดเริ่มต้นการถอดรหัสได้ยาก)

3.4 กรณีศึกษาระบบการระบุตำแหน่งและกำหนดหน้าที่ในยูคาริโอต

3.4.1 กรณีศึกษาระบบการระบุตำแหน่งและกำหนดหน้าที่ Ensembl เป็นโครงการของสถาบันชีวสารสนเทศแห่งยุโรปหรือ EMBL - European Bioinformatics Institute (EBI) กับสถาบันแซงเกอร์ทรัสต์เวลคัมหรือ Wellcome Trust Sanger Institure (WTSI) มีวัตถุประสงค์เพื่อที่จะพัฒนาระบบโปรแกรมสำหรับสร้างและดูแลรักษาข้อมูลการระบุตำแหน่งยีนแบบอัตโนมัติบนจีโนมยูคาริโอตในตอนแรกเขียนขึ้นมาเพื่อสร้างจีโนมมนุษย์อย่างคร่าวๆ (Draft Human Genome) ในแต่ละโคลน แต่ก็สามารถนำมาใช้ได้ดีกับการลำดับเบสทั้งจีโนมโดยใช้เทคนิค Shotgun กับหนูตัวโต (Mouse) กับหนูตัวเล็ก (Rat) และ *Anopheles gambie* Ensembl มีส่วนต่างๆ ประกอบด้วย 1) ส่วนเก็บและรับข้อมูลในระดับจีโนม 2) แสดงผลผ่านเว็บ 3) การระบุตำแหน่งยีนแบบอัตโนมัติโดยฮิวริสติค (Heuristic) กับมนุษย์, หนูตัวใหญ่, หนูตัวเล็ก, ยุง, ม้าน้ำ, *Fugu rubripes* และ *C. briggsae* กระบวนการระบุตำแหน่งยีนแบบอัตโนมัติของ Ensembl เริ่มจาก "คำนวณข้อมูลดิบ (Raw Compute)" โดย Ensembl ใช้โปรแกรม RepeatMasker, Genscan, tRNAscan, Eponine และ BLAST คำนวณข้อมูลดิบ (สาย

ลำดับที่สร้างขึ้นมาใหม่จากขั้นตอน Assembly) แล้วผลลัพธ์ที่ได้จะถูกเก็บในฐานข้อมูล Ensembl เพื่อที่จะดูผลลัพธ์ โครงสร้างยีน, Gene Family, Gene Expression และ Gene Ontologies ผ่านเว็บ

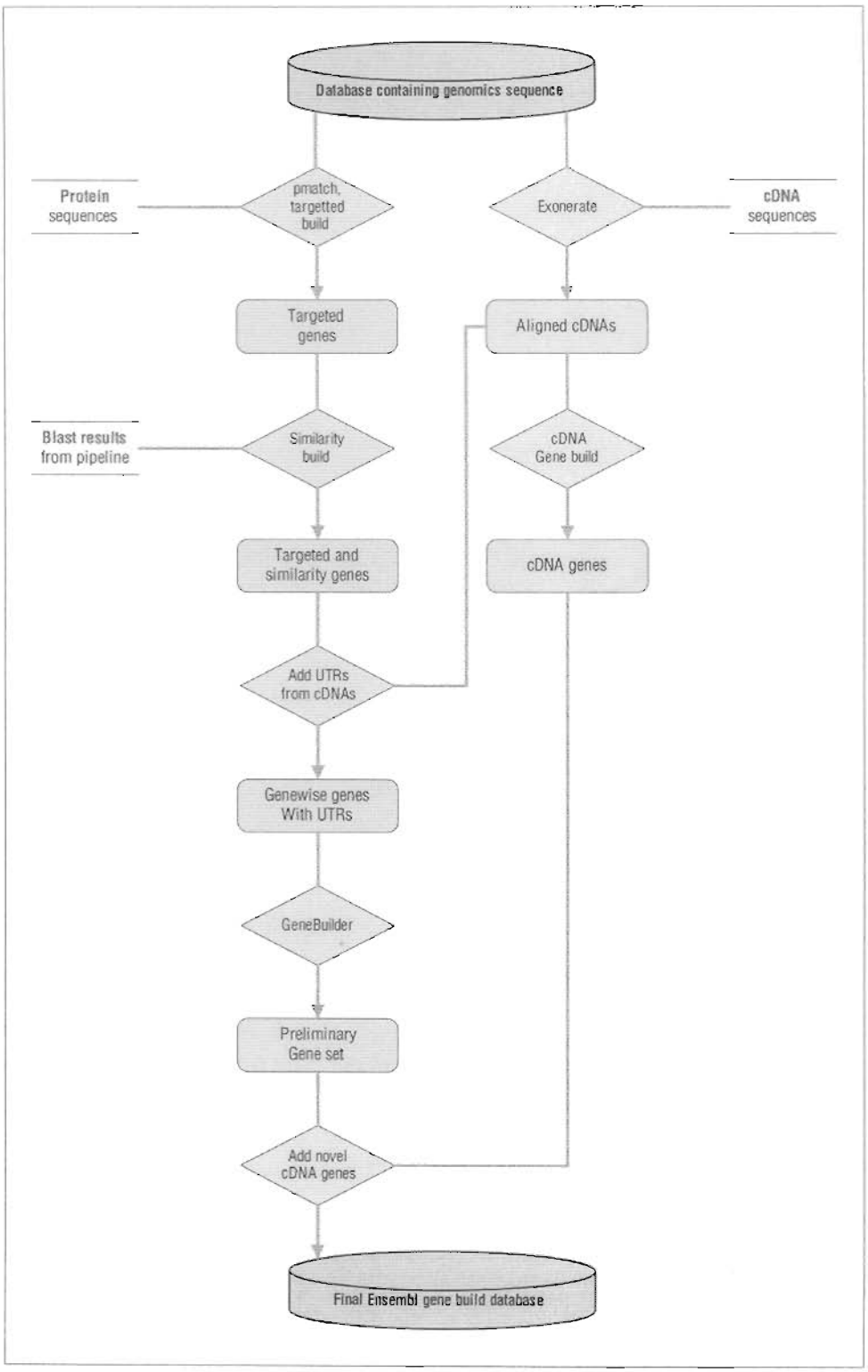
Ensembl ใช้แนวคิด 2 แบบทำการระบุตำแหน่งยีนกับจีโนมขนาดใหญ่ แบบแรกทำการลดขนาดของจีโนมที่จะค้นหา โดยเริ่มต้นการวิเคราะห์ด้วยการกวาดดูตลอดทั้งจีโนมโดยคร่าวๆ อย่างรวดเร็วด้วยวิธีนี้ก็จะได้ลำดับเบสที่ค่อนข้างสั้นได้อย่างแม่นยำและการวิเคราะห์ว่าลำดับเบสที่ค่อนข้างสั้นอย่างละเอียดต่อไป แบบที่สองคือ Ensembl จะใช้บางขั้นตอนการสร้างยีนมาใช้ บนชิ้นส่วนโครโมโซม ซึ่งจะยาวประมาณ 1-5 ล้านเบส และเรียกชิ้นส่วนโครโมโซมนี้ว่า "Slices" แต่ละชิ้น Slices นอกจากนี้ Ensembl มีกลเม็ดจัดการลักษณะผลลัพธ์, Exon หรือ ยีน,ที่ได้ในระดับ โคลน, คอนทีก, โครโมโซม ทำให้การเชื่อมโคลนและลักษณะของโคลนได้อย่างไร้รอยต่อ และ ดึงข้อมูล Exon, Genes ที่ได้จากฐานข้อมูลมาสร้างจุดเชื่อมระหว่าง Slices A กับ Slices B ด้วยวิธีนี้ทำให้ Ensembl สามารถแก้ไขยีนที่อยู่นอกบริเวณจุดเชื่อมต่อให้เข้ามาอยู่ในจุดเชื่อมต่อได้ โดยมีข้อแม้ว่าตำแหน่งของยีน ต้องอยู่เหนือจุดเชื่อมต่อ มิฉะนั้นจะเอายีนนั้นออกไป นอกจากนี้วิธี Whole-Genome Shotgun Assemblies (WGS) ยังเป็นที่น่าสนใจว่าจะสามารถนำไปใช้กับจีโนมที่ลำดับเบสเป็นครั้งแรกได้ดีเพียงใด เนื่องจากในการประกอบครั้งแรกคอนทีกจะยังมีชิ้นส่วนเล็กๆ อีกมากที่ประกอบผิดในกรณีนี้ สามารถใช้วิธีข้างต้นสั่งสร้างยีนให้อยู่บน Slices ที่กำหนดแล้วลองแก้ไขผลลัพธ์ที่ได้ว่าวิธีนี้ของ Ensembl ดีเพียงใด

นอกจากนี้ Ensembl ใช้เครื่องมือการทำนายยีนจำพวก *Ab Initio* หรือ การค้นหายีนจากภายในที่ใช้เฉพาะข้อมูลบนจีโนมเอง มายืนยันโครงสร้างยีนที่สร้างขึ้นมา และใช้ผลลัพธ์ที่ได้จากโปรแกรม BLAST มาช่วยในการทำนาย Exon เนื่องจากถึงแม้ว่าโปรแกรมที่ดีที่สุดอย่าง Genscan ก็ยังมีการทำนายยีนเกิน และมักจะหาพลาดเมื่อ Exon มีขนาดเล็ก ดังนั้นจึงนำโปรแกรม BLAST ที่สามารถระบุตำแหน่งของโปรตีนบน cDNA บนจีโนม มาค้นหายีนที่เหมือนกันหรือคล้ายคลึงกัน แต่อย่างไรก็ดี BLAST ก็ไม่เหมาะนำไปทำนายยีน เพราะโปรแกรม BLAST ไม่มีแบบจำลองสำหรับบริเวณตัด-แต่ง (Splice Sites) ทำให้โปรแกรม BLAST อย่างเดียวไม่สามารถระบุตำแหน่งยีนได้อย่างสมบูรณ์แบบ Ensembl จึงนำผลลัพธ์ที่ Homologous และผลลัพธ์อื่นๆ มาช่วยทำนายโครงสร้างยีนให้แม่นยำขึ้น ด้วยวิธีรวมข้อมูลโปรตีนจากหลายแหล่งข้อมูลที่เป็นอิสระต่อกันเพื่อที่จะทราบโครงสร้างการถอดรหัสที่สมบูรณ์

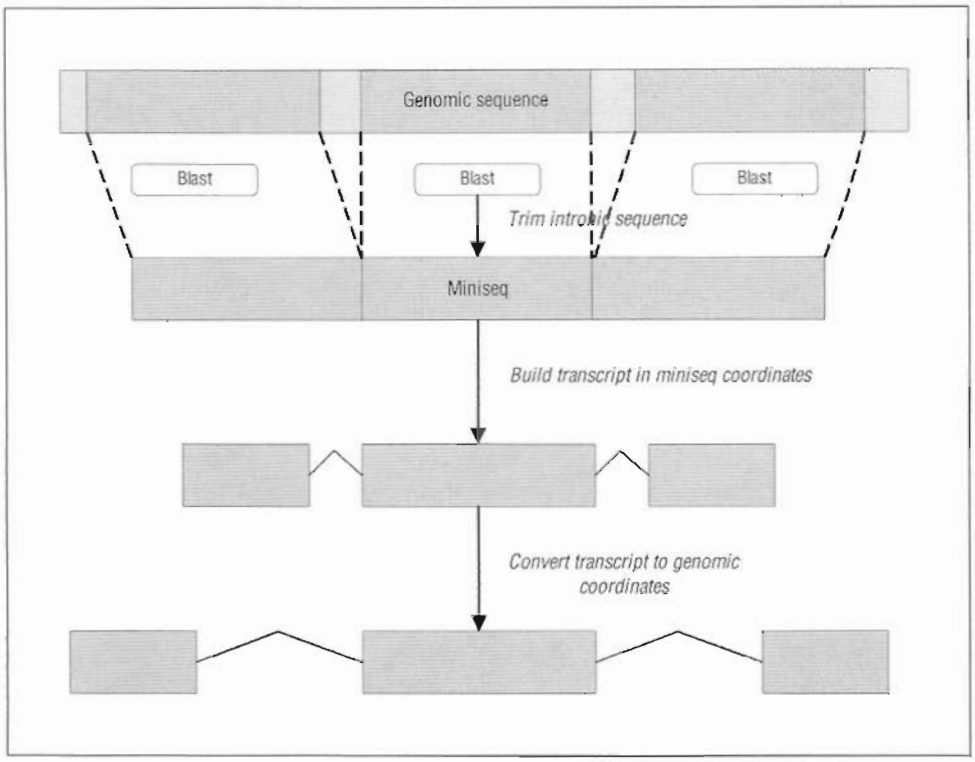
กระบวนการตัดสินใจที่สำคัญในการระบุตำแหน่งก็คือ การเลือกแหล่งข้อมูลสำหรับค้นหาความเหมือน โดย Ensembl เลือกใช้ข้อมูล cDNA และโปรตีนเฉพาะสกุล (Species-Specific Protein) แต่อย่างไรก็ดีข้อมูลของโปรตีนที่ Homology ก็มีประโยชน์ ดังนั้น Ensembl จึงได้แบ่งระดับความสำคัญก่อนหลัง ให้โปรตีนเฉพาะสกุลสูงกว่าโปรตีนที่ Homology โดย Ensembl จะระบุตำแหน่งโปรตีนเฉพาะสกุลและ cDNA บนจีโนมก่อน เพื่อที่จะสร้างแบบจำลองการถอดรหัส (Transcript Model) ส่วนโปรตีนพบจากสกุลอื่นก็ถูกใช้ด้วยเพื่อระบุการถอดรหัสที่ยังไม่พบบนจีโนมมาก่อนหน้านี้ หลังจากนั้น Ensembl จะทำการรวมโปรตีนที่หามาได้และ cDNA เข้าด้วยกัน เพื่อให้ได้การถอดรหัสพร้อม UTR (Untranslated Region) ส่วนข้อมูลที่ซ้ำกันจะละไว้ และยีนจะถูกสร้างโดยใช้การถอดรหัสเป็น

หลัก กระบวนการต่างๆ ของ Ensembl แสดงไว้ในภาพที่ 3-6 หลังจากการคำนวณข้อมูลดิบ ก็จะนำโปรตีนที่ทราบหน้าที่การทำงานและนำ cDNA ที่ได้จากจีโนมที่สนใจมาหาตำแหน่งบนจีโนม ขั้นตอนนี้เกี่ยวข้องกับการจัดเรียงตำแหน่งของโปรตีนและ cDNA ที่มีบริเวณตัดแต่งถูกต้องและ การถอดรหัสที่เชื่อมโยงเข้าไว้ด้วยกัน ขั้นตอนเหล่านี้ประกอบด้วย

3.4.1.1 การจัดวางโปรตีนบนจีโนม (Targeted Protein Alignments) Ensembl ใช้โปรแกรม Pmatch นำเอาโปรตีนไปจัดเรียงบนจีโนมโดยโปรแกรม Pmatch จะใช้ฐานข้อมูลโปรตีนจาก SwissProt/TrEMBL โดยหลักการแล้ว Pmatch จะค้นหา Exon ในโปรตีนของฐานข้อมูลที่ตรงกัน และใช้โปรแกรม Genewise สร้างโครงสร้างการถอดรหัส (Transcript Model) โปรแกรม Pmatch เป็นเพียงขั้นแรกในจัดเรียงโปรตีนเพราะว่า Pmatch ทำให้จีโนมจาก 3 Gb เหลือเพียง 1Mb (เป็นการลดขนาดจีโนมลง) โปรแกรม Genewise ถึงจะเป็นการจัดเรียงโปรตีนขั้นสุดท้าย Genewise จะทำงานในระดับโปรตีนและเผื่อไว้สำหรับบริเวณตัด-แต่งและการเลื่อนตำแหน่งการอ่านลำดับเบส (Frameshift) แต่อย่างไรก็ดี Genewise ก็มีข้อเสียถ้ายิ่งขนาดของจีโนมมากเท่าไหร่ ก็ยิ่งทำเสร็จช้าเท่านั้น Ensembl จึงใช้โปรแกรม BLAST ช่วยโดยการลดขนาดลง ด้วยการเอาชิ้นส่วนทั้งหมดที่พบโดย Pmatch ไป BLAST ฐานข้อมูลโปรตีน แล้วเพิ่มความยาวชิ้นส่วนจาก Pmatch โดยเอาผลจากโปรแกรม BLAST ที่ใช้ได้ (ที่ Hit) แล้วเอาลำดับเบสยาวประมาณ 200 bp ไปใช้ เพราะว่าจะได้เผื่อบริเวณตัด-แต่ง หลังจากนั้นชิ้นส่วนต่างๆ ที่ได้จาก BLAST ประมาณ 200 bp ทั้งหมดต่อเพื่อย่อลำดับเบสบนจีโนม ให้มีเฉพาะ Exon และ Intron ซึ่ง Ensembl เรียกว่า miniseq ในภาพที่ 3-7 ทำให้โปรแกรม Genewise ทำงานได้เพราะขนาดที่ Genewise จะทำงานลดลง ด้วย 3 ขั้นตอน 1) วางตำแหน่งยีนอย่างคร่าวๆ 2) วางตำแหน่ง Exon และ 3) การจัดเรียงท้ายสุด ช่วยลดเวลาทำงานเหลือเพียงสองสามชั่วโมง แต่อย่างไรก็ดีวิธีนี้ก็ทำให้ขาด Exon ที่มีขนาดเล็กไปเพราะว่าไม่ได้ใช้จีโนมทั้งหมด ซึ่งใช้เพียงบางส่วนเท่านั้น แนวคิด miniseq ของ Ensembl ไม่ได้ใช้เฉพาะโปรแกรม Genewise เท่านั้นแต่ใช้ได้กับโปรแกรมที่ต้องการลดขนาดจีโนมลงอย่าง Genewise ทำให้เพิ่มความเร็วได้ถ้าใช้กับ ขนาดจีโนมทั้งหมดรวมถึง est_genome และ Genomewise



ภาพที่ 3-6 ภาพรวมวิธีค้นหายีนของ Ensembl



ภาพที่ 3-7 Miniseq: miniseq เป็นตัวแทนของลำดับเบสทั้งหมดบนจีโนมจากหลายสถานะ

3.4.1.2 Similarity Alignments ในขั้นตอนนี้จะเริ่มจากเอาโปรแกรม RepeatMasker, Genscan, tRNAscan และ Eponine มาใช้กับโปรแกรม BLAST เพื่อค้นหาเปปไทด์จากโปรแกรมประเภท *Ab Initio* แล้วเลือกผลลัพธ์ที่ไม่ซ้ำกับขั้น Targeted หลังจากนั้นก็ใช้ BLAST หาโปรตีนเพื่อสร้าง miniseq จากภาพที่ 3-7 และใช้ Genewise หาโดเมนยีนหรือสร้างโครงสร้างการถอดรหัสเหมือนขั้นตอนที่แล้ว

3.4.1.3 DNA Alignments โดย Ensembl ใช้ฐานข้อมูล cDNA ในหลากหลายกรณี เช่น จีโนมมนุษย์ใช้ cDNA จาก EMBL และ RefSeq สำหรับจีโนมหนูใช้ชุดข้อมูล FANTOM2 นอกจากนี้ Ensembl ใช้โปรแกรม Exonerate ซึ่งช่วยให้การจัดเรียงของ cDNA บนจีโนมทำได้อย่างรวดเร็วในขั้นตอนเดียว โดยกำหนดค่า Coverage ที่ 90% และ Identity ที่ 97% และ Ensembl ละยีนที่น่าจะเป็น Pseudogene โดยถ้า ตัวที่มีการจัดเรียงที่ดีที่สุดเกิด Spliced

3.4.2 กรณีศึกษาระบบค้นหายีนและระบุตำแหน่ง RiceGAAS ข้าวเป็นพืชต้นแบบสำหรับใช้ค้นหายีน ในกลุ่มธัญพืชด้วยกัน เนื่องจากข้าวมีขนาดของจีโนมเล็กที่สุด ในขณะที่จีโนมของธัญพืชอื่นๆ มีขนาดใหญ่โตมาก เช่น ข้าวโพดมีประมาณ 7 เท่าของข้าว และข้าวสาลี 40 เท่าของข้าวดังแสดงในตารางที่ 3-1 ดังนั้นโอกาสที่จะมีโครงการจีโนมของพืชเหล่านี้จึงเป็นไปได้น้อย เพราะต้องลงทุนสูงพอๆ กับโครงการจีโนมมนุษย์ แต่เนื่องจากธัญพืชเหล่านี้มีวิวัฒนาการร่วมกับข้าวมา ดังนั้นจึงมีการกล่าวถึงความเป็นไปได้ที่จะใช้จีโนมข้าวสำหรับอ้างอิง (Reference Genome)

โครงการวิจัยจีโนมข้าวนานาชาติ (International Rice Genome Sequencing Project; IRGSP)

ตารางที่ 3-1 ขนาดของจีโนมจำพวกธัญพืช

ธัญพืช	ขนาดของจีโนม (ล้านเบส)
ข้าว	430
ข้าวฟ่าง	1,000
ข้าวโพด	3,000
ข้าวบาร์เลย์	5,000
ข้าวสาลี	16,000

ประกอบด้วยประเทศสมาชิกจำนวนทั้งสิ้น 10 ประเทศ คือ ญี่ปุ่น (ประเทศผู้นำกลุ่ม) สหรัฐอเมริกา อังกฤษ บราซิล ฝรั่งเศส จีน ไต้หวัน เกาหลี อินเดียและประเทศไทย โดยใช้ข้าวสายพันธุ์นิพพอนบาเร [ข้าวจาปอนิกาหรือข้าวญี่ปุ่น] เป็นต้นแบบในการศึกษา [สืบเนื่องจากญี่ปุ่นเป็นผู้เริ่มทำโครงการจีโนมข้าว โดยได้รับเงินจาก The Japan Racing (Horse) Association ในปี พ.ศ. 2534] 2) กลุ่มบริษัทเอกชน คือ บริษัทมอนซานโต้และบริษัทซินเจนต้า ใช้ข้าวสายพันธุ์นิพพอนบาเรเป็นต้นแบบในการศึกษา 3) ประเทศจีน โดย Beijing Genomics Institute ใช้ข้าวสายพันธุ์อินดิกาเป็นต้นแบบในการศึกษา ในขณะนี้มีข้อมูลจีโนมข้าวที่ The Rice Genome Research Program (RGP; ดู URL ได้ในตารางที่ 3-2) > 380Mb โดยสามารถดาวน์โหลดข้อมูลผ่านฐานข้อมูลดีเอ็นเอแห่งชาติญี่ปุ่น (DDBJ) โดย RiceGAAS มีวัตถุประสงค์เพื่อรวบรวมข้อมูลโดยเฉพาะอย่างยิ่งบนจีโนมข้าว

ลักษณะหน้าที่ RiceGAAS ประกอบไปด้วย 1) RiceGAAS รวบรวมข้อมูลจีโนมข้าวจากเว็บไซต์ GenBank ไว้ที่ RiceGAAS, 2) รวมกลุ่มโปรแกรมทำนายยีน (Gene Prediction) และโปรแกรมค้นหาความเหมือน (Homology Search) มาใช้ร่วมกัน, 3) RiceGAAS นำผลลัพธ์จากการวิเคราะห์โปรแกรมมาตีความบริเวณโค้ดยีนแบบอัตโนมัติ, 4) ทำการวิเคราะห์โปรแกรมทำนายยีนและโปรแกรมค้นหาความเหมือนอีกครั้งพร้อมทั้งแก้ไขปรับปรุงฐานข้อมูลที่ใช้อ้างอิงให้ทันสมัย ด้วยการตีความและการทำงานแบบอัตโนมัติ, 5) ท้ายสุดนี้ RiceGAAS นำเสนอข้อมูลกราฟฟิกผ่านเว็บ และ RiceGAAS สามารถใช้งานได้ที่ http://RiceGASS/dna.affrc.go.jp/.

ฐานข้อมูลของ RiceGAAS เก็บข้อมูลต่างๆ ประกอบไปด้วย

1. ข้อมูลจีโนมข้าวจาก GenBank

2. ข้อมูลที่ได้จากการวิเคราะห์ด้วยโปรแกรมสำหรับการทำนายยีน โปรแกรมการค้นหายีนบนฐานข้อมูลโปรตีนและฐานข้อมูลข้าว ESTs โปรแกรมทำการวิเคราะห์ Exon, บริเวณตัด-แต่ง (Splice Sites), บริเวณซ้ำ (Repeats) และ tRNA

3. ข้อมูลยีนที่ถูกทำนายและ Long Terminal Repeats (LTRs) ด้วยระเบียบวิธีแบบผสมจากหลากหลายโปรแกรม อาทิเช่น โปรแกรมการทำนายยีนร่วมกับผลลัพธ์ที่ได้จากการโปรแกรมค้นหาความเหมือน

4. นอกจาก RiceGAAS เก็บข้อมูลจีโนมข้าว ข้อมูลจากการวิเคราะห์ ข้อมูลยีนที่ถูกทำนายแล้ว ยังเก็บรายละเอียดต่างๆ ที่ใช้สำหรับทำการวิเคราะห์ยีน เช่น ข้อมูลความยาวของลำดับเบสที่กำหนด

ตารางที่ 3-2 โปรแกรมและลิงค์เชื่อมโยงข้อมูลที่มีของ RiceGAAS

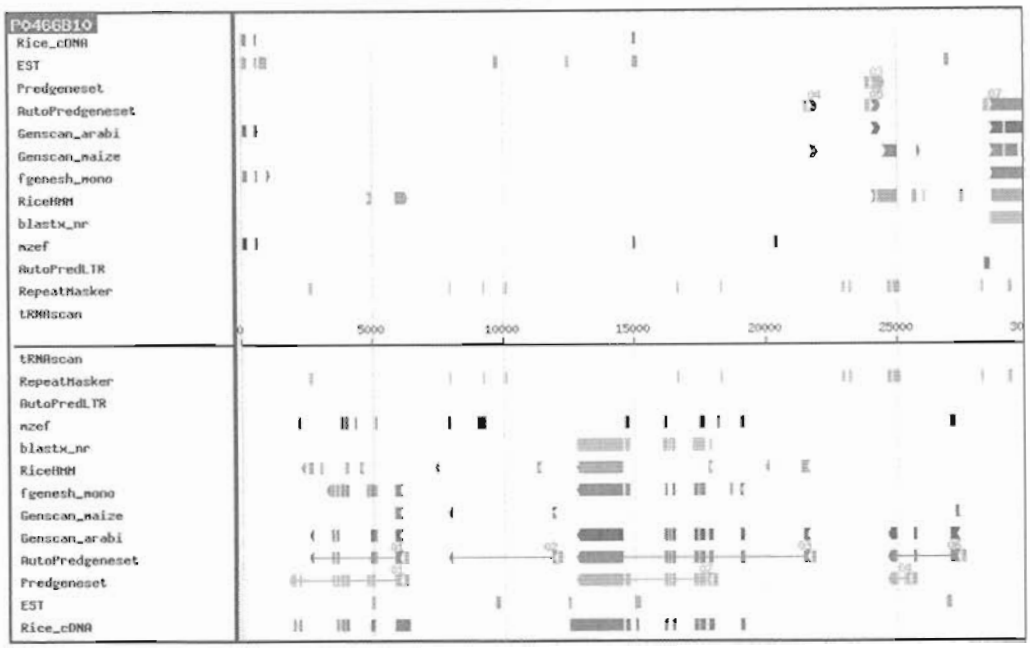
RiceGAAS	http://RiceGAAS.dna.affrc.go.jp
IRGSP	http://rgp.dna.affrc.go.jp/cgi-bin/statusdb/seqcollab.pl
RGP home page	http://rgp.dna.affrc.go.jp/
INE	http://rgp.dna.affrc.go.jp/giot/INE.html
Rice Genome Annotation	http://ricegaas.dna.affrc.go.jp/rgadb/
BLAST	http://www.ncbi.nlm.nih.gov/BLAST/
GENSCAN	http://genes.mit.edu/GENSCAN.html
RiceHMM	http://rgp.dna.affrc.go.jp/RiceHMM/index.html
MZEF	http://argon.cshl.org/genefinder/
SplicePredictor	http://bioinformatics.iastate.edu/cgi-bin/sp.cgi
printrepeats	http://www.sanger.ac.uk/Software/sequencing/docs/printrepeats/
RepeatMasker	http://ftp.genome.washington.edu/
tRNAscan	http://www.genetics.wustl.edu/eddy/tRNAscan-SE/
HMMER	http://pfam.wustl.edu/hmmsearch.shtml
ProfileScan	http://www.isrec.isb-sib.ch/software/PFSCAN_form.html
MOTIF	http://www.motif.genome.ad.jp/MOTIF3.html
PSORT	http://psort.ims.u-tokyo.ac.jp/
SOSUI	http://sosui.proteome.bio.tuat.ac.jp/sosuiframe0.html
PLACE-SignalScan	http://www.dna.affrc.go.jp/htdocs/PLACE/signalscan.html
MAFF DNA Bank	http://www.dna.affrc.go.jp/
ตารางการเปรียบเทียบการทำนายยีน	http://ricegaas.dna.affrc.go.jp/rga-bin/col_ accur.pl

หรือข้อมูลแบบจำลองยีนที่ใช้สำหรับหายีนด้วย เป็นต้น

เนื่องจากฐานข้อมูลมีการเก็บข้อมูลหลายแบบ RiceGAAS จึงแสดงผลการระบุตำแหน่งของในจีโนมข้าว ผ่านหน้าเว็บ 3 แบบด้วยกันประกอบไปด้วย

1. หน้าเว็บแสดงผลของตาราง Bacterial Artificial Chromosome (BAC) และ P1-Derived Artificial Chromosome (PAC) ของแต่ละโครโมโซม ในตารางจะบอกถึงชื่อโคลน หมายเลข Accession ตำแหน่งของโครโมโซม ขนาดโคลน วันและเวลาที่ลำดับเบสในโคลนถูกวิเคราะห์ นอกจากนี้ผลลัพธ์ที่ได้จากแต่ละโคลนสามารถค้นหาด้วยหมายเลขโคลนหรือ หมายเลขตำแหน่งของโครโมโซมจากหน้าฐานข้อมูลการระบุตำแหน่งจีโนมข้าว (Rice Genome Annotation) โดยสามารถเลือกดูผ่านเว็บในตารางที่ 3-2

2. แผนที่การระบุตำแหน่งของแต่ละโคลน (Clone) จะลิงค์จากชื่อโคลนโดย RiceGAAS จะแสดงผลลัพธ์การวิเคราะห์ทั้งหมดจะรวมอยู่ในหน้าแผนที่การระบุตำแหน่งของโคลนนั้น ดังแสดงในรูปภาพที่ 3-8 ประกอบด้วย โปรแกรม Predgeneset แสดงยีนจาก GenBank ที่ถูกระบุตำแหน่งแบบไม่อัตโนมัติ AutoPredgeneset แสดงยีนที่ถูกทำนายจาก RiceGAAS แบบอัตโนมัติ ส่วนจุดเริ่มต้นและจุดหยุดการแปลรหัส (Translation Initiation and Termination) แทนด้วยหางลูกศรและหัวลูกศรตาม



ภาพที่ 3-8 ภาพระบุตำแหน่งยีนบนโคลนหมายเลข P0466B10

ลำดับ ส่วนยีนที่ถูกทำนายด้วย Predgeneset หรือ AutoPredgeneset เมื่อกดเข้าไปจะเชื่อมสู่หน้ายีนที่ถูกทำนายนั้นๆ นอกจากนี้ผลลัพธ์ที่ได้จากโปรแกรม Genscan_arabi, Genscan_maize RiceHMM และโปรแกรมค้นหาความเหมือนอย่าง BLAST ก็แสดงอยู่ในหน้าแผนที่การระบุตำแหน่งด้วย ส่วน LTR แสดงถึง LTR ที่ถูกทำนายอยู่ใน GenBank ดังนั้น AutoPredLTR คือการทำนาย LTR โดย RiceGAAS แบบอัตโนมัติ RepeatMasker แสดงลำดับเบสที่ซ้ำกันด้วยโปรแกรม RepeatMasker ชื่อโปรแกรมต่างๆ ที่แสดงในหน้านี้สามารถเลือกได้เพื่อหาข้อมูลเพิ่มเติม

3. ข้อมูลสำหรับยีนที่ถูกทำนายจาก RiceGAAS ผลลัพธ์จากโปรแกรมต่างๆ 14 โปรแกรม ประกอบไปด้วย โปรแกรม BLAST ใช้ค้นหาความคล้ายคลึง ยังฐานข้อมูลโปรตีนจาก NCBI ด้วย BLASTX กับ BLASTP กับค้นหา EST จากฐานข้อมูลข้าว MAFF หรือ Ministry of Agriculture, Foresty and Fisheries DNA Bank ของธนาคารดีเอ็นเอ MAFF ด้วย BLASTN นอกจากนี้ยังค้นหา *Cis*-Element จากฐานข้อมูล PLACE

ใช้โปรแกรม Genscan กับ RiceHMM สำหรับการทำนายยีนโดเมน (Gene Domain); โปรแกรม MZEF สำหรับการทำนาย Exon; โปรแกรม SplicePredictor สำหรับทำนายบริเวณตัด-แต่ง; โปรแกรม Printrepeats และ RepeatMasker สำหรับตรวจจับลำดับเบสซ้ำ; ส่วนโปรแกรม tRNAscan สำหรับการทำนายทรานส์เฟอร์อาร์เอ็นเอ (Transfer RNA); โปรแกรม HMMER, ProfileScan และ MOTIF ใช้สำหรับค้นหารูปแบบกรดอะมิโนที่คล้ายคลึง โดย HMMER จะค้นหายังฐานข้อมูล Pfam ส่วน ProfileScan ค้นหาฐานข้อมูล PROSITE; โปรแกรม PSORT สำหรับการทำนายตำแหน่งการเกิดโปรตีน โปรแกรม SOSUI สำหรับการจำแนกและใช้สำหรับการทำนายโครงสร้างทุติยภูมิของเนื้อเยื่อโปรตีน โปรแกรม PLACE-SignalScan สำหรับตรวจจับ *Cis*-Element (ดูในตารางที่ 3-2); โปรแกรม

RiceHMM เป็นโปรแกรมที่ถูกพัฒนาขึ้นที่ RGP ใช้ทำนายโดเมนยีนโดยเฉพาะยีนของข้าว โดยสามารถปรับปรุงแบบจำลองยีนได้โดยการคำนวณจาก ESTs ของข้าวที่มี cDNAs ถึงเกือบ 15,000 cDNA

วิธีตีความบริเวณที่เป็นยีนโดยอัตตนมัติของ RiceGAAS

RiceGAAS มีหลักการค้นหายีน โดยทำการรวบรวมข้อมูลการวิเคราะห์จากหลายๆ โปรแกรม และทำการตีความบริเวณที่เป็นยีนซึ่งจัดตำแหน่งกรอบเปิดการอ่านที่ถูกต้อง มาทำการผสมผสานกันอย่างอัตโนมัติ ซึ่งวิธีการทำนายยีนของ RiceGAAS คือการรวบรวมการทำนายยีนจากหลากหลายโปรแกรมกับผลลัพธ์ที่ได้จากการค้นหาความเหมือนจากฐานข้อมูลต่างๆ และ RiceGAAS ยังพยายามทำนายยีนให้มากที่สุดเท่าที่จะมากได้เพื่อเพิ่มความเป็นไปได้ของยีนท่ามกลางยีนที่หามาได้จากโปรแกรมใน RiceGAAS ด้วยวิธี

1. การกำหนดคะแนน Exon จากโปรแกรม Genscan สำหรับ *Arabidopsis*, Genscan สำหรับ *maize* และ RiceHMM โดยขึ้นกับความน่าจะเป็นที่ชิ้นส่วน นั้นๆ น่าจะเป็น Exon โดยยิ่งถ้า Exon ที่ถูกทำนายจากโปรแกรมเหลื่อมกันมากเท่าไหร่ คะแนนของ Exon ก็ยิ่งมากขึ้นเท่านั้น

2. ทำการรวมรวมยีนที่ถูกทำนายซ้ำอีกครั้งซึ่งเหลื่อมกับ LTRs โดยแต่ละ Exon ที่เหลื่อมกับ LTR หลังจากนั้นก็จัดตำแหน่ง Exon ระหว่าง LTRs ใหม่

3. RiceGAAS ทำการเลือกแบบจำลองยีนแล้วเลือกยีนจากค่าคะแนนเฉลี่ยของ Exon ในข้อ 1 วิธีนี้จะไม่ทำนายยีนหลายตัวในแต่ละโดเมน

4. แทรก Exon ที่ทำนายด้วย MZEF โดยถ้า Exon จัดตำแหน่งกรอบเปิดการอ่านอย่างถูกต้อง

ท้ายนี้ RiceGAAS ได้ทำการประเมินโดยเปรียบเทียบยีนที่ RiceGAAS ทำนายโดยอัตโนมัติกับยีนที่ทำนายด้วยมือ โดย Sensitivity เปรียบเทียบจำนวนนิวคลีโอไทด์ (เทียบในระดับนิวคลีโอไทด์) และ Specificity เทียบจำนวนนิวคลีโอไทด์กันในระดับนิวคลีโอไทด์) ปรากฏว่า การทำนายยีนในระดับนิวคลีโอไทด์ระหว่าง RiceGAAS กับยีนที่ทำนายด้วยมือนั้นไม่แตกต่างกัน (ดูในตารางที่ 3-2 ตรงตารางการเปรียบเทียบการทำนายยีน)

จากระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนมของโปรคาริโอตและยูคาริโอตที่ผ่านมา แสดงให้เห็นถึงความเหมือนและความแตกต่าง เช่น วิธีการระบุตำแหน่งที่แตกต่าง ที่แต่ละระบบก็ไม่ได้บอกถึงรายละเอียดที่แท้จริงในการทำงานทั้งหมดให้ทราบ ตลอดจนถึงวิธีกำหนดหน้าที่ของยีนบนจีโนมก็เช่นเดียวกัน นอกจากนี้ในปัจจุบันยังไม่มีโปรเจคเกี่ยวกับระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนที่ครอบคลุมทั้งโปรคาริโอตและยูคาริโอต ทำให้ไม่สามารถนำโปรเจคที่มีอยู่สร้างระบบใหม่ขึ้นมา ในบทที่ 4 จะกล่าวถึงระบบที่สร้างจาก TurboGears ที่จะนำไปสู่การพัฒนาระบบที่ปัจจุบันยังไม่มีโปรเจคเกี่ยวกับการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนมใหม่ ด้วยแนวคิดแบบ Model-View-Controller รวมถึงสถาปัตยกรรมและ รายละเอียดต่อไป

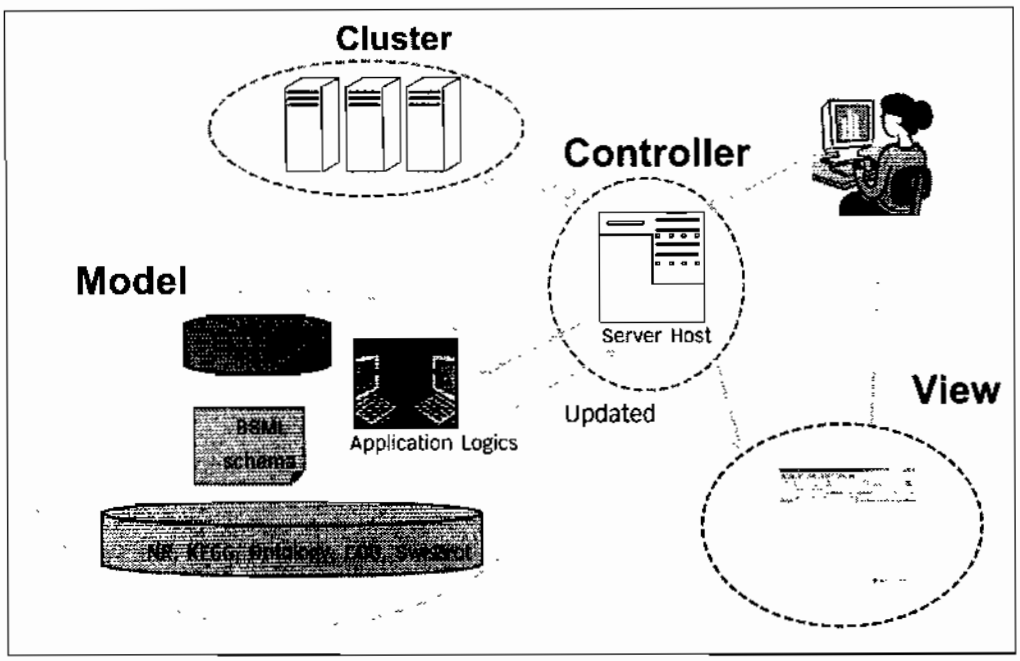
บทที่ 4
การระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนมแบบอัตโนมัติ

จากวิธีการค้นหายีนในบทที่ 2 นำไปสู่การระบุตำแหน่งและกำหนดหน้าที่ในบทที่ 3 และแสดงวิธีการระบุตำแหน่งและกำหนดหน้าที่ กรณีศึกษาในบทที่ผ่านมานั้นพยายามจะให้ข้อมูลของจีโนมที่ทำการศึกษามากที่สุดเท่าที่จะเป็นไปได้ เพราะว่านอกจากจะนำข้อมูลเหล่านั้นไปกำหนดหน้าที่ของยีนแล้ว ยังต้องการข้อมูลที่สามารถนำไปตั้งเป็นสมมุติฐานการวิจัย โดยถึงแม้ว่าไม่สามารถคัดกรองข้อมูลยีนได้อย่างร้อยเปอร์เซ็นต์ แต่ก็ยังมีสมมุติฐานการวิจัยอื่นๆ อีกที่ช่วยหาข้อมูลเพิ่มเติมได้ แสดงว่านอกจากการค้นหายีนแล้วควรจะทำการศึกษาหน้าที่ๆ เกี่ยวข้องของยีนเพื่อเพิ่มคุณภาพของกำหนดหน้าที่ของยีน ยังมีงานวิจัยอีกมากที่ทำการศึกษาวิจัยความสัมพันธ์ของยีน ซึ่งมีรายละเอียดมากเกินกว่าจะอธิบายในที่นี้ ดังนั้นจะลงรายละเอียดในตัวโปรแกรมต่างๆ ที่ถูกนำมาใช้ประกอบการศึกษายีนในจีโนม

ในวิทยานิพนธ์นี้นำเสนอการเชื่อมต่อลำดับเบส การทำนายยีน การระบุตำแหน่งและกำหนดหน้าที่แบบอัตโนมัติซึ่งถูกตรวสอบความถูกต้องของผลลัพธ์โดยนักชีววิทยา โดยรวมขั้นตอนต่อไปนี้เป็นหนึ่งเดียว อันประกอบด้วย 1) การเชื่อมต่อลำดับเบส (Fragment Assembly), 2) การตีความกรอบเปิดการอ่าน (Open Reading Frame Identification), 3) การระบุตำแหน่งและกำหนดหน้าที่กรอบเปิดการอ่านโดยหาความเหมือนในฐานข้อมูลต่างๆ (ORF Similarity Matching), สุดท้ายคือ 4) การตรวจสอบความถูกต้องโดยนักชีววิทยา (Human Verification) นอกจากที่กล่าวข้างต้นนักชีววิทยาสามารถวางกรอบหลักให้ปรับเปลี่ยนได้ตามความเหมาะสม โดยทั้ง 4 ขั้นตอนจะถูกส่งผ่านข้อมูลจากขั้นตอนหนึ่งสู่อีกขั้นตอนหนึ่งในฐานข้อมูลแล้วทำงานร่วมกันตามลำดับผ่าน TurboGears แล้วข้อมูลจะถูกเก็บอยู่ในรูปแบบเอกสาร XML ทำให้แต่ละขั้นตอนทำงานร่วมกันโดยการส่งผ่านข้อมูลได้อย่างถูกต้อง ในบทนี้จะอธิบายขั้นตอนที่วิทยานิพนธ์จะนำเสนอและออกแบบให้สามารถวางกระบวนการทำงานแบบอัตโนมัติ (Automated Workflow Model) ของขั้นตอนทั้งหมดอย่างเป็นระบบ และกล่าวถึงการออกแบบโครงสร้างข้อมูล (Data Model) กับโครงสร้างกระบวนการทำงาน (Process Model) ในระบบฐานข้อมูลที่จะนำเสนอในบทนี้ รวมถึงลักษณะการทำงาน และสถาปัตยกรรมของระบบในหัวข้อถัดไป

4.1 กระบวนการระบุตำแหน่งและกำหนดหน้าที่บนจีโนม

ในภาพที่ 4-1 แสดงโครงร่างการทำงานของระบบในส่วนต่างๆ ประกอบด้วย ระบบคอมพิวเตอร์ ระบบจัดการฐานข้อมูล ระบบควบคุม และระบบแสดงผล ซึ่งทำงานแตกต่างกันขึ้นกับหน้าที่รับผิดชอบ ในบทนี้จะกล่าวถึง การระบุตำแหน่งและกำหนดหน้าที่ซึ่งสัมพันธ์กับการจัดการฐานข้อมูลและระบบ



ภาพที่ 4-1 โครงร่างการทำงานของระบบในส่วนต่างๆ

ควบคุมในภาพที่ 4-1 และส่วนการแสดงผลจะเป็นหน้าที่ของระบบแสดงผล ระบบเครื่องคอมพิวเตอร์จะเกี่ยวข้องสถาปัตยกรรมระบบที่จะกล่าวต่อไปในบทนี้ภายหลัง เพื่อที่จะทำให้ระบบที่สร้างขึ้นทำงานอย่างเป็นระบบในบทนี้จึงแบ่งหัวข้อออกเป็น

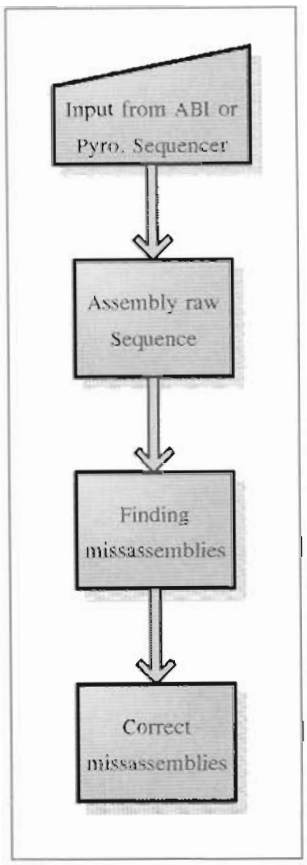
ขั้นตอนการทำงานทำงานในส่วนต่างๆ ประกอบด้วย การเชื่อมต่อลำดับเบส, การตีความกรอบเปิดการอ่าน, การระบุตำแหน่งและกำหนดหน้าที่กรอบเปิดการอ่านโดยหาความเหมือนในฐานข้อมูล, สุดท้ายคือ การตรวจสอบความถูกต้องโดยนักชีววิทยา

การออกแบบๆ MVC รวมถึงวิธีการทำงาน, วิธีการใช้งาน, การวางกระบวนการทำงานแบบอัตโนมัติ และสุดท้ายคือกล่าวถึงสถาปัตยกรรมของระบบ

4.1.1 การเชื่อมต่อลำดับเบส ในการเชื่อมต่อลำดับเบสใหม่ คือการนำลำดับเบสบนจีโนมซึ่งยังไม่เคยทำการระบุตำแหน่งยีนบนจีโนมจากห้องปฏิบัติการ ซึ่งผ่านกระบวนการจากเครื่องลำดับเบสอัตโนมัติ ได้เบสไฟล์และ Quality ไฟล์ของเบสไฟล์นั้น จากเบสไฟล์ที่เป็นตัวแทนลำดับเบสบนจีโนมนำมาผ่านโปรแกรมเชื่อมต่อลำดับเบสจากตารางที่ 1-1 จะได้ผลลัพธ์เป็น Consensus Sequence และได้ Contigs ย่อยซึ่งได้จากการเชื่อมต่อลำดับเบสขึ้นมาใหม่ ในขั้นตอนนี้เราได้แบ่งเป็น 3 ขั้นตอนจากภาพที่ 4-2

4.1.1.1 การคำนวณการเชื่อมต่อลำดับเบสจากข้อมูลดิบ ขั้นตอนนี้เป็นการคำนวณการเชื่อมต่อลำดับเบสขั้นต้นให้ได้มาซึ่ง Consensus Sequence และ Contigs ย่อยสำหรับจีโนมใหม่ที่ยังมีชิ้นส่วนที่เชื่อมต่อผิดอีกมาก ซึ่งอาจจะนำไปสู่การระบุตำแหน่งยีนและกำหนดหน้าที่ๆ ผิดพลาดได้

4.1.1.2 การหาการเชื่อมต่อลำดับเบสที่ผิดพลาด วิธีหาการเชื่อมต่อลำดับเบสผิดพลาดมีอยู่หลายวิธี เช่น หาจากการทดลองในห้องปฏิบัติการ ก็จะนำลำดับเบสที่ผ่านการเชื่อมต่อมาค้นหา



ภาพที่ 4-2 Assembly Pipeline

ความเหมือนจากลำดับเบสได้รับการพิสูจน์ ถ้าได้ตรงตามที่ต้องการ ก็จะเป็นหลักฐานยืนยันว่าเชื่อมต่อลำดับเบสที่ถูกต้อง แต่ในกรณีจีโนมใหม่อาจจะค้นหาความเหมือนจากจีโนมที่ใกล้ชิดกัน เป็นหลักฐานยืนยันได้ในระดับหนึ่ง

วิธีหลักที่จะนำมาใช้ในงานวิทยานิพนธ์นี้คือการเปรียบเทียบการเชื่อมต่อลำดับเบสจากโปรแกรม 2 โปรแกรมขึ้นไปเพื่อใช้เป็นหลักฐานยืนยันคุณภาพการเชื่อมต่อ โปรแกรมที่นำมาใช้เปรียบเทียบได้แก่ CAP3/PCAP และ Phrap โดยโปรแกรมแรกมี 2 แบบที่ทำงานคล้ายกันแต่ต่างกันที่ PCAP สามารถกระจายงานบนคลัสเตอร์ได้แต่ CAP3 ส่วนโปรแกรม Phrap ก็มีหลักการทำงานคล้ายโปรแกรมแรก แต่สามารถหา Consensus Sequence ได้ค่อนข้างยาวกว่าโปรแกรมแบบแรก

การเปรียบเทียบเพื่อที่จะหา 1) หาบริเวณซ้ำ (Repeated Regions) และ 2) หาการเชื่อมต่อลำดับเบสที่ผิดพลาด บริเวณซ้ำเกิดขึ้นบ่อยครั้งโดยเฉพาะอย่างยิ่งกับยูคาริโอต ในโปรคาริโอตมีบ้างแต่น้อยกว่า ดังนั้นทำให้การเชื่อมต่อลำดับเบสสำหรับยูคาริโอตจึงยากกว่า ปัญหาของบริเวณซ้ำก็คือทำให้ลำดับเบสที่หามาได้ไม่ครอบคลุม 100 เปอร์เซ็นต์ นอกจากนี้ก็เป็นสาเหตุที่โปรแกรมเข้าใจบริเวณที่เป็นยีนกับบริเวณซ้ำเป็นอย่างเดียวกัน ทำให้บริเวณที่น่าจะหาได้กลับหาไม่ได้เพราะโปรแกรมจะไม่เชื่อมต่อลำดับเบสที่เป็นบริเวณซ้ำ

การจัดการบริเวณซ้ำของโปรแกรม Phrap จัดการได้ไม่ดีเท่า CAP3/PCAP แต่ขนาด Consensus

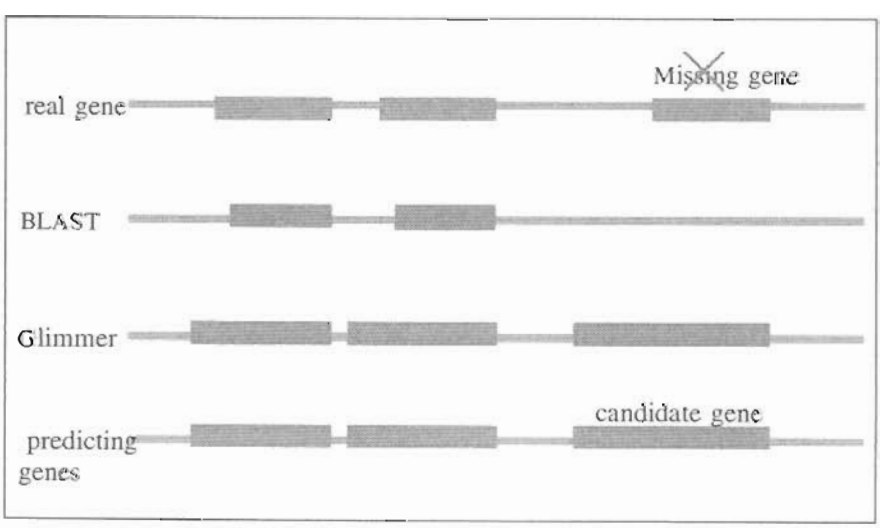
Sequence หรือ Contigs ของ CAP3/PCAP มักจะสั้นกว่า Phrap เมื่อนำทั้ง 2 โปรแกรมมาใช้เพื่อหาการเชื่อมต่อที่ผิดพลาด ย่อมดีกว่าใช้เพียงโปรแกรมเดียว ดังนั้นการเปรียบเทียบโปรแกรมทั้งคู่จึงถูกนำมาใช้ในงานวิทยานิพนธ์นี้เพื่อชดเชยข้อผิดพลาดในแต่ละโปรแกรม

4.1.1.3 การแก้ไขการเชื่อมต่อที่ผิดพลาด โดยขั้นตอนนี้นำผลลัพธ์ที่ได้จากขั้นที่แล้วมาแสดงจุดที่เชื่อมต่อผิดเพื่อที่จะทำการแก้ไขต่อไป การแก้ไขทำได้โดยนำหลักฐานที่ได้ไปแก้ไขบริเวณซ้ำที่ผิดพลาด และเชื่อมต่อลำดับเบสให้มีจำนวน Contigs น้อยที่สุด (โดยน้อยที่สุดคือ 1) ยิ่งจำนวน Contigs เยอะมากเท่าไหร่แสดงให้เห็นว่าโปรแกรมไม่สามารถเชื่อมต่อลำดับเบสได้อย่างสมบูรณ์ แต่อย่างไรก็ดีเราไม่สามารถใช้จำนวนที่โปรแกรมผลิต Contigs เป็นหลัก เช่น โปรแกรม Phrap มีจำนวน Contigs น้อยกว่า CAP3/PCAP แต่ในความเป็นจริงแล้วกลับมีการเชื่อมต่อผิดพลาดมากกว่า

จำนวนครั้งการแก้ไขการเชื่อมต่อที่ผิดพลาดขึ้นอยู่กับความต้องการของผู้ใช้ โดยส่วนมากขึ้นกับจำนวน Contigs ที่ได้ถ้ายอมรับได้ก็จะดำเนินการใช้ขั้นตอนของกระบวนการขั้นถัดไปคือ การตีความกรอบเปิดการอ่าน

4.1.2 การตีความกรอบเปิดการอ่าน ทำนองเดียวกับการเชื่อมต่อลำดับเบส วิทยานิพนธ์นี้นำโปรแกรมที่ใช้สำหรับตีความกรอบเปิดการอ่านหรือโปรแกรมค้นหายีนของยูคาริโอตหรือโปรคาริโอตมากกว่า 1 โปรแกรมมาใช้เพื่อเป็นหลักฐานยืนยันว่า ถ้าโปรแกรมหาผลลัพธ์ที่ได้ตรงกัน ก็จะเป็นหลักฐานดีกว่าผลลัพธ์ที่หาได้จากโปรแกรมเดียว [27] ดังแสดงในภาพที่ 4-3 ได้นำโปรแกรมค้นหายีน 2 แบบ 1) โปรแกรมค้นหาความเหมือนด้วยโปรแกรม BLAST, 2) โปรแกรมค้นหายีนแบบค้นหายีนภายใน (Intrinsic Approach)

4.1.2.1 โปรแกรมค้นหาความเหมือน เช่น BLAST หรือ FASTA โปรแกรมทั้งคู่ใช้หาความเหมือนในฐานข้อมูลต่างๆ ดังนั้นยิ่งฐานข้อมูลมีข้อมูลเยอะ ก็จะสามารถหาความเหมือนได้มาก ในทางกลับกันถ้าฐานข้อมูลนั้นมีข้อมูลน้อย ก็จะหาความเหมือนได้น้อย ดังนั้นฐานข้อมูลที่เราใช้ในวิทยานิพนธ์นี้จึงมีความสำคัญ ฐานข้อมูลที่ใช้ได้แก่ NR และ SwissProt ที่ได้จากเว็บ NCBI



ภาพที่ 4-3 การตีความกรอบเปิดการอ่านหรือ ORF Identification Outline

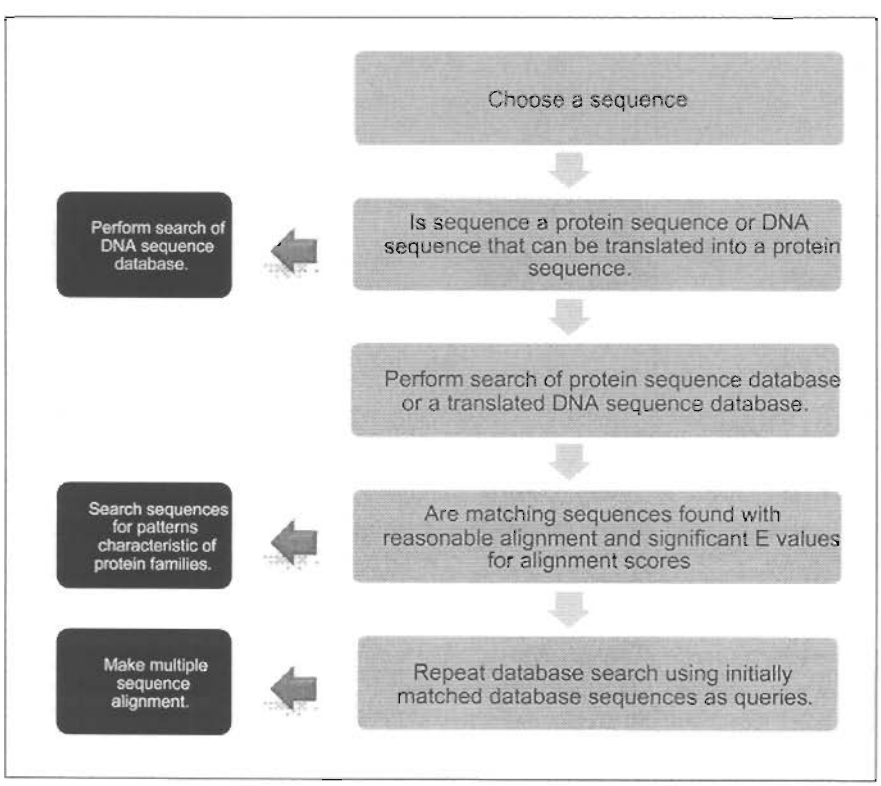
ฐานข้อมูล NR หรือ Non-Redundant เป็นฐานข้อมูลที่เก็บข้อมูลซึ่งไม่ซ้ำกันในเว็บ NCBI ให้โปรแกรมอย่างเช่น BLAST หรือ FASTA ค้นหาความเหมือนของลำดับเบสระหว่าง ลำดับเบสที่ต้องการศึกษากับฐานข้อมูลของชนิดลำดับเบสที่ต้องการค้นหาเช่น ลำดับโปรตีน, ลำดับนิวคลีโอไทด์

ฐานข้อมูล SwissProt เป็นฐานข้อมูลโปรตีนที่เราสามารถดาวน์โหลดได้จากเว็บ NCBI ดังนั้นเราจึงใช้ฐานข้อมูลทั้งคู่นี้มาใช้เพื่อค้นหาความเหมือน แต่อย่างไรก็ดีถึงแม้ใช้ฐานข้อมูลดังกล่าว ถ้าหากไม่มีข้อมูลอยู่บนนั้น ก็จะไม่สามารถหายีนบนกรอบเปิดการอ่านได้

4.1.2.2 โปรแกรมค้นหายีนแบบค้นหายีนภายใน เป็นการหากรอบเปิดการอ่านอีกวิธีหนึ่ง โดยสามารถใช้เพียงแค่โปรแกรมอย่างเดียวก็หากรอบเปิดการอ่านได้ โปรแกรมพวกนี้นิยมใช้พวก Markov Model มาใช้ เช่น Glimmer [28], [25]

ถ้าเรานำโปรแกรมทั้ง 2 แบบมาใช้ดังภาพที่ 4-3 นำโปรแกรม Glimmer และโปรแกรม BLAST มาใช้ร่วมกันเพื่อที่จะชดเชยการใช้โปรแกรมเหล่านั้นเพียงโปรแกรมเดียว โปรแกรม Glimmer ทำให้สามารถหากรอบเปิดการอ่านได้มากเท่าที่มากได้ ส่วนโปรแกรม BLAST จะช่วยยืนยันพิสูจน์ว่าสิ่งที่ Glimmer หามาได้คือบริเวณยีนดังแสดงในภาพที่ 4-3

4.1.3 การระบุตำแหน่งและกำหนดหน้าที่กรอบเปิดการอ่าน ทำได้โดยค้นหาความเหมือนของลำดับเบสซึ่งตรงกันในฐานข้อมูลที่ได้เลือกไว้ เนื่องจากค่าใช้จ่ายที่แพงทำให้ไม่สามารถทำการกำหนดหน้าที่โดยห้องปฏิบัติการได้ทั้งหมด ดังนั้นวิธีนี้จึงช่วยประหยัดและช่วยให้นักชีววิทยาตัดสินใจทำการ



ภาพที่ 4-4 การระบุตำแหน่งกรอบเปิดการอ่าน

ทดลองเพื่อศึกษาลำดับเบสที่ต้องการศึกษาต่อไปจากภาพที่ 4-4 การระบุตำแหน่งเริ่มที่

4.1.3.1 นำลำดับเบสที่ต้องการศึกษาไปแปลงเป็นลำดับโปรตีน ถ้าแปลงไม่ได้ จะนำลำดับเบสไปค้นหาในฐานข้อมูลลำดับเบส

4.1.3.2 นำลำดับโปรตีนที่แปลงไปค้นหาในฐานข้อมูลโปรตีนหรือ ฐานข้อมูลลำดับเบสที่แปลงเป็นฐานข้อมูลโปรตีน เพื่อค้นหาความเหมือน ในกรณีที่ค่าความเหมือนน้อยกว่าที่ตั้งไว้ ก็จะทำค้นหาความเหมือนของโปรตีนในสิ่งมีชีวิตอื่นๆ

4.1.3.3 สุดท้ายคือ การนำลำดับโปรตีนที่ได้ไปค้นหาในฐานข้อมูลโปรตีนอื่นๆ เพื่อค้นหาหน้าที่การทำงานของโปรตีนซึ่งยังไม่เคยค้นพบในสิ่งมีชีวิตที่เราต้องการศึกษา โดยศึกษาหน้าที่การทำงานของยีนที่เกี่ยวข้องกับการสร้างโปรตีนในสิ่งมีชีวิตอื่น แล้วนำยีนที่ได้จากสิ่งมีชีวิตอื่นมาค้นหาความเหมือนกับสิ่งมีชีวิตที่ต้องการศึกษาหน้าที่การทำงานแตกต่างอย่างไร เป็นการศึกษาเพิ่มเติมนอกจากศึกษายีนที่เกี่ยวข้องกับสิ่งมีชีวิตเพียง 1 สิ่งมีชีวิต แล้วทำซ้ำขั้นตอนเพื่อศึกษาเพิ่มเติมหรือเพิ่มความแม่นยำ

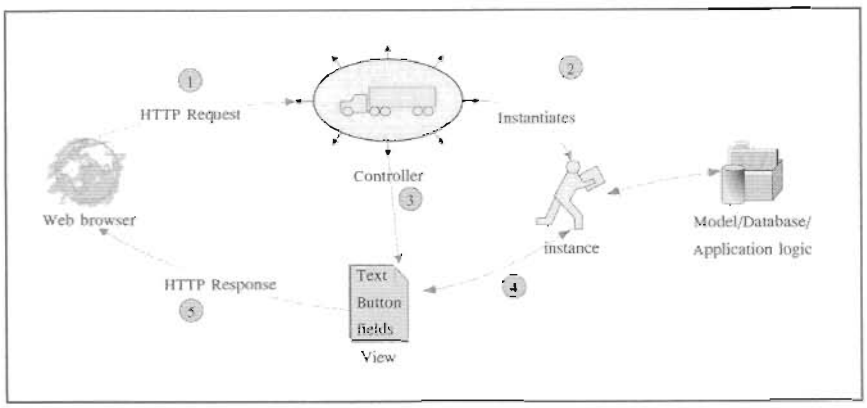
การกำหนดหน้าที่ทำได้โดยดาวน์โหลดไฟล์ข้อมูลในรูปแบบ GenBank แล้วแสดงข้อมูลเหล่านั้นในรูปแบบ HTML ดังแสดงภาพที่ 3-3 หรือเป็นรูปแบบตัวอักษรที่บอกให้ทราบถึงหน้าที่การทำงานต่างๆ ที่อยู่ภายในลำดับเบสที่ศึกษาหรือลำดับเบสที่เกี่ยวข้อง ในขั้นตอนนี้จะแสดงผ่านวิว ซึ่งเป็นส่วนแสดงผลใน TurboGears ที่จะกล่าวถึงในหัวข้อ MVC

4.1.4 การตรวจสอบความถูกต้องโดยนักชีววิทยา ทำได้โดยดึงข้อมูลจากขั้นตอนที่แล้ว ซึ่งเก็บอยู่ในรูปฐานข้อมูลหรือไฟล์ XML มาตรวจสอบโดยตรวจดูได้จากรายละเอียดข้อมูลที่เก็บไว้พร้อมกับข้อมูลนั้น เช่น เวอร์ชันโปรแกรม BLAST ที่ใช้, เวอร์ชันฐานข้อมูลที่ใช้ เพื่อบอกนักชีววิทยาให้ทราบถึงระยะเวลาของข้อมูลในฐานข้อมูลที่เก็บไว้, จำนวน Contigs ที่มาจากการเชื่อมต่อลำดับเบส ซึ่งก็จะบอกถึงจำนวนครั้งที่ต้องทำงาน ยิ่งจำนวน Contigs เยอะก็ต้องเชื่อมต่อลำดับเบสหลายครั้งจนกว่าจะเสร็จสมบูรณ์

ข้อมูลเหล่านี้จะถูกนำมาพิจารณาเพื่อตัดสินใจ ตัดแปลงแก้ไขข้อมูลหรือนำข้อมูลที่มีอยู่แล้วทำซ้ำอีกครั้ง เพื่อความถูกต้องแม่นยำมากยิ่งขึ้น การตัดแปลงแก้ไขข้อมูลเพื่อที่จะเพิ่มข้อมูลให้รายละเอียดมากยิ่งขึ้น เช่น เมื่อทำการศึกษากรอบเปิดการอ่านในครั้งแรกเสร็จ ข้อมูลที่ได้จะถูกนำไปใช้เป็นข้อมูลการสอนให้กับโปรแกรมอย่างเช่น Glimmer ที่สามารถบอกบริเวณที่เป็นยีนหรือบริเวณที่ไม่เป็นยีนในสิ่งมีชีวิตโปรคาริโอต เพื่อเพิ่มความแม่นยำที่จะระบุบริเวณที่เป็นยีนหรือไม่เป็นยีน

4.2 MVC: Model-View-Control

เพื่อที่จะให้การทำงานระหว่างขั้นตอน 1) การเชื่อมต่อลำดับเบส, 2) การตีความกรอบเปิดการอ่าน , 3) การระบุตำแหน่งและกำหนดหน้าที่กรอบเปิดการอ่านโดยหาความเหมือนในฐานข้อมูลต่าง, สุดท้ายคือ 4) การตรวจสอบความถูกต้องโดยนักชีววิทยา ทำงานร่วมกันได้อย่างราบรื่นวิทยานิพนธ์นี้ได้นำเสนอการทำงานที่แบ่งออกเป็นส่วน ได้แก่ 1) Model, 2) View และ 3) Controller เพื่อที่จะ



ภาพที่ 4-5 ภาพแสดงการทำงาน MVC แบบดั้งเดิม

แน่ใจว่าทุกขั้นทำงานร่วมกันอย่างถูกต้อง จึงเก็บข้อมูลระหว่างแต่ละขั้นตอนในรูปแบบ XML ที่เหมาะสำหรับส่งผ่านข้อมูลระบบเครือข่าย ที่งานวิทยานิพนธ์นำมาใช้เป็นแม่แบบส่งผ่านในแต่ละขั้นตอน

4.2.1 MVC และเว็บ จากกรณีศึกษาที่ผ่านมาในบทที่แล้ว บอกถึงการที่ผู้ใช้งานสนใจเพียงแค่ลักษณะหน้าตาหรือวิธีใช้งาน แต่สำหรับนักพัฒนาโปรแกรมคอมพิวเตอร์ จะสนใจการเขียนโปรแกรมเพื่อที่จะนำมางานอย่างไร มีรูปร่างหน้าตาแบบไหน และจะมีโครงสร้างของข้อมูลอย่างไร เพื่อนำไปพัฒนาให้ผู้อื่นมาใช้งานได้ ในงานวิจัยชิ้นนี้นำเสนอการทำงานแบบ MVC (Model-View-Control) ดังแสดงในภาพที่ 4-5 การวางกระบวนการทำงานต่างๆ ได้ทำการพัฒนาระบบการระบุตำแหน่งและกำหนดหน้าที่ด้วย TurboGears ซึ่งช่วยให้พัฒนาระบบได้อย่างรวดเร็ว จากภาพที่ 4-5 MVC และเว็บทำงานดังต่อไปนี้

4.2.1.1 เมื่อผู้ใช้เข้าเว็บและเริ่มทำงาน ทำการร้องขอยังเครื่องแม่ข่ายผ่านเว็บเบราเซอร์ (Web Browser) ส่งแบบฟอร์มข้อมูล อย่างเช่น รหัสผู้ใช้และรหัสผ่าน เครื่องแม่ข่ายก็จะทำหน้าที่รับข้อมูล ทำการถอดรหัสหรือวิเคราะห์ข้อมูลในแบบฟอร์ม

4.2.1.2 เครื่องแม่ข่ายจะทำหน้าที่เป็นตัวควบคุม (Controller) ทำหน้าที่ถอดรหัสและทำตามที่ร้องขอ ต่อจากนั้นก็จะร้องขอไปที่โมเดล (Model) ซึ่งทำหน้าที่คล้ายกับฐานข้อมูล โดยทั่วไปผลลัพธ์ที่ได้จากขั้นตอนนี้เรียกว่า Instance (ตัวแปร Object ที่สามารถกำหนดขึ้นแทน Object)

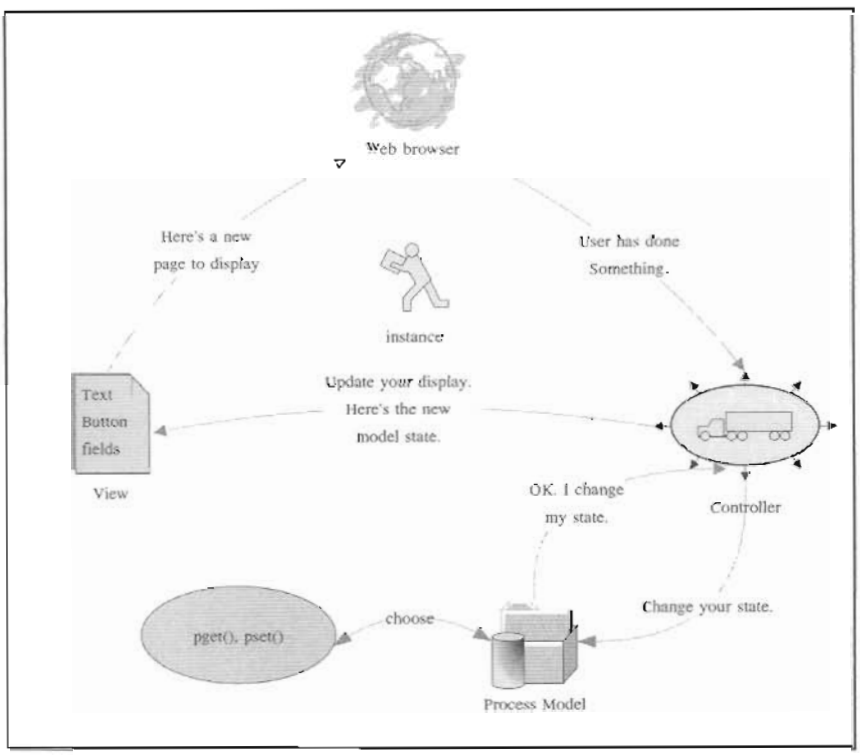
4.2.1.3 ตัวควบคุมจะทำการส่งรายละเอียดที่วิว (View) เพื่อที่จะแสดงรายละเอียดการทำงานออกมาเป็นรูปภาพ เช่น สร้างที่กด, ที่แสดงผลและ หน้าต่าง

4.2.1.4 ส่วนการแสดงผลขึ้นกับโมเดลที่ส่งต่อให้วิวทำการวิเคราะห์รายละเอียด

4.2.1.5 เมื่อวิวรับคำสั่งจากตัวควบคุม วิวจะแสดงผลลัพธ์สู่เว็บเบราเซอร์ เมื่อร้องขออีกครั้งการทำงานก็จะเริ่มที่ขั้นที่ 1-4 อีกครั้ง

4.2.2 โมเดล2 ข้อดีที่ใช้ TurboGears นอกจากพัฒนาได้อย่างรวดเร็วยังสามารถแบ่งเวอร์ชั่น (โมเดล2 ; Model2) การทำงานออกเป็น Developer กับ Producer ให้ TurboGears รับผิดชอบส่วนต่างๆ เพราะว่าแต่ก่อนเมื่อพัฒนาระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนที่ใครก็ตามที่มีอภิสิทธิ์เขียนกับอ่านไฟล์ก็สามารถที่จะแก้ไขโค้ดโปรแกรมของระบบได้แม้แต่ผู้ที่ไม่รู้การเขียนโปรแกรม

ในเชิงลึก แต่พอรู้จักการเขียน HTML แต่ไม่ใช่โค๊ดโปรแกรมของระบบ ด้วยวิธีแยกเวอร์ชั่นออกเป็น Developer ผู้มีอภิสิทธ์แก้ไขรายละเอียดเชิงลึกรวมถึงโค๊ดโปรแกรมได้ คือผู้ที่รู้จักการเขียนโปรแกรมของระบบกับ Producer ให้มีอภิสิทธ์แก้ไขรายละเอียดได้เฉพาะ HTML กับไฟล์โปรแกรมอีกเล็กน้อย ก็จะทำให้งานปลอดภัยขึ้น ดังแสดงในภาพที่ 4-6 กับวิธีออกแบบข้างล่างนี้



ภาพที่ 4-6 ภาพแสดงการทำงาน MVC แบบ Model2

4.2.2.1 จากภาพที่ 4-6 โมเดลกับโมเดล2 มีรายละเอียดแบบเดียวกัน เพราะทั้งคู่เป็นตัวเดียวกัน ทำให้สามารถนำกลับมาใช้ใหม่ได้ TurboGears ใช้โมดูล SQLObject ของ Python เป็นตัวติดต่อฐานข้อมูลนอกจากนี้ SQLObject สามารถจัดการฐานข้อมูลให้อยู่ในรูปเชิงวัตถุได้

4.2.2.2 สร้างตัวควบคุม TurboGears (TurboGears Controllers) เพื่อจะจัดการโค๊ดของระบบ โดยขั้นนี้ TurboGears สร้างโค๊ดโปรแกรมให้อัตโนมัติ เช่น การทำเครื่องแม่ข่ายให้สามารถจัดการกับการร้องขอบน HTTP หรือใช้โมดูล Catwalk สำหรับจัดการโมเดลกับใช้โมดูล CherryPy เพื่อจัดการเริ่ม (Start), จัดการหยุด (Stop) เว็บที่ติดตั้งไว้บนเครื่องแม่ข่าย

4.2.2.3 สร้างวิวแบบ HTML ในกรณีนี้ TGs ใช้โมดูล Kid สำหรับสร้างแม่แบบ HTML ซึ่งจะต่างจาก MVC แบบดั้งเดิมโดย HTML ที่สร้างจะรับมาจากตัวควบคุมซึ่งทำการบอกรายละเอียดที่ต้องการสร้างเป็น HTML เพื่อที่จะแสดงผลผ่านเว็บเบราเซอร์

Basic Root Controllers

```
import TurboGears
from TurboGears import Controllers, expose
class Root(Controllers.RootController):
    @expose(template="wiki20.templates.welcome")
    def index(self):
        return "<h1>Hello World</h1>"
```

ภาพที่ 4-7 ตัวอย่างระเบียบวิธีของตัวควบคุมใน TGs

ขั้นที่หนึ่ง: โมเดล

ใน MVC โมเดล, วิว และตัวควบคุมนั้นเป็นอิสระต่อกัน โมเดลไม่จำเป็นต้องรู้ว่าวิวคือ IE หรือว่าเป็น Firefox เช่นเดียวกันกับวิวก็ไม่จำเป็นต้องรู้ว่าใช้อะไรเป็นตัวจัดการฐานข้อมูล สิ่งที่ทั้งสามจำเป็นต้องการก็คือมีตัวสังเกตการณ์เพื่อที่จะแจ้งให้ทั้งสามทราบบอกสถานะในแต่ละช่วงให้กับทั้งสาม

ขั้นที่สอง: ตัวควบคุมของ TGs

จากข้างต้น TGs ใช้โมดูล CherryPy รับ/ส่ง ข้อมูลบน HTTP ผ่านเว็บเบราเซอร์ชนิดต่างๆ ซึ่งก็จะถูกถอดรหัส แล้วแปลให้ TGs ดำเนินการกับโมเดล แล้ววิวก็จะแสดงผลผ่านเว็บเบราเซอร์ ซึ่งส่งตัวควบคุมให้วิว ในกรณีนี้แสดงอยู่ในรูปแบบ Python ดังแสดงโค๊ดตัวอย่างที่นำไปใช้เป็นตัวควบคุมใน TGs ในภาพที่ 4-7 แสดงผลลัพธ์ออกมาเป็น "Hello World" บนเว็บเบราเซอร์

ขั้นที่สาม: วิว

วิวต้องการโค๊ด HTML ที่สร้างเว็บได้ TGs ใช้โมดูล Kid เป็นแม่แบบสร้าง HTML ดังแสดงโค๊ดตัวอย่างวิวในภาพที่ 4-8 สามารถอ่านรายละเอียดการใช้เพิ่มเติมจากหนังสือ “Rapid Web Applications with TurboGears - Using Python to Create Ajax-Power Sites” ในหนังสือ [29] จะอธิบายเพิ่มเติมในส่วนต่างๆ ของ MVC

จากภาพที่ 4-6 Instance บอกให้วิวรู้ถึงสิ่งที่ตัวควบคุมต้องการให้ทราบ ตัวอย่างนี้เป็นการสั่งงานโดยตัวควบคุมผ่าน Instance ที่ยังไม่ได้นำโมเดลเข้ามาร่วมด้วย แต่โดยปกติแล้วเมื่อมีโมเดลเข้ามาเกี่ยวข้อง ข้อมูลซึ่งอยู่ในโมเดลจะถูกส่งผ่านมายัง Instance ไม่ได้ส่งผ่านวิวโดยตรง ดังนั้นสิ่งที่วิวต้องการมีเพียงแค่การแจ้งการเปลี่ยนสถานะให้ทราบดังแสดงวิวในภาพที่ 4-6 เพราะฉะนั้นสิ่งที่วิวต้องการเมื่อโมเดลมีการเปลี่ยนสถานะนั่นก็คือการแจ้งสถานะปัจจุบันให้ทราบ

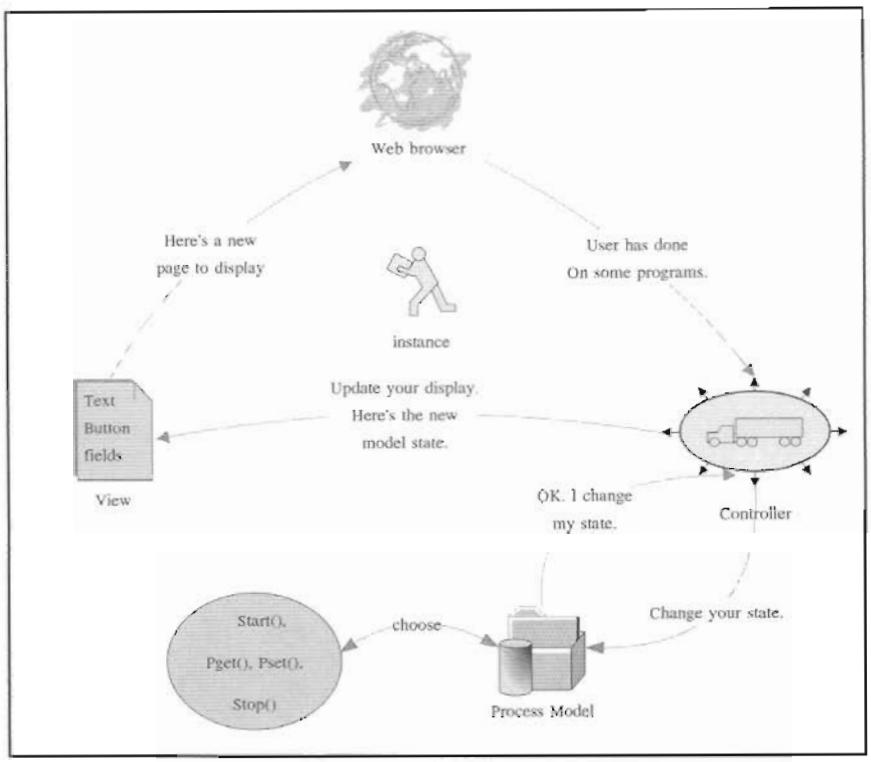
MVC หรือโมเดล2 ของ MVC คือการออกแบบในรูปแบบผสม [30] โดยหลักๆ แล้วมีรูปแบบการออกแบบประกอบไปด้วย รูปแบบตัวสังเกตุการณ์ (Observer), รูปแบบการวางแผน (Strategy) และมีรูปแบบแสดงผลแบบ Composite โดย MVC กับโมเดล2 ต่างที่วิวในโมเดล2 รู้สถานะโมเดลของในปัจจุบันผ่าน ตัวควบคุม (Controllers) แต่การออกแบบรูปแบบอื่นโดยส่วนมากไม่แตกต่าง ยกเว้นตัวควบคุมในโมเดล2 ที่แสดงสิ่งต่างๆ ของวิว ทำให้สามารถแสดงได้หลากหลายขึ้นเป็นอิสระขึ้น (โดย

Basic View TGs

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<?python import sitetemplate ?>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:py="http://purl.org/Kid/ns#" py:
extends="sitetemplate">
<head py:match="item.tag=='{http://www.w3.org/1999/xhtml}head'" py:attrs="item.
items()">
    <meta content="text/html; charset=UTF-8" http-equiv="content-type"
py:replace="''"/>
    <title py:replace="''">Your title goes here</title>
    <meta py:replace="item[:]"/>
</head>
<body py:match="item.tag=='{http://www.w3.org/1999/xhtml}body'" py:attrs="item.
items()">
    <div py:if="tg_flash" class="flash" py:content="tg_flash"></div>
    <div py:replace="[item.text]+item[:]"/>
    <p align="center"><img src="/static/images/tg_under_the_hood.png"
alt="TurboGears under the hood"/></p>
</body>
</html>
```

ภาพที่ 4-8 ตัวอย่างวิวใน TGs

สามารถเปลี่ยนวิวหรือตัวควบคุมโดยที่ไม่ต้องแก้ไขโค๊ดเพิ่มเติม) หรือแม้แต่เป็นระบบมากขึ้นโดยผ่านตัวควบคุม ส่วนโมเดลของทั้งสองแบบอาจจะเป็นฐานข้อมูลก็ได้หรือแม้แต่เป็นโค๊ดซึ่งใช้จัดการและดอยดูแลข้อมูลที่อยู่ภายในโมเดล ซึ่งหน้าที่นี้อาจจะสับสนกับของตัวควบคุมๆ ทำหน้าที่แปลรหัสที่รับเข้ามาจากวิวว่าต้องการให้ทำอะไรมากกว่าที่จะดอยจัดการข้อมูลในโมเดลดังนั้น ตัวควบคุมเป็นตัวแปลความหมายให้กับโมเดลว่าจะให้ทำสิ่งใด แล้วก็แปลต่อให้วิวทราบว่าวิวจะแสดงอะไรผ่านเว็บเบราเซอร์ นี่จึงเป็นหน้าที่หลักของตัวควบคุม นอกจากนี้ยังสามารถมีมากกว่าหนึ่งวิว โดยสามารถแยกคลาสของตัวควบคุม (Class Controllers) ออกมาซึ่งก็จะทำให้ง่ายต่อการดูแล



ภาพที่ 4-9 ภาพแสดงการร้องขอใช้โปรแกรม

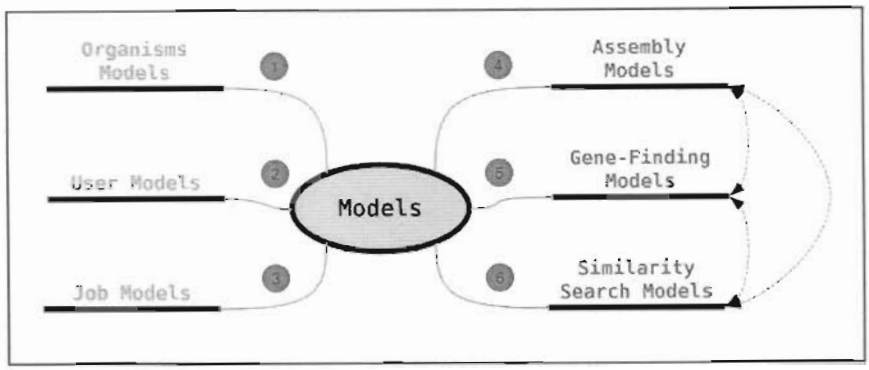
4.3 การวางกระบวนการทำงานแบบอัตโนมัติ

จากตัวอย่าง โปรแกรมบนระบบการระบุตำแหน่งและกำหนดหน้าที่ในบทที่ 3 โดยส่วนมากจะเก็บข้อมูลในลักษณะ สืบค้นฐานข้อมูล SQL, ไฟล์เอกสารในรูปแบบต่างๆ อาทิเช่น FASTA, ไฟล์ดัชนีอ้างอิงของ BLAST และยังเก็บข้อมูลในลักษณะเอกสาร XML ส่วนกระบวนการต่างๆ จะเขียนโปรแกรมทำงานตามที่ตั้งกระบวนการไว้ แต่ถ้ามองในแง่ของ MVC ลักษณะนี้จะนำเอาตัวควบคุมรวมกับโค๊ดที่ใช้จัดการและโค๊ดที่ใช้คอยดูแลข้อมูล มารวมเข้าด้วยกัน แล้วแยกโมเดลให้เก็บเฉพาะข้อมูลเท่านั้น ส่วนวิวก็จะแสดงผลผ่านให้กับโมเดลโดยตรงด้วยการดึงข้อมูลจากฐานข้อมูลหรือไฟล์เอกสารออกมา การเขียนโค๊ดแบบนี้อาจทำให้เกิดปัญหาดังต่อไปนี้

4.3.1 ไม่ยืดหยุ่น (Not Flexible) หากมีการเปลี่ยนแปลงแก้ไขโค๊ดซึ่งใช้ติดต่อกับฐานข้อมูล ยกตัวอย่างเช่น ถ้า BaSys ต้องการเพิ่มโปรแกรมเพื่อทำการคำนวณเปอร์เซ็นต์ GC ก็ต้องเปลี่ยนโค๊ดที่วิวและโค๊ดที่ ตัวควบคุม ดังนั้นเองทุกครั้งที่มีการเปลี่ยนแปลงแก้ไขอะไร ก็ต้องมาแก้ไขที่หน้าวิวกับตัวควบคุมทุกครั้งไป

4.3.2 ไม่ปลอดภัย (Not Encapsulated) เพราะต้องเปิดเผยรายละเอียดต่างๆ ที่อยู่บนวิว เช่น ชื่อข้อมูล, ที่อยู่ไฟล์บนเครื่องแม่ข่าย เหล่านี้เป็นต้น

4.3.3 ไม่เป็นโมดูล (Not Modular) ไม่สามารถนำวิวนี้มาใช้ซ้ำอีกครั้งได้เพราะการออกแบบอิงไปกับโค๊ดเป็นส่วนใหญ่ทำให้ยากแก่การนำวิวนั้นมาใช้อีกครั้ง



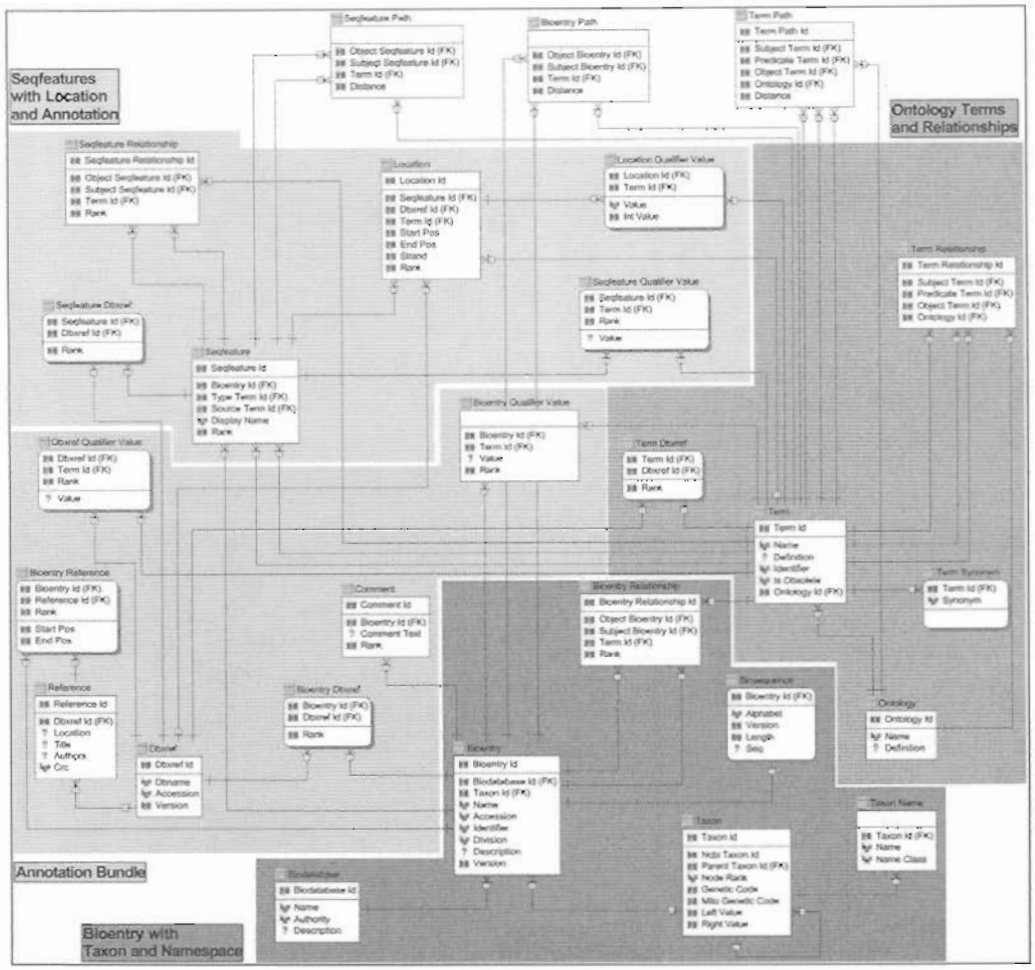
ภาพที่ 4-10 โมเดลแสดงคุณลักษณะต่างๆ ของข้อมูลภายในโมเดลและ Application Logic

ด้วยการออกแบบ MVC แบบโมเดล2 ทำให้สามารถแก้ไขปัญหาทั้งหมดได้ เพราะว่าโมเดลจะเป็นตัวจัดการเกี่ยวกับการติดต่อฐานข้อมูลทั้งหมด โดยไม่ต้องกังวลเกี่ยวกับเรื่องฐานข้อมูลอีกต่อไป นอกจากนี้โมเดลยังสามารถทำหน้าที่ดูแลการทำงานของโปรแกรมในระดับต่างๆ ของ Application Logic ได้ด้วยการเขียนโค๊ดเพิ่มให้จัดการในระดับนี้ได้อย่างเป็นอิสระออกจากวิว ทำให้สามารถออกแบบโปรแกรมต่างๆ ในระบบอย่างอิสระดังภาพที่ 4-9 เมื่อผู้ใช้ร้องขอการใช้โปรแกรม ตัวควบคุมจะทำการแปลความหมายที่ได้รับมา ส่งต่อให้โมเดลทราบ เพื่อที่จะดำเนินการในขั้นต่อไป เมื่อสถานะในโมเดลถูกเปลี่ยนจะเป็นการแจ้งให้วิวทราบโดยบอกผ่าน ตัวควบคุมทางอ้อม

ในส่วนของวิวไม่ต้องขึ้นกับโมเดลอีกต่อไป ทุกครั้งที่วิวติดต่อกับฐานข้อมูลตัวควบคุมจะเป็นตัวจัดการให้ทั้งหมดโดยดังภาพที่ 4-9 และทำหน้าที่แปลความหมายที่วิวต้อการ และไม่มีการแปะชื่อฐานข้อมูลหรือ ที่อยู่ไฟล์บนเครื่องแม่ข่ายที่หน้าวิวอีกต่อไป เพราะจะเป็นหน้าที่ของตัวควบคุมเป็นผู้ดำเนินการ ดังนั้นจึงมีความปลอดภัยระดับพอควร ให้นึกถึงการถอดปลั๊กกับการใช้เครื่องไฟฟ้าในบ้าน เมื่อต้องการใช้ก็เพียงแต่เดินไปเสียบปลั๊กเพื่อเปิดการใช้งาน ในทำนองเดียวกันด้วย MVC ก็เพียงเลือกโปรแกรมและส่งคำร้องขอใช้โปรแกรม แล้วการดำเนินการที่เหลือก็จะส่งต่อไปให้ ตัวควบคุมเป็นตัวจัดการต่อไป ทำให้ง่ายต่อการใช้งาน

จากภาพที่ 4-10 แบ่งโมเดลต่างๆ ออกเป็น 6 โมเดลโดยโมเดลหล่านี้ทำหน้าที่ตั้งแต่บอกลักษณะของข้อมูล วิธีใช้ข้อมูลที่โมเดลดูแล จนถึงโค๊ดของโปรแกรมสำหรับจัดการและดูแลรักษาข้อมูลของระบบ รายละเอียดของโมเดลทั้งหมดคือ

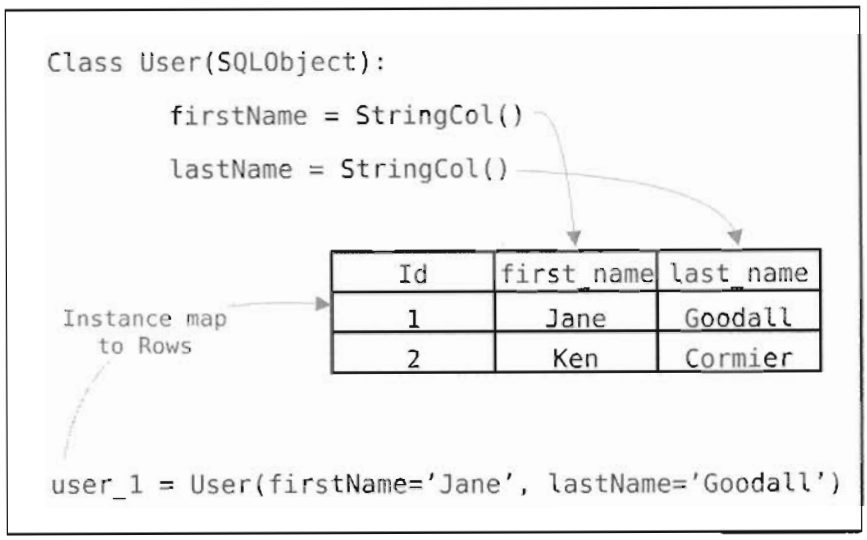
1. Organisms Models โมเดลนี้จะคอยเก็บฐานข้อมูลให้สืบค้นในรูปแบบต่างๆ อาทิเช่น ข้อมูลในฐานข้อมูล, ข้อมูลในไฟล์เอกสารเช่น เอกสาร XML และยังจัดการประเภทข้อมูลเชิงชีววิทยาด้วยฐานข้อมูล BioSQL กับ Chado อีกด้วย ดังแสดงตัวอย่างโครงสร้างฐานข้อมูล BioSQL ในภาพที่ 4-11 ประกอบด้วย 27 ตารางที่ทำงานทั้งหมด 4 รูปแบบ ได้แก่ 1) Bioentry with Taxon and Namespace 2) Seqfeatures with Location and Annotation 3) Annotation Bundle และ 4) Ontology Terms and Relationships โดยทั้ง 4 ข้อนี้ครอบคลุมสำหรับจัดการไฟล์ในรูปแบบ GenBank ที่โหลดมาจากเว็บของ NCBI



ภาพที่ 4-11 โครงสร้างฐานข้อมูล BioSQL

นอกจากนี้ข้อมูลในการระบุตำแหน่งและกำหนดหน้าที่ยีนอาจซ้ำกันได้ เนื่องจากสารชีวโมเลกุลที่เป็นผลผลิตของยีน (Gene Product) ส่วนใหญ่หมายถึงโปรตีนชนิดต่างๆ ที่เซลล์สร้างขึ้น นอกจากนี้ยังรวมถึงโมเลกุลของ RNA บางชนิดเช่น Ribosomal RNA และ tRNA เกิดจากยีนที่มีหน้าที่การทำงานคล้ายคลึงกัน เนื่องจากสายลำดับการสืบทอดมีความใกล้เคียงกันหรือเกิดจากบรรพบุรุษในสายลำดับการสืบทอดเดียวกัน ทำให้หนึ่งผลผลิตของยีนอาจจะมียีนที่เกี่ยวข้องมากกว่าหนึ่งยีน (One-to-Many) และตัวยีนเองในกรณี Operon หรือ Multigene Family ก็มีความสัมพันธ์กับยีนที่มีหน้าที่สร้างผลิตผลยีนอื่นๆ อีกกลายเป็นลักษณะความสัมพันธ์ (Many-to-Many) ถึงแม้ว่าในโครงสร้างฐานข้อมู BioSQL ไม่อนุญาตให้มีความสัมพันธ์แบบ m-m แต่ก็สามารถลิงค์ความสัมพันธ์โดยใช้ตารางชื่อ Term ใน Ontology Terms and Relationships ร่วมกับตารางชื่อ Bioentry Relationship ใน Bioentry with Taxon and Namespace เพื่อที่จะบ่งบอกถึงความสัมพันธ์ที่หลากหลายนี้

2. User Models จัดการเกี่ยวกับบัญชีผู้ใช้ ด้วยโมเดลกลุ่มผู้ใช้ (Group Model) เนื่องจากผู้ใช้งานมีหลายระดับ ตั้งแต่ผู้ดูแลระบบ (Administrator), คนทดลองใช้งาน (Guest) และอื่นๆ ตลอดจนโมเดลอภิสิทธิ์ (Permission Model) ในการใช้งาน เช่น แก้ไข, อ่าน และอื่นอีกเป็นต้น นอกจากนี้ยัง



ภาพที่ 4-12 การเก็บข้อมูลในตารางด้วย SQLObject

รวมถึงโมเดลสำหรับ Cookies ก็ถูกโมเดลด้วยเพื่อความปลอดภัยในการใช้งาน ในแง่ของ Application Logic สามารถให้ TGs สร้างฐานข้อมูลด้วยการเรียก SQLObject ดังภาพที่ 4-12 เก็บข้อมูลในตาราง อย่างอัตโนมัติด้วยคลาส User() ดังตัวอย่างที่เพิ่มชื่อ Jane นามสกุล Goodall ลงในตาราง

3. Job Models เป็นโมเดลวางกระบวนการทำงานแบบอัตโนมัติ (Automated Workflow Model) อันประกอบไปด้วย การวางกระบวนการจัดการลำดับงานที่ร้องขอเข้ามา คือการที่ผู้ส่งวางโปรแกรมค้นหาต่างๆ เพื่อที่จะทำงานอย่างเป็นกระบวนการ เช่น วางโปรแกรม PCAP ทำงาน ต่อมาให้โปรแกรม GLIMMER เข้ามาค้นหายีน ต่อจากนั้นวางโปรแกรม BLAST ค้นหาความเหมือนในฐานข้อมูล NR ของ NCBI เป็นต้น

4. Assembly Models ในโมเดลนี้จะคอยจัดการเกี่ยวกับ Application Logic ของโปรแกรมเกี่ยวกับการประกอบสายลำดับเบสขึ้นมาใหม่และยังรวมถึงโมเดลของโปรแกรมประเภทเดียวกันด้วย

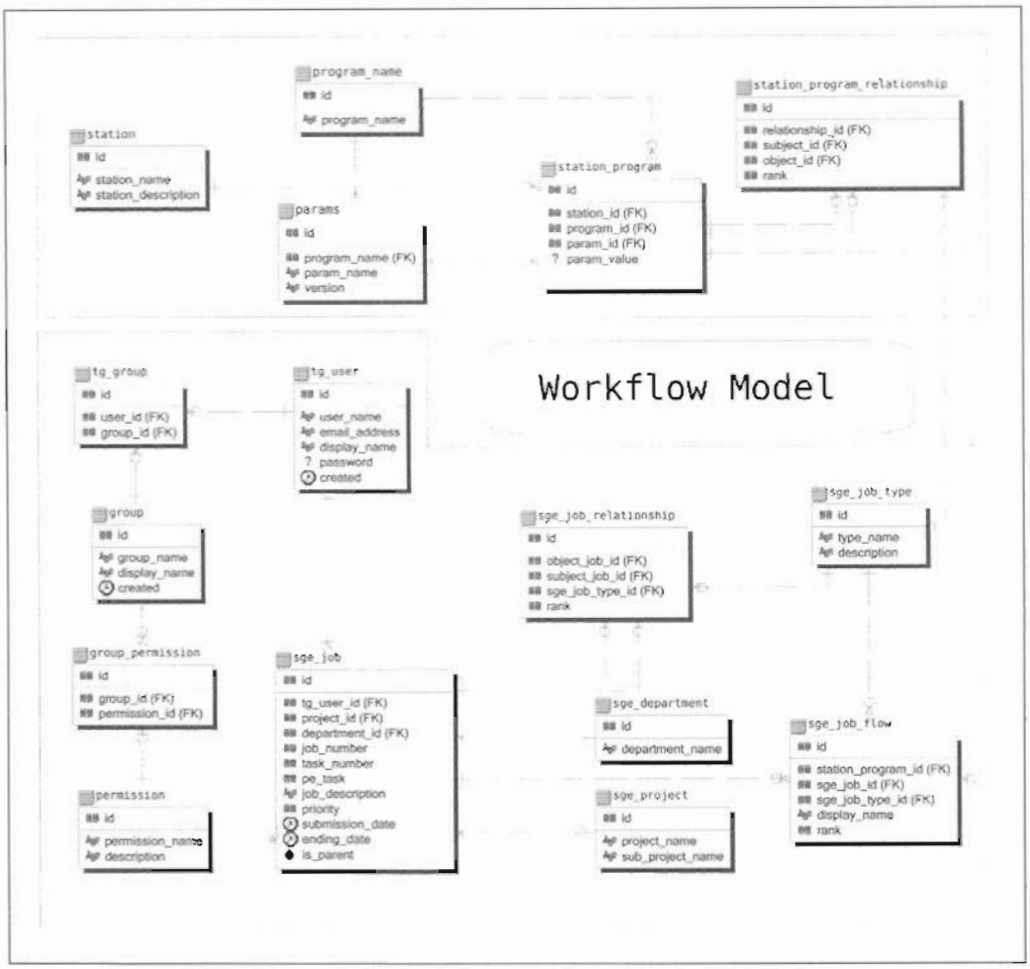
5. Gene-Finding Models ในโมเดลนี้จะคอยจัดการเกี่ยวกับ Application Logic ของโปรแกรมเกี่ยวกับการค้นหายีนและยังรวมถึงโมเดลของโปรแกรมประเภทเดียวกันด้วย

6. Similarity Search Models ในโมเดลนี้จะคอยจัดการเกี่ยวกับ Application Logic ของโปรแกรมเกี่ยวกับการค้นหาความเหมือนและยังรวมถึงโมเดลของโปรแกรมประเภทเดียวกันด้วย

ภาพที่ 4-13 คือโครงสร้างฐานข้อมูล Workflow ในฐานข้อมูลนี้ประกอบด้วย 16 ตาราง ซึ่งแต่ละตารางมีหน้าที่แตกต่างกันออกไป ตามแต่การใช้งาน ได้แก่

1. "tg_user" เก็บรายละเอียดชื่อล็อกอินเข้าใช้งาน, อีเมล์, ชื่อที่แสดง, รหัสผ่าน และ วันที่สร้างบัญชีผู้ใช้งาน โดยอีเมล์มีเพื่อที่จะส่งผลลัพธ์ให้ผู้ใช้ในกรณีที่งานใช้เวลานาน จึงต้องใช้อีเมล์แจ้งให้ทราบว่าเมื่องานเสร็จ

2. "tg_group" เก็บหมายเลขผู้ใช้งานกับหมายเลขกลุ่มในรูปแบบคีย์นอกหรือ Foreign Key โดยเมื่อมีการสร้างบัญชีผู้ใช้งาน ก็จะทำการกำหนดกลุ่มของผู้ใช้งานโดยอัตโนมัติ นอกจากนั้นมีการ



ภาพที่ 4-13 โครงสร้างของฐานข้อมูล Workflow

เพิ่มผู้ใช้งานแล้วตารางนี้ก็จะถูกเพิ่มโดยอัตโนมัติ

3. "group" คือ กลุ่มผู้ใช้งาน ประกอบไปด้วย ผู้ดูแลระบบหรือ Administrator, ผู้ดูแลรักษาหรือ Maintainer, ผู้เข้าเยี่ยมชมหรือ Guest เหล่านี้เป็นต้น

4. "group_permission" เก็บหมายเลขชนิดการใช้งาน (Permission) กับหมายเลขกลุ่มในรูปแบบคีย์นอกหรือ Foreign Key โดยเมื่อมีการกำหนดกลุ่มผู้ใช้งาน และกำหนดชนิดการใช้งานของกลุ่มแล้วตารางนี้ก็จะถูกเพิ่มโดยอัตโนมัติ

5. "permission" เป็นชนิดการใช้งานที่อนุญาตหรืออนุมัติให้ทำสิ่งใดสิ่งในกรอบที่กำหนด เช่น อนุญาตให้แก้ไขไฟล์ HTML ได้, อนุญาตแก้ไขโค้ดในระบบได้ เหล่านี้เป็นต้น

6. "sge_job" เมื่อผู้ใช้ร้องขอการใช้งานการระบุตำแหน่งและกำหนดหน้าที่มาที่เครื่องแม่ข่าย และเมื่อผ่านตรวจสอบแล้ว หลังจากนั้นเครื่องแม่ข่ายจะบันทึกการทำงาน โดยเริ่มที่ตารางนี้ ในตารางนี้ประกอบไปด้วย หมายเลขผู้ใช้ "tg_user_id", หมายเลขโครงการ "project_id" ใช้สำหรับระบุโครงการที่ทำงานเช่น โครงการระบุตำแหน่งและกำหนดหน้าที่หนูทั่วทั้งจีโนม), หมายเลขแผนก หรือ หมายเลขภาค "department_id", หมายเลขของงานที่ SGE สั่งทำงานได้ "job_number", รายละเอียดของงาน

ที่ทำ "job_description", ระดับความสำคัญของงานหรือ Priority, วันที่สั่งทำงาน "submission_date", วันที่ทำงานเสร็จสิ้น "ending_date" และตัวแปรชนิดไบนารีที่ใช้ระบุงานที่สั่งเป็นงานหลักในหลายขั้นตอนการทำงาน "is_parent"

7. "sge_job_relationship" บอกถึงความสัมพันธ์ระหว่างงานที่กำลังทำอยู่หรืองานที่ทำเสร็จแล้ว โดยใช้คีย์นอกที่อ้างอิงกับตาราง "sge_job" ประกอบด้วยชนิดของความสัมพันธ์ในตาราง "sge_job_type" อย่างความสัมพันธ์ "is_a" ยกตัวอย่างเช่น job1123 อยู่ในช่อง "object_job_id" เป็นความสัมพันธ์หลักหรือ Parent และ Job1124 อยู่ในช่อง "subject_job_id" เป็นความสัมพันธ์รองหรือ Child จะมีความหมายว่า Job1124 เกิดจาก Job1123 เป็นต้น

8. "sge_department" บอกถึงภาคหรือแผนกที่ทำงานอยู่จะมีหรือไม่มีก็ได้

9. "sge_project" เช่นเดียวกับ "sge_department" ใช้บอกโครงการที่ดำเนินการอยู่ในขณะนั้นและจำเป็นต้องมีเพื่อบอกถึงวัตถุประสงค์ที่กำลังทำ

10. "sge_job_type" ใช้บอกความสัมพันธ์ต่างๆ อาทิเช่น "is_a" ใช้บอกความสัมพันธ์ระหว่างพ่อกับลูก หรือ running ใช้บอกว่างานนั้นกำลังทำงานอยู่ หรือ waiting ใช้บอกว่างานที่ร้องขอยังไม่ถูกสั่งทำงาน หรือ finished ใช้บอกว่างานที่ทำเสร็จแล้ว นอกจากนี้ยังสามารถมีความสัมพันธ์อื่นอีกได้ เช่น ความสัมพันธ์ประกอบไปด้วย หรือ "compose_of" ใช้สำหรับบอกว่างานที่ทำอยู่หรือเสร็จแล้วก่อให้เกิดงานย่อยตามมา

11. "sge_job_flow" ถือว่าเป็นตารางที่เป็นหัวใจสำคัญเช่นเดียวกับ "sge_job" ที่ใช้สำหรับเก็บสถานะในปัจจุบันและสถานะในอดีตของโปรแกรมหรือสถานีที่เกี่ยวข้องกับงานนั้นๆ หมายความว่างานที่สั่งทำงานในระบบไม่ใช่ทำงานด้วยโปรแกรมใดโปรแกรมเดียวแต่เกิดจากการนำโปรแกรมหลายๆ โปรแกรมมาทำงานร่วมกัน เพราะฉะนั้นจึงต้องมีตารางสำหรับเก็บสถานะต่างจะได้ทราบว่างานที่ทำอยู่นี้ถึงไหนแล้ว ในตารางนี้จะอ้างคีย์นอกที่เป็นของตาราง "sge_job" หรือ "station_program" และ "sge_job_type" โดย "sge_job" จะบอกหมายเลขงานที่ SGE สั่งทำงาน ส่วน "station_program" ใช้บอกให้งานที่ทำอยู่ทราบว่าต้องใช้พาราเตอร์อะไรสำหรับทำงานและสุดท้ายก็คือ "sge_job_type" บอกให้ทราบถึงความสัมพันธ์ของงานที่ทำอยู่ให้ทราบ เป็นต้น

12. "station" ประกอบได้ด้วย การประกอบสายลำดับเบสใหม่หรือ Fragment Assembly, การค้นหายีนหรือ Gene-Finding และ การค้นหาความเหมือนหรือ Similarity Search ซึ่งเป็นชื่อในสถานีต่างๆ ซึ่งถูกใช้ในการระบุตำแหน่งและกำหนดหน้าที่ยีนทั่วทั้งจีโนม

13. "program_name" เป็นชื่อต่างๆ ของโปรแกรมที่อยู่ในสถานีในข้อที่แล้ว ซึ่งใช้สำหรับให้อ้างอิงตารางอื่นๆ และแสดงให้เห็นว่ามีโปรแกรมใดบ้างที่ได้จดทะเบียนอยู่ในระบบของบ้าง

14. "params" มีหมายเลข "program_name_id" และบอกให้ทราบว่าโปรแกรมในหมายเลข "program_name_id" นี้มีพารามิเตอร์อะไรบ้าง เช่นโปรแกรม BLAST มีพารามิเตอร์ "-m 7" บอกให้แสดงผลลัพธ์ออกมาในรูปแบบเอกสาร XML เป็นต้น ซึ่งในตารางนี้จะเก็บเพียงแต่พารามิเตอร์ ไม่ทำการเก็บค่าของพารามิเตอร์ลงตาราง ค่าพารามิเตอร์จะนำเก็บไว้ในตาราง "station_program"

15. "station_program" ตารางนี้จะเก็บค่าของพารามิเตอร์, คีย์นอกของตาราง params หรือ

"params_id" เพื่อเชื่อมโยงกันระหว่างโปรแกรมและพารามิเตอร์ของโปรแกรม ส่วนคีย์นอกอย่างเช่น "station_id" กับ "params_id" บอกให้ทราบว่าโปรแกรมที่กำลังทำงานนี้อยู่ในสถานีใด เพื่อที่จะส่งไปให้ตาราง "sge_job_flow" ทราบ

16. "station_program_relationship" เหมือนกับในตาราง "sge_job_relationship" ที่บอกความสัมพันธ์ระหว่างโปรแกรมในสถานีต่างๆ

เมื่อเปรียบเทียบรูปภาพที่ 4-13 และรูปภาพที่ 4-11 จะเห็นว่าภาพที่ 4-13 จะจัดการเกี่ยวกับการวางกระบวนการระบุตำแหน่งและกำหนดหน้าที่ขั้นพื้นฐาน ส่วนภาพที่ 4-11 จะจัดการการจัดเก็บข้อมูลเชิงชีววิทยา เมื่อนำทั้งสองมาใช้ร่วมกันก็จะเป็นการทำงานร่วมกัน ระหว่างการนำข้อมูลชีววิทยามาใช้กับการวางกระบวนการ ระบุตำแหน่งและกำหนดหน้าที่ ซึ่งจะเป็นอัตโนมัติหรือไม่ขึ้นการวางแผนการทำงานของผู้ออกแบบเอง จากที่ได้กล่าวมาในข้างต้นว่างานการระบุตำแหน่งและกำหนดหน้าที่เกิดจากการนำหลายโปรแกรมมาใช้ร่วมกัน เพื่อที่จะทำงานให้เป็นอัตโนมัติ ต้องเขียนโค๊ดขึ้นมาเพิ่มเติมนอกจากโครงสร้างฐานข้อมูลทั้งสองแบบที่กล่าวมาแล้ว ซึ่งในการออกแบบรูปแบบการทำงาน MVC คือการเขียนโค๊ดในส่วน application logic นั่นเอง หมายความว่าหากขาดส่วนใดส่วนหนึ่งก็จะไม่สามารถสร้างระบบแบบอัตโนมัติได้

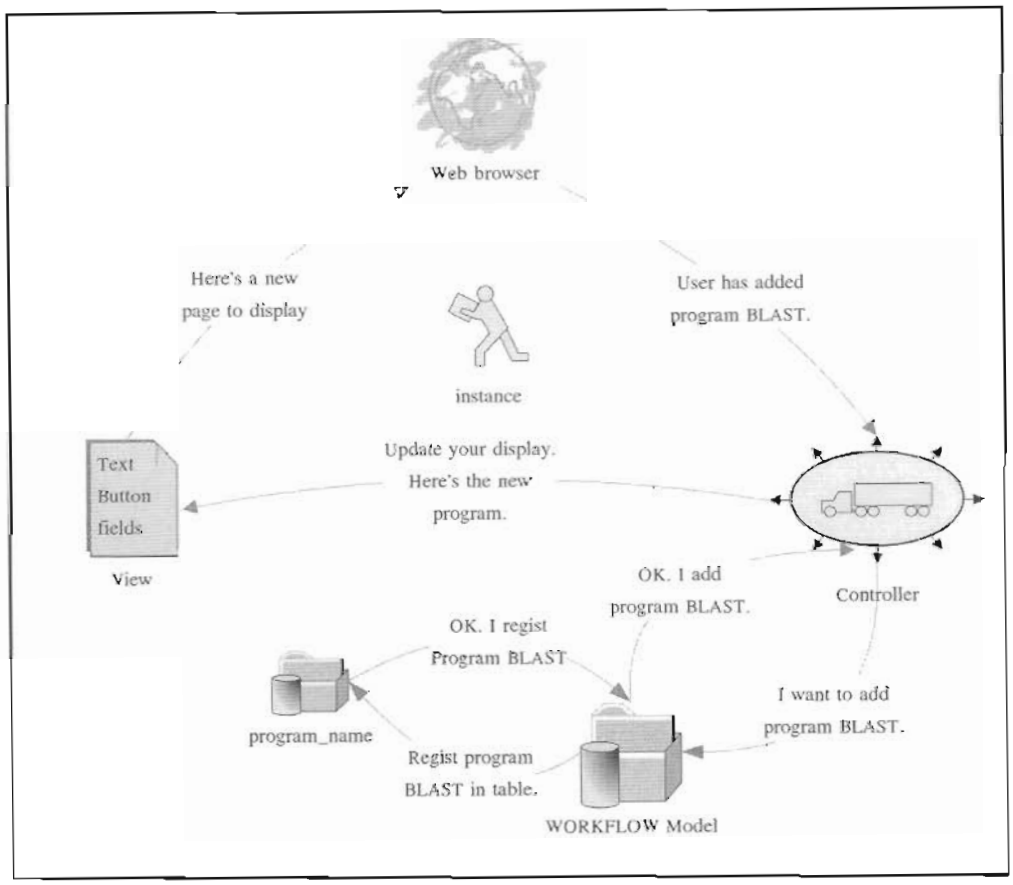
จุดประสงค์ของงานวิจัยชิ้นนี้ เพื่อที่จะ สร้างระบบการระบุตำแหน่ง และกำหนดหน้าที่ของยีนบนจีโนมใหม่แบบอัตโนมัติขึ้นมา ซึ่งก็ได้บอกรายละเอียดในองค์ประกอบต่างๆ เป็นที่เรียบร้อยแล้ว ไม่ว่าจะเป็นการออกแบบอย่างไรให้เป็นระบบ และสามารถนำโค๊ดกลับมาใช้ใหม่ได้อย่างไร ในหัวข้อต่อไปจะเป็นการแนะนำระบบระบบการระบุตำแหน่งและกำหนดหน้าที่ยีนแบบอัตโนมัติสำหรับวิธีการลำดับดีเอ็นเอบนจีโนม ที่พัฒนาขึ้นมาในรายละเอียดและสถาปัตยกรรมต่างๆ ที่เกี่ยวข้อง

4.4 วิธีการทำงานบนระบบ

การดูแลข้อมูลเข้าหรือออกในแต่ละกระบวนการที่แตกต่างกันออกไป ก็เป็นหัวใจสำคัญประการหนึ่ง ที่ผู้ดูแลระบบต้องคำนึงถึง แต่เนื่องจากข้อมูลชีวสารสนเทศเป็นข้อมูลที่อยู่อย่างกระจัดกระจาย นอกจากไม่เป็นระเบียบแล้วยังไม่สอดคล้องกัน จึงลำบากเมื่อต้องนำหลายโปรแกรมมาใช้งานร่วมกัน แต่ด้วยการออกแบบในรูปแบบ MVC ช่วยให้จัดระเบียบข้อมูลชีวสารสนเทศ และทำให้ข้อมูลเหล่านี้สอดคล้องกันได้ด้วย ตาราง "params" กับ "station_program" กับ "station" และสุดท้ายคือตาราง "program_name" ในภาพที่ 4-13 แสดงให้เห็นว่าลดจำนวนตารางโปรแกรมต่างๆ ให้เหลือเพียงแค่ 4 ตารางที่สัมพันธ์กัน ทำให้ไม่ต้องสร้างโปรแกรมต่อหนึ่งตาราง ยกตัวอย่างเช่น แต่ก่อนถ้าต้องการสร้างโมเดลโปรแกรม "BLAST" ก็ต้องสร้างตารางที่สัมพันธ์กับพารามิเตอร์ และก็ต้องสร้างตารางอย่างนี้เรื่อยๆ ถ้ามีโปรแกรมใหม่เข้ามา แต่ด้วยวิธีของๆ สามารถทำได้ดังนี้

1. การเพิ่มโปรแกรมใหม่เข้ามาก็คือการทะเบียนลงในตารางชื่อ "program_name" เป็นการแจ้งให้ทราบว่ามีโปรแกรมนี้

2. การที่สร้างโมเดลพารามิเตอร์ของโปรแกรม ใช้เพียงตาราง "params" เพื่อที่จะเก็บพารามิเตอร์ของโปรแกรมโดยอ้างอิงจากตาราง "program_name"

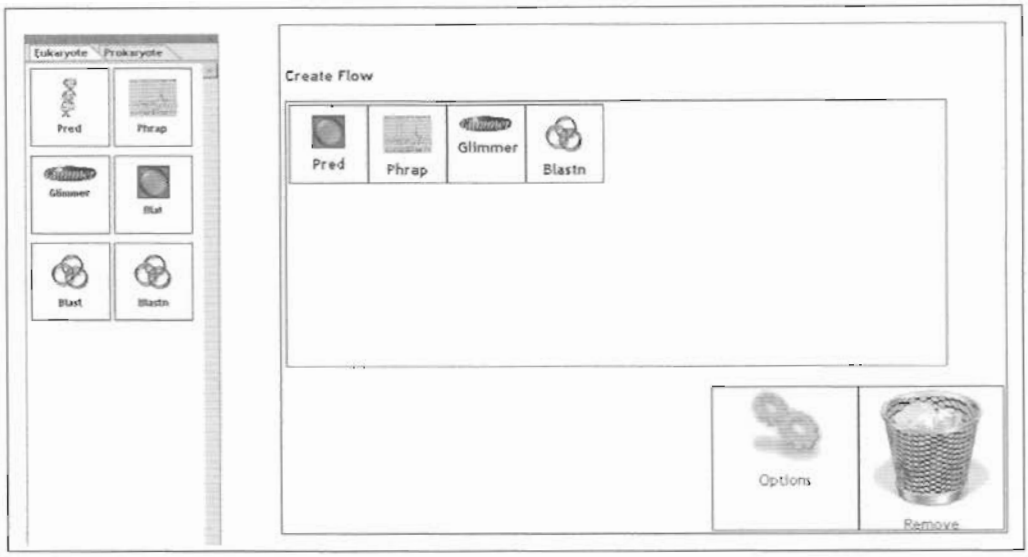


ภาพที่ 4-14 การลงทะเบียนในฐานข้อมูล Workflow

3. สุดท้ายแล้วเมื่อมีการเรียกใช้โปรแกรม ก็จะเรียกตาราง "station_program" เข้ามาเกี่ยวข้อง โดยการใส่ค่าพารามิเตอร์ที่รับเข้ามาโดยอิงกับชื่อโปรแกรมและชื่อของสถานีของโปรแกรมที่ใช้ ณ เวลานั้นๆ ดังแสดงในวิธีลงทะเบียนในภาพที่ 4-14 จากภาพแสดงการลงทะเบียนในฐานข้อมูล Workflow โดยเริ่มจากการร้องขอของผู้ใช้แล้ว ตัวควบคุมทำการตีความหมายเพื่อที่จะสั่งให้ลงทะเบียนโปรแกรม BLAST ลงในฐานข้อมูล จะเห็นได้ว่าวิธีการเพิ่มโปรแกรมเข้าไปในระบบเป็นไปตามวิธีในโมเดล2 ของ MVC ดังที่ได้อธิบายอย่างละเอียดในหัวข้อ โมเดล2 ที่ผ่านมา ขั้นตอนการทำงานของระหว่างโปรแกรมต่างๆ ประกอบด้วย

1. เริ่มที่การลงทะเบียนโปรแกรมดังแสดงในภาพที่ 4-14 และส่วนการเพิ่มพารามิเตอร์ของโปรแกรมก็คล้ายๆ กับการลงทะเบียนเพียงแต่เปลี่ยนจากตาราง "program_name" เป็นตาราง "params" (ขั้นตอนนี้ทำครั้งเดียวในครั้งแรกเท่านั้น)

2. ขั้นต่อมาผู้ใช้จะเลือกโปรแกรมที่ระบบจัดให้ดังแสดงในภาพที่ 4-15 วางเป็นกระบวนการทำงานที่ติดตั้งขึ้น โดยเมื่อผู้ใช้วางการทำงานเสร็จเรียบร้อย แล้วส่งไปให้ตัวควบคุมๆ ก็จะทำการตีความที่รับเข้ามา โดยเท่ากับจำนวนโปรแกรมที่ผู้ใช้วางการทำงานเอาไว้ หลังจากนั้นตัวควบคุมจะส่งชื่อสถานี, ชื่อโปรแกรมในแต่สถานี และพารามิเตอร์พร้อมค่าของมัน ให้ทำการเก็บลงฐานข้อมูล Workflow ใน



ภาพที่ 4-15 การเลือกโปรแกรมที่ระบบจัดให้

ตารางต่างๆ ทั้งหมด 16 ตาราง

3. หลังจากที่จัดเก็บลงลงฐานข้อมูล Workflow เป็นที่เรียบร้อย Workflow ก็จะทำการแจ้งที่ตัวควบคุม ให้ทราบเพื่อที่จะทำงานในขั้นตอนต่อไป โดยส่งข้อมูลต่างๆ ไปให้กับวิวเพื่อที่จะแสดงผลการร้องขอที่หน้าเว็บเพื่อที่จะแจ้งสถานะว่าผ่านหรือไม่ผ่าน ถ้าผ่านตัวควบคุมก็จะสั่งให้ Workflow ทำงานตามที่ผู้ใช้วางไว้ได้เลย แล้วก็ทำการบอกสถานะให้กับผู้ใช้ทางเว็บเป็นระยะ

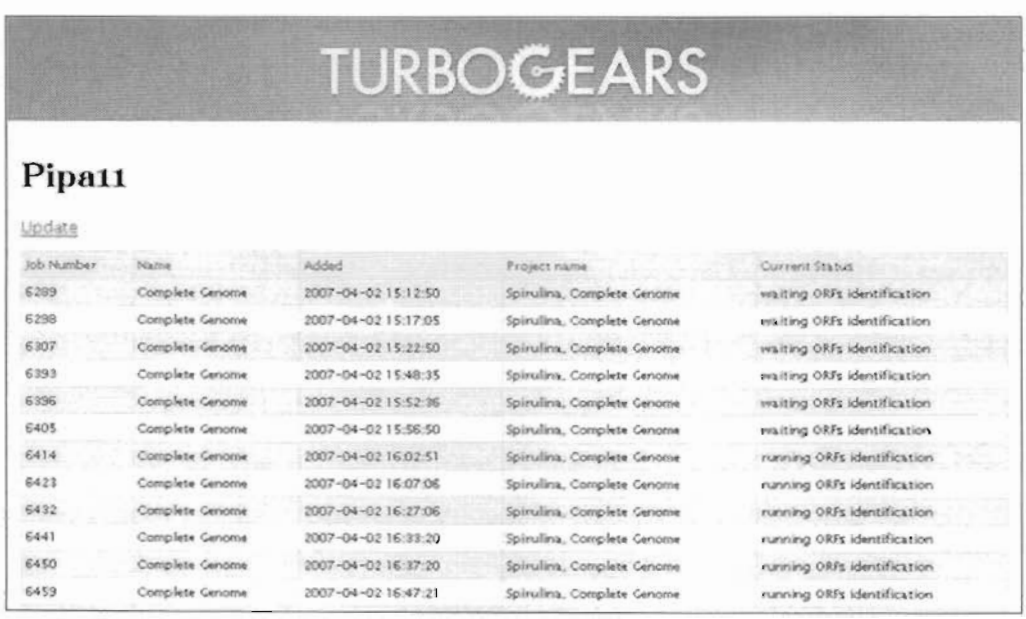
4. ขั้นตอนนี้จะเป็นการจัดการงานที่ผู้ใช้วางเอาไว้ เนื่องจากปัญหาของการที่ผลลัพธ์ของแต่ละโปรแกรมไม่สอดคล้องกัน จึงได้เลือกใช้ไฟล์เอกสารผลลัพธ์ในรูปแบบเอกสาร XML แทนที่จะนำข้อมูลจากฐานข้อมูลมาโดยตรง นั่นก็เพราะว่า ข้อมูลในฐานข้อมูลเหมาะสำหรับข้อมูลที่อยู่กับที่ กล่าวคือผู้ใช้เมื่อต้องการใช้ข้อมูลก็จะต้องมาที่ฐานข้อมูลแล้วทำการดึงข้อมูลไป แต่ด้วยธรรมชาติของข้อมูลในชีวสารสนเทศที่มีลักษณะกระจัดกระจาย ไม่สามารถทำอย่างนี้ได้เสมอไป ทำให้เลือกใช้ XML ที่ออกแบบมาสำหรับให้มีการเคลื่อนไหวของข้อมูลไปตามที่ต่างๆ อาทิเช่น แปลงเอกสาร XML ไปเก็บไว้ใน MySQL, Access, Postgres, Oracle หรือแปลงไปเป็นไฟล์เอกสารธรรมดาทั่วไปอย่างเช่น "*.txt" เป็นต้น

แต่อย่างไรก็ตามการเก็บข้อมูลลงฐานข้อมูลก็ยังมีข้อดีอีกมาก อาทิเช่น ความรวดเร็วในการสืบค้นข้อมูล ทำให้เลือกที่จะเก็บสถานะต่างๆ ของโปรแกรมในแต่ละสถานีไว้ที่ฐานข้อมูล ในโมเดลของ MVC นอกจาก Workflow ใช้เป็นฐานข้อมูลแล้ว ยังสามารถใช้เป็น Application Logic ได้อีกด้วย ดังนั้นถึงได้ออกแบบการทำงานในส่วนของโปรแกรมเอาไว้ที่โมเดล จากข้อที่แล้วเมื่อ Application Logic ของ Workflow ได้สั่งให้ Workflow ปฏิบัติงานตามที่ผู้ใช้วางไว้ หน้าที่อีกอย่างของ Workflow คือการแจ้งสถานะปัจจุบันให้ ตัวควบคุมทราบ เมื่อสถานะมีการเปลี่ยนแปลง เช่น เสร็จแล้ว หรือ เกิดข้อผิดพลาดบางอย่างที่ทำให้ทำงานต่อไม่ได้ ก็จะแจ้งมาที่ Workflow ให้ทราบ เพื่อที่ Workflow จะส่งข้อมูลเหล่านี้ไปที่ ตัวควบคุม แล้วส่งต่อไปให้วิว เพื่อแจ้งให้ผู้ใช้ทราบสถานะปัจจุบันที่เป็นอยู่

ด้วยวิธีการนี้ทำให้สามารถติดตามการทำงานที่ผู้ใช้วางไว้ได้ ซึ่งเพราะว่าข้อเสียอย่างหนึ่งของเว็บ HTML นั่นก็คือตัวมันเป็น Stateless หมายความว่า ไม่มีสถานะการทำงาน ทำให้ไม่สามารถติดตามสถานะการทำงานในแต่ละสถานีได้ ยกตัวอย่างเช่น สถานีประกอบสายลำดับเบสขึ้นมาใหม่, สถานีค้นหายีน และสถานีการค้นหาความเหมือน แต่ด้วยวิธีนี้สามารถติดตามได้โดยผ่าน ตัวควบคุมให้ตีความหมายหรือถอดรหัส หลังจากนั้นให้ โมเดลจัดการในเรื่องจัดเก็บข้อมูลตามแต่ความเหมาะสม หรือใช้จัดการโปรแกรมในระดับ Application Logic ให้ทำงานได้อย่างเป็นอัตโนมัติ แล้วให้วิวแสดงสถานะต่างๆ ให้ผ่าน HTML

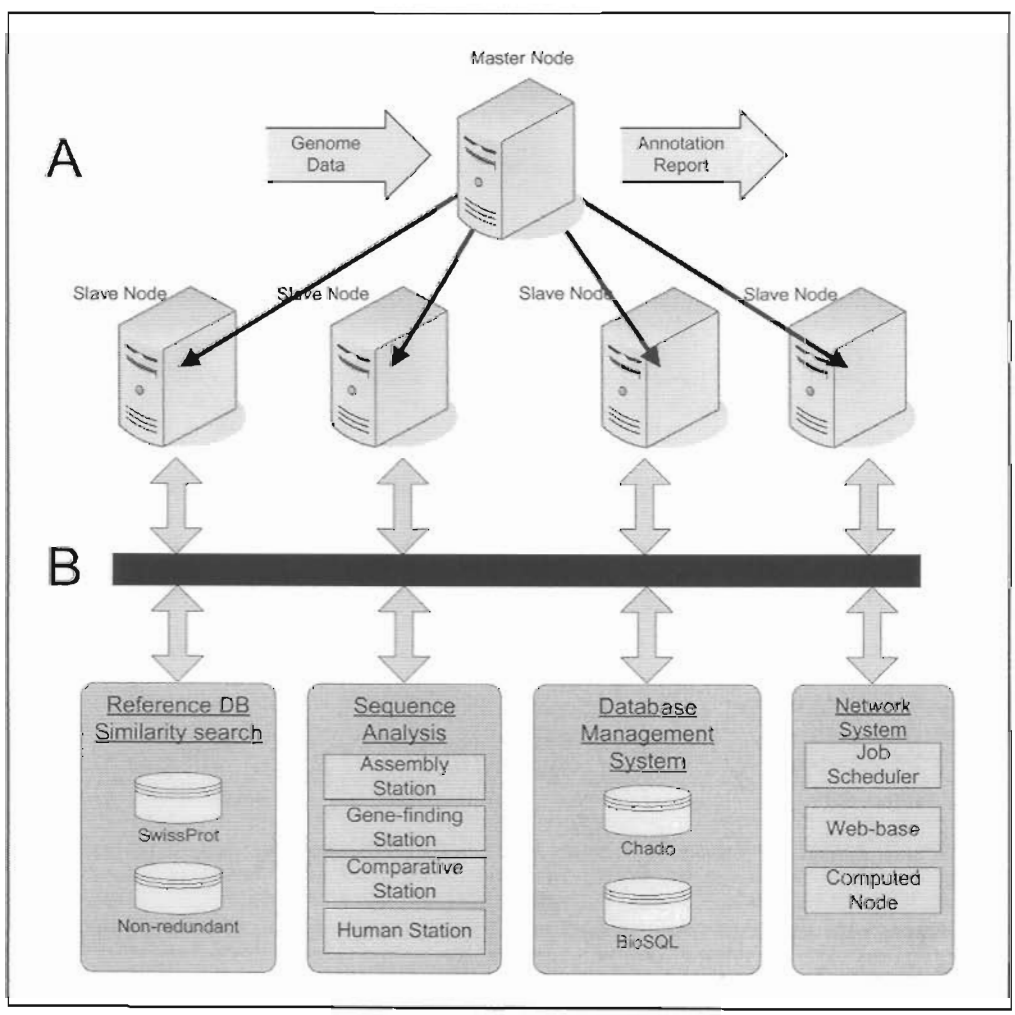
5. เมื่องานที่ผู้ใช้วางไว้ทำเสร็จแล้ว จะทำการเก็บลงฐานข้อมูลและเก็บอยู่ในรูป XML ถึงแม้ว่าจะเป็นการเก็บที่ซ้ำซ้อนกัน แต่ก็มีประโยชน์เมื่อเวลาเรียกใช้ สามารถนำมาใช้ได้ในทันที จึงสามารถชดเชยการเก็บที่ซ้ำซ้อนกันนี้ได้ ฐานข้อมูลที่ใช้คือ ฐานข้อมูล BioSQL ซึ่งระบบการระบุตำแหน่งและกำหนดหน้าที่ยีนแบบอัตโนมัติสำหรับวิธีการลำดับดีเอ็นเอบนจีโนมของ อิงกับผลลัพธ์ที่ได้จากการค้นหาความเหมือนบน NR ของ NCBI หากผลลัพธ์ที่ได้ตรงกับค่า E-Value กับค่า Bit Score ที่ต้องการก็จะทำการโหลดไฟล์ GenBank ของลำดับเบสที่ที่ตรงกันมาเก็บไว้ที่ระบบเอง หรือว่าจะโหลดเมื่อต้องการก็ได้

6. ฟอร์มข้อมูลต่างๆ ที่ต้องการแสดงผ่านวิว จะถูกส่งมาจาก Instance ซึ่งในกรณีนี้คือสิ่งที่ TGs สร้างขึ้นมาเพื่อคอยจัดการสำหรับแต่ละผู้ใช้ ให้สามารถใช้ ตัวควบคุมตัวเดียวกันแต่คนละ Instance การแสดงผลดังแสดงในภาพที่ 4-16 ประกอบด้วย สถานะปัจจุบัน, หมายเลขงาน, ชื่อ, เวลาที่สั่งทำงาน, ชื่อโครงการ เหล่านี้แสดงอยู่ในช่อง "Current Status", "Job Number", "Name", "Added" และ "Project Name" ตามลำดับ



ภาพที่ 4-16 ภาพแสดงหน้าต่างติดตามการทำงานด้วย TGs

สรุประบบการระบุตำแหน่งและกำหนดหน้าที่ยีนแบบอัตโนมัติสำหรับวิธีการล่าดับดีเอ็นเอบนจีโนม ที่ต้องการต้องสามารถรับมือกับความไม่สอดคล้องของข้อมูล, สามารถรักษาความปลอดภัยได้ในระดับหนึ่ง และสามารถนำโค๊ดกลับมาใช้ได้อีกครั้ง สามารถทำได้ด้วยระบบของ ด้วยการเก็บข้อมูลในรูปแบบเอกสาร XML และเก็บไว้ในฐานข้อมูลช่วยแก้ไขปัญหาเมื่อใช้ข้อมูลเพียงแต่รูปแบบใดแบบหนึ่งเท่านั้น นั่นก็เพราะว่าเอกสาร XML ถึงแม้ว่าจะถูกออกแบบให้ใช้งานสำหรับการแลกเปลี่ยนข้อมูลระหว่างระบบก็ตาม แต่ก็มีปัญหาที่ ถ้าในกรณีเอกสาร XML มีขนาดใหญ่มาก ก็จะทำให้การส่งข้อมูลเป็นไปอย่างล่าช้า ตรงกันข้ามกับฐานข้อมูล ที่ข้อมูลนอนรออยู่กับที่รอให้ผู้ใช้มาเรียกให้ และความรวดเร็วในการสืบค้นของฐานข้อมูลก็เร็วกว่ามาก เพียงแต่มันไม่เหมาะสำหรับการแลกเปลี่ยนข้อมูลกันระหว่างระบบ จึงใช้ข้อนี้ชดเชยข้อได้เปรียบเสียเปรียบซึ่งกันและกัน ด้วยการเลือกใช้กับข้อมูลให้เหมาะสม



ภาพที่ 4-17 สถาปัตยกรรมของระบบ

สามารถรักษาความปลอดภัยในการใช้งานได้โดยมีระบบล็อกอินเข้าระบบ ซึ่งเฉพาะผู้ที่มีสิทธิ์ใช้งานเท่านั้นจึงสามารถสั่งทำงานได้ และข้อมูลของผู้ใช้แต่ละคนถ้าหากผู้ใช้ไม่อนุญาตให้ ข้อมูลของตน

Retrieving GenBank file in 7 lines

```
from Bio import GenBank
gi=['16229']
ncbi_dict = GenBank.NCBIDictionary('nucleotide', 'genbank')
gb_record = ncbi_dict[gi[0]]
f = open('r001.gb','w')
f.write(gb_record)
f.close()
```

ภาพที่ 4-18 ตัวอย่างรับข้อมูล GenBank ด้วย Biopython

แบ่งปันร่วมกับคนอื่นหรือไม่มีผู้ร่วมงาน ข้อมูลก็จะถูกล็อคให้ไม่สามารถมองเห็น หรือมองเห็นได้แต่ไม่สามารถแก้ไขได้ตามแต่ความต้องการ นอกจากนี้ยังใช้ภาษา Python ในการเขียนขึ้นมาก็จะมีการโจมตีน้อยกว่าคนใช้ภาษา PHP ดังเห็นตัวอย่างมากมายที่มีแม้แต่แปะโค๊ดสำหรับเจาะเข้าไปในกระดานสาธารณะอย่างเช่น phpBB อย่างแพร่หลายอีกด้วย ทำให้ผู้ดูแลระบบจำเป็นต้องทำการปรับปรุงเวอร์ชั่นของโปรแกรมที่เขียนด้วยภาษานี้อยู่เสมอๆ

และด้วยความสามารถของ TGs ทำให้สามารถนำโค๊ดกลับมาใช้งานได้ใหม่ โดยที่ไม่ต้องไปแก้โค๊ด เพราะ Python เป็นภาษาในเชิงวัตถุ สามารถสืบทอดคลาส หรือเขียนคลาสขึ้นมาใหม่แต่เขียน Method ใหม่ขึ้นมาเองให้เหมาะกับงานที่ก็ได้ นอกจากนี้ TGs ยังจัดเครื่องมือต่างๆ มากมายช่วยพัฒนางานได้อย่างรวดเร็ว มีแม่แบบให้ใช้มากมายซึ่งช่วยย่นระยะเวลาในการออกแบบของได้เป็นอย่างมาก ตัวที่คล้ายๆ กับ TGs ก็มีอย่างเช่น Ruby on Rails ซึ่งเขียนด้วยภาษา Ruby ก็ใช้แนวคิด MVC เช่นเดียวกับ TGs ซึ่งในเรื่องนี้ก็ขึ้นกับนักพัฒนาว่าจะเลือกใช้ภาษาใดหรือเครื่องมืออะไรมาช่วยให้เขาทำงานได้อย่างประสิทธิภาพที่สุด

4.5 สถาปัตยกรรม

ระบบการระบุตำแหน่งและกำหนดหน้าที่ยีนแบบอัตโนมัติสำหรับวิธีการลำดับดีเอ็นเอบนจีโนมของทำงานอยู่บน ระบบคลัสเตอร์ ประกอบไปด้วยเครื่องคอมพิวเตอร์ 64 Bit Intel Core Duo จำนวน 4 โหนด และใช้ Sun Grid Engine สำหรับสั่งทำงานบนคลัสเตอร์ ดังแสดงในภาพที่ 4-17 ที่แต่ละโหนดจะทำการลงโปรแกรมที่เครื่องคอมพิวเตอร์เครื่องนั้น โดยเฉพาะอย่างยิ่ง ฐานข้อมูลที่ต้องลงไว้ที่โหนดลูกเพราะว่า จะได้โหลดอ่าน/เขียน ที่โหนดลูกได้เร็วกว่าการอ่าน/เขียนผ่าน NFS ซึ่งการทำแบบนี้จะช่วยลดภาระงานที่เครื่องแม่ข่ายได้เป็นอย่างมาก ให้โหนดลูกทำการคำนวณ หลังจากคำนวณเสร็จก็จะส่งผลลัพธ์กลับไปที่เครื่องแม่เพื่อทำงานในขั้นต่อๆ ไป

สรุปในสถาปัตยกรรมที่ออกแบบ เป็นการออกแบบที่กระจายงานไปที่เครื่องลูก ให้เครื่องลูก

เป็นเครื่องทำการคำนวณ โดยอาศัย Sun Grid Engine เป็นตัวกระจายงานที่สามารถตรวจสอบได้ว่าเครื่องลูกเครื่องใดมีงานทำอยู่ ก็จะกระจายงานไปให้เครื่องลูกอื่นที่ยังว่างอยู่ ทำให้สามารถกระจายงานได้อย่างมีประสิทธิภาพ นอกจากนี้ยังโหลดฐานข้อมูลสาธารณะจากที่ต่างๆ ดังแสดงอยู่ในภาพที่ 4-17 จะเป็นการช่วยเพิ่มความเร็วในการสืบค้นจากฐานข้อมูลให้มาค้นหาที่ระบบของเองแทนที่จะติดต่อผ่านเครือข่ายซึ่งขึ้นกับความเร็วของเครือข่าย ณ ขณะนั้น และยังเป็นเพิ่มความคงทนของข้อมูล ถึงแม้ว่าข้อมูลจากฐานข้อมูลสาธารณะล่ม แต่ข้อมูลที่เครือข่ายก็ยังมีอยู่ ในทางกลับกันก็เช่นกัน ทำให้เพิ่มต่อความทนทานของข้อมูล แต่ก็อาจจะมีปัญหาเรื่องเวอร์ชั่นของฐานข้อมูลที่มีการปรับปรุงทุก 3 เดือน หรือตามแต่ชนิดของฐานข้อมูล ทำให้ต้องการปรับปรุงให้ทันสมัยอยู่เสมอ

ส่วนระบบจัดการฐานข้อมูลมีการการจัดเก็บข้อมูลแบบ เก็บข้อมูลลงฐานข้อมูลกัน ในรูปแบบเอกสาร XML ให้ขึ้นกับชนิดของข้อมูลที่ใช้ ในด้านการวิเคราะห์สายลำดับเบสหรือ Sequence Analysis ของก็เป็นการรวมกันระหว่างสถานีต่าง 4 สถานีได้แก่ สถานีประกอบสายลำดับเบสขึ้นมาใหม่, สถานีการค้นหายีน, สถานีการค้นหาความเหมือนของสายลำดับเบสในฐานข้อมูล และ สถานีตรวจสอบความถูกต้องด้วยนักวิจัย เพื่อให้แน่ใจว่าข้อมูลที่ได้มีความถูกต้องตามที่ตั้งเอาไว้

บทที่ 5
สรุป

ด้วยความก้าวหน้าของเทคนิคการเชื่อมต่อลำดับเบส และการค้นหายีนทั่วทั้งจีโนมทั้งแบบการค้นหายีนแบบภายนอก, ภายใน หรือแบบผสม รวมถึงเทคนิคการค้นหาความเหมือน ได้นำมาซึ่งการสร้างระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนแบบต่างๆ ที่สามารถนำไปประยุกต์ใช้ในสิ่งมีชีวิตทั้งยูคาริโอต หรือโปรคาริโอต ได้แก่ Ensembl, Mage, RiceGAAS และ BaSys

วิทยานิพนธ์ฉบับนี้เสนอระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนแบบอัตโนมัติสำหรับวิธีการลำดับดีเอ็นเอบนจีโนมที่สามารถประยุกต์ได้ทั้งยูคาริโอตและโปรคาริโอต อันประกอบไปด้วย การลำดับเบส, การตีความกรอบเปิดการอ่าน, การระบุตำแหน่งและกำหนดหน้าที่ของกรอบเปิดการอ่านโดยหาความเหมือนในฐานข้อมูลต่างๆ และการตรวจสอบความถูกต้องโดยนักชีววิทยา เพื่อเพิ่มประสิทธิภาพในการค้นหายีนในจีโนมให้มีความแม่นยำ, รวดเร็ว และสามารถกระจายการทำงานได้อย่างมีประสิทธิภาพ

จากการศึกษาระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนมแบบอัตโนมัติที่ผ่านมา จนสร้างระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนมใหม่แบบอัตโนมัติด้วย TGs แบบ Model-View-Control มาประยุกต์ใช้สำหรับการระบุตำแหน่งและกำหนดหน้าที่ของยีนแบบอัตโนมัติช่วยพัฒนาในแง่ดังต่อไปนี้

การพัฒนาระบบพบว่าทำให้ระบบพัฒนาตัวได้อย่างรวดเร็ว โดยเฉพาะในด้านการวางกระบวนการทำงานแบบอัตโนมัติ การนำโปรแกรมมาใช้งาน ซึ่งเมื่อนำโปรแกรมมาใช้งานร่วมกับการกระจายการทำงานของข้อมูลได้แล้ว ก็จะยิ่งเพิ่มประสิทธิภาพในด้านความเร็วการระบุตำแหน่งและกำหนดหน้าที่ของยีนในจีโนม นอกจากนี้ยังสามารถนำเอาเทคนิคนี้ไปปรับปรุงเพื่อประยุกต์ใช้กับการออกแบบข้อมูลที่มีความแตกต่างกันให้ทำงานร่วมกัน โดยออกแบบการใช้งานด้วยโมเดลของฐานข้อมูล ร่วมกับโมเดลที่ใช้ควบคุมการทำงานของโปรแกรมมาจัดการให้เกิดความสมดุลของข้อมูลในระบบได้

ขั้นตอนการทำงาน 1) การเชื่อมต่อลำดับเบส เมื่อนำโปรแกรมค้นหาความเหมือนใช้งานร่วมกับการค้นหายีนภายใน ทำให้ทราบถึงข้อดีและข้อด้อยของการค้นหายีนที่นำโปรแกรมแบบใดแบบหนึ่งมาใช้งาน 2) การนำโปรแกรมค้นหาความเหมือนใช้ร่วมกับการเชื่อมต่อลำดับเบส พิสูจน์ให้ทราบถึงจุดเด่นของความเหมือน ซึ่งช่วยนำการเปรียบเทียบระหว่างโปรแกรมการเชื่อมต่อลำดับเบสมาแก้ไขการเชื่อมต่อที่ผิดพลาดได้อย่างมีประสิทธภาพ และ 3) การกำหนดหน้าที่ด้วยการค้นหาความหมือนโดยการแปลงลำดับเบสเป็นลำดับโปรตีน ซึ่งทำให้หาการหานัยสำคัญของข้อมูลได้อย่างถูกต้อง

การนำตัวควบคุมของ Model-View-Control ใน TGs และในส่วนของการใช้โมเดลทำงานร่วมกับโมเดลฐานข้อมูล BioSQL ในวิทยานิพนธ์ฉบับนี้ยังสามารถนำไปขยายผล เพื่อออกแบบระบบ

ต่างๆ ในการระบุตำแหน่งและกำหนดหน้าที่ของยีนจีโนมที่มีประสิทธิภาพและมีความเหมาะสมตามแต่สิ่งมีชีวิต

ความปลอดภัย โดยนำข้อมูลของผู้ใช้มาใช้งานที่ระบบส่วนตัว ทำให้แน่ใจที่ข้อมูลจะไม่ถูกนำไปประยุกต์ใช้หรือดัดแปลงแก้ไขโดยไม่ได้รับอนุญาต และยังสามารถกำหนดระดับอภิสิทธิ์การใช้งานได้เอง เพื่อที่จะกำหนดหน้าที่ๆ ทำได้กับทำไม่ได้ เพื่อที่จะให้ทำงานแบบโปรเจคได้อย่างเต็มประสิทธิภาพ

ปัจจุบันโปรแกรมซึ่งนำมาใช้งานในระบบมีไม่มากพอ โปรแกรมที่จะนำมาเพิ่มได้แก่ โปรแกรมเกี่ยวกับคำนวณการใช้โคดอน และการนำโปรแกรมค้นหาความเหมือน เช่น BLAST ทำงานร่วมกับการศึกษาเชิงเปรียบเทียบแบบทั่วทั้งจีโนม จะช่วยเพิ่มความสามารถการค้นหายีนของระบบมากยิ่งขึ้น

นอกจากนำ XML เป็นแม่แบบสำหรับส่งผ่านข้อมูลระหว่างกระบวนการ ในอนาคตจะมีการวางมาตรฐานกระบวนการทำงานใหม่ โดยนำ XML มาใช้เพื่อส่งต่อการทำงานระหว่างกระบวนการให้ได้อย่างถูกต้อง และนำ Javascript หรือ Ajax พัฒนาหน้าต่างการใช้งานของระบบให้สามารถรองรับการร้องขอการทำงานแบบ Asynchronous ได้อย่างมีประสิทธิภาพ

สุดท้ายพัฒนาการเขียนโปรแกรมที่ช่วยนักชีววิทยาทำงานในส่วนของตนเองได้อย่างเป็นระบบช่วยรองรับการเพิ่มหรือการนำโปรแกรมออกได้จริง โดยนำเทคโนโลยีเว็บเซอร์วิสมาประยุกต์ใช้ สร้างการติดต่อโปรแกรมระหว่างฝั่งผู้ใช้กับฝั่งเครื่องแม่ข่าย โดยเขียนรายละเอียดโปรแกรมที่ต้องการนำมาประยุกต์ให้ผู้ใช้สามารถนำไปใช้กับระบบคอมพิวเตอร์ของผู้ใช้ได้อย่างมีอิสระภาพ

วิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการและทำการลงมือปฏิบัติจริงเพื่อสร้างระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนมใหม่ ที่สามารถใช้งานได้จริง ซึ่งในปัจจุบันยังไม่มีโปรเจคสำหรับระบบดังกล่าวที่รองรับทั้งสิ่งมีชีวิตโปรคาริโอตและยูคาริโอต ทำให้การพัฒนาระบบดังกล่าวไม่สามารถพัฒนาได้อย่างรวดเร็ว ดังนั้นด้วยการนำ TurboGears ซึ่งใช้ภาษา Python ทำให้การพัฒนาระบบเป็นไปได้อย่างราบรื่นด้วยเทคนิค MVC มาประยุกต์ใช้ทำให้ ระบบมีประสิทธิภาพ, เป็นสัดส่วน และสามารถนำโค๊ดโปรแกรมกลับมาใช้ใหม่ได้

เอกสารอ้างอิง

1. Rouz, P., Pavy, N. and Rombauts, S. "Genome annotation : Which tools do we have for it?." *Plant Biology*. 2 (April 1999) : 90–95.
2. Gao, H. T., Hayes, J. H. and Cai, H. "Integrating Biological Research through Web Services." *IEEE computer*. (2005).
3. Fickett, J. W. "The Gene Identification Problem : An Overview for Developers." *Computers & Chemistry*. 20 (1996) : 103–118.
4. Nelson, D. L. and Cox, M. M. *Lehninger Principles of Biochemistry*. 2 ed. New York : W. H. Freeman. 1993.
5. Vallenet, D., et al. "MaGe : A Microbial Genome Annotation System Supported by Synteny Results." *Nucleic Acids Reseach*. 34 (2006) : 53–65.
6. Domselaar, G., et al. "BASys : A Web Server for Automated Bacterial Genome Annotation." *Nucleic Acids Reseach*. 33 (2005) : w455–w459.
7. Sakata, K., et al. "RiceGAAS : An automated annotation system and database for rice genome sequence." *Nucleic Acids Reseach*. 30 (2002) : 98–102.
8. Curwen, V., et al. "The Ensembl Automatic Gene Annotation System." *Genome Res*. 14 (2004) : 942–950.
9. Overbeek, R., et al. "The ERGO genomes analysis and discovery system." *Nucleic Acids Reseach*. 31 (2003) : 164–171.
10. Berriman, M. and Rutherford, K. "Viewing and annotating sequence data with Artemis." *Brief Bioinformatics*. 4 (2003) : 124–132.
11. Huang, X., Wang, J., Aluru, S., Yang, S. P. and Hillier, L. "PCAP : A Whole-Genome Assembly Program." *Genome Research*. 23 (2003) : 2164–2170.
12. Huang, X. and Madan, A. "CAP3 : A DNA Sequence Assembly Program." *Genome Research*. 9 (1999) : 868–877.
13. Sakata, K., et al. "RiceHMM : Gene domain prediction program for rice genome sequence." *Abstracts of 4th Annual Conference on Computational Genomics*. (2000).

14. อภิชาติ วรรณวิจิตร. *"5 ปีของการเข้าร่วมโครงการวิจัยจีโนมข้าวนานาชาติประเทศไทยได้อะไร."* [ออนไลน์] [สืบค้น วันที่ 10 เมษายน 2007] จาก http://knowledge. biotec.or.th/doc_-upload/20034410218.pdf.

15. Meyer, F., et al. "GenDB–An open source genome annotation system for prokaryotic genomes." *Nucleic Acids Reseach.* 31 (2003) : 2187–2195.

16. Bocs, S., Cruveiller, S., Vallenet, D., Nuel, G. and Medigue, C. "AMIGene : Annotation of Microbial Genes." *Nucleic Acids Reseach.* (2003) : 3723–3726.

17. Iliopoulos, I., et al. "Evaluation of annotation strategies using an entire genome sequence." *Bioinformatics.* (2003) : 717–726.

18. Borodovsky, M., E.Rudd, K. and V.Koonin, E. "Intrinsic and extrinsic approaches for detecting genes in bacterial genome." *Nucleic Acids Reseach.* (1994).

19. Fickett, J. W. "Finding genes by computer : The state of the art." *Trends in Genetics.* 12 (August 1996) : 316–320.

20. "Computational gene programs." [ออนไลน์] [สืบค้นวันที่ 23 มกราคม 2007] http://www.nslij-genetics.org/gene/programs.html.

21. Claverie, J. M. "Computational methods for the identification of genes in vertebrate genomic sequences." *Human Molecular Genetics.* 6 (1997) : 1735–1744.

22. Guigo, R. "Computational gene identification : An open problem." *Computers & Chemistry.* 21 (1997) : 215–222.

23. Haussler, D. "Computational genefinding." *Trends in Biotechnology.* 16 (1998) : 12–15.

24. Burge, C. B. and Karlin, S. "Finding the genes in genomic DNA." *Structural Biology.* 8 (June 1998) : 346–354.

25. Salzberg, S., et al. "Improved microbial gene identification with GLIMMER." *Nucleic Acids Reseach.* 27 (1999) : 4636–4641.

26. John Besemer, A. L. and Borodovsky, M. "GeneMarkS : A self-training method for prediction of gene starts in microbial genomes. Implication for finding sequence motifs in regulatory regions." *Nucleic Acids Reseach.* 29 (2001) : 2607–2618.

27. Rogic, S., Ouellett, B. F. and Mackworth, A. K. "Improving gene recognition accuracy by combining predictions from two gene-finding programs." *Bioinformatics.* 18 (2002) : 1034–1045.

28. Salzberg, S., Delcher, A., Kasif, S. and White, O. "Microbial gene identification using interpolated markov models." *Nucleic Acids Reseach.* 26 (1998) : 544–548.

29. Ramn, M., Dangoor, K. and Sayfan, G. *Rapid Web Application with TurboGears : Using Python to Create Ajax-Power Sites*. New York : Prentice Hall. 2006.

30. Freeman, E., Freeman, E., Sierra, K. and Bates, B. *Head First Design Patterns*. first ed. New York : O' Reilly. October 2004.

ประวัติผู้วิจัย

ชื่อ : นายพัทธพล เปยานนท์
ชื่อวิทยานิพนธ์ : ระบบการระบุตำแหน่งและกำหนดหน้าที่ของยีนบนจีโนมใหม่แบบอัตโนมัติ
สาขาวิชา : วิศวกรรมไฟฟ้า

ประวัติ

เกิดเมื่อวันที่ 16 ตุลาคม พ.ศ. 2522 ณ ประเทศไทย บิดาชื่อนายวิศิษฐ์ เปยานนท์ มารดาชื่อนางผ่องศรี เปยานนท์ มีพี่น้องรวมทั้งสิ้น 2 คน เป็นบุตรคนโต สำเร็จการศึกษาในระดับมัธยมศึกษาตอนต้นจากโรงเรียนวัดด่านสำโรง ระดับมัธยมศึกษาตอนปลายจากโรงเรียนสตรีสมุทรปราการ และระดับอุดมศึกษาในสาขาวิศวกรรมไฟฟ้าจากมหาวิทยาลัยเกษตรศาสตร์