THAI Bioinformatics
Vol 11 July 2010

Genomics
ปฐมบทแห่งชีววิทยาในยุค omics

เราจะทำ Reverse Complement
จาก DNA ด้วย Python ได้อย่างไร

CONTENTS

THAI BIOINFORMATICS Vol 11 July 2010

1 CONTENTS

2 EDITORIAL

HIGHLIGHT

4 Genomics
Genomics: ปฐมบทแห่งชีววิทยาในยุค omics
ชิเน ธำมรงค์ธรรม
เกิดอะไรขึ้น เมื่ออณูชีววิทยาแบบเดิมๆ ที่ศึกษายีนทีละยีน เปลี่ยนมาศึกษายีนทั้งจีโนม

SOFTWARE & PROGRAMING

7 Python Programing
ทำ reverse complement สาย DNA ด้วย Python
ประพัฒน์ สุริยผล
อีกหนึ่งโปรแกรมที่ใช้งานได้จริง กับการหา reverse complement ของ DNA

SCIENCE & SOCIETY

11 Joy and Fun
SUDOKU
ประเวช อรรจวัฒนวงศ์
Sudoku เกมยอดฮิต ที่หลายคนยังไม่รู้วิธีเล่น

4

7	9		1	2		8	3	
							4	
2	4		3		9		6	
	6			3				
	3			4	5	7	2	8
		7			2	3	9	
3					1			
5		4		6	8	9		
	2	9		7		4		

11

FROM THE COVER

Popular belief holds that chameleons change color to blend in with their surrounding habitat. However, in many chameleons color change can in fact indicate the reptile's "mood." Panther chameleons are known for their ability to shift among a range of bright colors such as turquoise, red and orange, making them prize pets for herp collectors. This juvenile panther chameleon caught at night has relatively dull coloration. Perhaps he was just bored?

Photo: Montagne d'Ambre National Park, Madagascar

EDITORIAL

Editor-in-chief
ประเวช อรรจวัฒนวงศ์

Editor
ประเวช อรรจวัฒนวงศ์

Assistant Editor
จิตสุพางค์ รอดบำเรอ
ศิริขวัญ พลประทีป

Editorial Advisor
ประพัฒน์ สุริยผล

Graphic Design
ประเวช อรรจวัฒนวงศ์

Online Editor
ประพัฒน์ สุริยผล

Facebook Moderator
ฝน นิลเขต

Advertising Team
จิตสุพางค์ รอดบำเรอ
ศิริขวัญ พลประทีป
ฝน นิลเขต

Academic Team
ประพัฒน์ สุริยผล
ภัสสร วรรณพินิจ
เจษฎา เด่นดวงบริพันธ์
ธีรพันธ์ เหล่าเมตตาจิตต์
คเณศ วงศ์ระวี
สหัสชัย ตั้งตรงทรัพย์
ธีรศักดิ์ เอโกมล

Cover Photo
Allison Perrigo

ในที่สุด **THAI** BIOINFORMATICS ฉบับนี้ก็วางแผงได้เสียที เล่นเอาผมเหงื่อตกกันเลยทีเดียวครับ สำหรับฉบับนี้ผมต้องขอออกตัวไว้ก่อนว่า เราทำออกมาเป็นฉบับเล็กๆ เนื่องจากทีมงานได้ใช้เวลากับการเตรียมงาน Thai Bioinformatics Network Meeting ครั้งที่ 1 ในวันที่ 30 กรกฎาคม พ.ศ. 2553 นี้ครับ

ราวๆ ต้นเดือนกรกฎาคมนี้ ผมมีโอกาสได้อ่านข่าว online จากคอลัมน์ CEO Dialogue ของหนังสือพิมพ์กรุงเทพธุรกิจ ฉบับวันที่ 18 มิถุนายน พ.ศ. 2553 ซึ่งเป็นเรื่องเล่าว่าคุณธนินท์ เจียรวนนท์ ได้กล่าวให้โอวาทกับนักเรียนโรงเรียนนานาชาติ คอนคอร์เดียน ซึ่งในตอนท้าย ผู้เขียนได้เล่าว่าที่หน้าโรงเรียนมีป้ายเขียนคำพูดไว้ประโยคหนึ่งว่า...

"Concordian promotes academic excellence while nurturing young people to become moral and intellectual leaders, people of dignity, integrity and compassion, who want to make a difference in the world."

ทำให้ผมนึกไปถึงนักเรียนไทยในหลายยุคหลายสมัยที่มีแต่การแข่งขันกันในเรื่องการเรียน แต่กลับหลงลืมไปว่าสิ่งที่สำคัญยิ่งกว่าการเรียนรู้ นั่นก็คือ คุณงามความดี จริยธรรม และการนำความรู้ไปช่วยเหลือสังคมด้วยความเสียสละและบริสุทธิ์ใจ ยิ่งผมเห็นการแข่งขันทางการศึกษาที่มีมากขึ้นในปัจจุบัน ก็แอบวิตกกังวลไปก่อนไม่ได้ว่าในอนาคต เด็กน้อยเหล่านี้จะเติบโตขึ้นมาเป็นผู้ใหญ่ที่เสียสละได้มากน้อยเท่าไหร่กัน แต่ในความกลัวของผม ก็มีแสงสว่างเรืองรองอยู่เช่นกัน ไม่ใช่ใครที่ไหนครับ ก็เป็นน้องๆ ที่ผมได้ร่วมงานด้วยนี่แหละครับ ที่ผมเขียนเช่นนี้ ไม่ได้มายกยอคนกันเองแต่อย่างใด แต่กลับชื่นชมด้วยความจริงใจ ที่อย่างน้อยคนไทยกลุ่มที่เรียกได้ว่าเป็นคนเก่งระดับแนวหน้า ก็ยังมีคนดีที่เอื้อเฟื้อเผื่อแผ่และเสียสละต่อสังคมอยู่ด้วย

สำหรับเนื้อหาในฉบับนี้ เราได้นำเสนอบทความเรื่อง Genomics เป็นเรื่องเด่นประจำเล่มครับ และ Python Programming ก็เป็นเรื่องของการเขียนโปรแกรมเพื่อเปลี่ยน DNA sequence ให้กลายเป็น DNA sequence แบบ reverse complement ด้วยโปรแกรมที่สั้นๆ ง่ายๆ และส่งท้ายกับเกมดังที่หลายคนชื่นชอบ นั่นก็คือ sudoku ครับ

ท่านผู้อ่านที่ที่มีข้อติชม แนะนำ หรืออยากให้เรานำเสนอเรื่องราวอะไรเป็นพิเศษ ก็เช่นเดิมครับ ท่านสามารถ post ข้อความเพื่อแนะนำและแสดงข้อคิดเห็นได้ตลอดเวลานะครับ ผมและทีมงานยินดีรับฟังเพื่อนำมาปรับปรุงให้ดียิ่งขึ้นครับ

Contact Us
www.thaibioinfonetwork.org

ร่วมเดินทางไปกับเรา

สู่โลกแห่งวิทยาการอันทันสมัย วิทยาศาสตร์แนวใหม่ที่ผสมผสานขึ้นมาจากศาสตร์แห่งการคำนวณและศาสตร์ที่ค้นคว้าหาคำตอบของชีวิต พบกับคอลัมน์น่าสนใจมากมาย อาทิเช่น ***Highlight*** *ที่นำเสนอเรื่องราวใหม่ๆ ในโลกแห่ง bioinformatics เรียนรู้การเขียนโปรแกรมคอมพิวเตอร์ไปกับ* ***Programing*** *และไปทำความรู้จักบุคคลในวงการกับ* ***Talk to...***

นอกจากนี้ยังมีคอลัมน์ใหม่ๆ อีกมากมายที่จะทำให้ท่านเพลิดเพลินไปกับโลกแห่ง bioinformatics

COMING SOON...

ติดต่อ:

THAI BIOINFORMATICS

www.thaibioinfonetwork.org

THAI Bioinformatics

นิตยสารออนไลน์ รายเดือน สำหรับคนไทยที่สนใจ bioinformatics

HIGHLIGHT

Genomics

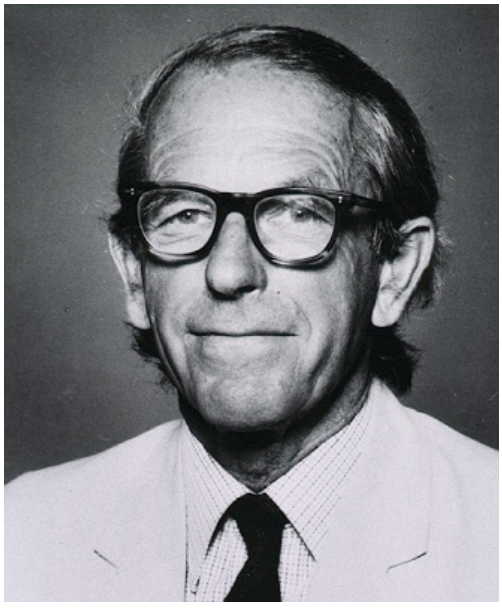
Genomics: ปฐมบทแห่งชีววิทยาในยุค omics

ชิเน ธำมรงค์ธรรม

เมื่อพูดถึงจีโนมิคส์ (genomics) ผมเชื่อว่าผู้อ่านเกือบทุกท่านเคยได้ยินคำนี้กันมาแล้ว หลายๆ ท่านอาจมีความรู้และความเข้าใจเกี่ยวกับจีโนมิคส์เป็นอย่างดี จีโนมิคส์เป็นเรื่องราวของการศึกษาเกี่ยวกับจีโนมของสิ่งมีชีวิต ทั้งในส่วนลำดับเบสทั้งหมดในสายดีเอ็นเอในเซลล์สิ่งมีชีวิต การศึกษาแผนที่พันธุกรรมในระดับจีโนมของสิ่งมีชีวิต และการศึกษาความสัมพันธ์ระหว่างดีเอ็นเอภายในเซลล์สิ่งมีชีวิต ยีนทั้งหมด และลักษณะทางพันธุกรรมของสิ่งมีชีวิตที่ถูกกำหนดโดยปัจจัยเหล่านี้ โดยจีโนมิคส์จะเป็นการศึกษาในระดับที่ต่างกันกับอณูชีววิทยาหรืออณูพันธุศาสตร์ ซึ่งเป็นวิทยาการที่ใช้ในการศึกษาโครงสร้าง บทบาทและหน้าที่ของยีนเพียงยีนเดียว หรือศึกษายีนจำนวนไม่มาก

นอกจากจีโนมิคส์จะเป็นการมองภาพกว้างในระดับจีโนมแล้ว ยังมีการเชื่อมโยงจีโนมิคส์เข้ากับวิทยาการแขนงอื่นด้วย จนเกิดเป็นวิทยาการอื่นๆ แขนงย่อยๆ ที่เกี่ยวเนื่องกับการศึกษาในระดับจีโนมตามมา เช่น (1) functional genomics (2) comparative genomics (3) structural genomics (4) metagenomics และ (5) pharmacogenomics เป็นต้น

จีโนมิคส์เริ่มเป็นที่กล่าวถึงพร้อมๆ กับวิทยาการสาขาหนึ่ง นั่นก็คือ ชีวสารสนเทศ (bioinformatics) ในวงการวิทยาศาสตร์



ศาสตราจารย์เฟดเดอริก แซงเกอร์

ดร. เฟรดเดอริค แซงเกอร์ นักชีวเคมีชาวอังกฤษ ผู้ได้รับรางวัลโนเบลสาขาเคมีถึง 2 ครั้ง ครั้งแรกในปี 1958 โดยได้รับจากผลงานการคิดค้นวิธีการหาชนิดของกรดอะมิโนตัวแรกในสายโพลิเปปไทด์ (N-terminal amino acid residue) โดยวิธีการที่คิดค้นได้สามารถนำมาใช้หาลำดับกรดอะมิโนของอินซูลิน ครั้งที่ 2 ในปี 1980 โดยได้รับจากผลงานการคิดค้นวิธีการหาลำดับเบสของสายดีเอ็นเอ (Sanger's Dideoxy chain termination DNA sequencing method) โดยในครั้งที่ 2 ได้รับรางวัลร่วมกับ ดร. วอลเตอร์ กิลเบิร์ท (Dr. Walter Gilbert) จากผลงานการคิดค้นวิธีการหาลำดับเบสของสายดีเอ็นเออีกวิธีหนึ่ง (Maxam-Gilbert chemical cleavage method) ทั้งสองวิธีเป็นการผลงานการคิดค้นจากนักวิทยาศาสตร์คนละกลุ่ม แต่ได้รับการพิจารณาให้ได้รับรางวัลร่วมกัน

นอกจากนี้เรื่องราวของจีโนมิคส์ก็ยังถูกนำไปเป็นส่วนหนึ่งในวงการบันเทิง หลายทุกคนยังจำภาพยนต์ฟอร์มยักษ์เรื่อง Jurassic Park ที่ออกฉายเมื่อสิบกว่าปีก่อนได้โดยถูกสร้างจากนวนิยายผจญภัยแนววิทยาศาสตร์ในชื่อเรื่องเดียวกัน ซึ่งถูกเขียนโดย Michael Crichton [ฮวน เอ็นริกเก้ ได้รับปริญญาโทสาขาบริหารธุรกิจ (MBA) จากมหาวิทยาลัยฮาร์วาร์ด เคยเป็นอาจารย์สอนหนังสือที่ Harvard Business School และเป็นผู้ก่อตั้ง (Founding Director) Life Sciences Project ของ Harvard Business School นิตยสาร Fortune ขนานนามเขาว่า Mr. Gene ปัจจุบัน ฮวน เอ็นริกเก้ เป็นประธานผู้ก่อตั้งและผู้บริหาร (Founder, Chairman and CEO: Chief Executive Officer) ของบริษัท Biotechonomy ซึ่งเป็นบริษัทที่ทำการวิจัยและดำเนินธุรกิจที่เกี่ยวกับวิทยาศาสตร์ชีวภาพ]

จีโนมิคส์ถูกพูดถึงกันมากในช่วง 10 กว่าปีที่ผ่านมา (แต่จริงๆ แล้วเรื่องราวจีโนมิคส์เริ่มมานานเกือบ 20 ปีแล้ว) ในประเทศไทยเอง นอกจากในวงการวิทยาศาสตร์แล้ว จีโนมิคส์ยังได้รับความสนใจจากบุคคลในหลายวงการ อดีตนายกรัฐมนตรีของไทยท่านหนึ่ง ยังให้ความสนใจเรื่องราวของจีโนมิคส์ ถึงขนาดเชิญ ฮวน เอ็นริกเก้ (Juan Enriquez) ผู้เขียนหนังสือเรื่อง "As the future catches you" มาเพื่อบรรยายเรื่องราวเกี่ยวกับจีโนมิคส์ที่ประเทศไทย "As the future catches you" เป็นหนังสือซึ่งได้รับการจัดให้เป็น the global best seller เล่มหนึ่ง พูดถึงเรื่องราวเกี่ยวกับจีโนมิคส์ รวมทั้งวิทยาศาสตร์ชีวภาพสาขาที่เกี่ยวเนื่องกัน ที่จะมีบทบาทและผลกระทบต่อวงการวิทยาศาสตร์และเทคโนโลยี การเมือง ธุรกิจ และสังคม

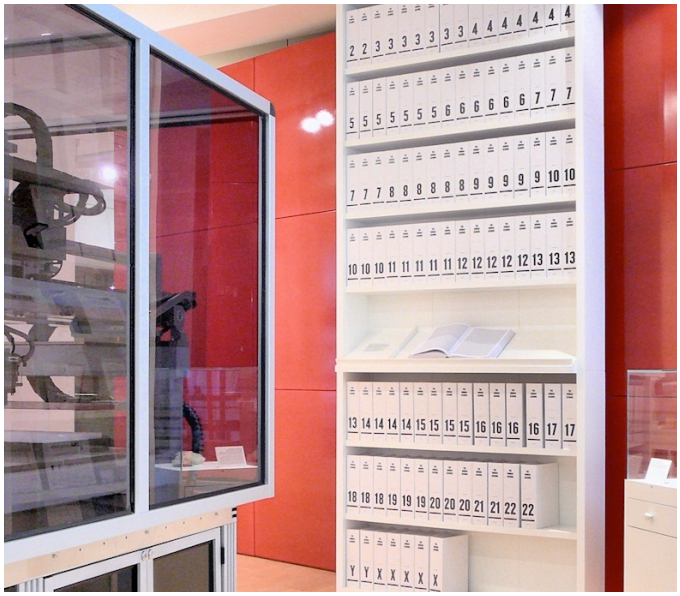
จีโนมิคส์มีเริ่มจุดกำเนิดมาจากผลงานของ ศาสตราจารย์เฟรดเดอริค แซงเกอร์ (Prof. Frederick Sanger) นักชีวเคมี

ชาวอังกฤษ ผู้ได้รับรางวัลโนเบลสาขาเคมีถึง 2 ครั้ง ผลงานดังกล่าวการหาลำดับเบสดีเอ็นเอของจีโนมไวรัส Bacteriophage Φ-X174 ได้สำเร็จในปี ค.ศ. 1977 โดยใช้วิธี random shot-gun sequencing รวมถึงได้ทำแผนที่จีโนมและจัดเก็บและวิเคราะห์ข้อมูลจีโนมที่หาลำดับเบสได้ ต่อมา ดร. แชงเกอร์ได้ทำการหาลำดับเบสดีเอ็นเอของจีโนมของไมโตคอนเดรียของมนุษย์ ได้สำเร็จในปี ค.ศ. 1981 ความสำเร็จทั้งสองส่วนนี้เป็นจุดเริ่มต้นของงานวิจัยด้านจีโนมิคส์ส่วนอื่นต่อๆ มา รวมถึงโครงการหาลำดับเบสของจีโนมมนุษย์

โครงการหาลำดับเบสของจีโนมมนุษย์ (Human Genome project) เป็นโครงการวิจัยขนาดใหญ่ ที่มีความร่วมมือในระดับนานาชาติ เริ่มในปี ค.ศ. 1990 นำโดย ดร. เจมส์ วัตสัน (Dr. James D. Watson) ซึ่งในขณะนั้นเป็นผู้อำนวยการ National Institute of Health (NIH) ของประเทศสหรัฐอเมริการ่วมกับนักวิทยาศาสตร์กลุ่มอื่นๆ จากประเทศสหรัฐอเมริกาและประเทศอื่นๆ อีกหลายประเทศทั่วโลก มีวัตถุประสงค์เพื่อที่จะหาลำดับเบสของสายดีเอ็นเอในจีโนมของมนุษย์ โดยโครงการนี้ได้รับการสนับสนุนจากกระทรวงพลังงาน (Department of Energy หรือที่เรียกย่อๆ ว่า DOE) ของสหรัฐอเมริกา โดยรัฐบาลสหรัฐอเมริกาให้การสนับสนุนทางด้านการเงินประมาณ 200 ล้านเหรียญสหรัฐ เป็นระยะเวลา 20 ปี แต่ในขณะนั้นเองก็ยังไม่มีความชัดเจนมากนักถึงแนวทางในการทำโครงการนี้ให้สำเร็จ ทั้งนี้เพราะในขณะนั้นการหาลำดับเบสของสายดีเอ็นเอยังมีขีดจำกัด ไม่ได้มีลักษณะที่เป็น high throughput ซึ่งสามารถหาลำดับเบสในปริมาณมากแบบที่มีอยู่ในปัจจุบันนี้ อีกทั้งจีโนมของมนุษย์มีขนาดใหญ่มาก ผู้อ่านทราบกันหรือไม่ว่าสายดีเอ็นเอของจีโนมมนุษย์มีความยาวประมาณ 2 เมตร ในขณะที่โครโมโซมของแบคทีเรียอีโคไลมีความยาวประมาณ 1.7 มิลลิเมตร ความยาวของสายดีเอ็นเอจากทุกเซลล์ในร่างกายมนุษย์มีความยาวมากกว่าระยะทางไปกลับระหว่างโลกและดวงอาทิตย์นับสิบๆ รอบ หนทางหนึ่งที่สามารถช่วยในการหาลำดับเบสของสายดีเอ็นเอในจีโนมของมนุษย์คือ การทำแผนที่จีโนมเพื่อให้ทราบถึงการเรียงลำดับของชิ้นดีเอ็นเอของทั้งจีโนมที่ถูกตัดเป็นส่วนย่อยๆ นอกจากนี้การศึกษาหาลำดับเบสในจีโนมของสิ่งมีชีวิตอื่นๆ ที่มีขนาดเล็กและมีความซับซ้อนน้อยกว่าจีโนมมนุษย์ เช่นจีโนมของจุลินทรีย์ทั้งกลุ่มโปรคาริโอต และยูคาริโอต ก็เป็นหนทางหนึ่งที่ทำให้ได้วิธีการที่สามารถนำไปพัฒนาต่อ เพื่อใช้ในการหาลำดับเบสในจีโนมของสิ่งมีชีวิตที่ซับซ้อนมากขึ้น รวมทั้งจีโนมของมนุษย์ อีกทั้งข้อมูลที่ได้จากการหาลำดับเบสในจีโนมของสิ่งมีชีวิตอื่นๆ จะทำให้ทราบและเข้าใจโครงสร้างและลักษณะของจีโนมของสิ่งมีชีวิตเหล่านั้น

สิ่งมีชีวิตที่ดำรงชีวิตได้โดยอิสระในธรรมชาติ หรือที่เราเรียกว่า free-living ที่นักวิทยาศาสตร์ได้ทำการหาลำดับเบสในจีโนมได้สำเร็จเป็นจีโนมแรกคือแบคทีเรียในสปีชีส์ *Haemophilus influenzae* โดยทำสำเร็จในปี ค.ศ. 1995 หลังจากนั้นจีโนมของแบคทีเรียอื่นๆ และสิ่งมีชีวิตอื่นๆ ก็ถูกหาลำดับเบสสำเร็จตามมา ขณะนี้มีจีโนมของแบคทีเรียมากกว่า 1,000 จีโนมที่ทำการหาลำดับเบสแล้ว รวมทั้งจีโนมของสิ่งมีชีวิตจำพวกยูคาริโอตอีกเป็นจำนวนมาก เทคนิคที่ใช้ในการหาลำดับเบสจีโนมของสิ่งมีชีวิตทั้งหมดที่ได้ทำสำเร็จ ได้ถูกนำมาพัฒนาจนทำให้สามารถหาลำดับเบสจีโนมมนุษย์ในระดับที่เป็น draft ได้สำเร็จ และได้มีการแถลงข่าวสำเร็จในระดับดังกล่าวเมื่อวันที่ 26 มิถุนายน ค.ศ. 2000 พร้อมกับตีพิมพ์ผลงานความสำเร็จในวารสาร Science และ Nature ในปีเดียวกันนั้น และหลังจากนั้นได้มีการหาลำดับเบสจีโนมมนุษย์จนเสร็จสมบูรณ์ในปี ค.ศ. 2003

ในปัจจุบันเทคโนโลยีของการหาลำดับเบสของสายดีเอ็นเอมีความก้าวหน้าไปมาก มีเทคนิคใหม่ๆ ที่ได้รับการพัฒนาขึ้นโดยอาศัยหลักการของ flow cell sequencing ร่วมกับ massively parallel approach ซึ่งมีลักษณะเป็น large-scale DNA sequencing ทำให้สามารถหาลำดับเบสได้เร็วขึ้นและได้ปริมาณเบสที่อ่านได้มากขึ้น (ultra-high-throughput) เทคโนโลยีนี้ถูกขนานนามว่าเป็น next-generation sequencing เทคนิคเหล่านี้ได้การพัฒนาต่อโดยบริษัทเอกชนจนเป็นเครื่อง flow cell sequencer ที่มีขายเป็นการค้า ได้แก่ 1) Illumina GA IIx (Solexa) 2) Roche GS-FLX (Pyrosequencing by 454) และ 3) Applied Biosystems ABI SOLiD โดยเครื่องของแต่บริษัทจะแตกต่างกันที่ flow cell configuration platform จริงๆ แล้วยังมีเครื่องจากบริษัทอื่นอีก 2-3 บริษัท จะเห็นได้ว่าเทคโนโลยีของการหาลำดับเบสดีเอ็นเอ ได้รับการพัฒนาให้มีความก้าวหน้าไปเร็วมาก ทำให้ในปัจจุบันสามารถหาลำดับเบสของจีโนมได้อย่างรวดเร็วและเสียค่าใช้จ่ายน้อยกว่าเดิมมาก ด้วยเทคโนโลยีในปัจจุบันสามารถหาลำดับเบสจีโนมมนุษย์เสร็จได้ภายใน 1 วัน ที่ Sanger Institute ประเทศอังกฤษมีเครื่อง Illumina 28 เครื่อง ABI SOLiD 8 เครื่อง และเครื่อง Roche 454 จำนวน 2



ตู้เอกสารที่บรรจุแฟ้มข้อมูลที่พิมพ์ลำดับนิวคลีโอไทด์ (DNA sequence) ทั้งหมดในจีโนมมนุษย์ จะเห็นว่าจีโนมของมนุษย์ที่บรรจุอยู่ภายในเซลล์เพียงเซลล์เดียว บรรจุข้อมูลไว้อย่างมากมายมหาศาล ตัวเลขบนสันแฟ้มแต่ละแฟ้มแสดงตัวเลขของโครโมโซมที่ลำดับนิวคลีโอไทด์นั้นบรรจุอยู่

เครื่อง ในระยะเวลา 1 เดือนสามารถหาลำดับเบสดีเอ็นเอได้ข้อมูลมากกว่า 1000 เทอราไบท์ (TeraByte; TB) จะเห็นได้ว่าปัจจุบันข้อจำกัดของงานวิจัยด้านจีโนมิคส์ไม่ได้อยู่ที่เวลาและค่าใช้จ่ายในการหาลำดับเบส แต่ข้อจำกัดจะเปลี่ยนมาอยู่ที่ขั้นตอนของการวิเคราะห์ข้อมูลปริมาณมหาศาลเหล่านี้ ซึ่งนักวิจัยทางชีวสารสนเทศจะมีบทบาทในส่วนนี้เป็นอย่างมาก

แม้ว่างานส่วนใหญ่ของจีโนมิคส์จะเป็นการการหาลำดับเบสดีเอ็นเอของจีโนม แต่ความรู้ที่ได้จากการวิเคราะห์ข้อมูลจีโนมทำให้เกิดวิทยาการแขนงย่อยของจีโนมิคส์นั่นคือ functional genomics ซึ่งส่วนใหญ่ จะเน้นไปที่การศึกษาการแสดงออกของยีนทั้งหมดในจีโนม (genome-wide approach) ที่มีการแสดงออกในสภาวะต่างๆ ทั้งในระดับ mRNA และระดับโปรตีน โดย functional genomics จะเป็นการศึกษาในลักษณะที่เป็นพลวัต (dynamic) เช่นการศึกษาการแสดงออกของยีนทั้งจีโนมในระดับ transcription (transcriptomics) และระดับ translation (proteomics) รวมทั้งการศึกษาปฏิสัมพันธ์ของโปรตีนทั้งจีโนมที่มีการแสดงออกในขณะนั้นๆ (protein-protein interaction) ในขณะที่ข้อมูลของจีโนมิคส์ส่วนใหญ่จะมีลักษณะแบบ Static เช่น ข้อมูลลำดับเบสของทั้งจีโนม ข้อมูลโครงสร้างของทั้งจีโนมและแต่ละโครโมโซม ข้อมูลแผนที่จีโนม เป็นต้น เทคนิคที่ใช้สำหรับงานวิจัยทาง Functional genomics ได้แก่ microarrays และ 2D-PAGE/MS ซึ่งมีลักษณะเป็น high-throughput method และใช้ร่วมกับเทคนิคทางชีวสารสนเทศ

วิทยาการแขนงย่อยอีกสาขาหนึ่งคือ comparative genomics เป็นการศึกษาจีโนมในลักษณะที่เปรียบเทียบกันระหว่างจีโนมของสิ่งมีชีวิตต่างชนิดกัน ต่างสปีชีส์ หรือสิ่งมีชีวิตชนิดเดียวกันแต่ต่างสายพันธุ์ โดยจะศึกษาในหลายประเด็นเช่น โครงสร้าง หน้าที่ของจีโนม ความสัมพันธ์เชิงวิวัฒนาการที่มีผลกระทบต่อจีโนมของสายพันธุ์ที่ทำการศึกษา ในการหาลำดับเบสจีโนมของสิ่งมีชีวิต ขั้นตอนหนึ่งที่สำคัญคือการหายีนซึ่งเป็นองค์ประกอบของจีโนมนั้นๆ และระบุว่าเป็นยีนอะไร (gene finding/annotation) ในขั้นตอนนี้ เทคนิคทาง comparative genomics เป็นวิธีการหนึ่งที่ถูกนำมาประยุกต์ใช้ในขั้นตอนนี้ เทคนิคอันหนึ่งซึ่งถูกใช้มากจนกลายเป็นหลักของงาน comparative genomics คือการเปรียบเทียบเพื่อหาความเหมือน ความคล้ายคลึงกัน ความแตกต่าง ของลำดับเบสของสายดีเอ็นเอ หรือลำดับกรดอะมิโนของสายโพลิเปปไทด์ของโปรตีนที่ทำการเปรียบเทียบ นอกจากยีนที่สร้างโปรตีนแล้ว สามารถถูกนำไปประยุกต์ใช้หาองค์ประกอบของจีโนมส่วนอื่นได้เช่น บริเวณควบคุม (regulatory region) ซึ่งทำหน้าที่ควบคุมการแสดงออกของยีน และอีกตัวอย่างหนึ่งคืออาร์เอ็นที่ไม่แปลรหัสเป็นโปรตีน (non-coding RNA ncRNA) เป็นที่ทราบกันดีว่าในจีโนมมนุษย์มีดีเอ็นเอส่วนที่เป็นยีนซึ่งมีรหัสเพื่อสร้างโปรตีนอยู่เพียงไม่เกิน 5% ส่วนที่เหลือประมาณสองพันเก้าร้อยกว่าล้านคู่เบสนั้น เดิมถูกเรียกว่าเป็น junk DNA ปัจจุบันพบว่าดีเอ็นเอบริเวณดังกล่าวสามารถมีการแสดงออก โดยถูกถอดรหัส (transcription) เป็นสายอาร์เอ็นเอและไปทำหน้าที่ต่างๆ ภายในเซลล์ได้โดยไม่ต้องถูกแปลรหัสเป็นโปรตีนก่อน ปัจจุบันพบว่าอาร์เอ็นเอเหล่านี้พบได้ในสิ่งมีชีวิตทั้งในกลุ่มโปรคาริโอตและยูคาริโอต จัดเป็น non-protein-coding RNA หรือ ncRNA โดยส่วนมากจะมีหน้าที่เกี่ยวข้องกับการควบคุมการแสดงออกของยีน โดยยีนสำหรับอาร์เอ็นเอกลุ่มนี้ ซึ่งก็จัดว่าเป็นส่วนประกอบของจีโนมเช่นกัน แต่จะไม่สามารถถูกหาได้โดยโปรแกรมสำหรับค้นหายีนทั่วไปที่ใช้หายีนสำหรับสร้างโปรตีน แต่ด้วยเทคนิคทาง comparative genomics และ functional genomics รวมทั้งชีวสารสนเทศทำให้สามารถค้นหายีนสำหรับอาร์เอ็นเอกลุ่มนี้ได้

ความรู้ที่ได้จากวิทยาการทางจีโนมิคส์ รวมทั้งข้อมูลจากโครงการ Human genome project ทำให้เกิดความรู้เพิ่มเติมขึ้นตามมามากมาย เช่น alternative splicing, expressed sequence tag (EST), single nucleotide polymorphism (SNP), non-protein-coding RNA (ncRNA) เป็นต้น นอกจากนี้ยังมีผลกระทบทำให้เกิดการเปลี่ยนแปลงและการพัฒนาในหลายๆ ด้าน วิทยาการหนึ่งที่เห็นได้ชัดเจนคือชีวสารสนเทศ (bioinformatics และ computational biology) ผู้อ่านส่วนมากคงจะเห็นบทบาทและความสำคัญของชีวสารสนเทศที่มีต่อจีโนมิคส์และวิทยาศาสตร์ชีวภาพในยุคปัจจุบัน โดยในส่วนหนึ่งเป็นเรื่องของ computational genomics ลองนึกภาพว่าถ้าไม่มีโปรแกรมสำหรับวิเคราะห์ข้อมูลจีโนม แต่ต้องการหาว่ามียีนอยู่จำนวนเท่าไหร่ในจีโนมมนุษย์ ซึ่งมีขนาดประมาณสามพันล้านคู่เบส หากใช้เวลาอ่านด้วยความเร็ว 1 คู่เบสใน 1 วินาที จะใช้เวลาถึงมากกว่า 90 ปี เพียงเพื่ออ่านข้อมูลจีโนม แต่ในความเป็นจริงเรามิได้เพียงต้องการอ่านลำดับเบสของจีโนมเท่านั้น แต่เราต้องการวิเคราะห์จีโนมเพื่อหาข้อมูลที่เป็นประโยชน์เพื่อการศึกษาเพิ่มเติมหลังจากการหาลำดับเบส ดังนั้นชีวสารสนเทศจึงมีความสำคัญเป็นอย่างมาก ในส่วนชีวสารสนเทศก็จะเชื่อมโยงไปถึงวิทยาการคอมพิวเตอร์ คณิต ศาสตร์ สถิติ เทคโนโลยีสารสนเทศ ซึ่งได้มีการพัฒนานำองค์ความรู้มาใช้ในการพัฒนาโปรแกรมและฐานข้อมูลเพื่อการวิเคราะห์และจัดเก็บข้อมูลจีโนม จะเห็นได้ว่าวิทยาการสาขาหนึ่งถูกพัฒนาเพื่อรองรับการพัฒนาวิทยาการอีกสาขาหนึ่ง ทำให้วิทยาการในปัจจุบันมีลักษณะเป็น สหสาขาวิชา (interdisciplinary) เช่นเดียวกับจีโนมิคส์ที่ต้องอาศัยความรู้ทั้งอณูชีววิทยา พันธุศาสตร์ และชีวเคมี เป็นต้น

จริงๆ แล้วเรื่องราวของจีโนมิคส์มีรายละเอียดอีกมากทั้งในเชิงกว้างและเชิงลึก ข้อมูลจีโนมิคส์ซึ่งถูกผลิตออกมาโดยเฉพาะจากการหาลำดับเบสจีโนมและจากเทคนิคอื่นๆนั้น ก็ต้องการการวิเคราะห์ ประมวลผล เพื่อให้ได้ข้อมูลในลักษณะที่เป็นองค์ความรู้ เพื่อให้เกิดประโยชน์ในการศึกษาค้นคว้าต่อไป ในที่นี้ผู้เขียนคงได้เพียงแต่เล่าให้ฟังคร่าวๆ ให้เห็นภาพความเป็นมา พัฒนาการความ ก้าวหน้าโดยรวมเท่านั้น รายละเอียดปลีกย่อยในแต่ละจุด ผู้สนใจสามารถหาอ่านได้จากแหล่งข้อมูลต่างๆ ทั้งตามวารสารต่างๆ และจากอินเตอร์เนต

SOFTWARE & PROGRAMING

Python Programing

ทำ reverse complement สาย DNA ด้วย Python

ประพัฒน์ สุริยผล

สวัสดีอีกครั้งครับ ครั้งที่แล้ว เราเริ่มจะใช้ Python ทำงานทางด้าน Bioinformatics กันแล้ว ถึงแม้ว่าโจทย์จะไม่ได้ยากหรือซับซ้อนอะไร แต่ก็ทำให้เราได้เรียนรู้การเขียนโปรแกรมไปใช้งานจริงได้พอสมควร ครั้งนี้ เรายังคงวุ่นวายอยู่กับสาย DNA ครับ สิ่งที่เราพบบ่อย คือเราต้องการดูเส้น reverse complement ของสาย DNA ที่เรามีอยู่ เช่น

```
5' ATTCGTAAGCATTAC 3'
3' TAAGCATTCGTAATG 5'
```

เมื่อเรามีสายด้านบนอยู่ เราต้องการแปลงให้ได้สายด้านล่าง ถ้าเส้น DNA ค่อนข้างยาว จะทำด้วยมือก็คงเสียเวลาและมีโอกาสผิดพลาดได้สูง วิธีแก้ปัญหาที่ส่วนใหญ่มักจะทำกันคือเข้าไปใช้บริการเว็บหลายๆ แห่งที่มีคำสั่งทำ reverse complement ให้ หรือไม่ก็เรียกใช้โปรแกรมบางตัวเช่น BioEdit ขึ้นมาช่วย

ต่อไปนี้ เราจะไม่ต้องเสียเวลาเข้าเว็บหรือเปิดโปรแกรม BioEdit อีกแล้ว เพราะเราจะเขียนโปรแกรมเล็กและสั้นด้วย Python ให้เราสามารถทำ reverse complement ได้อย่างรวดเร็ว ทุกครั้งที่ต้องการ

นิยามโจทย์ให้ชัดเจน

เราต้องตัดสินใจก่อนว่า เราต้องการ input คือค่าของสาย DNA จากทางไหน จะให้โปรแกรมเปิดไฟล์ที่มีข้อมูลของ DNA แล้วพิมพ์ผล reverse complement ออกทางหน้าจอ หรือจะให้โปรแกรมอ่านข้อมูลจากทางคีย์บอร์ดเลย การแสดงผลจะให้พิมพ์ออกหน้าจอหรือให้เก็บลงไฟล์ จะเห็นว่าการเขียนโปรแกรมถึงแม้ว่าจะเป็นโปรแกรมเล็กๆ ก็มีปัญหาเรื่องการนิยามโจทย์ให้ชัดเจน แต่หากเราเริ่มเชี่ยวชาญแล้ว เราอาจจะนิยามไว้ในใจก็ได้ถ้าเป็นโปรแกรมขนาดเล็ก แต่ถ้าเป็นโปรแกรมขนาดใหญ่ เราจำเป็นต้องนิยามให้ชัดเจน มิฉะนั้นจะมีปัญหาในตอนเขียนโปรแกรมครับ

เนื่องจากผู้ใช้โปรแกรมนี้คือตัวเราเอง ผมก็เลยสรุปไปเลยว่า ให้อ่านจากคีย์บอร์ดน่าจะสะดวกที่สุดในตอนนี้ เพราะผมมักจะ copy สาย DNA จากเว็บหรือจากเมล์ที่คนส่งมา แล้วต้องการหาเส้น reverse complement และการแสดงผล ผมกำหนดให้เป็นการพิมพ์คำตอบออกบนหน้าจอ ถ้าหากต้องการเก็บผลลงไฟล์ เราสามารถใช้ copy & paste ไปใส่ในโปรแกรมพวก notepad หรือ Microsoft Word ได้ หรือจะใช้วิธีที่สะดวกกว่าคือการ pipe ลงไฟล์ (ผมจะกล่าวถึงเรื่อง pipe ในครั้งต่อๆ ไปครับ)

เริ่มเขียนโปรแกรม

เช่นเคยครับ ผมลองเขียน pseudocode ออกมาก่อนดังด้านล่าง

```
1 dna = read_input()
2 reversed_dna = reverse_sequence(dna)
3 reverse_complement_dna = complement(reversed_dna)
4 print reverse_complement_dna
```

ขั้นตอนการทำงานครั้งนี้ดูง่ายมากใช่ไหมครับ มีแค่ 4 ขั้นตอน อ่านค่า reverse เส้น DNA แล้วแปลงค่า complementary base เสร็จแล้วก็พิมพ์ผลออกมา แต่สิ่งสำคัญที่เราจะเรียนครั้งนี้คือ function ครับ

ก่อนอื่น เรามาเริ่มต้นที่การรับ input จากผู้ใช้ก่อน เราจะใช้คำสั่ง raw_input ครับ เราเคยใช้กันไปครั้งหนึ่งแล้ว ตอนที่เราคำนวณพื้นรูปต่างๆ จำได้ไหมครับ ในครั้งนี้ เราเขียนโปรแกรมออกมาได้บรรทัดเดียวเป็น

```
dna = raw_input('Please enter DNA sequence -> ')
```

เขียนฟังก์ชัน (Function)

จากนั้น เรากำหนดค่าตัวแปร reversed_dna ด้วยการเรียกใช้คำสั่ง reverse_sequence ซึ่งเป็นคำสั่งที่ไม่มีอยู่ใน python เป็นหน้าที่ของเราที่จะต้องเขียนขึ้นมาเองครับ การเขียนคำสั่งใหม่หรือฟังก์ชันใหม่ใน python มีรูปแบบดังนี้ครับ

```
def ชื่อฟังก์ชัน(ตัวแปรต่างๆ ที่จะส่งมาให้ฟังก์ชัน):
    code ของฟังก์ชัน
```

ในกรณีนี้ฟังก์ชันของเราต้องการสาย DNA เข้ามา แล้วจะต้องทำงานส่งคืนสาย DNA อีกสายหนึ่งที่ reverse กับสายที่ส่งเข้ามา เราเขียนได้เป็น

```
1 def reverse_sequence(dna):
2     result = ""
3     for base in dna:
```

```
4          result = base+result
5      return result
```

เรามาลองไล่การทำงานของ code ในฟังก์ชันดูนะครับ

บรรทัดที่ 1 นิยามฟังก์ชัน โดยบอกว่าฟังก์ชันชื่อ `reverse_sequence` แล้วต้องการตัวแปรหนึ่งตัวชื่อ dna

บรรทัดที่ 2 กำหนดตัวแปรอีกหนึ่งตัวชื่อว่า `result` ที่ผมตั้งชื่อนี้ เพราะตัวแปรนี้จะเก็บผลลัพธ์ที่จะส่งคืน

บรรทัดที่ 3 – 4 เป็นการวนลูป โดยแต่ละครั้ง base จะเป็นตัวอักษรตัวหนึ่งใน dna (ที่เป็นข้อมูลตัวอักษรสตริง เพราะคำสั่ง `raw_input` จะคืนค่าที่เป็นตัวอักษรสตริงกลับคืนมา) เช่นถ้า dna มีค่าเป็น 'atgc' ในการวนลูปครั้งแรก base จะมีค่า a เมื่อทำงานมาถึงบรรทัดที่ 4 โปรแกรมจะนำเอาค่า base มาบวกกับค่าใน `result` ซึ่งในที่นี้ก็คือเอามาต่อกันนั่นเองแต่เนื่องจาก `result` มีค่าเป็น empty string (คือไม่มีตัวอักษรอะไรอยู่เลย จากบรรทัดที่ 2) ค่าที่ได้จึงเป็น a และเก็บค่าไว้ใน result

เมื่อวนลูปครั้งที่ 2 ตัวแปร base จะมีค่าเป็นตัวอักษรถัดไปคือ t เมื่อทำงานที่บรรทัดที่ 4 ก็จะเป็นการนำค่า base ปัจจุบันคือ t ไปรวมกับค่าใน `result` ซึ่งตอนนี้มีค่า a อยู่ นำผลลัพธ์ไปใส่ใน `result` ซึ่งจะได้เป็นค่า ta

เมื่อวนลูปครั้งที่ 3 `result` ก็จะถูกปรับค่าเป็น gta แล้วครั้งที่ 4 ก็จะได้ค่าเป็น cgta เมื่อวนลูปครั้งที่ 4 เสร็จแล้วปรากฏว่าไม่มีตัวอักษรให้วนลูปได้อีก จึงออกจากลูปมาที่บรรทัดที่ 5 ซึ่งมีการใช้คำสั่ง `return` เพื่อคืนค่าใน `result` กลับไปให้กับผู้เรียกใช้ ค่าใน `result` ขณะที่หลุดจากลูป คือ cgta ซึ่งเป็นสาย reverse ของ atgc ที่เราใส่เข้ามา ผลลัพธ์ที่ได้ก็จะไปอยู่ในตัวแปรชื่อ `reversed_dna`

ฟังก์ชันถัดไปที่เราต้องการก็คือ `complement` เขียนได้เป็น

```
1  def complement(dna):
2      result = ""
3      for base in dna:
4          if base == 'a':
5              result = result + 't'
6          elif base == 't':
7              result = result + 'a'
8          elif base == 'g':
9              result = result + 'c'
10         elif base == 'c':
11             result = result + 'g'
12         else:
13             result = result + 'n'
14     return result
```

การทำงานของฟังก์ชัน `complement` จะคล้ายกับฟังก์ชัน `reverse_sequence` ด้วยการวนลูปทีละ base แต่โปรแกรมที่อยู่ในลูปจะต่างออกไป สิ่งที่น่าสังเกตคือชื่อตัวแปรที่ส่งมาที่ฟังก์ชันในบรรทัด

```
reverse_complement_dna = complement(reversed_dna)
```

คือ `reversed_dna` แต่ชื่อที่เราตั้งไว้ตอนรับค่าในฟังก์ชัน `complement` คือ dna จะเห็นว่าไม่จำเป็นต้องเป็นชื่อเดียวกัน เมื่อมีการเรียกใช้ฟังก์ชัน ตัวแปร `dna` ที่เราใช้งานในฟังก์ชันจะมีค่าเท่ากับ `reversed_dna` ที่ส่งมา

สมมติว่า ตัวแปร reversed_dna มีค่า cgta เมื่อวนลูปครั้งแรก ตัวแปร `base` จะมีค่าเป็น c เมื่อเปรียบเทียบเงื่อนไขแล้ว จะไปตรงกับบรรทัดที่ 10 เมื่อเงื่อนไขเป็นจริง จึงไปทำงานที่บรรทัดที่ 11 ทำให้ค่า result เป็น g เพราะค่า `result` ตอนแรกเป็น empty string

เมื่อวนลูปครั้งที่ 2 ตัวแปร `base` เป็น g จะไปตรงกับเงื่อนไขบรรทัดที่ 8 ทำให้ค่า `result` เป็น gc เพราะว่าค่า `result` ก่อนหน้านี้คือ g เมื่อมาทำงานที่บรรทัดที่ 9 ก็จะเท่ากับว่า `result` = 'g' + 'c'

วนลูปครั้งที่ 3 ตัวแปร `base` เป็น t ตรงกับเงื่อนไขบรรทัดที่ 6 ทำงานที่บรรทัดที่ 7 ทำให้ค่า `result` เป็น gca เพราะค่าเดิมของ `result` จากลูปครั้งก่อนคือ gc เมื่อนำมาต่อด้วย a จึงได้เป็น gca

วนลูปครั้งที่ 4 เราก็จะได้ค่า `result` เป็น gcat

หลุดจากลูป ทำการคืนค่า `result` ในบรรทัดที่ 14

บรรทัดที่ 12 และ 13 เป็นการเขียนกันเอาไว้ ในกรณีที่ผู้ใช้อาจจะพิมพ์ค่าเบสอื่นๆ ที่ไม่ใช่ atgc ในกรณีนี้ผมตัดสินใจว่า ถ้าเกิดกรณีนี้ขึ้น ให้ใส่ค่า n สำหรับค่าที่เป็น complementary base เราอาจจะตัดสินใจหยุดโปรแกรมแล้วแสดงข้อความผิดพลาดเตือนผู้ใช้ก็ได้ แต่ครั้งนี้เพื่อให้ง่าย ผมจึงตัดสินใจแบบนี้ไปก่อน

การเขียนโปรแกรมจุดสำคัญอีกประการหนึ่ง โดยเฉพาะในส่วนของโปรแกรมที่เกี่ยวข้องกับการตัดสินใจ คือต้องคิดถึงกรณีต่าง ๆ ที่เป็นไปได้ให้รอบคอบ ถ้าหากเราไม่ได้คิดมากนัก เราก็อาจจะคิดว่าตัวเบสในสาย DNA มีเพียงแค่ atgc แต่เป็นไปได้ว่า ผู้ใช้อาจจะพิมพ์ผิด หรือตั้งใจพิมพ์ตัวอักษรอื่นที่ไม่ใช่ atgc ก็ได้ เราจึงต้องเขียนโปรแกรมเพื่อดักกรณีเหล่านี้ให้เรียบร้อย ทักษะในการคิดให้รอบคอบเช่นนี้มีความสำคัญและจำเป็นมากสำหรับการเขียนโปรแกรมให้ทำงานได้อย่างถูกต้อง

กลับมาที่ฟังก์ชัน เมื่อทำงานมาจนถึงบรรทัดสุดท้าย ค่าที่ประมาณได้จากฟังก์ชันก็จะถูกคืนค่ากลับมาที่ตัวแปร `reverse_complement_dna` ในบรรทัดที่ 3 และพิมพ์ผลลัพธ์ในบรรทัดที่ 4 ใน pseudocode ที่เราร่างไว้

```
3 reverse_complement_dna = complement(reversed_dna)
4 print reverse_complement_dna
```

สำหรับท่านผู้ที่เขียนโปรแกรมเป็นแล้วหรือพอเป็นบ้าง จะเห็นว่าโปรแกรมที่เขียนครั้งนี้ค่อนข้างยาว และไม่ “pythonic” เลย ประมาณว่าคนที่เขียน python เป็นแล้วมาเห็นโปรแกรมนี้จะร้องยี้ว่าเขียนเหมือนกับภาษา C ช่างเขียนได้ยาวและไม่ได้ใช้ความสามารถของ python เลย ขอให้ใจเย็นๆ ครับ ครั้งหน้าเราจะมาปรับและเรียนรู้ลึกลงไปในภาษา python ด้วยกันดูว่า โปรแกรมของเราจะออกมารูปร่างหน้าตาเป็นอย่างไร

จุดที่น่าสนใจในการเขียนโปรแกรมครั้งนี้คือการที่เราค่อยๆ เติมเบสในลูปเข้าไปในตัวแปร `result` ขอให้สังเกตความแตกต่างของการปรับค่า `result` ของทั้งสองฟังก์ชันให้ดีครับ ถ้าหากเราเขียนสลับตำแหน่ง ผลลัพธ์ที่ได้ก็ผิดพลาดได้ทันที สำหรับครั้งนี้ เราได้โปรแกรมที่ทำงานได้ถูกต้องตามที่เราต้องการแล้ว พบกันครั้งหน้า ขอให้มีความสุขกับการเขียนโปรแกรมทุกๆ ท่านครับ

Listing program ทั้งหมด

```
1   def reverse_sequence(dna):
2       result = ""
3       for base in dna:
4           result = base+result
5       return result
6
7   def complement(dna):
8       result = ""
9       for base in dna:
10          if base == 'a':
11              result = result + 't'
12          elif base == 't':
13              result = result + 'a'
14          elif base == 'g':
15              result = result + 'c'
16          elif base == 'c':
17              result = result + 'g'
18          else:
19              result = result + 'n'
20      return result
21
22  dna = raw_input('Please enter DNA sequence -> ')
23  reversed_dna = reverse_sequence(dna)
24  reverse_complement_dna = complement(reversed_dna)
25  print reverse_complement_dna
```

THAI BIOINFORMATICS NETWORK

เครือข่ายของนักวิจัยไทยและผู้สนใจ bioinformatics

เตรียมพบกับบ้านของเรา เร็วๆ นี้

SCIENCE & SOCIETY

Joy and Fun

SUDOKU (数独)

ประเวช อรรจวัฒนวงศ์

สวัสดีครับ อย่าเพิ่งแปลกใจนะครับที่เห็นคอลัมน์แบบนี้ใน e-magazine ผมแค่อยากให้เพื่อนสมาชิกมีเกมสนุกๆ ได้เล่นกันเวลาอ่านนิตนสารก็เท่านั้นเองครับ ฉบับนี้ผมเลยอยากแนะนำเกมสนุกๆ ที่หลายต่อหลายคนคงเล่นกันจนเก่งแล้วครับ ผมเองก็เป็นคนหนึ่งที่ชอบเล่นเกมนี้ครับ เกมที่ว่าก็คือ sudoku ครับ

หลายคนอาจคิดว่า sudoku เป็นเกมที่คิดค้นขึ้นโดยคนญี่ปุ่น เพราะชื่อที่เรียกนั้นเป็นภาษาญี่ปุ่น แต่อันที่จริงแล้วเกมนี้กำเนิดขึ้นในยุโรปครับ ในช่วงปี ค.ศ. 1979 โดย Howard Garnes ซึ่งเขาได้แนวความคิดมาจาก ผลงานของนักคณิตศาสตร์ (และเขายังเป็นนักฟิสิกส์อีกด้วย) ชาวสวิตชื่อ Leonhard Euler ที่ได้คิดหลักการทางคณิตศาสตร์ขึ้นมา ชื่อว่า “ลาติน สแควร์” (latin square) พอพูดถึงคณิตศาสตร์หลายคนอาจจะเบื่อแล้วพาลทำให้ไม่อยากอ่านต่อ แต่ผมขอให้ใจเย็นๆ ก่อนนะครับ เดี๋ยวผมจะอธิบายหลักการของลาตินสแควร์อย่างง่ายๆ ให้ฟังครับ ลองนึกภาพตามผมดูนะครับ สมมติว่ามีตาราง

1	2	3
2	3	1
3	1	2

ตารางตัวเลขแบบลาตินสแควร์ขนาด 3x3 แต่ละแถวมีตัวเลข 1, 2 และ 3 เติมอยู่ และตัวเลขในแต่ละคอลัมน์จะต้องไม่ซ้ำกันด้วย

ตัวเลขอยู่ตารางหนึ่งขนาด 3x3 แบบในรูปด้านขวามือเลยครับ แล้วในแต่ละแถวก็ใส่ตัวเลข 1 ถึง 3 ไว้ครับ แต่ว่ามีเงื่อนไขอยู่อย่างหนึ่งคือ ในแต่ละคอลัมน์ต้องไม่ให้ตัวเลข 1 2 และ 3 ซ้ำกัน ตารางแบบนี้แหละครับ ทางคณิตศาสตร์เขาเรียกตารางแบบนี้ว่าลาตินสแควร์ แต่โดยหลักการแล้วตารางของลาตินสแควร์ไม่ได้จำกัดอยู่ที่ขนาด 3x3 เท่านั้นนะครับ แต่สามารถเป็นไปได้ถึงขนาด nxn เลยทีเดียว

โม้กันมาเยอะแล้วครับ ย้อนกลับไปที่ Howard Garnes ผู้สร้างเกม sudoku ขึ้นมาครับ แล้วก็นำไปตีพิมพ์ลงในนิตยสาร

7	9		1	2		8	3	
							4	
2	4		3		9		6	
	6			3				
	3			4	5	7	2	8
		7			2	3	9	
3					1			
5		4		6	8	9		
	2	9		7		4		

← แถว
← กล่อง
← ช่อง
↑ คอลัมน์

7	9	6	1	2	4	8	3	5
1	8	3	7	5	6	2	4	9
2	4	5	3	8	9	1	6	7
8	6	2	9	3	7	5	1	4
9	3	1	6	4	5	7	2	8
4	5	7	8	1	2	3	9	6
3	7	8	4	9	1	6	5	2
5	1	4	2	6	8	9	7	3
6	2	9	5	7	3	4	8	1

(ซ้าย) ตาราง sudoku มาตรฐาน ประกอบด้วยกล่อง 9 กล่อง ในแต่ละกล่องประกอบด้วยช่องสี่เหลี่ยมจำนวน 9 ช่อง

(บน) เฉลย sudoku ในโจทย์ด้านซ้ายมือ

Dell Magazine ในคอลัมน์ “Dell Pencil Puzzles and Word Games” แต่ว่าตอนนั้นเขาไม่ได้ชื่อว่า sudoku นะครับ แต่เป็นชื่อ “Number Place” ต่อมาในปี ค.ศ. 1984 สำนักพิมพ์นิโคลิ (Nikoli) เป็นผู้เปิดตัวเกมนี้ในประเทศญี่ปุ่น ด้วยชื่อ sudoku ซึ่งตั้งโดยนายคาจิ มากิ (Kaji Maki) ประธานกลุ่มนิโคลิ

เอาล่ะครับ เรารู้จัก sudoku กันมาพอหอมปากหอมคอแล้วก็มารู้จักกับกติกาการเล่นดีกว่าครับ ก่อนอื่นผมขอแนะนำให้รู้จักกับตารางของเกม sudoku กันก่อนดีกว่านะครับ ตารางของเกมจะเป็นตารางขนาด 9x9 ครับ โดยภายในตารางนี้จะประกอบไปด้วย **“กล่อง”** จำนวน 9 กล่องครับ ภายในกล่องแต่ละกล่องก็จะมี **“ช่อง”** สี่เหลี่ยมขนาดเล็กจำนวน 9 ช่องครับ เมื่อมองดูตาราง sudoku ในภาพรวมทำให้เราเห็นตารางนี้แบ่งเป็นช่องจำนวน 81 ช่อง ซึ่งในจำนวนนี้แบ่งได้เป็น 9 แถว และ 9 คอลัมน์ครับ

วิธีการเล่นก็ง่ายมากๆ ครับ คุณจะต้องเติมตัวเลข 1 ถึง 9 ลงในแต่ละแถวครับ ทำเช่นนี้ทุกแถวจะครบตารางครับ แต่มีข้อแม้อยู่ว่า หลังจากที่คุณเติมตัวเลขลงในตารางครบแล้ว ตัวเลข 1 ถึง 9 ในแต่ละแถวจะต้องไม่ซ้ำกัน และตัวเลข 1 ถึง 9 ในแต่ละคอลัมน์ก็จะต้องไม่ซ้ำกัน นอกจากนี้ ในแต่ละกล่องจะใช้ตัวเลข 1 ถึง 9 ได้เพียงตัวละครั้งเท่านั้น เอาล่ะครับรู้กติกากันแล้ว ผมก็ขอให้เพื่อนๆ ได้ลองซ้อมมือกับเกม sudoku กันหน่อยนะครับ แล้วพบกันใหม่ครับ

ระดับ: ง่าย

4		6	9			2		8
	7		1			4		
5	9			3	8	6		
7		3		1				
		5		7		3		
				8	9		4	7
8	3	4					2	
				2		9	6	
2					5			

ระดับ: ง่าย

5	1	3	8			7		6
6			1					
				3	2			
		4		1			8	3
	6				8	4	1	
9				4	5			7
8	3					6	5	
						9		
2	5		7			8		

ระดับ: ปานกลาง

7	8		2					
	9		6			3		
6		4		8	9		7	
				5				1
	7	3				6	4	
		5	9	3		7		
	4	9		7			6	
3			4			2		9
1	6					4		

ระดับ: ยาก

3					7		1	
			6	9				8
4	6		1		3			
7				5				3
	4				8		2	
6	8		9	1				5
8						9		
	7		5			3	6	
9				3		2		4