โครงการ

การพัฒนาวิธีการสร้างเครือข่ายควบคุมการแสดงออกของยีนแบบมีเงื่อนไขด้วยเทคนิคบูลีน

Constraint-based Boolean network for reconstructing gene regulatory network

คณะผู้วิจัย

หัวหน้าโครงการ: นางสาวสมคิด บุมี
ผู้ร่วมโครงการ: ผศ. ดร. อัศวิน มีชัย

สถาบันพัฒนาและฝึกอบรมโรงงานต้นแบบ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

งานวิจัยนี้ได้รับทุนอุดหนุนงบประมาณแผ่นดิน มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ปีงบประมาณ 2554

รายงานการวิจัยฉบับสมบูรณ์

ประจำปีงบประมาณ 2554

---

1. ชื่อโครงการวิจัย

ชื่อเรื่อง (ภาษาไทย) การพัฒนาวิธีการสร้างเครือข่ายควบคุมการแสดงออกของยีนแบบมีเงื่อนไขด้วยเทคนิคบูลีน

(ภาษาอังกฤษ) Constraint-based Boolean network for reconstructing gene regulatory network

2. หน่วยงานหลักที่รับผิดชอบงานวิจัย

ชื่อหน่วยงาน สถาบันพัฒนาและฝึกอบรมโรงงานต้นแบบ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
เลขที่ 49 ซอยเทียนทะเล 25 ถนนบางขุนเทียน - ชายทะเล
แขวงท่าข้าม เขตบางขุนเทียน, กรุงเทพฯ 10150

3. คณะผู้วิจัย

3. 1 หัวหน้าโครงการ นางสาวสมคิด บุมี
ตำแหน่ง ผู้ช่วยนักวิจัย
สถานที่ทำงาน สถาบันพัฒนาและฝึกอบรมโรงงานต้นแบบ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี (บางขุนเทียน)
83 หมู่ 8 แขวงท่าข้าม เขตบางขุนเทียน กรุงเทพฯ 10150
โทรศัพท์ 02-4707531
โทรสาร 02-4523455
ประสบการณ์ในงานวิจัย Systems biology and Bioinformatics
ความรับผิดชอบต่อโครงการที่เสนอ คิดเป็น 90% ของงานทั้งหมด

3.2 ผู้ร่วมโครงการ ผศ.ดร. อัศวิน มีชัย
ตำแหน่ง ผู้ช่วยศาสตราจารย์
สถานที่ทำงาน: ภาควิชาวิศวกรรมเคมี คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
126 ถนนประชาอุทิศ ทุ่งครุ กรุงเทพฯ 10140
โทรศัพท์: 02-4709616
โทรสาร: 02-4279623
ประสบการณ์ในงานวิจัย Systems biology, Metabolic engineering & Bioinformatics
ความรับผิดชอบต่อโครงการที่เสนอ คิดเป็น 10% ของงานทั้งหมด

บทคัดย่อ

ปัจจุบันเทคนิคไมโคอะเรย์ทำให้เราสามารถศึกษาเครือข่ายการแสดงออกของยีนในสิ่งมีชีวิตได้ และวิธีการทางคอมพิวเตอร์ที่มีการพัฒนาขึ้นมานั้นโดยส่วนมากใช้ศึกษาเครือข่ายควบคุมการแสดงออกของยีนโดยปราศจากการคำนึงถึงเงื่อนไขของกลไกการทำงานที่มีในสิ่งมีชีวิต ในการศึกษาครั้งนี้ ผู้วิจัยได้นำเสนอวิธีการสร้างเครือข่ายควบคุมการแสดงออกของยีนแบบมีเงื่อนไขโดยใช้เทคนิคบูลีน เพื่อใช้ศึกษาเครือข่ายควบคุมการแสดงออกของยีนในกระบวนการสลายน้ำตาลกาแลกโตส ภายใต้เงื่อนไขการเปลี่ยนแปลงสภาวะจากการหายใจแบบไม่ใช้ออกซิเจนโดยใช้กลูโคสเป็นแหล่งพลังงานเป็นการหายแบบใช้ออกซิเจน จากข้อมูลการแสดงออกของยีนที่ได้มาจากเทคนิคไมโครอะเรย์ วิธีการดังกล่าวประกอบด้วย 3 ขั้นตอนหลัก ขั้นตอนแรก คือ การแปลงข้อมูลจริงของการแสดงออกของยีนมาเป็นข้อมูลเพียงสองระดับคือ 0 และ 1 โดยเปรียบเทียบ 3 วิธีด้วยกัน คือ Max-x%Max Mean และ Sign of log ratio ซึ่งจะใช้ค่าความเหมือนกันระหว่างข้อมูลที่มีการแปลงเป็นสองระดับและข้อมูลจริงเป็นตัวตัดสินว่าควรใช้วิธีการใดในการแปลงข้อมูลก่อนที่จะสร้างเครือข่ายควบคุมการแสดงออกของยีน ขั้นตอนที่สองคือ การสร้างเครือข่ายควบคุมการแสดงออกของยีนจากข้อมูลที่แปลงเป็น 0 และ 1 ด้วยวิธีการสร้างเครือข่ายควบคุมการแสดงออกของยีนแบบมีเงื่อนไขด้วยเทคนิคบูลีนที่ได้พัฒนาขึ้น จากภาษา C^{++} เรียกโปรแกรมนี้ว่า *เทคนิคบูลีนแบบมีเงื่อนไข* ขั้นตอนสุดท้ายคือการสร้างเครือข่ายควบคุมการแสดงออกของยีนในรูปแบบกราฟที่แสดงถึงการควบคุมแบบกระตุ้นและการควบคุมแบบยับยั้ง ซึ่งพบว่าเทคนิคบูลีนแบบมีเงื่อนไขดังกล่าวสามารถใช้สร้างเครือข่ายควบคุมการแสดงออกของยีนได้รวมทั้งมีความแม่นยำมากขึ้นกว่าเครือข่ายควบคุมการแสดงออกของยีนที่สร้างด้วยวิธีเทคนิคบูลีนแบบไม่พิจารณาเงื่อนไขสามารถลดความซับซ้อนของเรือข่ายควบคุมการแสดงออกของยีน โดยการลดความผิดพลาดในการทำนายความสัมพันธ์ โดยสามารถเพิ่มความถูกต้องขึ้น 25%

นอกจากนี้ เทคนิคบูลีนแบบมีเงื่อนไขที่พัฒนาขึ้นนั้นยังสามารถนำไปประยุกต์ใช้สร้างเครือข่ายควบคุมการแสดงออกของยีนในกระบวนการสร้างแป้งในพืช *A. thaliana* ภายใต้สภาวะกลางวันและกลางคืนได้ โดยสามารถทำนายโปรตีนควบคุมที่เกี่ยวข้องกับเอนไซม์ในกระบวนการสร้างแป้งทั้งกระบวนการสังเคราะห์แป้ง และกระบวนการสลาย พบว่า โปรตีนควบคุม 2 ตัว คือ *At4g39780* and *At2g43010* เกี่ยวข้องกับ เอนไซม์ในการะบวนการสลายแป้ง นั่นคือ a cytosolic disproportionating enzyme like protein (*At2g40840*; *DPE2*) alpha-amylase like 3 (*At1g69830*; *AMY3*) และ a plastidic alpha-glucan phosphorylase (*At3g29320*; *PHS1*) และมีโปรตีนควบคุม 1 ตัว คือ *At2g28200* ที่เกี่ยวข้องกับเอนไซม์ 5 ตัว คือ คือ phosphoglucomutase (*At5g51820*; *PGM1*) Starch synthase II (*At3g01180*; *STS2*) Starch branching enzyme III (*At2g36390*; *SBE3*) Glucan water dikinase 1 (*At1g10760*; *GWD1/SEX1*) และ Glucanotransferase (*At5g64860*; *DPE1*) จะเห็นได้ว่า เทคนิคบูลีนแบบมีเงื่อนไขนั้นสามารถใช้สร้างเครือข่ายควบคุมการแสดงออกของยีนจากข้อมูลไมโครอะเรย์ และทำให้ได้เครือข่ายควบคุมการแสดงออกของยีนที่สอดคล้องกับข้อมูลทางชีววิทยามากขึ้น

คำสำคัญ : ไมโคอะเรย์ / เครือข่ายควบคุมการแสดงออกของยีน / เทคนิคบูลีนแบบมีเงื่อนไข

Abstract

The availability of large-scale expression microarray data allows the inference of underlying gene regulatory network of organisms. Various computational techniques developed for this task often infer the circuits directly from the expression data without taking into consideration of biological information *in priori*. Here, we present a constraint-based Boolean approach that integrates biological constraints into a Boolean algorithm to formulate a genetic circuit of *Saccharomyces cerevisiae* during the diauxic shift in a glucose-limited culture from time-series expression data. This approach includes three main steps. First, the gene expression data was discretized to either '0' or '1' based on three discretization methods, Max-x%Max, Mean, and Sign of log ratio. A similarity score was used as a criterion for selecting a suitable discretization method. Second, the discretized data was employed to determine gene relationships using a developed constraint-based Boolean algorithm. This algorithm implemented by C^{++} programming considers all possibility of gene circuits and then specifically selects the circuits that correspond with both the expression data and a set of pre-defined biological constraints, so called *"constraint-based Boolean network"*. Last, the gene regulatory network was depicted as a directed graph model describing activation and inhibition among genes. The evaluating result from inferred network between constraint and non-constraint-based algorithm shows that our developed algorithm can reduce false prediction and gain 25% increase of accuracy over non-constraint-based algorithm.

Moreover, we demonstrated the validity of the technique by employing it to infer the gene regulatory network of starch metabolism from time series expression data of *Arabidopsis thaliana* taken diurnally. Inferred network showed that two transcription factors, *At4g39780* and *At2g43010,* co-regulate three enzymes in starch degradation, a cytosolic disproportionating enzyme like protein (*At2g40840*; *DPE2*), alpha-amylase like 3 (*At1g69830*; *AMY3*), and a plastidic alpha-glucan phosphorylase (*At3g29320*; *PHS1*). There is only one transcription factor, *At2g28200*, regulates transcription of five enzymes in starch metabolism, phosphoglucomutase (*At5g51820*; *PGM1*) Starch synthase II (*At3g01180*; *STS2*) Starch branching enzyme III (*At2g36390*; *SBE3*) Glucan water dikinase 1 (*At1g10760*; *GWD1/SEX1*) และ Glucanotransferase (*At5g64860*; *DPE1*) The results show constraint-based Boolean network shall enable the better gene regulatory network from large-scale gene expression data which is corresponding to biological data.

Keywords: Microarray / gene regulatory network / constraint-based Boolean network

สารบัญเรื่อง

สารบัญเรื่อง (ต่อ)

สารบัญตาราง

สารบัญรูป

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหาที่ทำการวิจัย

สิ่งมีชีวิตจำเป็นต้องมีการควบคุมการสังเคราะห์โปรตีนหรือการแสดงออกของยีนเพื่อให้โปรตีนที่ได้มีปริมาณเพียงพอสำหรับเมตาบอลิซึมต่างๆนั้น การแสดงออกของยีน คือ การสร้างผลิตผลจากยีนที่เป็นข้อมูลทางพันธุกรรมบนสายดีเอนเอโดยจะต้องมีการถอดรหัสและการแปลรหัสพันธุกรรม เพื่อสังเคราะห์โปรตีนที่จำเป็นต้องใช้ภายในเซลล์หรือระหว่างเซลล์ การแสดงออกของยีนเพื่อสร้างโปรตีนแต่ละชนิดเกิดขึ้นในเวลาและปริมาณที่แตกต่างกัน ทั้งนี้ขึ้นอยู่กับชนิดและบทบาทหน้าที่ของโปรตีนนั้น โดยการควบคุมการสังเคราะห์โปรตีนในร่างกายนั้นเกี่ยวข้องกับความสมดุลของหลายกระบวนการที่สำคัญได้แก่ กระบวนการถอดรหัส (transcription) และการตกแต่งหลังการถอดรหัส (post transcriptional modification) กระบวนการแปลรหัส (translation) และการตกแต่งหลังการแปลรหัส (post translational modification) รวมทั้งกระบวนการสลาย mRNA (mRNA degradation) และการสลายโปรตีน (protein degradation) [2] การควบคุมการสังเคราะห์โปรตีน คือ การควบคุมอัตราการผลิตและการสลายของโปรตีนเพื่อให้โปรตีนที่ได้มีปริมาณเพียงพอต่อขบวนการเมตาบอลิซึมต่างๆ หากมีการกระตุ้นหรือยับยั้งขั้นตอนใดๆ ในกระบวนการเหล่านี้จะมีผลกระทบต่อปริมาณโปรตีนที่เป็นผลิตผลทั้งสิ้น ทั้งนี้ขึ้นอยู่กับสภาพแวดล้อม หน้าที่ และความต้องการของเซลล์ชนิดนั้น การควบคุมเมตาบอลิซึมในเซลล์โปรคาริโอต (prokaryote) มีกลไกที่แตกต่างจากเซลล์ยูคาริโอต เนื่องจากโปรคาริโอตมีพันธุกรรมที่ไม่ซับซ้อนอีกทั้งกระบวนการสังเคราะห์อาร์เอนเอ (RNA) และโปรตีนเกิดขึ้นในไซโตพลาสซึม (cytoplasm) ซึ่งใช้เวลาที่ใกล้เคียงกัน ในขณะที่การควบคุมการแสดงออกของยีนในยูคาริโอต เช่น พืชและสัตว์ จะซับซ้อนกว่าในโปรคาริโอต เช่น แบคทีเรีย เพราะในยูคาริโอต (eukaryote) มียีนอยู่มากมายหลายชนิดบนโครโมโซมหลายอันทั้งนี้ขึ้นอยู่กับชนิดของสิ่งมีชีวิตนั้น การควบคุมการสังเคราะห์โปรตีนในโปรคาริโอตส่วนใหญ่เกิดขึ้นที่ระดับการถอดรหัสพันธุกรรม การควบคุมนี้เป็นแบบโอเพอรอน (operon) สำหรับในยูคาริโอตการควบคุมการสังเคราะห์โปรตีนเกิดขึ้นได้ทั้งระดับยีน (gene) ระดับการถอดรหัส (transcriptional control) และการควบคุมการแปลรหัส (translational control) ซึ่งการควบคุมในระดับการถอดรหัสนั้นเกิดจากโปรตีนควบคุม (transcription factors) เข้ามาทำงานเพื่อกระตุ้นหรือยับยั้งกระบวนการถอดรหัส จะเห็นว่า การควบคุมการแสดงออกของยีนในระดับการถอดรหัสนั้นสำคัญมาก อีกทั้งยังเป็นสาเหตุสำคัญที่ก่อให้เกิดโรคทางพันธุกรรมได้ [3, 4]

จะเห็นได้ว่า กลไกการควบคุมการแสดงออกของยีนในระดับการถอดรหัสนั้นสำคัญมากต่อการดำรงชีวิตของสิ่งมีชีวิตต่างๆ นักวิทยาศาสตร์จึงพยายามที่จะเข้าใจกลไกเหล่านี้จากข้อมูลด้านจีโนมที่มีมากขึ้น กอปรกับเทคโนโลยีด้านไมโครอะเรย์ (microarray) ได้ส่งผลให้นักวิทยาศาสตร์ได้พยายามคิดค้นวิธีการต่างๆ เพื่อมาศึกษาการแสดงออกของยีนของโปรคาริโอตและยูคาริโอต [5, 6] โดยการอนุมานลักษณะการควบคุมการแสดงออกของยีนจากข้อมูลดีเอ็นเอจากเทคนิคไมโครอะเรย์ด้วยวิธีทางคอมพิวเตอร์ [7-9] เช่น ordinary differential equations (ODE), partial differential equations (PDE) Bayesian networks [10] และ Boolean networks [1, 9, 11, 12] เป็นต้น ซึ่งวิธีที่ง่ายที่สุดที่ใช้ศึกษาเครือข่ายควบคุมการแสดงออกของยีน (gene regulatory network) ด้วยเทคนิคบูลีน (Boolean network) แต่อย่างไรก็ตาม การศึกษาเครือข่ายควบคุมการแสดงออกของยีนในระดับการถอดรหัสโดยใช้บูลีนนั้นมีข้อจำกัดคือ สามารถวิเคราะห์ข้อมูล (input) ที่มีค่าเพียง 2 ระดับ นั่นคือ 0 หรือ 1 ดังนั้น

สำหรับการวิเคราะห์เครือข่ายควบคุมการแสงดออกของยีนนั้น จำเป็นต้องมีการแปลงค่าการแสดงออกของยีน (raw data) ให้อยู่เพียง 2 ระดับ (binary values) คือ 0 หรือ 1 ซึ่ง 0 อาจหมายถึงไม่มีการแสดงออกของยีนนั้นๆ และ 1 คือ มีการแสดงออกของยีนนั้นๆ เรียกขั้นตอนนี้ว่า “discretization” โดยใช้วิธีการทางสถิติ (discretization method) จะเห็นได้ว่า การแปลงค่าการแสดงออกของยีนซึ่งเป็นข้อมูลจริง มาเป็นข้อมูลเพียง 2 ระดับนั้น อาจไม่มีเพียงพอ และทำให้เกิดการทำนายที่ผิดพลาด (false prediction) เครือข่ายควบคุมการแสงดออกของยีนมีความซับซ้อน ยากต่อการวิเคราะห์ต่อไป

วิธีการที่ใช้ในการแปลงข้อมูลการแสดงออกของยีนในขั้นตอนนี้มีหลากหลายแบบ อย่างไรก็ตามในการศึกษาเครือข่ายควบคุมการทำงานของยีนด้วยเทคนิคบูลีนที่มีมาก่อนหน้านี้นั้น ส่วนมากเน้นการปรับปรุงวิธีการเทคนิคบูลีนมากกว่าการพิจารณาถึงการเลือกวิธีการที่ใช้ในการแปลงข้อมูล ซึ่งอาจเป็นปัจจัยหลักหนึ่งที่ทำเครือข่ายควบคุมการแสดงออกของยีนนั้นไม่สอดคล้องกับข้อมูลที่มีอยู่ นอกจากนี้การปรับปรุงเทคนิคบูลีนที่สามารถสร้างเครือข่ายควบคุมการแสดงออกของยีนที่มีความซับซ้อนน้อยลงให้สอดคล้องกับข้อมูลทางชีววิทยาน่าจะทำให้เครือข่ายควบคุมการแสงดออกของยีนที่สร้างขึ้นนั้นมีความซับซ้อนน้อยลง การจะศึกษาเครือข่ายควบคุมการแสดงออกของยีนโดยใช้เทคนิคบูลีนที่มีมาก่อนหน้านี้นั้น ยังขาดการพิจารณาถึงเงื่อนไขทางด้านชีววิทยา (biological constraint) เช่น โปรตีนควบคุม (transcription factors) สามารถควบคุมการแสดงออกของยีนเป้าหมายในระดับเมตาบอลิก (เอนไซม์) ได้ แต่เอนไซม์ไม่ควรควบคุมการแสดงออกของโปรตีนควบคุม เป็นต้น การใส่ biological constraint ให้กับการวิเคราะห์ด้วยเทคนิคบูลีนน่าจะลดความซับซ้อนของเครือข่ายควบคุมการแสดงออกของยีน โดยการลดความผิดพลาดของการทำนายที่มักจะเกิดขึ้นจากการวิเคราะห์ด้วยวิธีนี้ ทำให้เครือข่ายที่สร้างขึ้นนั้นง่ายต่อการศึกษาและวิเคราะห์ต่อไป

ดังนั้น การวิจัยในครั้งนี้ ผู้วิจัยมีความมุ่งหวังที่พัฒนาและปรับปรุงเทคนิควิธีการสร้างเครือข่ายการแสดงออกของยีนแบบมีเงื่อนไขโดยใช้เทคนิคบูลีน เรียกว่า Constraint-based Boolean network พร้อมทั้งคำนึงถึงการประเมินหาวิธีการที่เหมาะสมต่อการแปลงข้อมูลการแสดงออกของยีนให้เป็นข้อมูล 2 ระดับ เพื่อช่วยให้สามารถสร้างเครือข่ายควบคุมการแสดงออกของยีนให้สอดคล้องกับระบบของสิ่งมีชีวิตมากยิ่งขึ้น ซึ่งสามารถลดความซับซ้อนของเครือข่ายควบคุมการแสดงออกของยีน นอกจากนี้สามารถนำเอาวิธีที่พัฒนาขึ้นไปประยุกต์ใช้วิเคราะห์เครือข่ายควบคุมการแสดงออกของยีนของกระบวนการในพืช เช่น กระบวนการสังเคราะห์แป้ง ซึ่งเป็นแหล่งพลังงานหลักของมนุษย์

1.2 วัตถุประสงค์ของโครงการ

เพื่อปรับปรุงและพัฒนาวิธีการในการสร้างเครือข่ายควบคุมการแสดงออกของยีนแบบมีเงื่อนไขด้วยเทคนิคบูลีน เพื่อนำไปใช้ศึกษาเครือข่ายควบคุมการแสดงออกของยีนในกระบวนการสังเคราะห์แป้ง

1.3 ขอบเขตของโครงการวิจัย

1.3.1 ปรับปรุงและพัฒนาโปรแกรมคอมพิวเตอร์ (Constraint-based Boolean network) เพื่อสร้างเครือข่ายควบคุมการแสดงออกของยีนโดยใช้ข้อมูลจากไมโครอะเรย์ของยีสต์ภายใต้สภาวะการเลี้ยงแบบ diauxic shift

1.3.2 ประเมินหาวิธีการที่เหมาะสมที่สุดที่ใช้แปลงค่าการแสดงออกของยีนเป็นข้อมูล 2 ระดับ (discretization method)

1.3.3 ประยุกต์ใช้โปรแกรมคอมพิวเตอร์ที่พัฒนาขึ้น เพื่อสร้างและศึกษาเครือข่ายการแสดงออกของยีนในกระบวนการสังเคราะห์แป้ง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 โปรแกรมคอมพิวเตอร์ที่ใช้ในการสร้างและศึกษาเครือข่ายควบคุมการแสดงออกของยีนโดยใช้ข้อมูลไมโครอะเรย์

1.4.2 เครือข่ายควบคุมการแสดงออกของยีนในกระบวนการสังเคราะห์แป้ง เพื่อนำไปสู่แนวทางในการปรับปรุงสายพันธุ์พืชที่มีความสามารถในการสังเคราะห์แป้งเช่น ข้าวและมันสำปะหลัง ที่สามารถสังเคราะห์แป้งให้มีคุณสมบัติตามต้องการ เหมาะต่อการนำไปใช้ในอุตสาหกรรมต่างๆ และนำไปสู่การเพิ่มมูลค่าของผลิตภัณฑ์ทางการเกษตร

1.4.3 มีการผลิตบุคลากรที่มีความเชี่ยวชาญเฉพาะทางด้านเทคโนโลยีชีวสารสนเทศ (Bioinformatics)

1.5 แผนการถ่ายทอดเทคโนโลยีหรือผลการวิจัยสู่กลุ่มเป้าหมาย

1.5.1 เผยแพร่ผลงานวิจัยในรูปสิ่งตีพิมพ์และงานประชุมวิชาการในระดับชาติและระดับนานาชาติ

1.5.2 ร่วมมือกับกลุ่มวิจัย

1.6 แผนการดำเนินงานตลอดโครงการ

ตารางที่ 1.1 แผนการดำเนินงาน

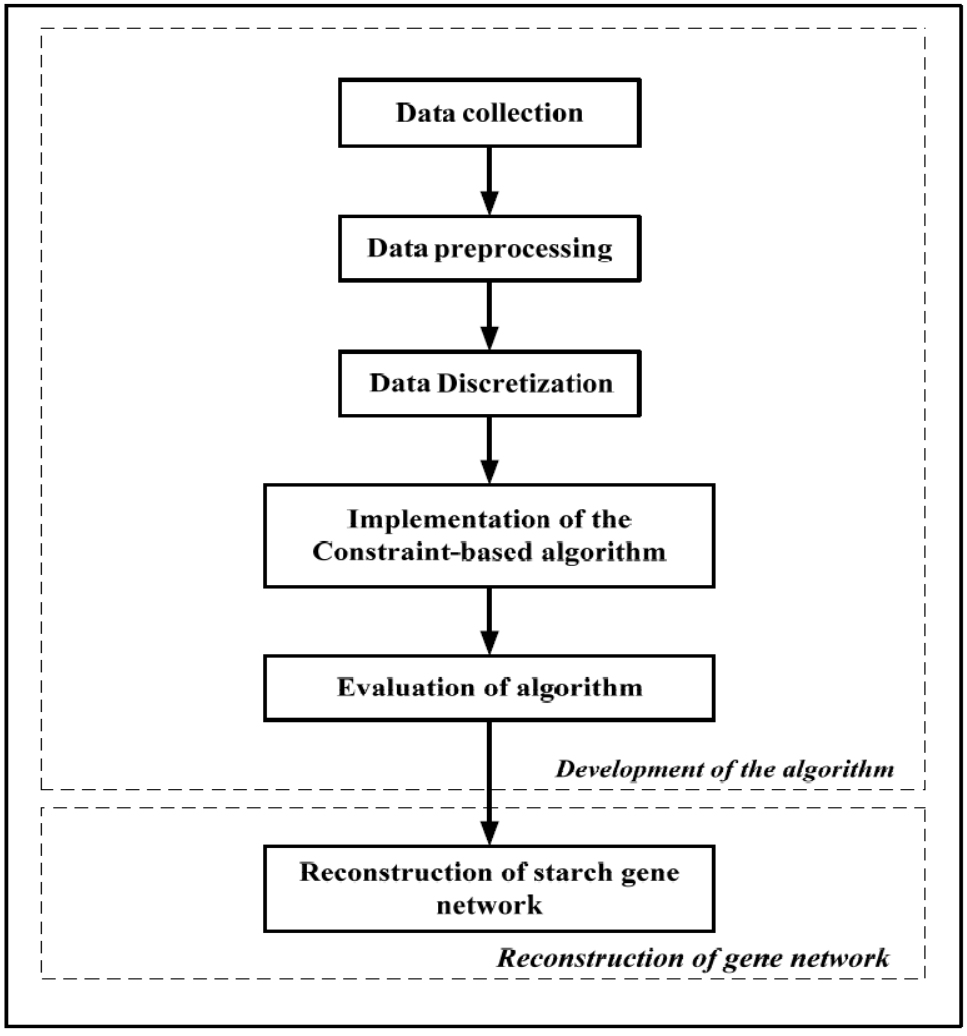
ขั้นตอนการวิจัย	เดือนที่ 1-3	เดือนที่ 4-6	เดือนที่ 7-9	เดือนที่ 10-12
1. การเก็บและรวบรวมข้อมูล				
2. การเตรียมข้อมูล				
3.ประเมินวิธีการที่ใช้ในการแปลงข้อมูลการแสดงออกของยีน				
4. พัฒนาโปรแกรมที่ใช้ในการสร้างเครือข่ายการแสดงออกของยีนแบบมีเงื่อนไข				
5. การทดสอบและประเมินโปรแกรม				
6. การประยุกต์ใช้โปรแกรมเพื่อสร้างและศึกษาเครือข่ายควบคุมการแสดงออกของยีนในกระบวนการสังเคราะห์แป้ง				
7. สรุปผลและเขียนรายงาน				

บทที่ 2
ระเบียบวิธีวิจัย

การศึกษาวิจัยแบ่งออกเป็น 6 ขั้นตอนหลัก ดังแสดงในรูปที่ 2.1 ซึ่งมีรายละเอียดดังต่อไปนี้

2.1 การเก็บและรวบรวมข้อมูล (Data collection)

การทดลองนี้จะเก็บรวบรวมข้อมูลไมโครอะเรย์ 2 แบบ คือ cDNA microarrays ของยีสต์ และ oligoncleotide microarray (Affymetrix GeneChips) จากฐานข้อมูล Stanford Microarray และ NASC's International Affymetrix Service (NASCArrays) ของพืช *Arabidopsis thaliana*



รูปที่ 2.1 วิธีการดำเนินวิจัยโดยสังเขป

2.2 การเตรียมข้อมูล (Data preprocessing)

ก่อนที่จะนำข้อมูลการแสดงออกของยีนไปใช้นั้น ต้องมีการเตรียมข้อมูลให้เหมาะสม รวมทั้งมีการปรับค่าการแสดงออกของยีนให้อยู่ในมาตรฐานเดียวกันตามเทคโนโลยีไมโครอะเรย์ที่ใช้ เพื่อสามารถนำมาเปรียบเทียบหรือวิเคราะห์ในขั้นตอนต่อไป ข้อมูลที่ใช้ในงานวิจัยนี้มี 2 แบบ จาก cDNA

microarrays และ oligoncleotide microarray (Affymetrix GeneChips) การเตรียมข้อมูลก็จะแตกต่างกัน ข้อมูลการแสดงออกของยีนที่ผ่านกระบวนการเตรียมข้อมูลแล้วนั้นจะแปลงเป็น 2 ระดับ (binary values) คือ เป็น 0 หรือ 1 สำหรับวิเคราะห์ด้วยเทคนิคบูลีนต่อไป

2.3 ประเมินวิธีการที่ใช้ในการแปลงข้อมูลการแสดงออกของยีน (Data Discretization)

ขั้นตอนที่สำคัญในการวิเคราะห์เครือข่ายควบคุมการแสดงออกของยีนด้วยเทคนิคบูลีน คือ การแปลงค่าการแสดงออกของยีนเป็นข้อมูล 0 และ 1 เพื่อใช้ในการสร้างฟังก์ชั่นบูลีน (Boolean function) ในการหาความสัมพันธ์ระหว่างคู่ยีนว่าเป็นความสัมพันธ์แบบกระตุ้น (activation) หรือยับยั้ง (inhibition) ซึ่งวิธีการทางสถิติที่ใช้ในการแปลงข้อมูลเป็น 0 หรือ 1 นั้นมีหลายแบบ จึงควรมีการประเมินหาวิธีการที่เหมาะสมใช้ในการแปลงข้อมูลการแสดงออกของยีน เพื่อให้ได้เครือข่ายควบคุมการแสดงออกของยีนที่มีความสอดคล้องกับระบบของสิ่งมีชีวิต โดยจะประเมินวิธีการด้วยการเปรียบเทียบค่าความเหมือน (Similarity score) ระหว่างเดนโดแกรม (dendogram) ของ raw data และ binary values ว่ามีความเหมือนหรือต่างกันเพียงใด โดยเดนโดแกรมนั้นสร้างจากวิธีการ hierarchical clustering วิธีการที่ให้ค่า similarity score ระหว่าง dendogram ของ raw data และ binary values สูงสุด ควรจะเป็นวิธีการที่ดีที่สุดที่จะใช้ในการแปลงข้อมูล และสามารถช่วยสร้างเครือข่ายควบคุมการแสดงออกของยีนด้วยเทคนิคบูลีนที่สอดคล้องกับข้อมูลทางชีววิทยามากที่สุด [13]

2.4 พัฒนาโปรแกรมที่ใช้ในการสร้างเครือข่ายการแสดงออกของยีนแบบมีเงื่อนไข (Constraint-based Boolean network)

ด้วยข้อจำกัดของเทคนิคบูลีนที่อาจสร้างเครือข่ายควบคุมการแสดงออกของยีนที่มีความผิดพลาด เพราะสร้างมาจากข้อมูลเพียง 2 ระดับ คือ 0 และ 1 ดังนั้น นอกจากการพิจารณาถึงความสำคัญของ dicretization method ก่อนการวิเคราะห์ด้วยเทคนิคบูลีนแล้ว ควรปรับปรุงเทคนิคบูลีนให้สามารถสร้างเครือข่ายควบคุมการแสดงออกของยีนที่มีความซับซ้อนน้อยลง ลดความผิดพลาดในการทำนาย การใส่เงื่อนไขทางชีววิทยาให้กับการวิเคราะห์ด้วยเทคนิคบูลีน น่าจะลดความซับซ้อนและลดการทำนายที่ผิดพลาดที่มักจะเกิดขึ้นจากการวิเคราะห์ด้วยวิธีนี้ ทำให้ง่ายต่อการศึกษาและวิเคราะห์เครือข่ายที่สร้างขึ้นได้ต่อไป ดังนั้น งานวิจัยนี้จึงพัฒนาวิธีการที่ใช้สร้างเครือข่ายการแสดงออกของยีนแบบมีเงื่อนไข เรียกว่า Constraint-based Boolean network ด้วยภาษา C^{++} ซึ่งจะใช้ข้อมูลการแสดงออกของยีนใน galactose pathway ของยีสต์เป็นข้อมูลในการพัฒนาโปรแกรม

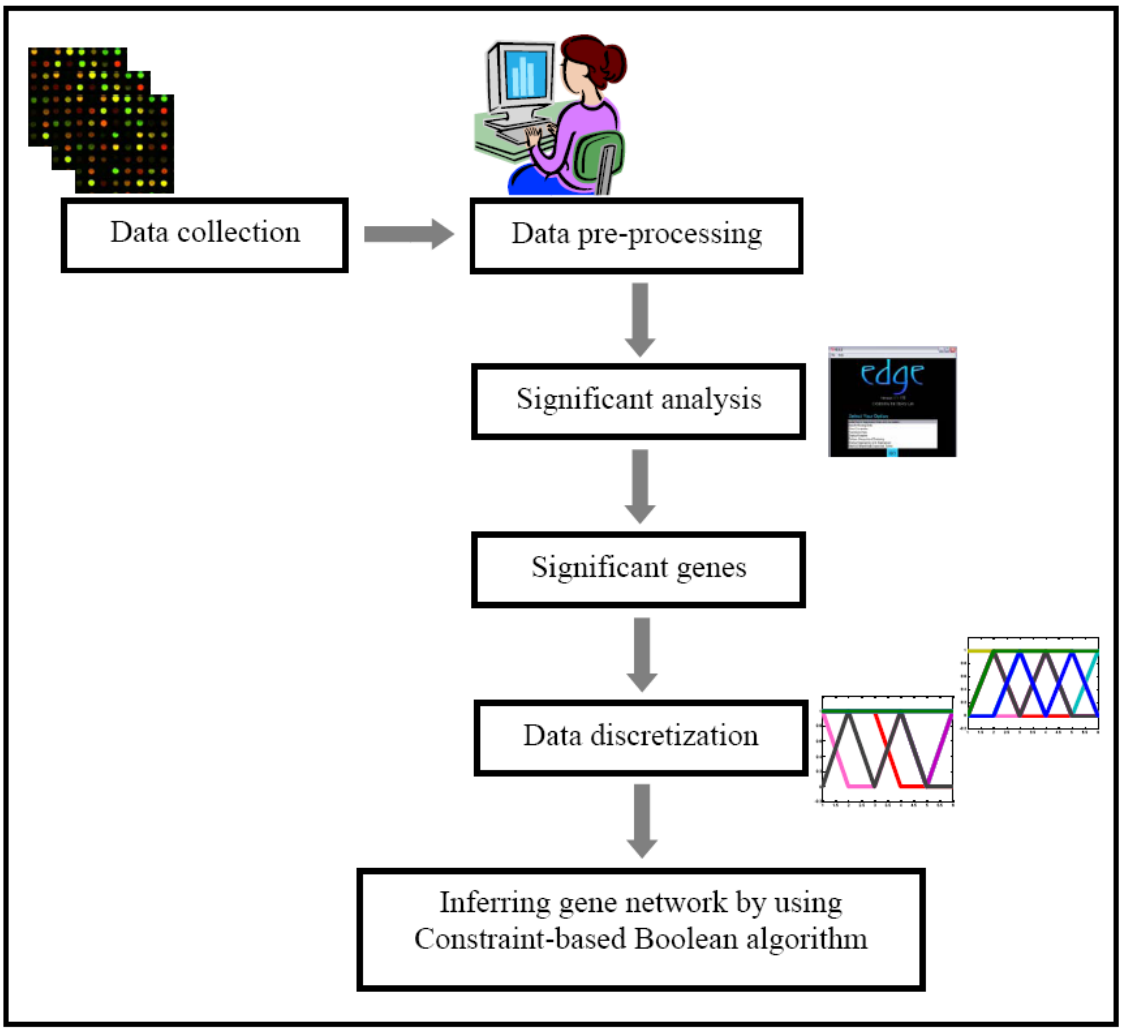
2.5 การทดสอบและประเมินโปรแกรม (Model validation)

หลังจากการพัฒนาโปรแกรมคอมพิวเตอร์เพื่อสร้างเครือข่ายควบคุมการแสดงออกของยีนแบบมีเงื่อนไข โปรแกรมที่พัฒนาขึ้นนั้นจำเป็นต้องมีการทดสอบเพื่อวัดประสิทธิภาพของโปรแกรมที่พัฒนาขึ้นโดยการเปรียบเทียบเครือข่ายควบคุมการแสดงออกของยีนอ้างอิง [14] ใน galactose pathway และเครือข่ายควบคุมการแสดงออกของยีนที่สร้างขึ้นจาก Constraint-based Boolean network ด้วยค่าสถิติ เช่น ค่าความถูกต้อง (accuracy) ค่าความแม่นยำ (precision) เป็นต้น

2.6 การประยุกต์ใช้โปรแกรมเพื่อสร้างและศึกษาเครือข่ายควบคุมการแสดงออกของยีนในกระบวนการสังเคราะห์แป้ง (Case study for starch application)

ในขั้นการทดลองนี้ จะนำเอาโปรแกรมที่พัฒนาขึ้นมาใช้สร้างเครือข่ายควบคุมการทำงานของยีนในกระบวนการสังเคราะห์แป้ง เพื่อทำความเข้าใจเกี่ยวกับกลไกการทำงานของกระบวนการสังเคราะห์แป้งมากขึ้น โดยเลือกเอาข้อมูลการแสดงออกของยีนในพืชต้นแบบ (plant model) นั่นคือ *Arabidopsis thaliana* ซึ่งข้อมูลการแสดงออกของยีนต้องผ่านกระบวนการ pre-processing และหา

ยีนที่มีการแสดงออกอย่างมีนัยสำคัญ (Significant genes) และหายีนที่เกี่ยวข้องกับกระบวนการสังเคราะห์แป้งโดยใช้ข้อมูลยีนในงานวิจัยของ [14] ข้อมูลที่ผ่านการ discretization จะนำไปสร้างเครือข่ายควบคุมการทำงานของยีนในกระบวนการสังเคราะห์แป้งด้วย Constraint-based Boolean network



รูปที่ 2.2 ขั้นตอนการวิเคราะห์เครือข่ายควบคุมการแสดงออกของยีนในกระบวนการสังเคราะห์แป้ง

บทที่ 3
ผลดำเนินงานวิจัย

3.1 การเก็บและรวบรวมข้อมูล (Data collection)

การทดลองนี้จะใช้ข้อมูล 2 ชุด ดังตารางที่ 3.1 ประกอบด้วย 1) ข้อมูลการแสดงออกของยีนในสภาวะ diauxic shift ของ *Saccharomyces cerevisiae* [15, 16] จากฐานข้อมูล Stanford Microarray (http://cmgm.stanford.edu/pbrown/explore/array.txt) เป็นข้อมูลทดสอบสำหรับการวิเคราะห์เริ่มต้นและการพัฒนาโปรแกรมสร้างเครือข่ายการแสดงออกของยีนแบบมีเงื่อนไข (test data) และ 2) ข้อมูลการแสดงออกภายใต้สภาวะ diurnal cycle ของ *Arabidopsis* [14] จากฐานข้อมูล NASC's International Affymetrix Service (NASCArrays) (http://affymetrix.arabidopsis.info/) ใช้เป็น case study สำหรับสร้างและศึกษาเครือข่ายควบคุมการแสดงออกของยีนในกระบวนการสังเคราะห์แป้ง

ตารางที่ 3.1 ข้อมูลการแสดงออกของยีนที่ใช้ในงานวิจัยนี้

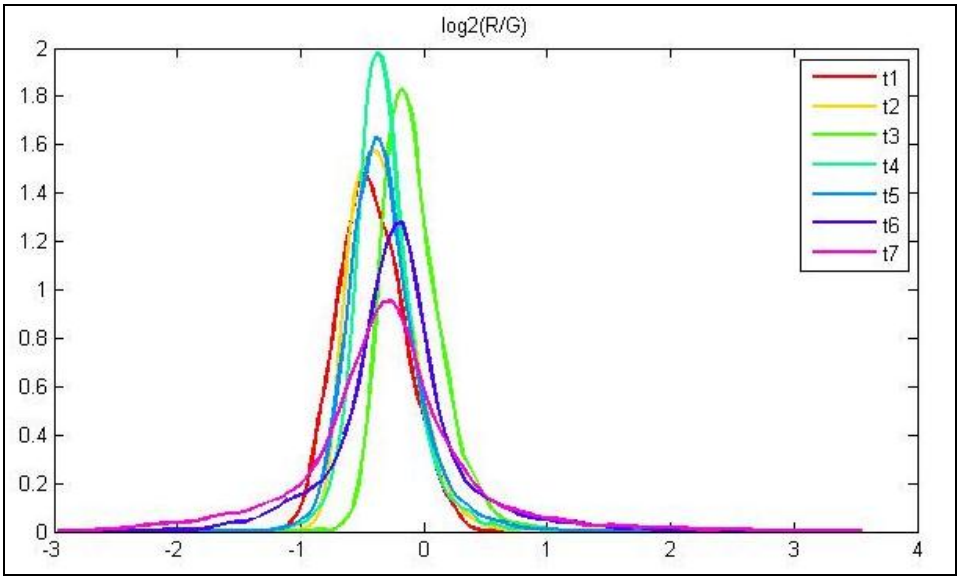
ข้อมูล	สิ่งมีชีวิต	เทคโนโลยีไมโครอะเรย์	จำนวนยีนในชุดข้อมูล	Database
Test data	*S. cerevisiae*	cDNA microarray	6,153 ยีน	Stanford Microarray Database http://cmgm.stanford.edu/pbrown/explore/array.txt
Case study	*A. thalian*	Affymetrix GeneChips	22,810 ยีน	NASCArrays http://affymetrix.arabidopsis.info/narrays/experimentpage.pl?experimentid=60

ข้อมูลการแสดงออกของยีนของ test data เป็นข้อมูลการศึกษาการแสดงออกของยีน 6,153 ยีน ภายใต้สภาวะ diauxic shift คือ การเปลี่ยนกระบวนการหมักเป็นการหายใจแบบใช้ออกซิเจนใน 7 ช่วงเวลา (time points; T) คือ ณ เวลาเริ่มต้น (0 ชั่วโมง; T_1) 9.5 ชั่วโมง (T_2), 12 ชั่วโมง (T_3), 13 ชั่วโมง (T_4), 15 ชั่วโมง (T_5), 17 ชั่วโมง (T_6) และ 18.5 ชั่วโมง (T_7) ซึ่งเป็นช่วงเวลาศึกษาการแสดงออกของยีนเมื่อได้รับปริมาณกลูโคสปริมาณ คือ T_1 =19, T_2 =18.7, T_3 =17.6, T_4 =14, T_5 =7.5, T_6 =0.2, และ T_7 =0 g/l Glucose

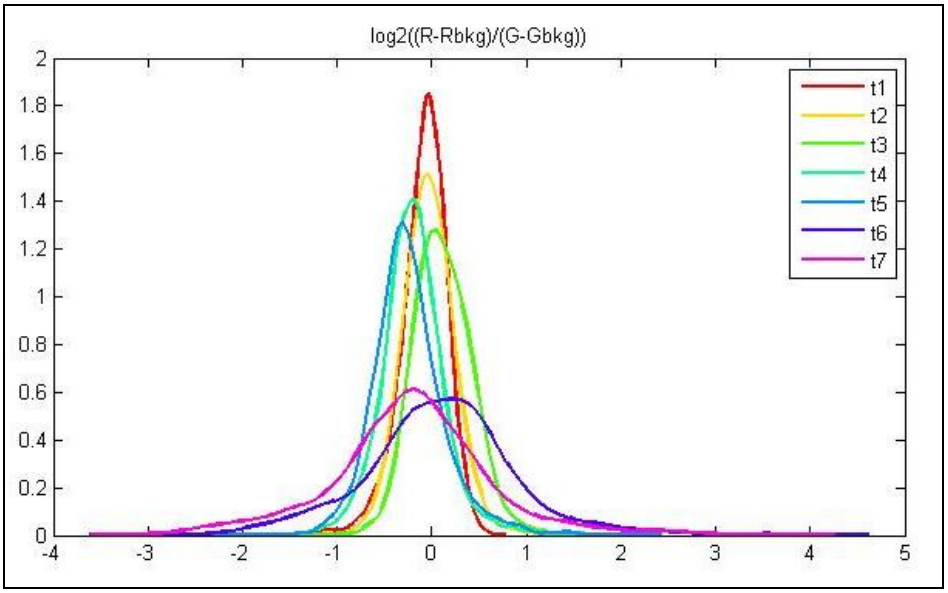
3.2 การเตรียมข้อมูล (Data preprocessing)

ก่อนที่จะนำข้อมูลการแสดงออกของยีนไปใช้นั้น ต้องมีการเตรียมข้อมูลให้เหมาะสม รวมทั้งมีการปรับค่าการแสดงออกของยีนให้อยู่ในมาตรฐานเดียวกัน เรียกว่า normalization เพื่อสามารถนำเอาการแสดงออกของยีนแต่ละยีนมาเปรียบเทียบกันได้ โดยเตรียมข้อมูล galactose pathway ให้อยู่ในรูปแบบ $\log_2$ratio และพิจารณาการกระจายตัวของข้อมูลการแสดงออกของยีนในรูปแบบ $\log_2$ratio (รูปที่ 3.1) ว่ามีการกระจายตัวของข้อมูลเป็นแบบปกติ (normal distribution) ซึ่งจะต้องมีค่าเฉลี่ยอยู่ที่ 0 แต่จากข้อมูลพบว่าข้อมูลมีการกระจายตัวออกห่าง 0 จึงต้องมีการปรับค่าของข้อมูลให้การกระจายตัวของข้อมูลเข้าใกล้ 0 คือ โดยการหักลบความผิดพลาดเนื่องมาจาก dye bias ด้วยการหักลบค่า background intensity (background correction) ของการแสดงออกของยีนออก (รูปที่ 3.2) อย่างไรก็ตาม จะเห็น

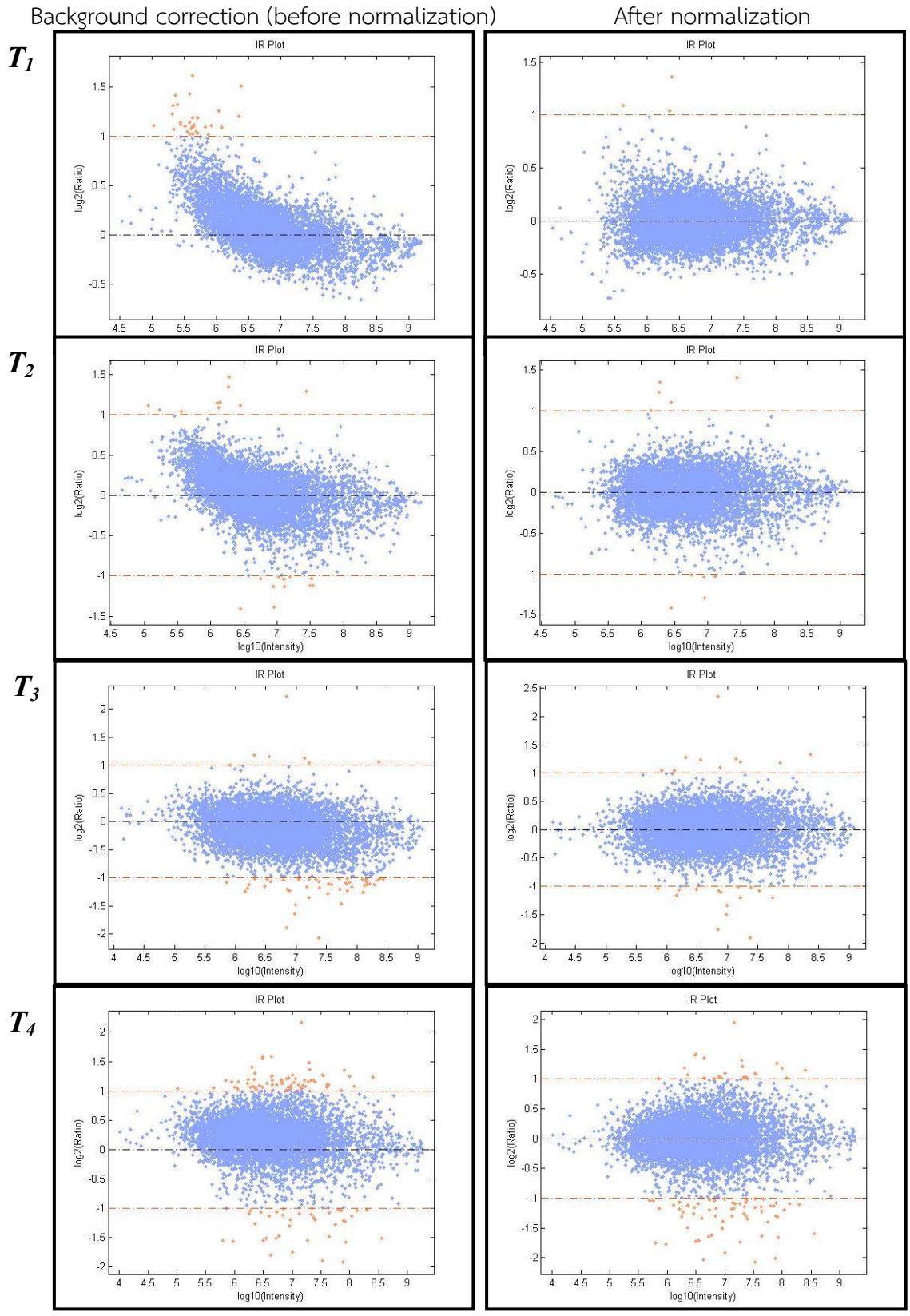
ว่า แม้มีการปรับค่า background ออกแล้ว ยังมีบางช่วงเวลาที่มีการกระจายตัวแตกต่างออกไป ดังนั้น จึงมีการปรับข้อมูล (normalization) ด้วยวิธี *lowess* เพื่อปรับให้ข้อมูลอยู่มาตรฐานเดียวกัน คือ เข้าใกล้ 0 ให้สามารถเปรียบเทียบข้อมูลการแสดงออกของยีนระหว่างช่วงเวลาได้ (รูปที่ 3.3-3.5)



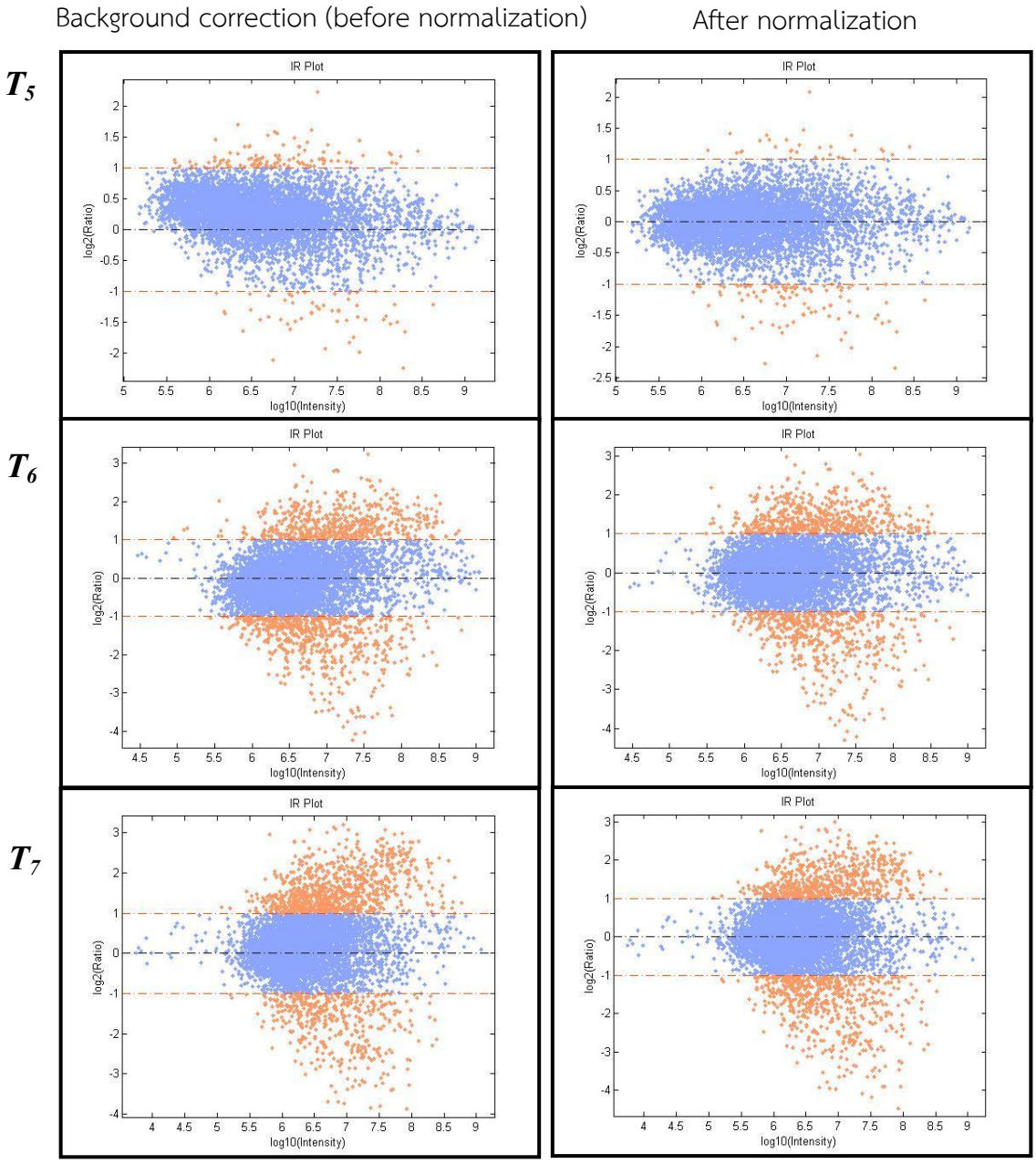
รูปที่ 3.1 การกระจายตัวของข้อมูลการแสดงออกของยีนของทุกช่วงเวลา (T_1-T_7) ในรูปแบบ $\log_2$ratio



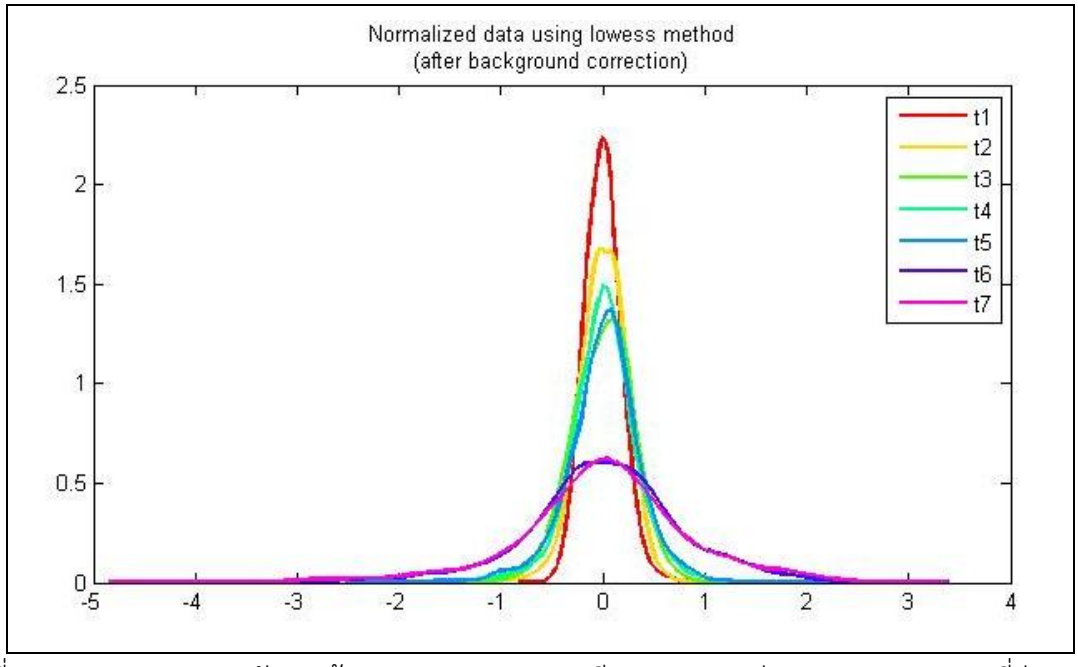
รูปที่ 3.2 การกระจายตัวของข้อมูลการแสดงออกของยีนของทุกช่วงเวลา (T_1-T_7) ที่ผ่านการ normalization ด้วย background correction แล้ว



รูปที่ 3.3 เปรียบเทียบ IR plot (ใช้ Bioinformatics Toolbox ของ MATLAB) ของข้อมูลการแสดงออกของยีนแต่ละช่วงเวลา (T_1-T_4) ในรูปแบบ $\log_2$ratio ที่ผ่าน background correction และ ที่ผ่านการ normalization ด้วยวิธี *lowess*



รูปที่ 3.4 เปรียบเทียบ IR plot (ใช้ Bioinformatics Toolbox ของ MATLAB) ของข้อมูลการแสดงออกของยีนแต่ละช่วงเวลา (T_5-T_7) ในรูปแบบ $\log_2$ratio ที่ผ่าน background correction และที่ผ่านการ normalization ด้วยวิธี *lowess*



รูปที่ 3.5 การกระจายตัวของข้อมูลการแสดงออกของยีนของของทุกช่วงเวลา (T_1-T_7) ที่ผ่านการ normalization ด้วยวิธี *lowess*

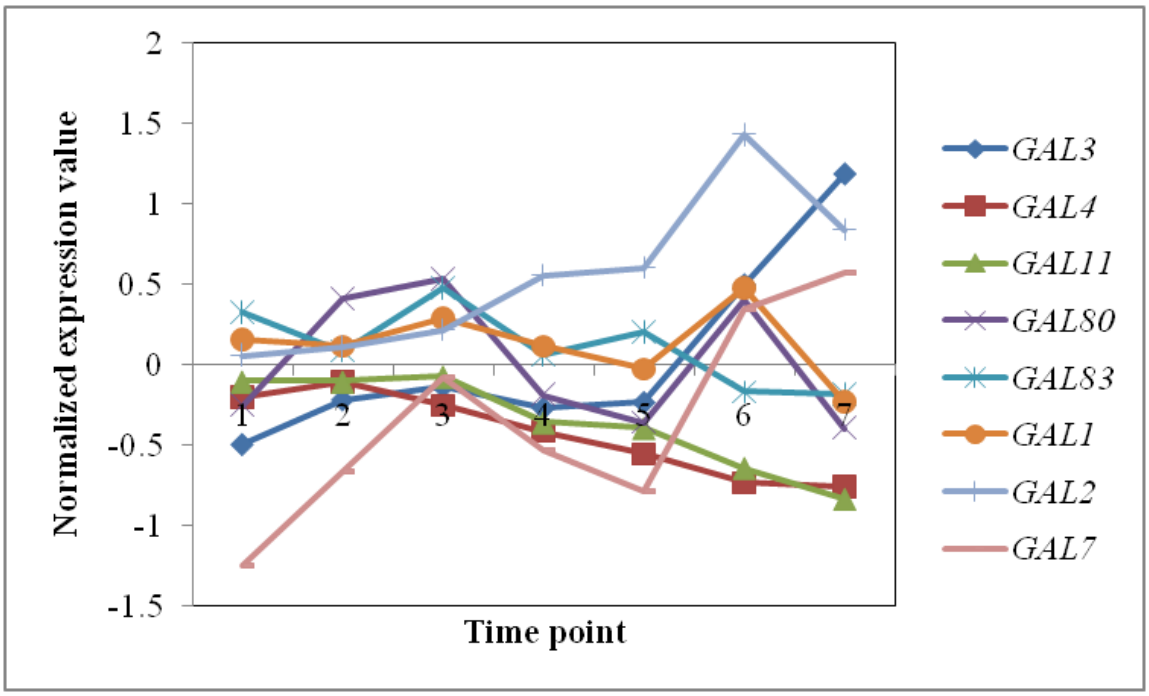
ส่วนข้อมูล Affymetrix GeneChips เป็นข้อมูลการแสดงออกของยีนของพืช Arabidopsis ภายใต้สภาวะกลางวันและกลางคืน มีทั้งหมด 11 ช่วงเวลา (time points; *T*) คือ ณ เวลาเริ่มต้นในช่วงเวลา (0 ชั่วโมง; T_1) 1 ชั่วโมง (T_2), 2 ชั่วโมง (T_3), 4 ชั่วโมง (T_4), 8 ชั่วโมง (T_5), 12 ชั่วโมง (T_6) 13 ชั่วโมง (T_7), 14 ชั่วโมง (T_8), 16 ชั่วโมง (T_9) 20 ชั่วโมง (T_{10}) และ 24 ชั่วโมง (T_{11}) โดยช่วงเวลา T_1 - T_5 คือ ช่วงเวลากลางคืน T_6 – T_{10} คือ ช่วงเวลากลางวัน ส่วน T_6 เป็นช่วงเวลาซ้อนทับช่วงเวลา T_1 เตรียมข้อมูลการแสดงออกของยีนชุดนี้ใช้ภาษา R [http://www.r-project.org/] ใน Bioconductor [http://www.bioconductor.org] [17, 18] โดยเตรียมข้อมูลการแสดงออกของยีนให้อยู่ในรูปแบบ $\log_2$intensity และหายีนที่มีการแสดงออกอย่างมีนัยสำคัญด้วยโปรแกรม EDGE [19] เวอร์ชั่น 1.1.175 [http://faculty.washington.edu/jstorey/edge/] ก่อนนำไปวิเคราะห์สร้างเครือข่ายควบคุมการแสดงออกของยีน

3.3 แปลงค่าการแสดงออกของยีนและการประเมินวิธีการที่ใช้ในการแปลงข้อมูลการแสดงออกของยีน (Data Discretization)

ข้อมูลที่ผ่านการ Normalization แล้วจะนำมาแปลงค่าการแสดงออกยีนเป็น 2 ระดับ (binary values) คือ 0 และ 1 ซึ่งหมายถึงไม่มีการแสดงออกของยีน และมีการแสดงออกของยีน ตามลำดับ เพื่อสะดวกต่อการวิเคราะห์ด้วยเทคนิคบูลีนในขั้นตอนต่อไป ขั้นตอนนี้เรียกว่า “discretization” ในงานวิจัยในสนใจการวิเคราะห์การแสดงออกของยีนที่เกี่ยวข้องกับ galactose pathway โดยเลือกเอาเฉพาะกลุ่มยีนที่เกี่ยวข้องกับ galactose pathway (ตารางที่ 3.2) ที่มีข้อมูลของการควบคุมเครือข่ายการแสดงออกของยีน [15] มาวิเคราะห์ต่อไป เพราะจะได้สามารถประเมินประสิทธิภาพของงานวิจัยนี้ได้ ข้อมูลการแสดงออกของกลุ่มยีนดังกล่าว ดังรูปที่ 3.6

ตารางที่ 3.2 ยีนที่เกี่ยวข้องกับ galactose pathway ที่ใช้ในงานวิจัยนี้ [15]

Gene name	Protein	Enzyme	Function
GAL2	*gal2p*	permease	Transports galactose into the cell
GAL1	*gal1p*	Galactokinase	Conversion of intracellular galactose
GAL7	*gal7p*	Uridylyltransferase	Conversion of intracellular galactose
GAL10	*gal10p*	Epimerase	Conversion of intracellular galactose
GAL4	*gal4p*	Regulatory gene	Promotes transcription of *GAL* genes
GAL80	*gal80p*	Regulatory gene	Binds *gal4p* and inhibits its activity (absent of galactose)
GAL3	*gal3p*	Regulatory gene	Associate with *gal80p* to release its repression of *gal4p*



รูปที่ 3.6 การแสดงออกของกลุ่มยีนที่เกี่ยวข้องกับ galactose pathway ในทุกช่วงเวลา (T_1-T_7) ที่ผ่านการ normalization ด้วยวิธี *lowess*

วิธีที่ใช้ในการแปลงค่าการแสดงออกของยีนให้เป็น 0 หรือ 1 (discretization method) ในงานวิจัยนี้คือ

- Mean คือ ค่าเฉลี่ยของการแสดงออกของยีนของทุกๆช่วงเวลา ถ้าค่าการแสดงออกของยีนนั้นๆ มีค่ามากกว่า Mean ก็มีค่าเท่ากับ 1 ถ้าน้อยกว่ามีค่าเท่ากับ 0
- Max-x%Max โดย Max คือ ค่าการแสดงออกของยีนแต่ละยีนที่มีค่ามากที่สุดของทุกๆช่วงเวลา (time points), x คือ 10, 20, 30, ..., 90 ถ้าค่าการแสดงออกของยีนนั้นๆ มีค่ามากกว่า Max-x%Max ก็มีค่าเท่ากับ 1 ถ้าน้อยกว่ามีค่าเท่ากับ 0

- Sign of $\log_2$ratio คือ ถ้าค่าการแสดงออกของยีนมีค่าน้อยกว่าศูนย์จะให้เท่ากับ 0 ถ้ามีค่ามากกว่าศูนย์ จะให้เท่ากับ 1

จะเห็นว่าเรามีการพิจารณาวิธีที่ใช้แปลงค่าการแสดงออกของยีนหลายวิธี ดังนั้น จึงมีการวิเคราะห์เพื่อหาวิธีที่เหมาะสมที่สุดที่นำมาใช้แปลงค่าการแสดงออกของยีนที่ผ่านการ normalization (raw data) มาเป็นข้อมูล binary values (0 หรือ 1) ก่อนนำไปวิเคราะห์ต่อไปด้วยเทคนิคบูลีน ข้อมูลดิบที่ผ่านการ normalization (raw data) และข้อมูล binary values ของยีนที่เกี่ยวข้องกับ galactose pathway โดยใช้วิธีการ Mean Max-x%Max และ Sign of $\log_2$ratio แสดงดังตารางที่ 3.3-3.6 ตามลำดับ

ตารางที่ 3.3 ข้อมูลการแสดงออกของยีนที่เกี่ยวข้องกับ galactose pathway ที่ผ่านกระบวนการ normalization แล้ว

	T_1	T_2	T_3	T_4	T_5	T_6	T_7
GAL3	-0.49411	-0.21759	-0.13606	-0.26882	-0.23447	0.4957	1.18269
GAL4	-0.20091	-0.1047	-0.25154	-0.41504	-0.55639	-0.73697	-0.76121
GAL11	-0.1047	-0.1047	-0.074	-0.35845	-0.39593	-0.64386	-0.8365
GAL80	-0.25154	0.41143	0.53605	-0.18442	-0.35845	0.40054	-0.39593
GAL83	0.32193	0.08406	0.47508	0.05658	0.20163	-0.16812	-0.18442
GAL1	0.15056	0.11103	0.28688	0.11103	-0.02915	0.47508	-0.23447
GAL2	0.05658	0.11103	0.21412	0.55582	0.60407	1.43296	0.83996
GAL7	-1.25154	-0.66658	-0.074	-0.53533	-0.78588	0.34483	0.57531
GAL10	-0.55639	-0.39593	0.66903	-0.13606	-0.76121	0.64155	0.78241

ตารางที่ 3.4 ข้อมูล Binary values ของยีนที่เกี่ยวข้องกับ galactose pathway ด้วยวิธี Mean

	T_1	T_2	T_3	T_4	T_5	T_6	T_7
GAL3	0	0	0	0	0	1	1
GAL4	1	1	1	1	0	0	0
GAL11	1	1	1	1	0	0	0
GAL80	0	1	1	0	0	1	0
GAL83	1	0	1	0	1	0	0
GAL1	1	0	1	0	0	1	0
GAL2	0	0	0	1	1	1	1
GAL7	0	0	1	0	0	1	1
GAL10	0	0	1	0	0	1	1

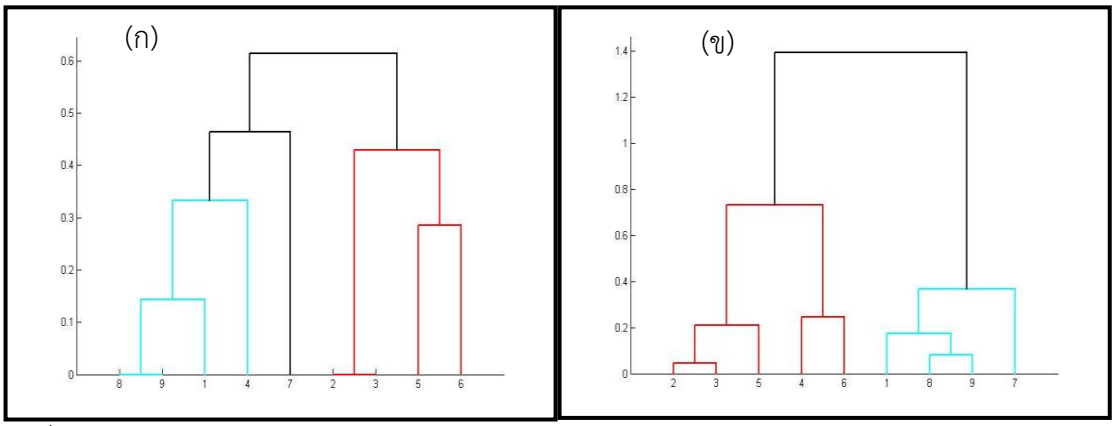
ตารางที่ 3.5 ข้อมูล Binary values ของยีนที่เกี่ยวข้องกับ galactose pathway ด้วยวิธี Max-x%Max

	T_1	T_2	T_3	T_4	T_5	T_6	T_7
GAL3	0	0	0	0	0	1	1
GAL4	0	0	0	0	0	0	0
GAL11	0	0	0	0	0	0	0
GAL80	0	1	1	0	0	1	0
GAL83	1	1	1	1	1	0	0
GAL1	1	1	1	1	0	1	0
GAL2	1	1	1	1	1	1	1
GAL7	0	0	0	0	0	1	1
GAL10	0	0	1	0	0	1	1

ตารางที่ 3.6 ข้อมูล Binary values ของยีนที่เกี่ยวข้องกับ galactose pathway ด้วยวิธี Sign of $\log_2$ratio

	T_1	T_2	T_3	T_4	T_5	T_6	T_7
GAL3	0	0	0	0	0	1	1
GAL4	0	0	0	0	0	0	0
GAL11	0	0	0	0	0	0	0
GAL80	0	1	1	0	0	1	0
GAL83	1	0	1	0	1	0	0
GAL1	0	0	1	0	0	1	0
GAL2	0	0	0	0	1	1	1
GAL7	0	0	0	0	0	1	1
GAL10	0	0	1	0	0	1	1

เพื่อให้ได้เครือข่ายการควบคุมการแสดงออกของยีนที่สอดคล้องกับข้อมูลในห้องปฏิบัติการมากที่สุด งานวิจัยนี้ได้การเปรียบเทียบค่าความเหมือน (Similarity score) ของเดนโดแกรม (dendogram) ของการจัดกลุ่มความเหมือนของกลุ่มยีนที่เลือกมาในชุดข้อมูล raw data และเดนโดแกรมของข้อมูล binary values ว่ามีความเหมือนหรือต่างกันเพียงใด โดยการสร้างเดนโดแกรมด้วยวิธี hierarchical clustering โดยเขียนโปรแกรมเพื่อคำนวณหา similarity score ใน MATLAB [*ภาคผนวก ก*] ตัวอย่างการเปรียบเทียบความเหมือนของเดนโดแกรมของข้อมูลทั้งสองแบบ โดยมีสมมติฐานว่า discretization method ที่สามารถเปลี่ยนข้อมูลดิบเป็น binary values แล้วทำให้ข้อมูลทั้งสองชุดนี้คล้ายกันด้วยค่า similarity score สูงสุด น่าจะเป็นวิธีการที่เหมาะสมที่สุดสำหรับแปลงค่าการแสดงออกของยีน เพื่อวิเคราะห์ด้วยเทคนิคบูลีนแบบมีเงื่อนไข (Constraint-based Boolean network) เพราะให้ค่าสอดคล้องกับความเป็นจริง (raw data) มากที่สุด ตัวอย่างเดนโดแกรมของ raw data และ binary profiles แสดงในรูปที่ 3.7

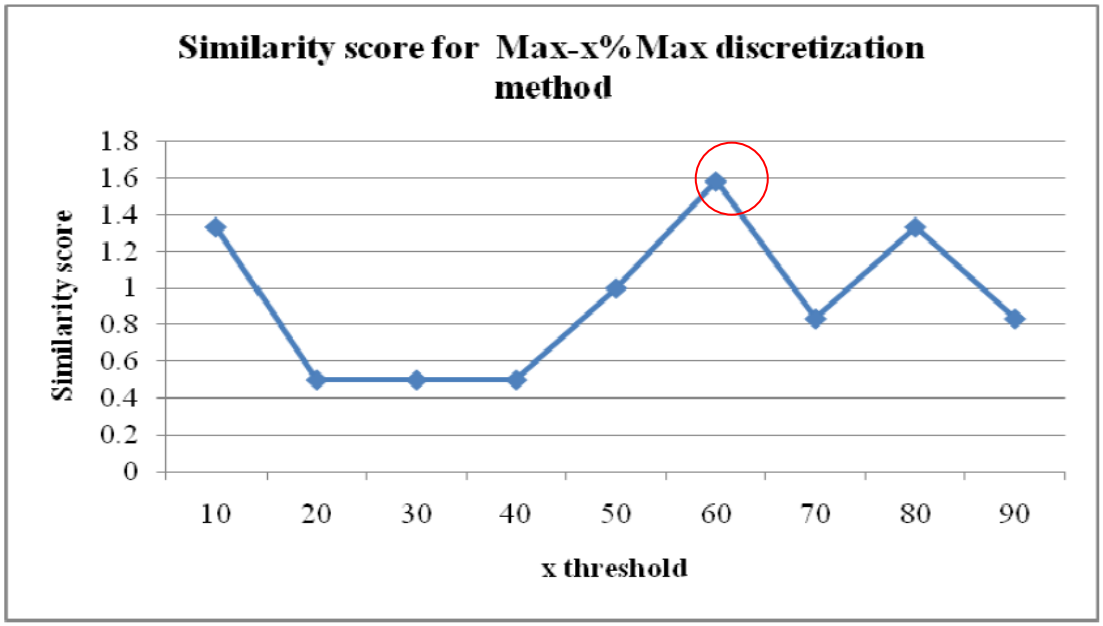


รูปที่ 3.7 การเปรียบเทียบความเหมือนกันระหว่างเดนโดแกรมของข้อมูล (ก) raw data/reference และ (ข) ข้อมูล binary profiles ที่ใช้วิธี Mean มาแปลงค่าการแสดงออกของยีนเป็น 0 หรือ 1

จากรูปที่ 3.7 ลำดับเลขด้านล่างของเดนโดแกรมแต่ละรูปคือ ยีน ตัวอย่างเช่น รูป 3.7ก เมื่อเปรียบเทียบคู่ยีนในข้อมูล raw data พบว่ายีน 2 และ ยีน 3 มีความใกล้เคียงกันมากที่สุด เมื่อเทียบเคียงกับคู่ยีนคู่อื่นๆ ก็จะถูกจัดอยู่ในกลุ่มเดียวกันก่อน และไล่จัดกลุ่มต่อไปตามหลักการของ hierarchical clustering เมื่อได้เดนโดแกรมของ raw data ก็จะนำมาเปรียบเทียบกับเดนโดแกรมของ binary values ที่แปลงโดย discretization method ต่างๆ (ในรูป 3.7 นี้ใช้วิธีการ mean) (รูป 3.7ข) ด้วยค่า similarity score และจากการเปรียบเทียบค่า similarity score พบว่า Max-60%Max นั้นให้ค่า similarity score สูงสุด ดังตารางที่ 3.7 และรูปที่ 3.8 ดังนั้น จึงเลือกวิธี Max-60%Max เพื่อไปแปลงข้อมูลการแสดงออกของยีนเป็น binary values ก่อนการสร้างเครือข่ายควบคุมการแสดงออกของยีนด้วยเทคนิคบูลีนแบบมีเงื่อนไข (ที่พัฒนาขึ้น) ต่อไป

ตารางที่ 3.7 เปรียบเทียบ Similarity score ของการเปรียบเทียบเดนโดแกรมของ reference/reference กับ reference -binary profiles ที่ได้จาก discretization methods ทั้ง 3 วิธี

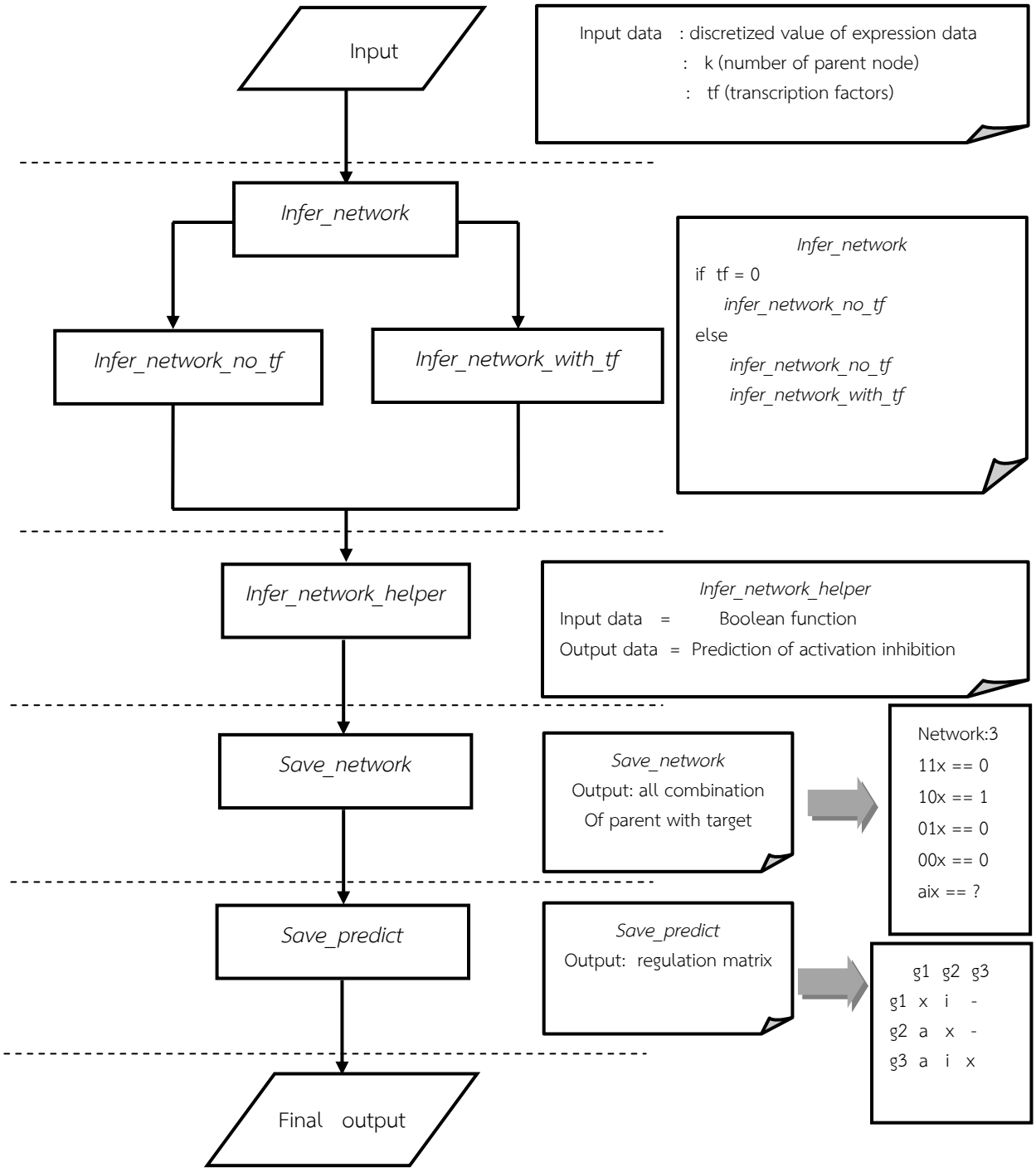
Comparing dendogram	Similarity score	Normalized similarity score
Reference/reference	2.6167	1
Reference/Mean	1.3333	0.51
Reference/Sig of Log_2Ratio	0.8333	0.32
Reference/Max-60%Max	1.5833	0.61



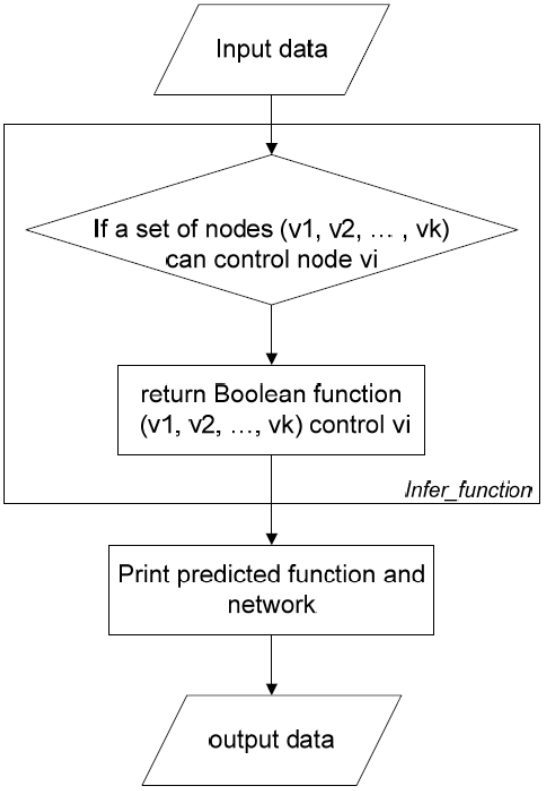
รูปที่ 3.8 Similarity score ของการเปรียบเทียบเดนโดแกรมของ raw data/reference กับ binary values ที่ได้จากวิธีการ Max-x%Max เมื่อ x = 10, 20, 30,..., 90

3.4 พัฒนาโปรแกรมที่ใช้ในการสร้างเครือข่ายการแสดงออกของยีนแบบมีเงื่อนไข (Constraint-based Boolean network)

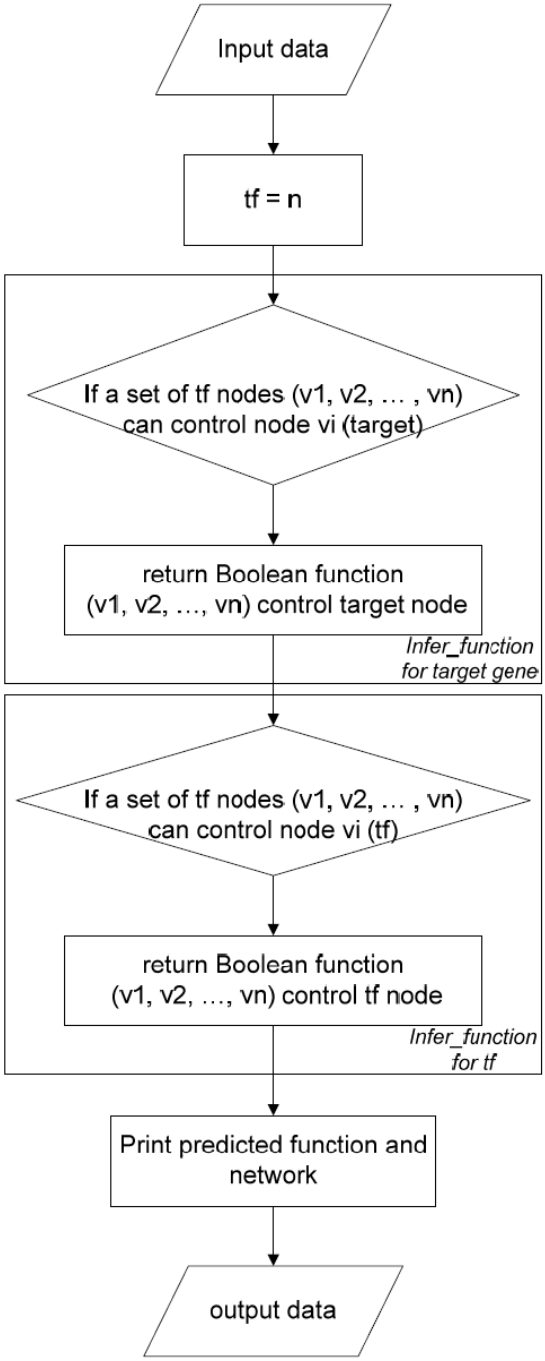
การจะศึกษาเครือข่ายการแสดงออกของยีนโดยใช้ Boolean network ที่มีการแปลงข้อมูลเพียง 2 ระดับ คือ 0 และ 1 ทำให้ได้เครือข่ายควบคุมการแสดงออกของยีนที่มีความซับซ้อน เนื่องจากได้ false positive มาก ดังนั้น จึงพัฒนาโปรแกรมเพื่อสร้างเครือข่ายควบคุมการแสดงออกของยีน โดยเพิ่มเงื่อนไขทางชีววิทยา (biological constraint) บนพื้นฐานของหลักการ genetic circuit และข้อมูลทางห้องปฏิบัติการ เพื่อลดความผิดพลาดและความซับซ้อนของเครือข่ายที่สร้างขึ้น ซึ่งได้มีการเพิ่มเงื่อนไขทางชีววิทยาว่าเอนไซม์ไม่สามารถควบคุมโปรตีนควบคุม (transcription factors) ได้ เพื่อเป็นการลดความซับซ้อนของ network ลง ในงานวิจัยนี้ ใช้ภาษา C^{++} ในการพัฒนา Constraint-based Boolean network โดยใช้ข้อมูล galactose pathway [15] ที่ผ่านขั้นตอน 3.3 ขั้นตอนการทำงานของโปรแกรมที่พัฒนาขึ้นแสดงดังรูปที่ 3.9-3.10 ส่วนโค้ดสคริป C++ สำหรับสร้าง gene regulatory network ของกลุ่มยีนที่สนใจด้วยวิธี Constraint-based Boolean network [*ภาคผนวก ข*]



รูปที่ 3.9 ขั้นตอนการพัฒนาโปรแกรมการสร้างเครือข่ายการแสดงออกของยีนแบบมีเงื่อนไข



Non-constraint-based Boolean Network (classical Boolean network) [1]



Constraint-based Boolean network

รูปที่ 3.10 เปรียบเทียบ Non-constraint-based Boolean Network (classical Boolean network) และ Constraint-based Boolean network

จากรูป 3.9 โปรแกรมจะรับค่าข้อมูล binary values ผ่านทาง command line โดยการเรียกใช้โปรแกรมด้วยคำสั่ง

```
>./boolean.out  หรือ
>./boolean.out <k-value> <tf-value> <input> <output>
```

โดย *<k-value> <tf-value> <input> <output>* เป็นพารามิเตอร์ที่ผู้วิเคราะห์กำหนด มีรายละเอียดดังนี้

- *k-value* คือ จำนวนยีนที่มาควบคุม (parent genes) กำหนดมากสุด คือ 5
- *tf-value* คือ โปรตีนควบคุม (Transcription factors: TF) ซึ่งจำเป็นต้องเป็นเงื่อนไขทางชีววิทยา จากตัวอย่างไฟล์ข้อมูลดังตารางที่ 3.8 ถ้ากำหนด TF = 2 นั่นคือ g1 และ g2 สามารถควบคุม g3 แต่ g3 ไม่สามารถควบคุม g1 และ g2 ในการสร้างเครือข่ายควบคุมการแสดงออกของยีนอาจมี TF หรือไม่มีก็ได้ หากไม่มี TF โปรแกรมก็จะสร้างเครือข่ายควบคุมการแสดงออกของยีนโดยไม่ได้คำนึงถึงเงื่อนไขทางชีววิทยาที่ใส่เข้าไป แต่หากมีการใส่ TF ในการสร้างเครือข่ายควบคุมการแสดงออกของยีนด้วย โปรแกรมจะมีการพิจารณาว่า เอนไซม์ไม่ควรควบคุมการแสดงออกของ TF ได้ ดังนั้น โปรแกรมจะไม่พิจารณาฟังก์ชั่นบูลีนของเอนไซม์ที่มีต่อ TF นั้นๆ
- *input* คือ ชื่อไฟล์ข้อมูล binary values (.csv) ตัวอย่างไฟล์ข้อมูลดังตารางที่ 3.8 โดยจัด TF ไว้ column ต้นๆ แล้วตามด้วยเอนไซม์ ดังนั้น หากไฟล์ข้อมูล มี TF 2 ตัว เช่น g1 และ g2 ก็ต้องจัด g1 ไว้คอลัมน์แรก แล้วตามด้วย g2 ที่คอลัมน์สอง ส่วนคอลัมน์สามเป็นต้นไป คือ เอนไซม์ โปรแกรมจะรู้อัตโนมัติว่า TF อยู่ตรงตำแหน่งคอลัมน์ใดจากการกำหนดจำนวน TF ให้โปรแกรม
- *output* คือ ชื่อไฟล์ผลที่ได้จากการวิเคราะห์ด้วย Constraint-based Boolean network (.csv) ตัวอย่างไฟล์ผลที่ได้จากการวิเคราะห์ด้วย Constraint-based Boolean network ดังตารางที่ 3.9 โดยแถว คือ ยีนหรือโปรตีนควบคุม (parent genes) ส่วนคอลัมน์ คือ ยีนเป้าหมาย (target genes) “i” คือ inhibition เป็นความสัมพันธ์ระหว่างคู่ยีนแบบยับยั้ง (inhibition) ส่วน “a” คือ activation เป็นความสัมพันธ์ระหว่างคู่ยีนแบบกระตุ้น “-” คือ ไม่มีความสัมพันธ์ระหว่างคู่ยีน โดยการวิจัยนี้ตัดการพิจารณาการควบคุมยีนตัวเองออก ซึ่งก็คือ “x” ในผล จากตัวอย่างในตาราง 3.9 จะเห็นว่าเอนไซม์ g3 ถูกยับยั้งด้วยยีน g1 และกระตั้นด้วยยีน g2 ดังนั้นฟังก์ชั่นบูลีนของความสัมพันธ์นี้คือ g3 = g2 AND NOT g1 ซึ่งผลที่ได้จะแสดงเป็นเครือข่ายควบคุมการแสดงออกของยีนด้วยโปรแกรม Cytoscape [20]

จากการเปรียบเทียบขั้นตอนการทำงานโปรแกรม Non-constraint-based Boolean Network (classical Boolean network) [1] และ Constraint-based Boolean network ที่พัฒนาขึ้นมาในงานวิจัยนี้ แสดงดังรูปที่ 3.10

ตารางที่ 3.8 ตัวอย่างไฟล์ข้อมูล binary values (.csv) ที่ใช้วิเคราะห์ด้วย Constraint-based Boolean network

g1	g2	g3
0	1	1
0	0	0
1	0	0
1	1	1
0	1	0
0	0	0

ตารางที่ 3.9 ตัวอย่างไฟล์ผล (.csv) จากการวิเคราะห์ด้วย Constraint-based Boolean network

g1	g2	g3
g1	x	i
g2	a	x
g3	a	i

3.5 การทดสอบและประเมินโปรแกรม (Model validation)

หลังจากการพัฒนาโปรแกรมคอมพิวเตอร์เพื่อสร้างเครือข่ายควบคุมการแสดงออกของยีนแบบมีเงื่อนไขแล้ว โปรแกรมที่พัฒนาขึ้นนั้นจำเป็นต้องมีการทดสอบเพื่อวัดประสิทธิภาพของโปรแกรม Constraint-based Boolean network ที่พัฒนาขึ้น ทั้งนี้ มีการประเมินความถูกต้องของเครือข่ายควบคุมการแสดงออกของยีนใน galactose pathway ที่สร้างขึ้นด้วยข้อมูล binary values จากทุกวิธี คือวิธี Mean Max-x%Max และ Sign of $\log_2$ratio เพื่อทดสอบสมมติฐานที่ตั้งไว้ว่า discretization method ที่ให้ค่า similarity score ของการเปรียบเทียบเดนโดนแกรมของ binary values เทียบกับ raw data สูงที่สุดจะเป็นวิธีที่เหมาะสมที่สุดในการสร้างเครือข่ายควบคุมการแสดงออกของยีนด้วยเทคนิคบูลีน ข้อมูลที่ได้จากโปรแกรมที่พัฒนาขึ้นจากภาษา C^{++} นั้น ผลที่ได้จากโปรแกรมจะสามารถระบบรูปแบบการควบคุมกันของยีนว่า ยีนหนึ่งสามารถควบคุมแบบการแสดงออกของยีนอื่นแบบยับยั้ง (i; inhibition) หรือแบบกระตุ้น (a; activation) ส่วน x คือ ไม่มีการกระตุ้น/ยับยั้งกันระหว่างคู่ยีนนั้นๆ ท้ายสุดจะนำผลเครือข่ายควบคุมการแสดงออกของยีนที่สร้างจาก Constraint-based Boolean network ที่ผ่านขั้นตอน discretization ด้วยวิธีการต่างๆ มาเปรียบเทียบกับเครือข่ายควบคุมการแสดงออกของยีนอ้างอิง (reference network) แสดงในตารางที่ 3.10 ซึ่งเป็นข้อมูลทางห้องปฏิบัติการหรือเป็นข้อมูลที่มีการกล่าวอ้างมาก่อน โดยประเมินประสิทธิภาพของโปรแกรม Constraint-based Boolean network ด้วยค่าทางสถิติ [21] ดังนี้

1. False positive rate (FPR) คือ ค่าอัตราส่วนการที่โปรแกรมทำนายว่ามีความสัมพันธ์กันของคู่ยีนทั้งที่ความจริงไม่มี ต่อความสัมพันธ์ของคู่ยีนที่ไม่มีใน reference network คำนวณได้จาก FP/(TN+FP)
2. False negative rate (FNR) คือ ค่าอัตราส่วนการที่โปรแกรมทำนายว่าไม่มีความสัมพันธ์กันของคู่ยีนที่ถูกต้องจริงๆ ต่อในความสัมพันธ์ของคู่ยีนที่สามารถทำนายได้ทั้งหมด คำนวณได้จาก FN/(TP+FN)
3. False discovery rate (FDR) คือ อัตราส่วนการที่โปรแกรมทำนายผิดพลาด จากความสัมพันธ์ทั้งหมดที่ทำนายได้ คำนวณได้จาก FP/(TP+FP)
4. ค่าความถูกต้อง (accuracy) คือ ค่าความถูกต้องของโปรแกรม คำนวณได้จาก (TP+TN)/(TP+TN+TP+FN)
5. ความแม่นยำ (precision) คือ ค่าความแม่นยำของโปรแกรมในการสร้างเครือข่ายควบคุมการแสดงออกของยีน คำนวณได้จากรูปแบบความสัมพันธ์ของการ regulation คือ TR/(TR+FR+FI)
6. ความจำเพาะ (specificity) คือ ค่าความจำเพาะต่อของการทำนายความสัมพันธ์ในเครือข่ายที่ไม่มีอยู่จริง คำนวณได้จากรูปแบบความสัมพันธ์ของการ regulation คือ TZ/(TZ+FR)
7. ความไว (sensitivity) คือ สัดส่วนของการทำนายความสัมพันธ์ที่มีอยู่จริง ในการทำนายหรือสร้างเครือข่ายควบคุมการแสดงออกของยีน คำนวณได้จากรูปแบบความสัมพันธ์ของการ regulation คือ TR/(TR+FZ+FI)

โดยคำนวณจากค่าการวัดผลลัพธ์ออกมาเป็นคลาส positive หรือ negative ต่างๆ ดังนี้

- True Positive (TP) คือ สิ่งที่โปรแกรมทำนายว่าเป็นการควบคุมแบบ (i; inhibition) หรือกระตุ้น (a; activation) และสอดคล้องกับ reference network ว่าเป็นแบบยับยั้ง (i; inhibition) หรือกระตุ้น (a; activation) เช่นเดียวกัน
- True Negative (TN) คือ สิ่งที่โปรแกรมทำนายว่าไม่มีความสัมพันธ์กันระหว่างยีนและใน reference network ก็ไม่มีความสัมพันธ์ระหว่างยีนเช่นเดียวกัน
- False Positive (FP) คือ สิ่งที่โปรแกรมไม่สามารถทำนายความสัมพันธ์ระหว่างคู่ยีนได้ นั่นคือ ไม่มีความสัมพันธ์กันของคู่ยีน แต่ใน reference network มีความสัมพันธ์ระหว่างคู่ยีนเกิดขึ้นไม่ว่าจะเป็นการยับยั้งหรือกระตุ้น
- False Negative (FN) คือ สิ่งที่โปรแกรมทำนายว่ามีความสัมพันธ์ระหว่างคู่ยีนว่ามีความสัมพันธ์กันระหว่างคู่ยีน แต่ใน ใน reference network ไม่มีความสัมพันธ์ระหว่างคู่ยีนนั้น หรือ ความสัมพันธ์ของคู่ยีนแตกต่างจากใน reference network เช่น โปรแกรมทำนายว่าความสัมพันธ์เป็นการกระตุ้น แต่ใน reference network มีความสัมพันธ์ระหว่างคู่ยีนเป็นแบบยับยั้ง หรือโปรแกรมทำนายว่าความสัมพันธ์เป็นแบบยับยั้ง แต่ใน reference network มีความสัมพันธ์ระหว่างคู่ยีนเป็นการกระตุ้น

รูปแบบความสัมพันธ์ของการ regulation ในเครือข่ายโดยพิจารณารูปแบบความสัมพันธ์เป็น activation หรือ inhibition โดยสามารแบ่งการวัดค่าผลลัพธ์ของการ regulation ในเครือข่ายได้เป็น

- True regulation (TR) คือ โปรแกรมทำนายว่าเป็นการควบคุมแบบ (i; inhibition) หรือกระตุ้น (a; activation) และสอดคล้องกับ reference network ว่าเป็นแบบยับยั้ง (i; inhibition) หรือกระตุ้น (a; activation) เช่นเดียวกัน
- True zero (TZ) คือ โปรแกรมทำนายว่าไม่มีความสัมพันธ์กันระหว่างคู่ยีน สอดคล้องกับใน reference network ว่าไม่มีความสัมพันธีระหว่างคู่ยีนนั้นเช่นเดียวกัน
- False regulation (FR) คือ โปรแกรมทำนายว่ามีความสัมพันธ์กันระหว่างคู่ยีน ไม่ว่าจะเป็น activation หรือ inhibition แต่ไม่มีความสัมพันธ์นั้นใน reference network
- False zero (FZ) โปรแกรมทำนายว่าไม่มีความสัมพันธ์กันระหว่างคู่ยีน แต่ใน reference network มีความสัมพันธ์เป็นแบบ activation
- False interaction (FI) โปรแกรมทำนายว่ามีความสัมพันธ์กันระหว่างคู่ยีนแบบ inhibition แต่ใน reference network มีความสัมพันธ์เป็นแบบ activation

ผลของการวัดผลลัพธ์ออกมาเป็นคลาส positive หรือ negative ของเครือข่ายควบคุมการแสดงออกของยีนด้วย Constraint-based Boolean network โดยใช้วิธี Max-x%Max Mean และ Sign of $\log_2$ratio ดังตารางที่ 3.11-3.13 ตามลำดับ เครือข่ายควบคุมการแสดงออกของยีนด้วยโปรแกรม Constraint-based Boolean network โดยใช้ข้อมูล binary values จากทั้ง 3 วิธีการ แสดงดังรูปที่ 3.11 และผลการประเมินประสิทธิภาพของโปรแกรม Constraint-based Boolean network ในการสร้างส่วนเครือข่ายควบคุมการแสดงออกของยีนใน galactose pathway แสดงในรูป 3.12

ตารางที่ 3.10 Reference network ของ galactose pathway [16]

		Regulator genes						
		GAL3	*GAL4*	*GAL80*	*GAL1*	*GAL2*	*GAL7*	*GAL10*
Target genes	*GAL3*	x	a	-	-	-	-	-
	GAL4	-	x	i	-	-	-	-
	GAL80	i	a	x	-	-	-	-
	GAL1	a	a	i	x	-	-	-
	GAL2	a	a	i	-	x	-	-
	GAL7	a	a	i	-	-	X	-
	GAL10	a	a	i	-	-	-	x

ตารางที่ 3.11 คลาส positive และ negative prediction ของการประเมินประสิทธิภาพของโปรแกรมในการสร้างเครือข่ายควบคุมการแสดงออกของยีนใน galactose pathway จากวิธีการ Max-x%Max

	GAL3	*GAL4*	*GAL80*	*GAL1*	*GAL2*	*GAL7*	*GAL10*
GAL3	TN	FN	TN	TN	TN	TN	TN
GAL4	FP	TN	TP	TN	TN	TN	TN
GAL80	FN	FN	TN	TN	TN	TN	TN
GAL1	FN	FN	FN	TN	TN	TN	TN
GAL2	FN	FN	FN	TN	TN	TN	TN
GAL7	TP	TP	FN	TN	TN	TN	TN
GAL10	FN	FN	FN	TN	TN	TN	TN

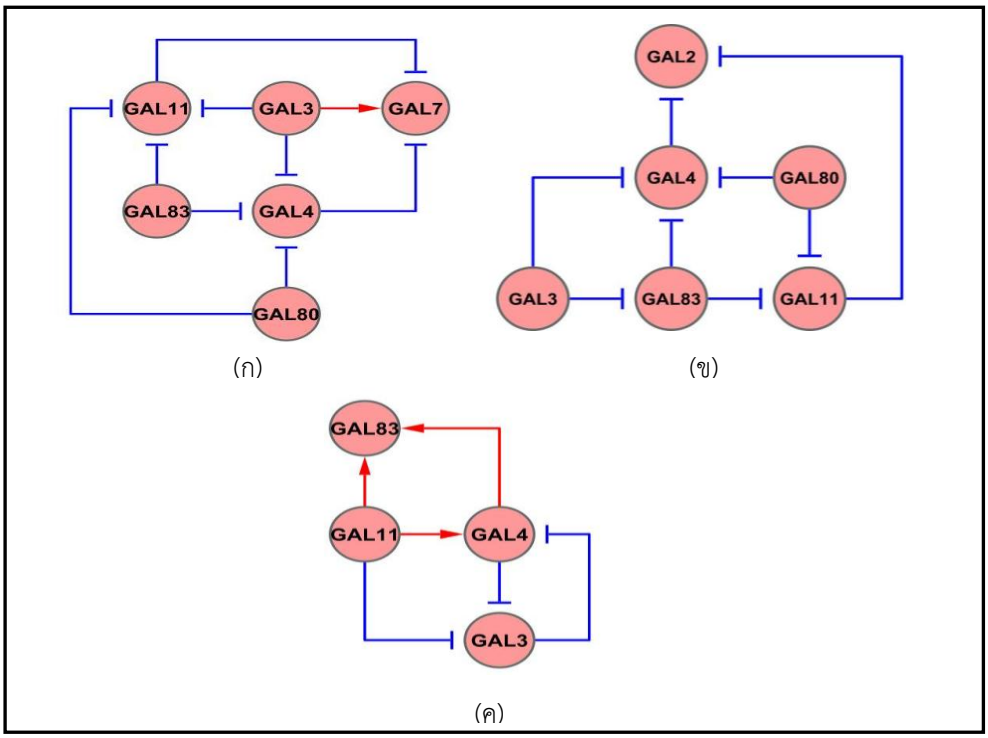
ตารางที่ 3.12 คลาส positive และ negative prediction ของการประเมินประสิทธิภาพของโปรแกรมในการสร้างเครือข่ายควบคุมการแสดงออกของยีนใน galactose pathway จากวิธีการ Mean

	GAL3	*GAL4*	*GAL80*	*GAL1*	*GAL2*	*GAL7*	*GAL10*
GAL3	TN	TP	FP	TN	TN	TN	TN
GAL4	TN	TN	FN	TN	TN	TN	TN
GAL80	FN	FN	TN	TN	TN	TN	TN
GAL1	FN	FN	FN	TN	TN	TN	TN
GAL2	FN	FN	FN	TN	TN	TN	TN
GAL7	FN	FN	FN	TN	TN	TN	TN
GAL10	FN	FN	FN	TN	TN	TN	TN

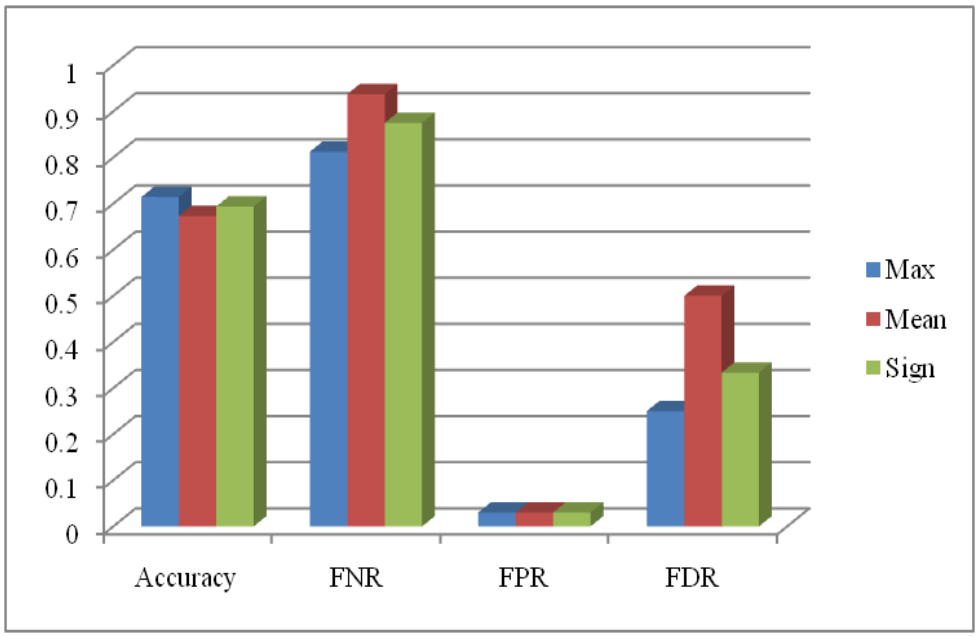
ตารางที่ 3.13 คลาส positive และ negative prediction ของการประเมินประสิทธิภาพของโปรแกรมในการสร้างเครือข่ายควบคุมการแสดงออกของยีนใน galactose pathway จากวิธีการ Sign of $\log_2$ratio

	GAL3	*GAL4*	*GAL80*	*GAL1*	*GAL2*	*GAL7*	*GAL10*
GAL3	TN	FN	TN	TN	TN	TN	TN
GAL4	FP	TN	TP	TN	TN	TN	TN
GAL80	FN	FN	TN	TN	TN	TN	TN
GAL1	FN	FN	FN	TN	TN	TN	TN
GAL2	FN	TP	FN	TN	TN	TN	TN
GAL7	FN	FN	FN	TN	TN	TN	TN
GAL10	FN	FN	FN	TN	TN	TN	TN

จากรูปที่ 3.12 จะเห็นได้ว่า discretization method ที่ให้ค่า similarity score ของการเปรียบเทียบเดนโดนแกรมของ binary values เทียบกับ raw data สูงที่สุด นั่นคือ Max-60%Max ไม่ใช่วิธีที่เหมาะสมที่สุดในการสร้างเครือข่ายควบคุมการแสดงออกของยีนด้วยเทคนิคบูลีน เพราะไม่สามารถสร้างเครือข่ายควบคุมการแสดงออกของยีนใน galactose pathway ที่ถูกต้องแม่นยำที่สุด แต่กลับพบว่าการใช้วิธี Mean แปลงค่าการแสดงออกของยีนเป็น 0 หรือ 1 นั้นสามารถช่วยให้การสร้างเครือข่ายด้วย Constraint-based Boolean network ได้ถูกต้องแม่นยำกว่าวิธีการอื่น นอกจากนี้ เมื่อลองเปรียบเทียบประสิทธิภาพของการเครือข่ายควบคุมการแสดงออกของยีนที่สร้างจากโปรแกรม Constraint-based Boolean network ที่พัฒนาขึ้นเปรียบเทียบกับ Non-Constraint-based Boolean network (classical Boolean algorithm) [1] โดยการใช้ข้อมูล binary values จากวิธี mean พบว่าการเพิ่มเงื่อนไขทางชีววิทยาใน Constraint-based Boolean network สามารถเพิ่มความถูกต้อง (accuracy) โดยลดความผิดพลาด (FDR) ของการทำนายลง ดังตารางที่ 3.14 รวมทั้งช่วยเพิ่มความแม่นยำ (precision) ความจำเพาะ (specificity) แม้ว่าความไวของโปรแกรม (sensitivity) ในการสร้างเครือข่ายควบคุมการแสดงออกของยีนของ Non-Constraint-based Boolean network (classical Boolean algorithm) [1] และ Constraint-based Boolean network จะไม่แตกต่างกันก็ตาม ดังรูปที่ 3.13 และรูปแบบความสัมพันธ์ของการ regulation ของเครือข่ายควบคุมการแสดงออกของยีนด้วย Mean Non-Constraint-based Boolean network (classical Boolean algorithm) [1] และ Constraint-based Boolean network แสดงดังตารางที่ 3.15-3.16



รูปที่ 3.11 เครือข่ายควบคุมการแสดงออกของยีนด้วยโปรแกรม Constraint-based Boolean network ด้วยข้อมูล binary values จากวิธีการ Max-x%Max (ข) Sign of $\log_2$ratio และ (ค) Mean



รูปที่ 3.12 ผลของการประเมินประสิทธิภาพของโปรแกรม Constraint-based Boolean network ในการสร้างเครือข่ายควบคุมการแสดงออกของยีนด้วยวิธี Mean Max-x%Max และ Sign of $\log_2$ratio

ตารางที่ 3.14 การเปรียบเทียบประสิทธิภาพของการเครือข่ายควบคุมการแสดงออกของยีนที่สร้างจากโปรแกรม Constraint-based Boolean network และ Classical Boolean algorithm [1]

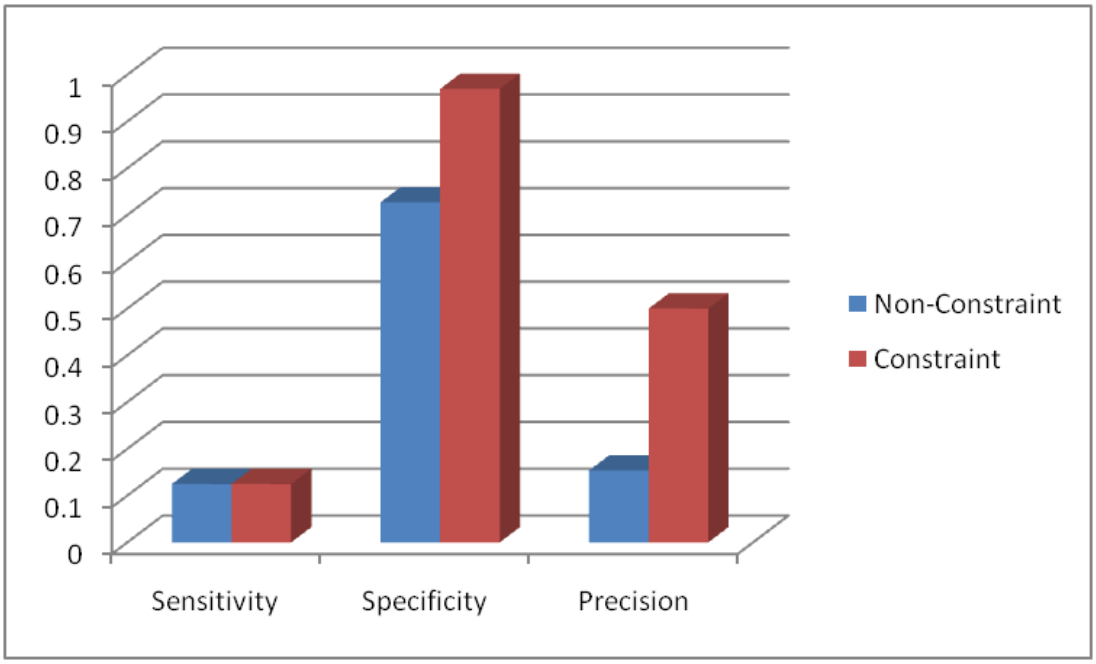
	Non-Constraint	Constraint-based
Accuracy	0.57	0.71
FNR	0.75	0.81
FPR	0.27	0.03
FDR	0.69	0.25

ตารางที่ 3.15 รูปแบบความสัมพันธ์ของการ regulation จากการประเมินประสิทธิภาพของโปรแกรม Constraint-based Boolean network ในการสร้างเครือข่ายควบคุมการแสดงออกของยีนใน galactose pathway จากวิธีการ Mean

	GAL3	*GAL4*	*GAL80*	*GAL1*	*GAL2*	*GAL7*	*GAL10*
GAL3	TZ	FZ	TZ	TZ	TZ	TZ	TZ
GAL4	FR	TZ	TR	TZ	TZ	TZ	TZ
GAL80	FZ	FZ	TZ	TZ	TZ	TZ	TZ
GAL1	FZ	FZ	FZ	TZ	TZ	TZ	TZ
GAL2	FZ	FZ	FZ	TZ	TZ	TZ	TZ
GAL7	TR	FI	FZ	TZ	TZ	TZ	TZ
GAL10	FZ	FZ	FZ	TZ	TZ	TZ	TZ

ตารางที่ 3.16 รูปแบบความสัมพันธ์ของการ regulation จากการประเมินประสิทธิภาพของโปรแกรม Classical Boolean algorithm [1] ในการสร้างเครือข่ายควบคุมการแสดงออกของยีนใน galactose pathway จากวิธีการ Mean

	GAL3	*GAL4*	*GAL80*	*GAL1*	*GAL2*	*GAL7*	*GAL10*
GAL3	FR	FI	TZ	TZ	FR	FR	TZ
GAL4	FR	TZ	TR	FR	FR	FR	FR
GAL80	FZ	FZ	TZ	TZ	TZ	TZ	TZ
GAL1	FZ	FZ	FZ	TZ	TZ	TZ	TZ
GAL2	FZ	FZ	FZ	TZ	TZ	TZ	TZ
GAL7	TR	FI	FZ	TZ	FR	TZ	TZ
GAL10	FZ	FZ	FZ	TZ	TZ	TZ	TZ



รูปที่ 3.13 การเปรียบเทียบประสิทธิภาพของการเครือข่ายควบคุมการแสดงออกของยีนที่สร้างจากโปรแกรม Constraint-based Boolean network ที่พัฒนาขึ้นเปรียบเทียบกับ Non-Constraint-based Boolean network (classical Boolean algorithm) [1]

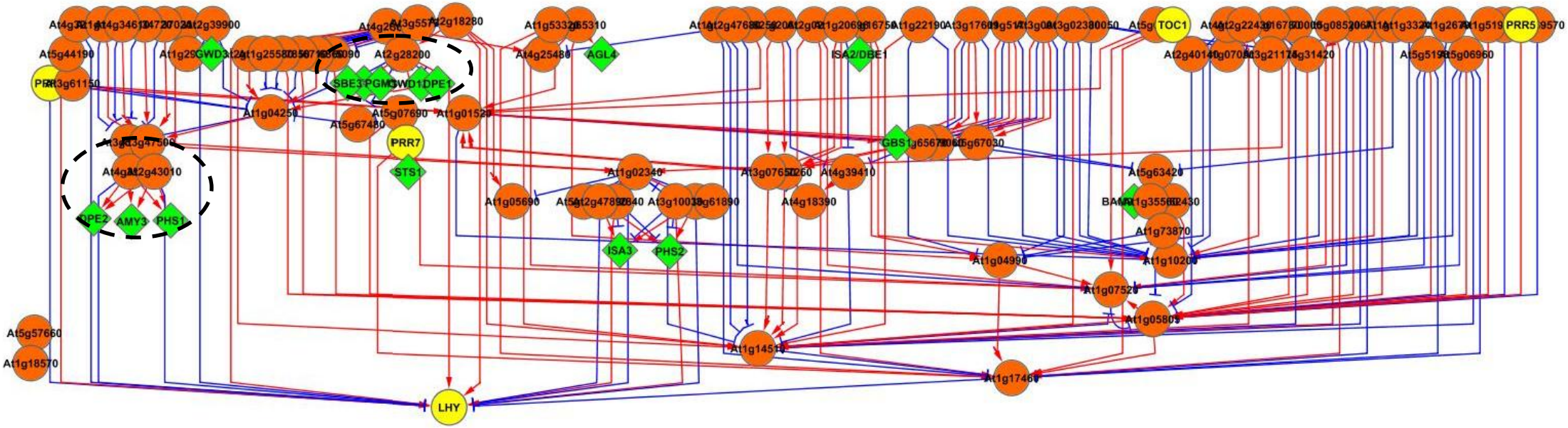
3.6 การประยุกต์ใช้โปรแกรมเพื่อสร้างและศึกษาเครือข่ายควบคุมการแสดงออกของยีนในกระบวนการสังเคราะห์แป้ง (case study)

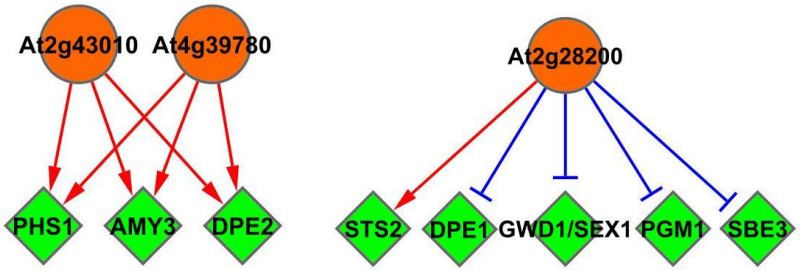
ในขั้นการทดลองนี้ จะนำเอา Constraint-based Boolean network ที่พัฒนาขึ้นมาใช้สร้างเครือข่ายควบคุมการทำงานของยีนในกระบวนการสังเคราะห์แป้ง เพื่อทำความเข้าใจเกี่ยวกับกลไกการทำงานของกระบวนการสังเคราะห์แป้งมากขึ้น โดยใช้ข้อมูลการแสดงออกของยีนในพืชต้นแบบ นั่นคือ *Arabidopsis thaliana* ภายใต้สภาวะการเลี้ยงแบบ diurnal cycle นั่นคือ ภายใต้สภาวะการเลี้ยงแบบ

กลางวัน-กลางคืน (12-12 ชั่วโมง) หลังจากผ่านขั้นตอนการ pre-process [17, 18] ด้วยภาษา R [http://www.r-project.org/] ใน Bioconductor [http://www.bioconductor.org] ในการจัดการข้อมูล โดยเตรียมข้อมูลการแสดงออกของยีนให้อยู่ในรูปแบบ log_2intensity และหายีนที่มีการแสดงออกอย่างมีนัยสำคัญด้วยโปรแกรม EDGE [19] เวอร์ชั่น 1.1.175 [http://faculty.washington.edu/jstorey/edge/] หลังจากนั้น จึงเลือกกลุ่มยีนทำหน้าที่เป็นเอนไซม์ที่เกี่ยวข้องกับการสังเคราะห์แป้ง (Starch genes) โดยอ้างอิง [14] และโปรตีนควบคุม (Transcription factors; TF) และยีนที่เกี่ยวข้องการเวลา (clock genes) มาสร้างเครือข่ายควบคุมการแสดงออกของยีนในกระบวนการสังเคราะห์แป้ง พบว่า มี Starch genes 21 ยีน *[ภาคผนวก ค1]* และ TF และ clock genes รวม 113 ยีน *[ภาคผนวก ค2]* ดังนั้น กลุ่มยีนที่จะนำไปสร้างเครือข่ายควบคุมการแสดงออกของยีนด้วย Constraint-based Boolean network มีทั้งหมด 134 ยีน โดยใช้ข้อมูล binary values ที่ได้จากวิธีการ Sign of Log_2Ratio (ค่า similarity score สูงสุด) แสดงในรูปที่ 3.14

เครือข่ายควบคุมการแสดงออกของยีนของกระบวนการสังเคราะห์แป้งในพืช Arabidopsis ที่สร้างขึ้น ประกอบด้วย 112 nodes และ 234 directed edges โดย nodes แสดงถึงยีน และ directed edges คือ รูปแบบความสัมพันธ์ระหว่างยีนที่ได้จาโปรแกรม Constraint-based Boolean network ซึ่งเป็นได้ทั้งแบบกระตุ้นหรือยับยั้ง โดยความสัมพันธ์แบบกระตุ้นแสดงเป็นเส้นสีแดง หรือยับยั้งแสดงเป็นเส้นสีน้ำเงิน โดยเครือข่ายที่สร้างขึ้นประกอบด้วย starch genes จำนวน 16 ยีน (สี่เหลี่ยมขนมเปียกปูนสีเขียว) transcription factors (TF) จำนวน 91 ยีน (วงกลมสีส้ม) และ clock genes จำนวน 5 ยีน (วงกลมสีเหลือง) เมื่อพิจารณาเครือข่ายย่อย (sub-network) ระหว่าง starch genes และ TF พบว่า มี 2 เครือข่ายย่อย ดังรูปที่ 3.13 โดยเครือข่ายแรกเป็นความสัมพันธ์ของเอนไซม์ที่เกี่ยวข้องกับการสลายแป้ง (starch degradation) 3 เอนไซม์ คือ A cytosolic disproportionating enzyme like protein (*At2g40840*; *DPE2*) Alpha-amyla like 3 (*At1g69830*; *AMY3*) และ A plastidic alpha-glucan phosphorylase (*At3g29320*; *PHS1*) โดยทั้ง 3 เอนไซม์นี้เกี่ยวข้องกับ 2 โปรตีนควบคุม (TF) คือ *At4g39780* และ *At2g43010* ซึ่งเป็นโปรตีนควบคุมที่มี AP2 domain

ส่วน sub-network ที่สองเป็นความสัมพันธ์ของเอนไซม์ 5 เอนไซม์ คือ phosphoglucomutase (*At5g51820*; *PGM1*) Starch synthase II (*At3g01180*; *STS2*) Starch branching enzyme III (*At2g36390*; *SBE3*) Glucan water dikinase 1 (*At1g10760*; *GWD1/SEX1*) และ Glucanotransferase (*At5g64860*; *DPE1*) โดยทั้ง 5 เอนไซม์นี้เกี่ยวข้องกับโปรตีนควบคุม 1 ตัว คือ At2g28200 which has zinc-finger domain โดย *PGM1 STS2* และเอนไซม์ *SBE3* เป็นเอนไซม์ที่เกี่ยวข้องกับการสร้างแป้ง (Starch biosynthesis) โดยเฉพาะ amylopectin *PGM1* เป็นเอนไซม์ที่เปลี่ยน alpha-D-glucose-6-phosphate ไปเป็น alpha-D-glucose-1-phosphate เพื่อเปลี่ยนเป็น ADP-glucose ต่อไปด้วยเอนไซม์ ADP-glucose pyrophosphorylase (*AGPase*) หลังจากนั้น Starch synthase II *(STSII)* จะต่อเชื่อม ADP-glucose เป็นสายยาวด้วยการเชื่อมพันธะชนิดอัลฟา-1-4 (1->4)-alpha-D-glucosyl) ส่วนเอนไซม์ *SBE3* จะเป็นเอนไซม์ที่สร้างกิ่งด้วยพันธะ ชนิดอัลฟา-1-6 (1->6)-alpha-D-glucosyl) ในขณะที่ *GWD1/SEX1* และ *DPE1* เป็นเอนไซม์ที่สำคัญนกระบวนการสลายแป้งเพื่อเปลี่ยนเป็น maltose แล้วลำเลียงออกนอก cytosol อย่างไรก็ตาม แม้ว่าจะสามารถสร้างเครือข่ายควบคุมการแสดงออกของ starch genes และ transcription factors ได้ แต่ข้อมูลดังกล่าวยังต้องพิสูจน์ในห้องปฏิบัติการว่าเอนไซม์เหล่านี้เกี่ยวข้อง หรือถูกควบคุมการแสดงออกด้วย transcription factors เหล่านี้หรือไม่





รูปที่ 3.14 (ก) เครือข่ายควบคุมการแสดงออกของยีนในกระบวนการสังเคราะห์แป้งซึ่งประกอบด้วยยีน starch genes จำนวน 16 ยีน (สี่เหลี่ยมขนมเปียกปูนสีเขียว) transcription factors (TF) จำนวน 91 ยีน (วงกลมสีส้ม) และ clock genes จำนวน 5 ยีน (วงกลมสีเหลือง) เส้นสีน้ำเงิน คือ ความสัมพันธ์แบบยับยั้ง (inhibition) เส้นสีแดง คือ ความสัมพันธ์แบบกระตุ้น วงกลมเส้นประนั้น คือ sub-network ของ starch-TF network (ข) Sub-network ของ starch-TF gene network

บทที่ 4
สรุปและเสนอแนะ

4.1 สรุปผลการดำเนินงาน

กระบวนการต่างๆที่เกิดขึ้นในสิ่งมีชีวิตนั้น มีรูปแบบการควบคุมการทำงานหรือการแสดงออกของยีนเป็นเครือข่าย หรือเรียกว่า เครือข่ายควบคุมการแสดงออกของยีน (genetic network) ซึ่งมีการพัฒนาวิธีการต่างๆให้สามารถนำมาวิเคราะห์ข้อมูลไมโครอะเรย์ ซึ่งงานวิจัยนี้ได้พยายามพัฒนาโปรแกรมที่ช่วยสร้างเครือข่ายควบคุมการแสดงออกของยีนด้วยเทคนิคบูลีนจากข้อมูลไมโครอะเรย์ โดยการเพิ่มเงื่อนไขทางชีววิทยา เรียกว่า Constraint-based Boolean network พร้อมกับการพิจารณาความสำคัญของเทคนิคหรือวิธีการที่ใช้ในการแปลงข้อมูลการแสดงออกของยีนให้อยู่ในสองระดับ คือ 0 หรือ 1 ซึ่งเมื่อเปรียบเทียบ Constraint-based Boolean network และ Non-Constraint-based Boolean network พบว่า การเพิ่มเงื่อนไขทางชีววิทยานั้น สามารถลดความซับซ้อนของเครือข่ายควบคุมการแสดงออกของยีนที่สร้างขึ้นได้ โดยการลดความผิดพลาดที่เกิดจาก false positive prediction ลง ทำให้มีความถูกต้องและแม่นยำมากขึ้น โดยสามารถเพิ่มความถูกต้องขึ้น 25%

นอกจากนี้ Constraint-based Boolean network ที่พัฒนาขึ้นนั้นยังสามารถนำไปประยุกต์ใช้สร้างเครือข่ายควบคุมการแสดงออกของยีนในกระบวนการสร้างแป้งในพืช *A. thaliana* ได้ โดยสามารถทำนายโปรตีนควบคุม (transcription factors) ที่เกี่ยวข้องกับเอนไซม์ในกระบวนการสร้างแป้งทั้งกระบวนการสังเคราะห์แป้ง (starch biosynthesis) และกระบวนการสลายแป้ง (starch degradation) พบว่า โปรตีนควบคุม 2 ตัว คือ *At4g39780* and *At2g43010* เกี่ยวข้องกับ เอนไซม์ในการะบวนการสลายแป้ง นั่นคือ A cytosolic disproportionating enzyme like protein (*At2g40840*; *DPE2*) Alpha-amylase like 3 (*At1g69830*; *AMY3*) และ A plastidic alpha-glucan phosphorylase (*At3g29320*; *PHS1*) และมีโปรตีนควบคุม 1 ตัว คือ *At2g28200* ที่เกี่ยวข้องกับเอนไซม์ 5 ตัว คือ คือ phosphoglucomutase (*At5g51820*; *PGM1*) Starch synthase II (*At3g01180*; *STS2*) Starch branching enzyme III (*At2g36390*; *SBE3*) Glucan water dikinase 1 (*At1g10760*; *GWD1/SEX1*) และ Glucanotransferase (*At5g64860*; *DPE1*) แต่อย่างไรก็ตาม ข้อมูลเหล่านี้ยังต้องมีการพิสูจน์ทางห้องปฏิบัติการ เพื่อยืนยันว่าโปรตีนควบคุมเหล่านี้มีความเกี่ยวข้องหรือควบคุมเอนไซม์ดังกล่าว

4.2 ข้อเสนอแนะ

การสร้างเครือข่ายควบคุมการแสดงออกของยีนด้วย Constraint-based Boolean network ยังไม่เป็นระบบอัตโนมัติ ซึ่งอาจมีความยุ่งยากซับซ้อนในสร้างเครือข่าย ควรมีการปรับปรุงโปรแกรมให้สามารถสร้างเครือข่ายควบคุมการแสดงออกของยีนแบบอัตโนมัติได้ นอกจากนี้ ควรระวังในการเพิ่มเงื่อนทางชีววิทยา การใส่เงื่อนไขที่ว่าเอนไซม์ไม่ควรควบคุมการแสดงออกของโปรตีนควบคุมนั้น อาจไม่จริงเสมอไป เพราะในความเป็นจริงเอนไซม์บางตัวก็สามารถมีการควบคุมการแสดงออกของโปรตีนควบคุมได้ โดยเฉพาะอย่างยิ่งในระดับหลังการแปลรหัสโปรตีน (post-translation)

เอกสารอ้างอิง
(References)

1. Martin S, Zhang Z, Martino A, Faulon J-L: **Boolean Dynamics of Genetic Regulatory Networks Inferred from Microarray Time Series Data**. *Bioinformatics* 2007, **23**:866-874.
2. Cseke LJ, Kirakosyan A, Kaufman PB, Warber SL, Duke JA, Brielmann HL: **Natural Products from Plants**. *Molecular regulation* 2006:587.
3. Theuns, Jessie, VanBroeckhoven, Christine: **Transcriptional regulation of Alzheimer's disease genes: implications for susceptibility** *Human Molecular Genetics* 2000, **9**(16):2383-2394.
4. Villard J: **Transcription regulation and human diseases**. *SWISS MED WKLY* 2004, **134**:571-579.
5. Long TA, Brady SM, Benfey PN: **Systems Approaches to Identifying Gene Regulatory Networks in Plants**. *Annual Review of Cell and Developmental Biology* 2008, **24**:81-103
6. Shen-Orr SS, Milo R, Mangan S, Alon U: **Network motifs in the transcriptional regulation network of *Escherichia coli***. *nature genetics* 2002, **31**:64-68.
7. Babu MM: **Computational approaches to study transcriptional regulation**. *Biochemical Society Transactions* 2008, **36**:758-765.
8. Barrett CL, Palsson BO: **Iterative Reconstruction of Transcriptional Regulatory Networks: An Algorithmic Approach**. *PLoS Computational Biology* 2006, **2**(5):429-438.
9. Kwon AT, Hoos HH, Ng R, : **Inference of transcriptional regulation relationships from gene expression data**. In: *Proceedings of the 2003 ACM symposium on Applied computing 2003; Melbourne, Florida* ACM New York, NY, USA 135 - 140
10. Perrin B, Ralaivola L, Mazurie A, Bottani S, Mallet J, d'Alché-Buc F: **Gene networks inference using dynamic Bayesian networks**. *Bioinformatics* 2003, **9**(2):138-148.
11. Kauffman SA: **The Origins of Order: Self-Organization and Selection in Evolution**. Oxford University Press, New York; 1993.
12. Mehra S, Hu W-S, Karypis G: **A Boolean algorithm for reconstructing the structure of regulatory networks**. *Metabolic Engineering* 2004, **6**:326-339.
13. Pensa R, Leschi C, Besson J, Boulicaut J-F: **Assessment of discretization techniques for relevant pattern discovery from gene expression data**. In: *In 4th ACM SIGKDD Workshop on Data Mining in Bioinformatics.* In Press.; 2004: 24-30.

14. Smith SM, Fulton DC, Chia T, Thorneycroft D, Chapple A, Dunstan H, Hylton C, Zeeman SC, Smith AM: **Diurnal Changes in the Transcriptome Encoding Enzymes of Starch Metabolism Provide Evidence for Both Transcriptional and Posttranscriptional Regulation of Starch Metabolism in Arabidopsis Leaves**. *Plant Physiology* 2004, **136**:2687-2699.
15. DeRisi JL, Iyer VR, Brown PO: **Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale**. *Science* 1997, **278**:680-686.
16. Ideker T, Thorsson V, Ranish JA, Christmas R, Buhler J, Eng JK, Bumgarner R, Goodlett DR, Aebersold R, Hood L: **Integrated genomic and proteomic analyses of a systematically perturbed metabolic network**. *Science* 2001, **292**(5518):929-934.
17. Li C, Wong WH: **Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection**. *Proceedings of the National Academy of Sciences of the United States of America* 2001, **98**(1):31-36.
18. Workman C, Jensen LJ, Jarmer H, Berka R, Gautier L, Nielser HB, Saxild HH, Nielsen C, Brunak S, Knudsen S: **A new non-linear normalization method for reducing variability in DNA microarray experiments**. *Genome biology* 2002, **3**(9):research0048.
19. Storey JD, Xiao W, Leek JT, Tompkins RG, Davis RW: **Significance analysis of time course microarray experiments**. *Proceedings of the National Academy of Sciences of the United States of America* 2005, **102**(36):12837-12842.
20. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T: **Cytoscape: a software environment for integrated models of biomolecular interaction networks**. *Genome Res* 2003, **13**(11):2498-2504.
21. Hache H, Lehrach H, Herwig R: **Reverse engineering of gene regulatory networks: a comparative study**. *EURASIP journal on bioinformatics & systems biology* 2009:617281.

ผลงานตีพิมพ์

Bumee S., Liamwirat C., Saithong T., and Meechai A., (2010) Extended Constraint-Based Boolean Analysis: A Computational Method in Genetic Network Inference, Communications in Computer and Information Science, 1, Vol. 115, Computational Systems-Biology and Bioinformatics (CSBio2010), p. 71-82. (ภาคผนวก ง)

ลงชื่อ_____

วันที่_____

ภาคผนวก

ภาคผนวก ก

โค้ดสคริป MATLAB สำหรับคำนวณหา similarity score ในการเปรียบเทียบข้อมูลดิบของการแสดงออกของยีน (raw data) กับข้อมูลที่แปลงค่าเป็น 0 หรือ 1 แล้ว (Discretized data) ซึ่งประกอบด้วย 2 ฟังก์ชั่น

```
function [leaf_set, intern_set] = find_leaf_set(data,dist,method)
% Usage: [leaf_set, intern_set] = find_leaf_set(data,'correlation','average')

Y = pdist(data,dist);
Z = linkage(Y,method);
leaf_amt = size(data,1);
leaf_set = cell(leaf_amt-1,1);
intern_set = cell(leaf_amt-1,1);
for i=1:leaf_amt-1
    if (Z(i,1) > leaf_amt)
        leaf_set[2] = union(leaf_set[2],leaf_set{Z(i,1)-leaf_amt});
        intern_set{i} = union(intern_set{i},Z(i,1)-leaf_amt);
        intern_set{i} = union(intern_set{i},intern_set{Z(i,1)-leaf_amt});
    else
        leaf_set{i} = union(leaf_set{i},Z(i,1));
    end
    if (Z(i,2) > leaf_amt)
        leaf_set{i} = union(leaf_set{i},leaf_set{Z(i,2)-leaf_amt});
        intern_set{i} = union(intern_set{i},Z(i,2)-leaf_amt);
        intern_set{i} = union(intern_set{i},intern_set{Z(i,2)-leaf_amt});
    else
        leaf_set{i} = union(leaf_set{i},Z(i,2));
    end
end
```

```
function [SB, SB_self] = cal_BScore(leaf_set_ref,leaf_set,intern_set)
% Usage: [SB, SB_self] = cal_BScore(leaf_set_ref,leaf_set,intern_set)

SB = zeros(length(leaf_set),1);
SB_self = zeros(length(leaf_set),1);
```

```
    for i=1:length(leaf_set)
        for j=1:length(leaf_set_ref);
            disp(['[' num2str(i) ',' num2str(j) ']']);
            if (isempty(setdiff(leaf_set_ref{j},leaf_set{i})) &&
isempty(setdiff(leaf_set{i},leaf_set_ref{j})))
                disp('yes');
                SB_self(i) = 1/length(leaf_set{i});
            end

            if isempty(intern_set{i}) == 0
                bset = intern_set{i};
                SB(i) = sum(SB_self(bset));
            end
        end
                                                    end
```

ภาคผนวก ข

โค้ดสคริป C++ สำหรับสร้าง gene regulatory network ของกลุ่มยีนที่สนใจด้วยวิธี Constraint-based Boolean network

```
//-------------------------------------------------------------

#include <algorithm>
#include <iostream>
#include <iomanip>
#include <fstream>
#include <string>
#include <vector>
#include <map>

#include "../flx/algutl.hpp"
#include "../flx/iosutl.hpp"
#include "../flx/strutl.hpp"

//-------------------------------------------------------------

#define IS_TF( i, tfbeg, tfend ) ( i >= tfbeg && i <= tfend )
#define LINE_PREDICT_NETWORK 5
#define LINE_INFER_CASE      1

//-------------------------------------------------------------

enum regulation
{
  type_none =  0,
  type_gene =  1,
  type_join =  2,
  type_up   =  4,
  type_down =  8,
  type_actv = 16,
  type_inhb = 32,
  type_nspc = 64,
};
```

```
//-------------------------------------------------------------

struct expression
{
  std::vector< int > expr; // gene expression
  std::string        name; // gene name
};

typedef std::vector< expression >      profile;
typedef std::vector< int >             pattern;
typedef std::map< pattern, int >       inferer;
typedef std::pair< inferer, pattern > predict; // [ inferer, conclusion ]
typedef std::vector< predict >         network;

//-------------------------------------------------------------

char regulate_to_char( int val )
{
  return ( val == type_none ? '-' :
         ( val == type_gene ? 'x' :
         ( val == type_join ? 'v' :
         ( val == type_up   ? '1' :
         ( val == type_down ? '0' :
         ( val == type_actv ? 'a' :
         ( val == type_inhb ? 'i' : '-' ) ) ) ) ) ) );
}

//-------------------------------------------------------------

bool load_profile( std::istream & istr,
                   profile & prof )
{
  if( false == istr.good( ) )
    return false;

  // Parse header
  while( false == istr.eof( ) )
```

```
    {
      std::string line;
      if( false == std::getline( istr, line ) )
        continue;

      std::string trim = flx::str_trim( line );
      if( "" == trim )
        continue;

      std::vector< std::string > names = flx::str_split( trim, "," );
        prof.resize( names.size( ) );

      for( size_t i = 0 ; i < names.size( ) ; ++i )
        prof[ i ].name = names[ i ];

      break;
    }

    // Parse expression
    while( false == istr.eof( ) )
    {
      std::string line;
      if( false == std::getline( istr, line ) )
        continue;

      std::string trim = flx::str_trim( line );
      if( "" == trim )
        continue;

      // Split each value in current line
      std::vector< std::string > temp = flx::str_split( trim, "," );

      // Fill each gene expression value
      for( size_t i = 0 ; i < temp.size( ) ; ++i )
        prof[ i ].expr.push_back( flx::val_cast< int >( temp[ i ] ) > 0 ? type_up : type_down
);
    }
```

```
  return true;
}

//------------------------------------------------------------

bool save_profile( std::ostream & ostr,
                   const profile & prof )
{
  if( false == ostr.good( ) )
    return false;

  size_t size_prof = prof.size( );
  if( 0 == size_prof )
    return false;

  size_t size_expr = prof[ 0 ].expr.size( );
  if( 0 == size_expr )
    return false;

  // Print header name
  for( size_t i = 0 ; i < size_prof ; ++i )
  {
    ostr << prof[ i ].name;
    for( size_t j = 0 ; j < size_expr ; ++j )
      ostr << ", " << regulate_to_char( prof[ i ].expr[ j ] );
    ostr << std::endl;
  }

  return true;
}

void save_pattern( const pattern & p,
                   std::ostream  & ostr )
{
  pattern::const_iterator
    beg = p.begin( ),
    end = p.end( );
```

```
  for( beg; beg != end; ++beg )
    ostr << regulate_to_char( *beg );
}

void save_inferer( const inferer & f,
                   std::ostream  & ostr )
{
  inferer::const_iterator
    beg = f.begin( ),
    end = f.end( );

  for( beg; beg != end; ++beg )
  {
    save_pattern( beg->first, ostr );

    ostr << " == "
         << regulate_to_char( beg->second )
         << std::endl;
  }
}

void save_network( const network & n,
                   std::ostream  & ostr )
{
  for( size_t i = 0; i < n.size( ); ++i )
  {
    ostr << "Network: " << ( i + 1 ) << std::endl;
      save_inferer( n[ i ].first, ostr );
      save_pattern( n[ i ].second, ostr );
    ostr << " == ?" << std::endl << std::endl;
  }
}

void save_predict( const network & n,
                   std::ostream  & ostr )
{
  ostr << "Result:" << std::endl << std::endl << "    ";
  for( size_t i = 0; i < n.size( ); ++i )
```

```
    ostr << "v" << std::left << std::setw( 2 ) << ( i + 1 ) << " ";
  ostr << std::endl;

  for( size_t i = 0; i < n.size( ); ++i )
  {
    pattern::const_iterator
      beg = n[ i ].second.begin( ),
      end = n[ i ].second.end( );

    ostr << "g" << std::left << std::setw( 2 ) << ( i + 1 ) << " ";
    for( beg; beg != end; ++beg )
      ostr << std::left << std::setw( 4 ) << regulate_to_char( *beg );
    ostr << std::endl;
  }
}

//-------------------------------------------------------------

bool infer_function( const profile & p,
                     const pattern & v,
                          inferer & f )
{
  f.clear( );

  size_t size_prof = p.size( );
  if( 0 == size_prof )
    return false;

  size_t size_expr = p[ 0 ].expr.size( );
  if( 0 == size_expr )
    return false;

  pattern t; t.resize( size_prof, (int)type_none );
  for( size_t i = 0 ; i < size_expr - 1 ; ++i )
  {
     int pval = type_none;

     for( size_t j = 0 ; j < size_prof ; ++j )
```

```
    {
      if( type_gene == v[ j ] )
      {
        pval = p[ j ].expr[ i + 1 ];
        t[ j ] = type_gene;
      }
      else
      if( type_join == v[ j ] )
      {
        t[ j ] = p[ j ].expr[ i ];
      }
      else
      {
        t[ j ] = type_none;
      }
    }

    if( f[ t ] == type_none )
    {
      f[ t ] = pval;
    }
    else
    if( f[ t ] != pval )
    {
      f.clear( );
      return false;
    }
  }

  return true;
}

//-------------------------------------------------------------

void infer_network_helper( const profile & p,
                           const pattern & v,
                           const inferer & f,
                                 pattern & conc )
```

```
{
  inferer::const_iterator
    beg = f.begin( ),
    end = f.end( );

  size_t size_prof = p.size( );
  for( beg; beg != end; ++beg )
  {
    const pattern & pat = beg->first;
    int             res = beg->second;

    for( size_t i = 0; i < size_prof; ++i )
    {
      if( v[ i ] == type_gene )
        conc[ i ] = type_gene;
      else
      if( pat[ i ] == type_up )
      {
        if( conc[ i ] == type_none )
          conc[ i ] = res == type_up ? type_actv : type_inhb;
        else
        if( conc[ i ] == type_actv )
          conc[ i ] = res == type_up ? type_actv : conc[ i ];
        else
        if( conc[ i ] == type_inhb )
          conc[ i ] = res == type_up ? type_actv : conc[ i ];
      }
      else
      if( pat[ i ] == type_down )
      {
        if( conc[ i ] == type_none )
          conc[ i ] = res == type_up ? type_inhb : conc[ i ];
        else
        if( conc[ i ] == type_actv )
          conc[ i ] = res == type_up ? type_nspc : conc[ i ];
      }
    }
  }
```

```
}

//-------------------------------------------------------------

void infer_network_with_starch( profile & p,
                                network & n,
                                size_t    k,
                                size_t    tf )
{
  size_t size_prof = p.size( );

  k = k > tf ? tf : k;
  for( size_t r = tf; r < size_prof; ++r )
  {
    for( flx::combinator comb( tf, k );
         comb.valid( ); comb.next( ) )
    {
      pattern v;
        v.resize( size_prof, (int)type_none );

      v[ r ] = type_gene;
      for( size_t i = 0 ; i < k ; ++i )
        v[ comb[ i ] ] = type_join;

      inferer f;
      if( true == infer_function( p, v, f ) )
      {
          pattern & conc = n[ r ].second;
          inferer & memo = n[ r ].first;

          infer_network_helper( p, v, f, conc );
          memo.insert( f.begin( ), f.end( ) );
      }
    }
  }
}

//-------------------------------------------------------------
```

```
void infer_network_no_starch( profile & p,
                              network & n,
                              size_t    k,
                              size_t    tf )
{
  size_t size_prof = p.size( );

  k = k + 1 > tf ? tf : k + 1;
  for( flx::combinator comb( tf, k );
       comb.valid( ); comb.next( ) )
  {
    for( size_t r = 0 ; r < k ; ++r )
    {
      pattern v;
        v.resize( size_prof, (int)type_none );

      v[ comb[ r ] ] = type_gene;
      for( size_t i = 1 ; i < k ; ++i )
        v[ comb[ ( i + r ) % k ] ] = type_join;

      inferer f;
      if( true == infer_function( p, v, f ) )
      {
        pattern & conc = n[ comb[ r ] ].second;
        inferer & memo = n[ comb[ r ] ].first;

        infer_network_helper( p, v, f, conc );
        memo.insert( f.begin( ), f.end( ) );
      }
    }
  }
}

//-------------------------------------------------------------

bool infer_network( profile & p,
                    network & n,
```

```
                    size_t    k,
                    size_t    tf )
{
  n.clear( );

  size_t size_expr = p[ 0 ].expr.size( );
  if( 0 == size_expr )
    return false;

  size_t size_prof = p.size( );
  if( 0 == size_prof )
    return false;

  n.resize( size_prof );
  for( size_t i = 0; i < size_prof; ++i )
    n[ i ].second.resize( size_prof, (int)type_none );

  if( tf == 0 )
  {
    // no tf, do all combinations
    infer_network_no_starch( p, n, k, size_prof );
  }
  else
  {
    infer_network_with_starch( p, n, k, tf );
    infer_network_no_starch( p, n, k, tf );
  }

  return true;
}

//------------------------------------------------------------
int proc( int argc, char ** argv )
{
  if( argc != 5 )
  {
    std::cerr << argv[ 0 ]
              << " <k-value> <tf-value> <input> <output>"
```

```
            << std::endl;

  return -1;
}

size_t k = -1;
if( false == flx::val_cast< size_t >( argv[ 1 ], k ) )
{
  std::cerr << "Error: Invalid k-value ("
            << argv[ 1 ]
            << ") !!!"
            << std::endl;

  return -1;
}

size_t tf = -1;
if( false == flx::val_cast< size_t >( argv[ 2 ], tf ) )
{
  std::cerr << "Error: Invalid tf-value ("
            << argv[ 2 ]
            << ") !!!"
            << std::endl;

  return -1;
}

std::ifstream input( argv[ 3 ] );
if( false == input.is_open( ) )
{
  std::cerr << "Error: Open file "
            << argv[ 3 ]
            << " failed !!!"
            << std::endl;

  return -1;
}
```

```
  std::ofstream output( argv[ 4 ] );
  if( false == output.is_open( ) )
  {
    std::cerr << "Error: Create file "
              << argv[ 4 ]
              << " failed !!!"
              << std::endl;

    return -1;
  }

  profile p;
  if( false == load_profile( input, p ) )
  {
    std::cout << "Error: Load profile failed !!!"
              << std::endl;

    return -1;
  }

  network n;
  if( false == infer_network( p, n, k, tf ) )
  {
    std::cout << "Error: Infer network failed !!!"
              << std::endl;

    return -1;
  }
  save_network( n, output );
  save_predict( n, output );
  return 0;
}

int main( int argc, char ** argv )
{
  return proc( argc, argv );
}
//------------------------------------------------------------
```

ภาคผนวก ค

ยีนที่ใช้ในการสร้างเครือข่ายควบคุมการแสดงออกของยีน (gene regulatory network) ของกระบวนการสังเคราะห์แป้ง ซึ่งแยกเป็นกลุ่มยีนที่ทำหน้าที่เป็นเอนไซม์ที่เกี่ยวข้องกับการสร้างแป้ง (starch genes) และกลุ่มยีนที่ทำหน้าที่เป็นโปรตีนควบคุม (transcription factors) และยีนที่เกี่ยวข้องกับเวลา (clock genes)

ค1: Starch genes

No.	ID	Gene Name	Enzyme
1	At5g51820	PGM1	Phosphoglucomutase
2	At5g24300	STS1	Starch synthase I
3	At3g01180	STS2	Starch synthase II
4	At4g18240	STS4	Starch synthase IV
5	At1g32900	GBS1	Granule-bound starch synthase
6	At2g36390	SBE3	Starch branching enzyme III
7	At2g39930	ISA1	Starch debranching enzyme: Isoamylase I
8	At1g03310	ISA2/DBE1	Starch debranching enzyme: Isoamylase II
9	At4g09020	ISA3	Starch debranching enzyme: Isoamylase III
10	At1g10760	GWD1/SEX1	Glucan water dikinase 1
11	At5g26570	GWD3	Glucan water dikinase-like 3
12	At5g64860	DPE1	Glucanotransferase
13	At2g40840	DPE2	Trasglucosidase
14	At3g29320	PHS1	Glucan phosphorylase (plastidial)
15	At3g46970	PHS2	Glucan phosphorylase (cytosolic)
16	At1g76130	AMY2	α-Amylase2
17	At1g69830	AMY3	α-Amylase3
18	At4g17090	BAM3/BMY8	β-Amylase3
19	At2g32290	BAM6	β-Amylase6
20	At5g18670	BAM9/BMY3	β-Amylase9
21	At5g11720	AGL4	α-Glucosidase-like 4

ค2: Transcription factors and clock genes

No.	Gene ID	family name
1	At1g01060	MYB-related
2	At1g01520	MYB-related
3	At1g02340	bHLH
4	At1g04250	AUX-LAA
5	At1g04990	C3H
6	At1g05690	TAZ
7	At1g05805	bHLH
8	At1g07050	C2C2-CO-like
9	At1g07520	GRAS
10	At1g10200	LIM
11	At1g14510	Alfin
12	At1g17460	MYB-related
13	At1g18570	MYB
14	At1g19700	HB
15	At1g20693	HMG
16	At1g20696	HMG
17	At1g22070	bZIP
18	At1g22190	AP2-EREBP
19	At1g25580	NAC
20	At1g26790	C2C2-Dof
21	At1g28050	C2C2-CO-like
22	At1g29160	C2C2-Dof
23	At1g33240	Trihelix
24	At1g35560	TCP
25	At1g47270	TLP
26	At1g49560	GARP-G2-like
27	At1g50420	GRAS
28	At1g51700	C2C2-Dof
29	At1g51950	AUX-IAA

No.	Gene ID	family name
30	At1g53320	TLP
31	At1g56170	CCAAT-HAP5
32	At1g58110	bZIP
33	At1g69570	C2C2-Dof
34	At1g70000	MYB-related
35	At1g73870	C2C2-CO-like
36	At1g76590	PLATZ
37	At1g77850	ARF
38	At2g02070	C2H2
39	At2g18280	TLP
40	At2g20570	GARP-G2-like
41	At2g22430	HB
42	At2g28200	C2H2
43	At2g28550	AP2-EREBP
44	At2g31070	TCP
45	At2g34720	CCAAT-HAP2
46	At2g35940	HB
47	At2g39900	LIM
48	At2g40140	C3H
49	At2g42400	VOZ
50	At2g43010	BHLH
51	At2g46830	MYB-related
52	At2g47680	C3H
53	At2g47890	C2C2-CO-like
54	At3g02380	C2C2-CO-like
55	At3g02830	C3H
56	At3g06160	ABI3-VP1
57	At3g07650	C2C2-CO-like
58	At3g09600	MYB-related
59	At3g10030	Trihelix
60	At3g17609	bZIP
61	At3g21175	ZIM

No.	Gene ID	family name
62	At3g28910	MYB
63	At3g47500	C2C2-Dof
64	At3g50700	C2H2
65	At3g55770	LIM
66	At3g58680	MBF1
67	At3g59060	BHLH
68	At3g61150	HB
69	At3g61890	HB
70	At4g00050	BHLH
71	At4g16750	AP2-EREBP
72	At4g16780	HB
73	At4g17490	AP2-EREBP
74	At4g18390	TCP
75	At4g25480	AP2-EREBP
76	At4g28610	GARP-G2-like
77	At4g31420	C2H2
78	At4g32280	AUX-IAA
79	At4g34610	HB
80	At4g36730	bZIP
81	At4g39410	WRKY
82	At4g39780	AP2-EREBP
83	At5g02810	C2C2-CO-like
84	At5g02840	MYB-related
85	At5g05090	GARP-G2-like
86	At5g06770	C3H
87	At5g06960	bZIP
88	At5g07690	MYB
89	At5g08520	MYB
90	At5g12840	CCAAT-HAP2
91	At5g15850	C2C2-CO-like
92	At5g17300	MYB-related
93	At5g18680	TLP

No.	Gene ID	family name
94	At5g24470	C2C2-CO-like
95	At5g37020	ARF
96	At5g37260	MYB-related
97	At5g38140	CCAAT-HAP5
98	At5g39760	ZF-HD
99	At5g44190	GARP-G2-like
100	At5g46710	PLATZ
101	At5g48250	C2C2-CO-like
102	At5g51980	C3H
103	At5g56860	C2C2-GATA
104	At5g57660	C2C2-CO-like
105	At5g60100	C2C2-CO-like
106	At5g60850	C2C2-Dof
107	At5g61380	C2C2-CO-like
108	At5g62430	C2C2-Dof
109	At5g63420	Trihelix
110	At5g65310	HB
111	At5g65670	AUX-IAA
112	At5g67030	FHA
113	At5g67480	TAZ

ภาคผนวก ง

Extended Constraint-Based Boolean Analysis: A Computational Method in Genetic Network Inference

Somkid Bumee[1], Chalothorn Liamwirat[2], Treenut Saithong[1,3], and Asawin Meechai[1,4]

[1] Systems Biology and Bioinformatics Research Laboratory, Pilot Plant Development and Training Institute
[2] Division of Biotechnology, School of Bioresources and Technology
[3] Bioinformatics and Systems Biology Program
[4] Department of Chemical Engineering, Faculty of Engineering,
King Mongkut's University of Technology Thonburi, Bangkok, Thailand
somkid@pdti.kmutt.ac.th, chalothorn09@yahoo.com,
treenut.sai@kmutt.ac.th, asawin.mee@kmutt.ac.th

Abstract. Reconstruction of a genetic network, which describes gene regulation of cellular response processes, has been widely studied by using various approaches. Some of which are computational expensive and require enormous efforts. Herein, we proposed an *extended constraint-based Boolean* to infer genetic network. Our method incorporated the specific constraints for a particular system in addition to the general conceptual constraints of a typical genetic circuit, to improve the performance of the existing constraint-based Boolean algorithm. This method was demonstrated in inference of the genetic network underlying circadian rhythms from microarray time series data. The results showed that the proposed method provides good accuracy, specificity, and precision under the trade-off of computational efforts. Moreover, the resulting network showed that prior knowledge is a useful bias for modeling genetic network. The proposed method is therefore a promising alternative approach for inferring genetic network from high-throughput data, such as microarray.

Keywords: Genetic network, extended constraint-based Boolean, conceptual constraints, specific constraints.

1 Introduction

Relationship between gene in a genetic network is important information in understanding the cellular response processes, which involve the regulation of gene expression [1]. The regulation of gene expression lies on a huge number of components that comprise a genetic network, multiple levels of regulation as well as the elaborated interaction between levels [2]. Though the number of network constituents is a barrier of network reconstruction, the (differential) expression of such components is often employed in network inference. This strategy becomes more and more popular once the measurement of thousands of gene components (or whole genome) can simultaneously be performed with the aid of microarray techniques.

J.H. Chan, Y.-S. Ong, and S.-B. Cho (Eds.): CSBio 2010, CCIS 115, pp. 71–82, 2010.

In the last decade, availability of high-throughput technologies, allowing the levels of transcripts to be measured for the whole genome at the same time, enables scientists to understand cellular system by reconstructing genetic network [3, 4]. Various computational approaches have been developed on the purpose of genetic network reconstruction, such as Boolean network [5-7], graphical Gaussian model [8], and Bayesian network [7, 9]. Among these approaches Boolean network and Bayesian network methods are mostly used in the context of reconstructing genetic network from microarray data [10]. These two approaches have distinct advantages and disadvantages. Bayesian network provides a more accurate result yet with a huge requirement of prior data and computational efforts in an iterative learning algorithm, while Boolean network is the simpler method to reconstruct genetic network. Under the trade-off of computational effort, Boolean network is considerably a competitive method to Bayesian network.

Boolean network was originally introduced by Kauffman [5,11]. Later, Shmulevich and Zhang used Boolean network to infer genetic network of cell cycle regulation based on gene expression data [12]. In Boolean network, gene expression is simply considered as binary values, ON or OFF, and the regulation between genes is set by Boolean function. Boolean network was then extended to be Probabilistic Boolean network [13]. This model consists of a family of Boolean networks that combine more than one transition Boolean functions. Inferred network which composes of a set of Boolean functions is selected by using the highest score based on probability. In 2007, Martin and colleagues [6] used Boolean dynamics to infer genetic regulatory network. Possible Boolean networks were generated from microarray time series data. The genetic network was then inferred from selection of possible Boolean networks by using steady-state dynamics. However, the result from Boolean network often includes a number of false positive, resulting in a complex inferred network. To resolve such a problem of Boolean network, recently, our group proposed a constraint-based Boolean network to formulate genetic network by taking prior knowledge into account [14]. The prior knowledge in this work, called the *conceptual constraints*, *i.e.*, enzymatic coding genes do not control and regulate regulatory genes, were included in the filtering process before generating Boolean functions. The result showed the achievement of this model to reduce the complexity of the inferred genetic network by eliminating a certain false prediction.

One of the most studied genetic networks is circadian clock system (*i.e.* a genetic circuit generates about 24h rhythm or circadian rhythm) because it is an important system controlling many biological processes in a wide range of organisms, including plants. Circadian clock in plants has mostly been studied in *Arabidopsis thaliana* [15, 16] in which a certain network components and regulations are revealed. The core circadian clock composes of multiple interlocked feedback loops such as interlock with the timing of cab expression 1 (*TOC1*)/CIRCADIAN AND CLOCK ASSOCIATED1 (*CCA1*)/LATE ELONGATED HYPOCOTYL (*LHY*) loop and (Pseudo-response regulator; *PRR5/PRR7/PRR9*)/*CCA1/LHY* loop. Experimental results show that *CCA1* and LHY are partially redundant genes which are negative regulators of *TOC1* [15]. *CCA1* and *LHY* are also positive regulators of two *TOC1* relatives, *PRR7*, and *PRR9* [16], while *TOC1* acts as a positive regulator of *CCA1* and *LHY*.

The circadian clock network has been used for computational method demonstration in various works [7, 14, 17], mainly due to the appropriate size of the network and (microarray) data availability (http://www.ncbi.nlm.nih.gov/geo/). Needham and colleagues [7] inferred a relationship between circadian-clock genes from an initial set of key genes and iteratively learned to increase network members around genes. A circadian clock was also used as a seed for inferring genetic network by finding co-regulation patterns between gene pairs in circadian clock [8]. The genetic network of *Arabidopsis* genome was performed by using an iterative random sampling strategy.

In this work, we extended the previous constraint-based Boolean analysis [14] to acquire a more accurate network inference by focusing on the filtering process. The specific biological constraints derived from prior knowledge of a particular system under study were introduced to the Boolean network in addition to the conceptual constraints. The algorithm takes a set of genes in a standard input format that is easy in the data preparation process. The extended method was demonstrated in inference of a genetic network underlying circadian rhythms in *A. thaliana* using microarray time series data. Finally, the inferred circadian network was validated with literature [15, 16, 18] and the performance of the extended algorithm was evaluated. The genetic network inferred from our algorithm was compared with that of the constraint-based Boolean approach. The results showed that our algorithm, which considers both conceptual and specific constraints, can increase the accuracy, specificity, and precision of the inferred network. Also the degree of complexity of the inferred network is substantially reduced, resulting more understandable results. The proposed method is therefore a promising alternative approach for inferring larger-scale genetic network from high-throughput microarray time-series data.

2 Methods for Reconstructing Constraint-Based Boolean Network of Circadian Rhythms

The overview of methodology is shown in Fig. 1A. Briefly, expression data was pre-processed before discretization. The binary values from discretization step are the inputs for the extended constraint-based Boolean program to generate Boolean functions and types of regulating genes, i.e., activation and inhibition. The output from the constraint-based Boolean program is Boolean relationship which can be visualized by Cytoscape [19].

2.1 Microarray Data and Data Pre-Processing

Gene expression time series datasets from the Affymetrix microarray under diurnal changes of *Arabidopsis* leaves were downloaded from NCBI database (http://www.ncbi.nlm.nih.gov, experiment reference number is GSE8365) [20]. *Arabidopsis* were grown in light/dark cycles for 7 days and then transferred to constant light. After 24 hours in constant light, 12 samples were harvested at four hours intervals over the next 44 hours for RNA extraction and hybridization on Affymetrix microarrays. The expression data were preprocessed using a package of Bioconductor [21, 22].

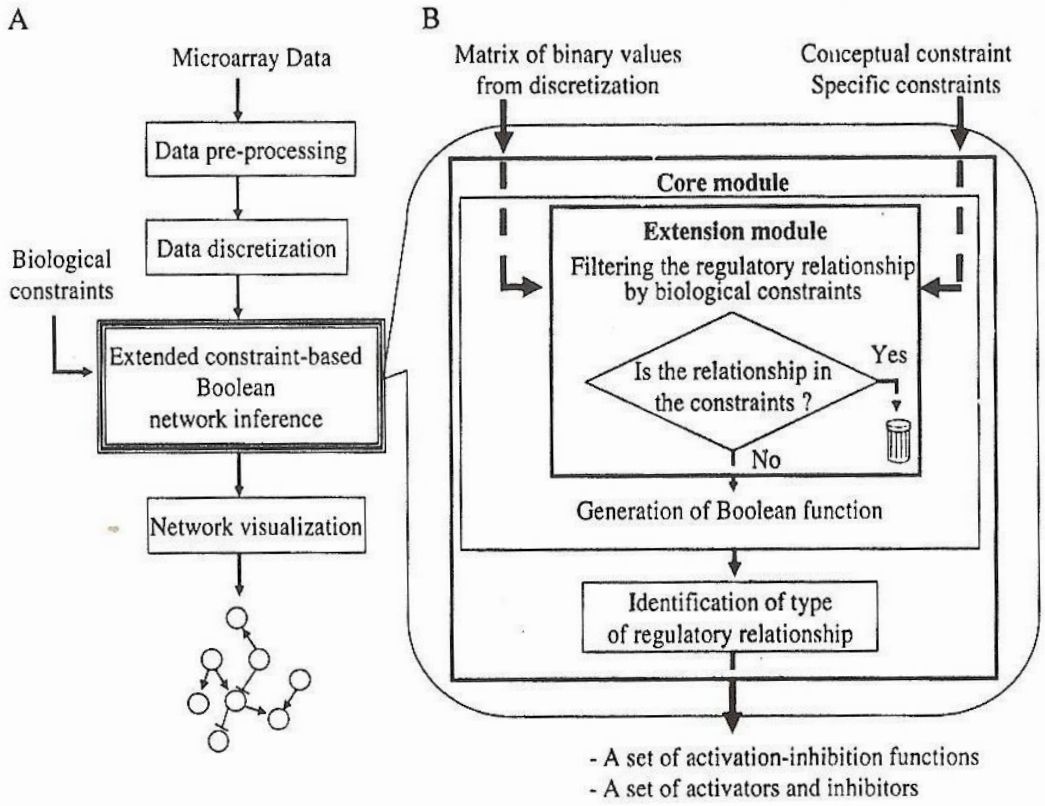


Fig. 1. Overall methodology for genetic network reconstruction by using (A) our constraint-based Boolean algorithm where the extension to the previous method is described in (B)

2.2 Data Discretization

The continuous expression level of gene was discretized into two levels, either '0' or '1', based on the concept of Boolean analysis that, herein, represents the strength of expression, or state, of gene at particular time. In other words, the expression level of gene was converted into either '0' for weak expression or '1' for strong expression. In this work, we used the maximum value of expression level as the simple criteria for discritization that the expression level of gene greater than the determined percentage of the maximum value was discretized into '1', '0' otherwise. The discretized value, $s_{i,t}$, for gene i at time t is defined as Equation (1).

$$s_{i,t} = \begin{cases} 1 & \text{if } x_{i,t} > \text{Max}(G_i) - r \cdot \text{Max}(G_i) \\ 0 & \text{others,} \end{cases} \quad (1)$$

where $x_{i,t}$ is the expression level of gene i at time t, G_i is the set of all expression levels of gene i over the time series, and r is the percentage of the maximum value of expression level of gene i. Here, the expression level greater than 30% of the highest value was converted to 1, *i.e.* r = 0.3. The data matrix of discretized values of all gene expression, *i.e.* $S = [s_{i,t}]$, is called matrix of binary values. It was then passed to extended constraint-based Boolean algorithm to generate Boolean functions

representing the regulatory relationship that is necessary for the construction of the Boolean network.

2.3 Constraint-Based Boolean Network Inference

Our method, called the *extended constraint-based Boolean algorithm* was adapted and implemented based on the previous published works [6, 14]. The algorithm consists of two modules, core and extension modules (Fig. 1B). The core module slightly adapted from the algorithm in Martin et al. [6] includes generation of Boolean function and identification of type of regulatory relationship, i.e. either activation or inhibition. The latter module is filtering the relationship of genes by biological constraints provided by a user. It is the extension we added in order to improve the performance of the classical Boolean algorithm by reduction of the number of relationships considered in the core module. Nevertheless, the network inference can be performed without this module if a set of biological constraints is not submitted.

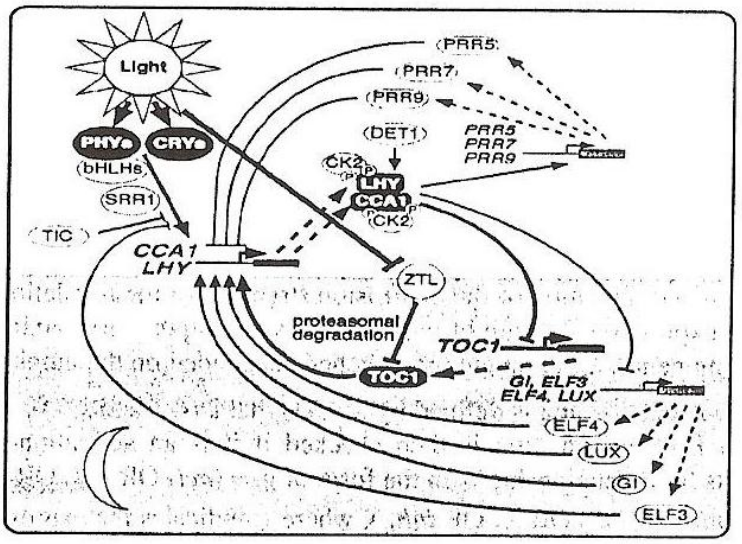
Core module: Generation of Boolean functions and identification of types of regulatory relationship. In brief, at first, the matrix of binary values of all interesting genes is passed into the first module to extract the regulatory relationship between the set of regulating genes and single target gene. The gene expression data, *i.e.* '0' or '1', of the target gene at time t is influenced by the expression strength of the regulating genes at previous time, t-1. This relationship is in the form of Boolean functions having a logic combination of the expression strengths of the regulating genes as an input and the expression strength of target gene as an output. The maximum number of the regulating genes, k, for single target gene is depended on the number of time points of expression data, T, and is defined by Max(k) that $2^k < T$ and $k > 0$.

Each Boolean function is then checked if it is an activation-inhibition function which its logic relationship is in the form of $g_i = (act_1 \text{ OR } act_2 \text{ OR } \ldots \text{ OR } act_{A_i})$ AND NOT $(inh_1 \text{ OR } inh_2 \text{ OR } \ldots \text{ OR } inh_{I_i})$, where g_i indicates the expression strength of the i^{th} target gene, *act* and *inh* indicates the expression strength of activators and inhibitors for the i^{th} target gene respectively, and A_i and I_i are the number of activators and inhibitors for the i^{th} target gene respectively. The activators and inhibitors of each target gene are simultaneously identified in this checking step. The type of regulatory relationship, either activation or inhibition, is finally assigned based on the activation-inhibition function. Other Boolean functions that are not the activation-inhibition function are ignored. Consequently, the output from the algorithm is a set of activation-inhibition functions and a set of activators and inhibitors for each target gene.

Extension module: Filtering the regulatory relationship by biological constraints. In the extension module, the relationship of genes is filtered by a set of biological constraints that are considered as the prior knowledge for a specific system. In this work, two types of biological constraints, *i.e.*, conceptual and specific constraints are added in the program. The conceptual constraint is first added to the previous algorithm [14]. It includes the general concept of the regulation in the transcriptional level, for example, (i) transcription factors directly regulate the gene expression by binding at promoter of genes; and (ii) enzymes and transporters do not regulate the gene expression although some of their downstream products do. Here, this type of

constraints is set based on a presumption that there is not any regulation by products of enzyme-encoding genes within duration of study. The relationship with this type of genes is hence discarded. The specific constraint introduced in this work includes prior knowledge, hypothesis, or existing experimental data indicating the regulatory relationship between the specific set of genes. Here, this set of constraints is specified by pairs of genes having no regulatory relationship which is supported by biological evidences. For example, the relationship between CONSTANTS (*CO*) and phosphoglycerate kinase (PGK) was set as null because the genes have distinct functions in different pathway and there is no experimental data inferring their relationship. The Boolean relationship between these two genes generated was hence

A



B

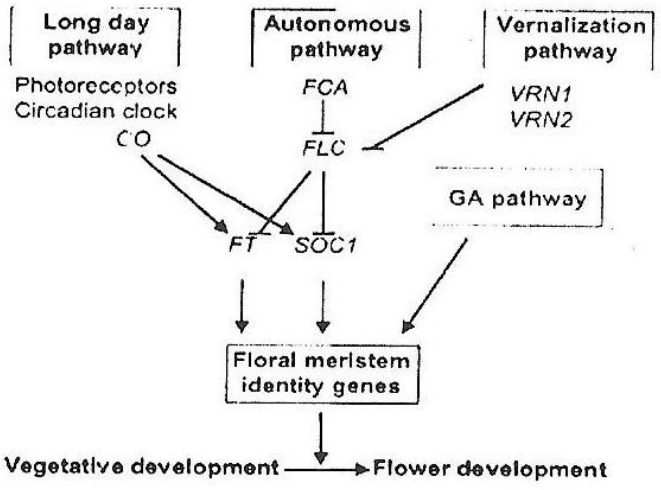


Fig. 2. Gene relationship in (A) circadian clock [18] and (B) flowering pathway [23]

discarded by consideration of that specific constraint. However, the network inference can be performed without this module if a set of biological constraints is not submitted.

2.4 Evaluation of Genetic Network

The obtained genetic network were evaluated by using standard measures that were accuracy (ACC) calculated by (TP+TN)/(TP+TN+FP+FN), specificity (SPC) calculated by TN/(FP+TN), precision or positive prediction value (PPV) calculated by TP/(TP+FP), and false discovery rate (FDR) calculated by FP/(FP+TP), where *TP* refers to correctly inferred edges, either activation or inhibition. *FP* refers to false predicted relationship, including wrong type of regulation. *TN* refers to missing edges in both the inferred and the reference networks. *FN* refers to missing edges, which exist in the reference network, in the inferred network. The network that was used as a reference was based on selected genes in this study [18, 23] (Fig. 2).

Accuracy indicates the percentage of correct predictions. Specificity indicates the percentage of negative predictions which are correctly inferred. Precision indicates the percentage of positive predictions which are correctly inferred. False discovery rate indicates the percentage of false predictions among all predictions.

3 Results and Discussion

3.1 Data Discretization

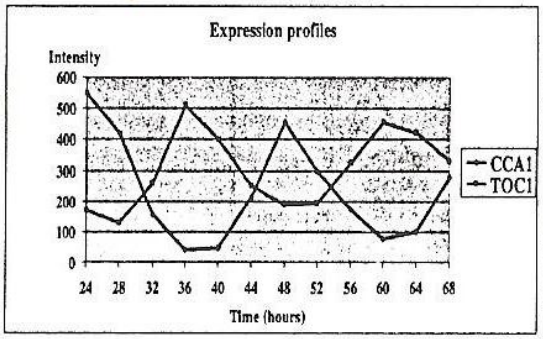
To demonstrate the algorithm, the expression data of fourteen genes were selected from microarray data [20], including seven known core-circadian-clock genes and seven non-circadian genes (*i.e.* genes in glycolysis and flowering pathways). The selected genes and their molecular functions are shown in Table 1.

Table 1. List of genes and functions used in this study [18, 24]

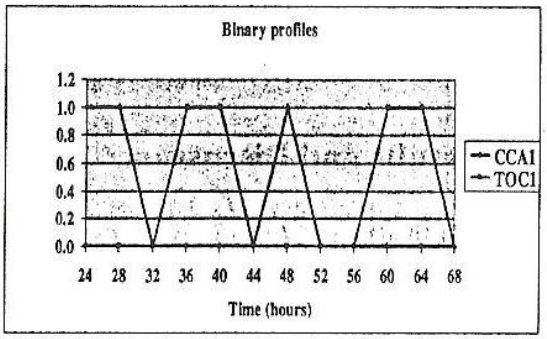
Gene	Function
CCA1	Single Myb domain Transcription factor
ELF4	Transcription factor
GI	Unknown
LHY	Single Myb domain transcription factor
PRR5	Pseudo-response regulator
PRR7	Pseudo-response regulator
TOC1	Pseudo-response regulator
CO	CONSTANTS (CO) promotes flowering under long days
FT	Flowering locus promotes flowering
SOC1	Suppressor of overexpression of CO1
PGI1	Phospho-glucose (Glc) isomerase
TPI	Triosephosphate isomerase
PGK	Phosphoglycerate kinase
PGM	Phosphoglycerate mutase

Max-30%max was used as a threshold for the data discretization in this study. An example of the characteristics of discretized data is shown in Fig. 3. Fig. 3A shows the expression data of two selected circadian clock-genes, *CCA1* and *TOC1*. Based on the discretization method, the expression data of *CCA1* and *TOC1* across time points were discretized into 0 or 1. So, each gene consists of a series of binary values, called binary profile. Fig. 3B shows the binary profiles of such genes. The selection of the discretization method may affect the final result; however the employed discretization method was proven to be the most appropriate one for the system under studied (unpublished data). The matrix of binary values representing binary profiles of a set of genes was an input for the algorithm, see Fig. 3C. The matrix of binary values is delimited text format. The first column is the gene name. The following columns are binary values of each gene across time points. This is a standard format that is convenient for users in the data preparation process.

A



B



C

CCA1	1	0	0	0	0	0	0	0	0	0	0	0
ELF4	0	0	0	0	1	0	0	0	0	0	0	0
GI	0	0	1	1	0	0	0	0	0	0	0	0
LHY	1	0	0	0	0	0	0	0	0	0	0	0
PRR5	0	0	1	1	0	0	0	0	1	1	0	0
PRR7	0	0	1	0	0	0	0	1	1	0	0	0
TOC1	0	0	0	1	0	0	0	0	0	0	0	0
CO	0	0	0	0	0	1	0	0	0	0	0	0
FT	1	0	0	0	0	0	0	0	0	0	0	0
SOC1	0	0	0	0	0	0	0	0	1	1	0	0
PGI1	1	1	1	0	1	0	0	1	1	1	1	1
TPI	0	0	0	1	1	1	0	1	0	0	0	1
PGK	1	1	0	0	1	1	0	0	0	0	1	0
PGM	1	1	1	0	0	0	1	1	0	0	0	0

Fig. 3. Characteristics of data expression profiles (A), example of discretized data (B), and the matrix of binary values (C) of genes in circadian clock, glycolysis, and flowering mechanism

3.2 Genetic Network of Circadian Rhythms

The genetic network of circadian clock was inferred by using our method, the extended constraint-based algorithm with taking consideration of both conceptual and specific constraints into account. The network result by our method was compared with those by the classical Boolean without any biological constraint and the constraint-based Boolean with only conceptual constraints. All these three algorithms can

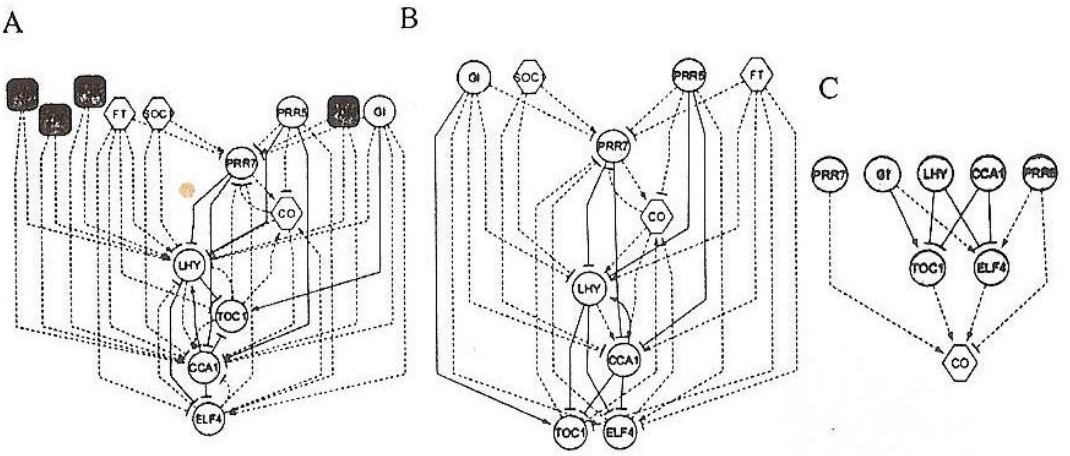


Fig. 4. Genetic networks of 14 genes in circadian rhythm and flowering mechanism using classical Boolean, without consideration of biological constraint *in priori* (A); constraint-based Boolean with conceptual constraints (B); constraint-based Boolean with both conceptual and specific constraints (C). A blue rectangular represents an enzymatic gene; a yellow hexagon represents a flowering gene; a pink circle represents a circadian gene. Black solid lines represent predicted relationship corresponding to known biological knowledge, while broken lines represent predicted relationship that exceeds the current knowledge.

infer directed networks describing the types of gene regulatory relationship, either activation or inhibition (Fig. 4). The node is a gene and the edge is the relationship between genes. The types of the relationships are represented by an arrow and a T-shape arrow for activation and inhibition, respectively.

Figs. 4A-C show the networks inferred by using classical Boolean, constraint-based Boolean with conceptual constraints, and constraint-based Boolean with both conceptual and specific constraints, respectively. All these inferred genetic networks can describe regulations between *TOC1/CCA1/LHY* and (*PRR5/PRR7/PRR9*) */CCA1/LHY* loops. All three inferred network show that *CCA1* and *LHY* are inferred as negative regulators of *TOC1* and *ELF4* which corresponds to known biological knowledge [15, 18], while *PRR5* and *PRR7* are inferred as positive regulators of *CCA1* and *LHY* [16] in the inferred network by using classical Boolean and constraint-based Boolean with conceptual constraints. However, among three inferred networks, the two networks from the previously developed methods show significantly higher in complexity and false predictions than the one from our method, indicated by the number of solid and broken line edges. Adding conceptual constraints before generating Boolean function can greatly reduce the complexity of the network (Fig. 4B). That means it can reduce false predicted relationships that are caused by regulations by products of enzyme-encoding genes as shown in Fig. 4A. Fig. 4C shows the inferred genetic network by using our method with adding conceptual and specific constraints before generating Boolean function. The algorithm can identify regulations in the core oscillator of circadian mechanism and also substantially reduce the false predicted relationships. This resulted in less complex inferred network that is reasonable for further analysis and making a biological sense.

3.3 Network Evaluation

The inferred genetic networks were evaluated through a set of coefficients: ACC, SPC, PPV, and FDR. These coefficients allow us to assess the performance of our algorithm in comparison with those of the previously developed methods which are the classical Boolean and the constraint-based Boolean algorithms with conceptual constraints.

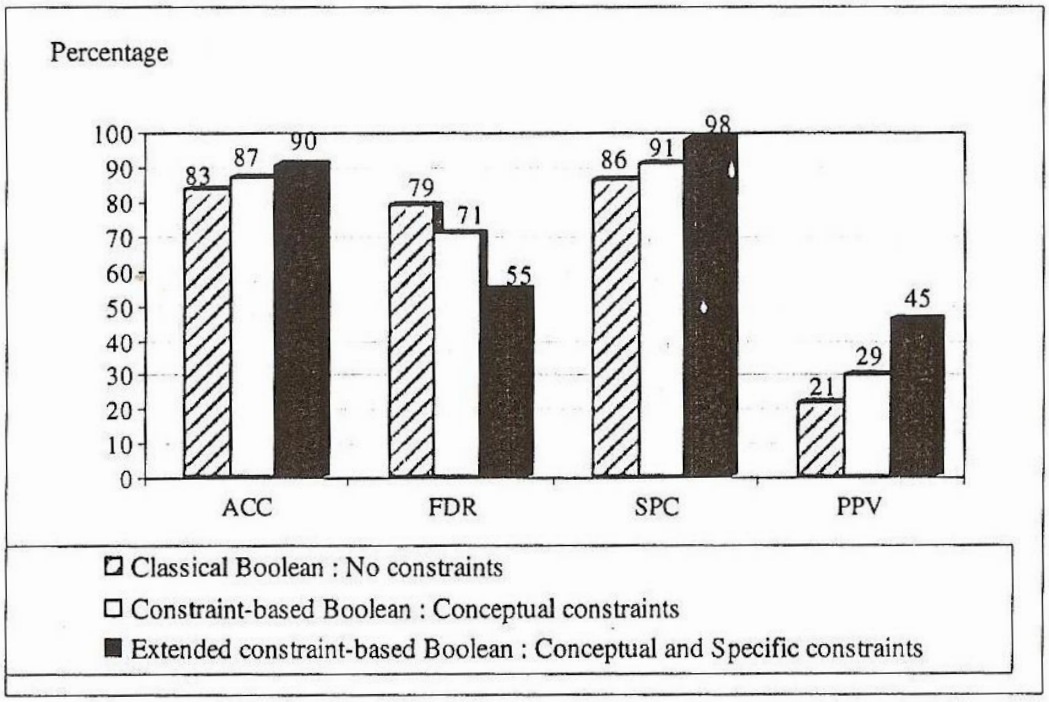


Fig. 5. Comparing the performances of the extended constraint-based Boolean, constraint-based Boolean, and classical Boolean algorithms

Fig. 5 shows that the Boolean network taking into consideration of biological constraints gives better accuracy, specificity, and precision. In comparison with the classical Boolean network, the extended constraint-based Boolean algorithm provides 90% accuracy, 98% specificity, and 45% precision, which are 8%, 13% and 114% improvement, respectively. Moreover, the false discovery rate (FDR) is decreased from 79% to 55% (31% improvement). When considering the Boolean network taking only the conceptual constraints into account, the percent improvement over the classical Boolean network are 4%, 5%, and 38% for accuracy, specificity, and precision, respectively. These results clearly show that taking more consideration of biological constraints in priori can provide better accuracy, specificity, and precision. Besides the improve accuracy, the extended constraint-based Boolean algorithm provides a result with a low level of false prediction. The extension of the constraint-based method by incorporating the specific constraint is thus not only advantage in term of reduction in, but also great decrease in computational burden due to Boolean functions calculation. Not only genetic network inference of circadian clock, the

algorithm was also applied to infer genetic network of galactose pathway using microarray data [25]. The results show that our algorithm provides both high (>70%) accuracy and (>80%) specificity (unpublished data). Therefore, the extended constraint-based Boolean algorithm might be an alternative strategy for genetic network inference. Also, this method might be employed to infer a large-scale genetic network, whose result might be used as seed information for further network analysis or hypothesis development. Although the incorporation of prior knowledge into Boolean network is not yet systematic, this can help scientists to understand simpler genetic network inferred by using this method. However, it will be great to develop it as more systematic approach.

In this work, we have shown the advantages and successes of incorporation prior knowledge (in terms of specific constraint) into the Boolean network though implantation of the constraint in not yet systematic. For the next step, computational technique including systematic incorporation of the constraint will be improved to have the capability of the algorithm to support the large-scale data analysis.

4 Conclusion

The regulation of gene expression lies on a huge number of components that comprise a genetic network. Understanding of this regulation system is often studied by inference of genetic networks from microarray data. We have proposed an algorithm so-called extended constraint-based Boolean algorithm to infer genetic network. The algorithm considers both conceptual constraints of a typical genetic circuit and specific constraints of a particular system before generating Boolean functions. The method was demonstrated in inference of a genetic network underlying circadian rhythms in Arabidopsis thaliana from microarray time series data. The inferred circadian network was validated with literature and the performance of the novel algorithm was evaluated. The resulted network showed that prior knowledge is an useful bias for modeling genetic network. Moreover, the results showed that the proposed method provides good accuracy, specificity, and precision under the trade-off of computational efforts. The proposed method is therefore a promising alternative approach for inferring genetic network from high-throughput microarray time series data. In the future, this method will be applied to infer genetic network from different conditions of microarray data.

References

1. Karlebach, G., Shamir, R.: Modelling and analysis of gene regulatory networks. Nat. Rev. Mol. Cell. Biol. 9, 770–780 (2008)
2. Zhang, S.-Q., Ching, W.-K., Ng, M.K., Akutsu, T:: Simulation study in Probabilistic Boolean Network models for genetic regulatory networks. Int. J. Data Min. Bioinform. 1, 217–240 (2007)
3. Kwon, A.T., Hoos, H.H., Ng, R.: Inference of transcriptional regulation relationships from gene expression data. Bioinformatics 19, 905–912 (2003)
4. Kervestin, S., Amrani, N.: Translational regulation of gene expression. Genome Biol. 5, 359 (2004)

5. Kauffman, S.A.: Metabolic stability and epigenesis in randomly constructed genetic nets. J. Theor. Biol. 22, 437–467 (1969)
6. Martin, S., Zhang, Z., Martino, A., Faulon, J.L.: Boolean dynamics of genetic regulatory networks inferred from microarray time series data. Bioinformatics 23, 866–874 (2007)
7. Needham, C.J., Manfield, I.W., Bulpitt, A.J., Gilmartin, P.M., Westhead, D.R.: From gene expression to gene regulatory networks in Arabidopsis thaliana. BMC Syst. Biol. 3, 1–18 (2009)
8. Ma, S., Gong, Q., Bohnert, H.J.: An Arabidopsis gene network based on the graphical Gaussian model. Genome Res. 17, 1614–1625 (2007)
9. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol. Biol. Cell 9, 3273–3297 (1998)
10. Li, P., Zhang, C., Perkins, E.J., Gong, P., Deng, Y.: Comparison of probabilistic Boolean network and dynamic Bayesian network approaches for inferring gene regulatory networks. BMC Bioinformatics 8 (suppl. 7), 13 (2007)
11. Kauffman, S.A.: The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, New York (1993)
12. Shmulevich, I., Zhang, W.: Binary analysis and optimization-based normalization of gene expression data. Bioinformatics 18, 555–565 (2002)
13. Shmulevich, I., Dougherty, E.R., Kim, S., Zhang, W.: Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics 18, 261–274 (2002)
14. Munkung, W., Liamwirat, C., Bumee, S., Meechai, A.: A constraint-based Boolean approach to inferring genetic circuits. In: The 13th International Annual Symposium on Computational Science and Engineering, pp. 427–431 (2009)
15. Alabadí, D., Oyama, T., Yanovsky, M.J., Harmon, F.G., Más, P., Kay, S.A.: Reciprocal regulation between TOC1 and LHY/CCA1 within the Arabidopsis circadian clock. Science 293, 880–883 (2001)
16. Farre, E.M., Harmer, S.L., Harmon, F.G., Yanovsky, M.J., Kay, S.A.: Overlapping and distinct roles of PRR7 and PRR9 in the Arabidopsis circadian clock. Curr. Biol. 15, 47–54 (2005)
17. Du, P., Gong, J., Syrkin Wurtele, E., Dickerson, J.A.: Modeling gene expression networks using fuzzy logic. IEEE Trans. Syst. Man Cybern. B Cybern. 35, 1351–1359 (2005)
18. McClung, C.R.: Plant circadian rhythms. Plant Cell 18, 792–803 (2006)
19. Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. 13, 2498–2504 (2003)
20. Covington, M.F., Harmer, S.L.: The circadian clock regulates auxin signaling and responses in Arabidopsis. PLoS Biol. 5, e222 (2007)
21. Workman, C., Jensen, L.J., Jarmer, H., Berka, R., Gautier, L., Nielser, H.B., Saxild, H.H., Nielsen, C., Brunak, S., Knudsen, S.: A new non-linear normalization method for reducing variability in DNA microarray experiments. Genome Biol. 3, research0048 (2002)
22. Li, C., Wong, W.H.: Model-Based Analysis of Oligonucleotide Arrays: Expression Index Computation and Outlier Detection. In: Conference Model-Based Analysis of Oligonucleotide Arrays: Expression Index Computation and Outlier Detection, pp. 31–36 (2001)
23. Blazquez, M., Koornneef, M., Putterill, J.: Flowering on time: genes that regulate the floral transition. EMBO reports 2, 1078–1082 (2001)
24. The Arabidopsis Information Resource, `http://www.arabidopsis.org/`
25. DeRisi, J.L., Iyer, V.R., Brown, P.O.: Exploring the metabolic and genetic control of gene expression on a genomic scale. Science 278, 680–686 (1997)