แนวคิดใหม่ภายใต้วิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค เพื่อระบุส่วนที่ยึดจับปัจจัยการถอดรหัสในสายดีเอ็นเอที่หลากหลาย

A NOVEL PSO-BASED APPROACH TO IDENTIFY TRANSCRIPTION FACTOR BINDING SITES IN A VARIETY OF DNA SEQUENCES

ศราวุธ โสมอินทร์
SARAWOOT SOM-IN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2562
KMITL-2019-SC-D-001-009

แนวคิดใหม่ภายใต้วิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค เพื่อระบุส่วนที่ยึดจับปัจจัยการถอดรหัสในสายดีเอ็นเอที่หลากหลาย

A NOVEL PSO-BASED APPROACH TO IDENTIFY TRANSCRIPTION FACTOR BINDING SITES IN A VARIETY OF DNA SEQUENCES

ศราวุธ โสมอินทร์
SARAWOOT SOM-IN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2562
KMITL-2019-SC-D-001-009

A NOVEL PSO-BASED APPROACH TO IDENTIFY TRANSCRIPTION FACTOR BINDING SITES IN A VARIETY OF DNA SEQUENCES

SARAWOOT SOM-IN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY PROGRAM IN COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2019

KMITL-2019-SC-D-001-009

หัวข้อวิทยานิพนธ์ แนวคิดใหม่ภายใต้วิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคเพื่อระบุส่วนที่ยึดจับปัจจัยการถอดรหัสในสายดีเอ็นเอที่หลากหลาย
ชื่อนักศึกษา นายศราวุธ โสมอินทร์
รหัสประจำตัว 55650552
ปริญญา ปรัชญาดุษฎีบัณฑิต
ภาควิชา วิทยาการคอมพิวเตอร์
คณะ วิทยาศาสตร์
มหาวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
พ.ศ. 2561
อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ.ดร.วรางคณา กิ้มปาน

บทคัดย่อ

การตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสถือเป็นปัญหาที่สำคัญสำหรับงานวิจัยด้านชีวสารสนเทศศาสตร์ (Bioinformatics) เนื่องจากรูปแบบอักขระที่คล้ายกันของกลุ่มสายโมทีฟ ทำให้มีการพัฒนาขั้นตอนวิธีด้านคอมพิวเตอร์ในการตรวจหาเพื่อลดทรัพยากรที่ใช้ในห้องปฏิบัติการลง โดยการตรวจหามีสองแบบ ได้แก่ การตรวจหาแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า และการตรวจหาแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า ซึ่งหนึ่งในแนวคิดที่นิยมนำมาประยุกต์ใช้เพื่อการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสคือ แนวคิดความฉลาดแบบกลุ่ม (Swarm Intelligence) อย่างไรก็ตามความผิดพลาดในการตรวจหายังคงเกิดขึ้นได้ เนื่องจากรูปแบบอักขระระหว่างสายโมทีฟที่เป็นส่วนที่ยึดจับปัจจัยการถอดรหัส มีรูปแบบของอักขระที่แตกต่างกัน ดังนั้นเพื่อเป็นการแก้ไขข้อผิดพลาดในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัส งานวิจัยนี้ได้นำเสนอ 2แนวคิด แนวคิดแรกคือการพัฒนาความแม่นยำในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า โดยประยุกต์ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (PSO) กับขั้นตอนสานสัมพันธ์ (Nexus) ซึ่งเป็นขั้นตอนที่คิดขึ้นใหม่ เรียกว่าขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาค (NexusPSO) โดยความแม่นยำของขั้นตอนวิธีที่นำเสนอวัดได้จากการเปรียบเทียบค่าอินฟอร์เมชั่นคอนเทนท์กับขั้นตอนวิธีที่เกี่ยวข้อง ซึ่งทดสอบกับข้อมูลดีเอ็นเอของอีโคไล (Escherichia Coli: E.Coli) จากผลการทดลองแสดงให้เห็นว่าขั้นตอนวิธีที่นำเสนอมีความถูกต้องสูงสุด สำหรับแนวคิดที่สอง เป็นการนำเสนอขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคร่วมกับระยะทางแฮมมิง (PSO_HD) ซึ่งเป็นการประยุกต์ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค ร่วมกับหลักการระยะทางแฮมมิง (Hamming Distance) เพื่อการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า ซึ่งจากผลการทดลองแสดงให้เห็นว่าขั้นตอนวิธีที่นำเสนอสามารถพัฒนาการตรวจหาที่แม่นยำและถูกต้องครอบคลุมมากขึ้น

คำสำคัญ : ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค ส่วนที่ยึดจับปัจจัยการถอดรหัสในสายดีเอ็นเอ ระยะทางแฮมมิง การตรวจหาสายโมทีฟ คะแนนความคล้ายแบบเมทริกซ์

Thesis Title	A Novel PSO-Based Approach to Identify Transcription Factor Binding Sites in a Variety of DNA Sequences
Student Name	Mr. Sarawoot Som-in
Student ID	55650552
Degree	Doctor of Philosophy Program
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Year	2018
Thesis Advisor	Asst.Prof.Dr. Warangkhana Kimpan

Abstract

The detection of Transcription Factor Binding Sites (TFBSs) is an important problem for bioinformatics research. According to similar patterns of motif sequences, computational algorithms for detection have been improved to reduce resources used in laboratory. There are 2 types of the detection: to detect one motif sequence per one input sequence and to detect multiple sequences per one input sequence. One of the well-known methods commonly used for TFBSs detection is Swarm Intelligence. However, errors in TFBSs detection can be caused by different binding sites in the same genome sequence. Therefore, to fix the error in TFBSs detection, this research presents two concepts: The first purpose of this research is to improve the effectiveness and accuracy in the detection of TFBSs with one motif sequence per one input sequence by applying Particle Swarm Optimization algorithm (PSO) and the newly developed Nexus procedure, called NexusPSO. The accuracy of the proposed algorithm was measured and compared with related algorithms, using information content (IC) as an indicator and test with DNA data of Escherichia coli. The experimental result indicated that the proposed algorithm is the most accurate. The second purpose of this research is to propose PSO_HD by applying PSO and Hamming distance to improve the efficiency of TFBSs detection in multiple motif sequences per one input sequence. The experiments indicate that the proposed algorithm can improve the efficiency of detecting TFBSs with more precision and recall.

Keywords : Particle Swarm Optimization (PSO), Transcription Factor Binding Sites (TFBSs), Hamming Distances (HD), Motif Detection, Matrix Similarity Score

กิตติกรรมประกาศ

การศึกษาในระดับปริญญาเอกในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เป็นความใฝ่ฝันของนักศึกษาในสาขาวิทยาการคอมพิวเตอร์หลายๆ คน เนื่องจากเป็นสถาบันที่มีชื่อเสียงและได้รับการยอมรับในด้านการให้ความรู้และการนำความรู้นั้นไปพัฒนาใช้ ผู้เขียนรู้สึกเป็นเกียรติอย่างมากที่ได้รับโอกาสในการเข้ามาศึกษาเล่าเรียนและสามารถสำเร็จการศึกษาในระดับสูงสุดจากสถาบันแห่งนี้ ท่ามกลางความชื่นชมยินดีของครอบครัวและคนใกล้ชิด ความสำเร็จนี้เกิดจากคณาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ และให้ความเมตตาผู้เขียนตลอดมานับตั้งแต่วันแรกที่ได้เข้ามาศึกษาในสถาบันแห่งนี้

วิทยานิพนธ์นี้สามารถสำเร็จลุล่วงได้ด้วยดี จากความกรุณาอย่างยิ่งของ ผู้ช่วยศาสตราจารย์ ดร. วรางคณา กิ้มปาน อาจารย์ที่ปรึกษา ในการให้คำเสนอแนะ แนวคิด ตลอดจนแก้ไขข้อบกพร่องจนกระทั่งผลงานวิชาการและวิทยานิพนธ์เล่มนี้เสร็จสมบูรณ์ จึงขอขอบพระคุณเป็นอย่างสูง และขอขอบพระคุณประธานกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร. ชัยพร ใจแก้ว กรรมการสอบวิทยานิพนธ์ รองศาสตราจารย์ ดร. จีรพร วีระพันธ์ ดร. รุ่งรัตน์ เวียงศรีพนาวัลย์ และ ดร. กุลสวัสดิ์ จิตขจรวานิช ที่กรุณาให้คำแนะนำเพื่อปรับปรุงแก้ไขโดยเฉพาะอย่างยิ่งในส่วนของผลการทดลอง ทำให้วิทยานิพนธ์นี้มีความสมบูรณ์แบบมากยิ่งขึ้น ขอขอบคุณ นาย จิรัฏฐ์ สำราญสุข และ Mr. Graysen Ortega ที่ช่วยให้คำปรึกษา ปรับปรุงแก้ไขไวยากรณ์ภาษาอังกฤษในบทความวิชาการเพื่อส่งตีพิมพ์ในระดับนานาชาติ ขอขอบคุณ นางสาวทิฆัมพร บุริมสิทธิกุล ที่ช่วยหาข้อมูลที่จำเป็นเพื่อสนับสนุนการเขียนบทความวิชาการ ทำให้เนื้อหาในบทความวิชาการมีความต่อเนื่อง และมีข้อมูลอ้างอิงที่เชื่อถือได้ สุดท้ายขอขอบคุณ ครอบครัว คุณพ่อ คุณแม่ พี่ชาย และภรรยา ที่อยู่เคียงข้าง มอบกำลังใจและให้คำปรึกษา จนกระทั่งผ่านเรื่องราวและอุปสรรคต่างๆ มาได้อย่างดี ขอบคุณพระเจ้า ผู้ไม่เคยทำให้ผู้เขียนหมดความหวังใจและท้อถอยในการทำผลงานวิชาการและวิทยานิพนธ์ให้ดีที่สุด เพื่อเป็นประโยชน์ต่อสังคมต่อไป

ศราวุธ โสมอินทร์

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

คำย่อ/สัญลักษณ์

คำย่อ/สัญลักษณ์	คำอธิบาย
TFBSs	ส่วนที่ยึดจับปัจจัยการถอดรหัสในสายโปรตีน
CoM	ผลลัพธ์สายโมทีฟแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า
MultiM	ผลลัพธ์สายโมทีฟแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า
PWM	รูปแบบค่าน้ำหนักตำแหน่งแบบเมทริกซ์
Nexus	ขั้นตอนสานสัมพันธ์ เป็นแนวคิดที่พัฒนาขึ้นใหม่เพื่อการเตรียมข้อมูล
HD	ระยะทางแฮมมิง เป็นแนวคิดที่นำมาคำนวณความเหมือนระหว่างสายโมทีฟ
IC	อินฟอร์เมชั่นคอนเทนท์ เป็นสมการสำหรับคำนวณรูปแบบของโมทีฟที่อยู่ในอไลนเมนท์ซึ่งพิจารณาส่วนที่ไม่เป็นสายโมทีฟ
CS	คอนเซนซัสสกอร์ เป็นสมการสำหรับคำนวณรูปแบบของโมทีฟที่อยู่ในอไลนเมนท์ซึ่งไม่พิจารณาส่วนที่ไม่เป็นสายโมทีฟ
Cut-offs	การตัดผลลัพธ์ที่ไม่ถึงมาตรฐานออก
mSS	เป็นสมการในการคำนวณความเหมือนระหว่างอไลนเมนท์กับสายโมทีฟที่เป็นไปได้
F-score	เอฟสกอร์ เป็นสมการสำหรับวัดผลลัพธ์เรื่องความแม่นยำและครอบคลุม
PS	ค่าแม่นยำ เป็นค่าที่บ่งชี้ความแม่นยำของผลลัพธ์
RC	ค่าความถูกต้องครอบคลุม เป็นค่าที่บ่งชี้ความถูกต้องครอบคลุมของผลลัพธ์
RM	สายโมทีฟที่อยู่นอกเหนือจากอไลเมนท์
S	สายข้อมูลที่นำเข้า
M	สายโมทีฟ หรือสายโมทีฟที่เป็นไปได้ทั้งหมด

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

สายดีเอ็นเอ (Deoxyribonucleic Acid: DNA) มีส่วนประกอบที่สำคัญส่วนหนึ่งที่เป็นลำดับเบสสั้นๆ เรียกว่า ส่วนที่ยึดจับปัจจัยการถอดรหัส (Transcription Factor Binding Sites: TFBSs) ซึ่งเป็นส่วนที่ทำให้เกิดกระบวนการถอดรหัสยีน (Transcription Process) ส่งผลให้เกิดการสังเคราะห์โปรตีน (Protein Synthesis) ส่วนที่ยึดจับปัจจัยการถอดรหัสประกอบด้วยสายข้อมูลย่อย (Subsequences) หรือเรียกว่าสายโมทีฟ (Motif Sequences) ซึ่งมีคุณลักษณะ (รูปแบบอักขระ) ที่ใกล้เคียงกัน ข้อมูลส่วนที่ยึดจับปัจจัยการถอดรหัสคือข้อมูลที่ช่วยให้นักวิจัยด้านชีววิทยาสามารถทราบบริเวณที่ทำให้เกิดการสังเคราะห์โปรตีนของสายดีเอ็นเอ ซึ่งเป็นประโยชน์ต่อการศึกษาและพัฒนางานวิจัยที่เกี่ยวข้องอื่นๆ การตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสสามารถดำเนินการได้โดยการใช้อุปกรณ์ในห้องทดลอง [1] แต่เนื่องจากต้นทุนในการดำเนินการสูง จึงได้มีการพัฒนาแอปพลิเคชันทางด้านคอมพิวเตอร์เพื่อลดต้นทุนการตรวจหาส่วนที่ยึดจับปัจจัยถอดรหัสแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า โดยประยุกต์ขั้นตอนวิธีการสุ่มตัวอย่างแบบกิบส์ (Gibb Sampling) [2] คิดค้นโดย C. E. Lawrence และคณะในปี ค.ศ. 1993 และขั้นตอนวิธีอื่นๆ อาทิ ขั้นตอนวิธีค่าคาดหมายสูงสุด (Multiple Expectation-maximization for Motif Elicitation: MEME) [3] คิดค้นโดยโดย T.L. Bailey และ C. Elkan ในปี ค.ศ. 1994 ขั้นตอนวิธีคอนเซนซัส (Consensus) [4] คิดค้นโดย G.Z. Hertz และ G.D. Stormo ในปี ค.ศ. 1999 ขั้นตอนวิธีอไลเอซีอี (AlignACE) [5] คิดค้นโดย J.D. Hughes และคณะในปี ค.ศ. 2000 ขั้นตอนวิธีไบโอโพรสเปคเตอร์ (BioProspector) [6] คิดค้นโดย X. Liu และคณะในปี ค.ศ. 2001 และขั้นตอนวิธีเอ็มดีสแกน (MDScan) [7] คิดค้นโดย X.S. Liu และคณะในปี ค.ศ. 2002 เป็นต้น เมื่อวิเคราะห์รูปแบบการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัส สามารถระบุได้ว่าเป็นปัญหาที่ไม่สามารถแก้ได้ในขอบเขตของเวลา (NP-hard Problems) เช่นเดียวกับปัญหาการเดินทางของพนักงานขาย (Travelling Salesman Problem: TSP) [8] ปัญหาของการจัดตารางการผลิตแบบตามสั่ง (Job Shop Scheduling Problem: JSP) [9] ปัญหาการผลิตแบบไหลลื่น (Flow Shop Scheduling Problem: FSP) [10] ปัญหาการเชื่อมต่อระหว่างลำดับย่อยร่วมยาวสุด (Longest Common Subsequence Problem: LCS) [11] และปัญหาอื่นๆ อีกจำนวนมาก [12] จึงมีการพัฒนาขั้นตอนวิธีทางคอมพิวเตอร์เพื่อแก้ไขปัญหาเหล่านี้ อาทิ ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization: PSO) [13] คิดค้นโดย J.Kennedy และ R.Eberhart ในปี ค.ศ. 1995 และขั้นตอนวิธีการหาค่าที่เหมาะสมที่สุดด้วยระบบอาณาจักรมด (Ant Colony Optimization: ACO) [14] คิดค้นโดย M. Dorigo และคณะในปี ค.ศ. 1996 เป็นต้น แต่ความจำกัดของขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคและขั้นตอนวิธีการหาค่าที่เหมาะสมที่สุดด้วยระบบอาณาจักรมด คือการติดอยู่ในค่าที่ดีที่สุดเฉพาะที่ (Local Optimums) อย่างไรก็ตามยังมีการประยุกต์ใช้ขั้นตอนวิธีเหล่านี้ผสมผสานแนวคิดอื่นๆ เพื่อลดการติดอยู่ในค่าที่ดีที่สุดเฉพาะที่ ในปัญหาการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัส อาทิ ขั้นตอนวิธีเอซีอาร์ไอ (ACRI) [15] คิดค้นโดย W. Liu และคณะในปี ค.ศ. 2013 ขั้นตอนวิธีไอพีเอสโอ (IPSO) [16] คิดค้นโดย W. Zhou และคณะในปี ค.ศ. 2005 ซึ่งทั้งสองขั้นตอนวิธี

ให้ผลลัพธ์ที่ดีขึ้นแต่ยังมีข้อจำกัดเรื่องความแม่นยำในการตรวจหาสายโมทีฟที่มีรูปแบบของนิวคลีโอไทด์ที่แตกต่าง

สำหรับการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสอีกประเภทหนึ่งคือ คือการตรวจหาแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า ซึ่งมีการประยุกต์ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคร่วมกับหลักสถิติแบบเบย์เพื่อการตรวจหา (A Bayesian Scoring Scheme based Particle Swarm Optimization algorithm to identify transcription factor binding sites: PSO-variants) [17] คิดค้นโดย M. Karabulut และ T. Ibrikci ในปี ค.ศ. 2012 ในขณะที่ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm: GA) ถูกนำมาประยุกต์เพื่อแก้ปัญหาการตรวจหาเช่นกัน อาทิ ขั้นตอนวิธีจีเอเอ็มอี (GAME) [18] คิดค้นโดย Z. Wei และ S.T. Jensen ในปี ค.ศ. 2006 ซึ่งเป็นการประยุกต์ขั้นตอนวิธีเชิงพันธุกรรมร่วมกับหลักสถิติแบบเบย์ เพื่อการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสให้มีความครบถ้วน หลังจากนั้นขั้นตอนวิธีเชิงพันธุกรรมถูกนำมาประยุกต์ร่วมกับแนวคิดอื่นๆ เพื่อพัฒนาการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสที่มีความแม่นยำและครอบคลุมยิ่งขึ้น อาทิ จีเอแอลเอฟพี (GALF_P) [19] คิดค้นโดย T.M. Chan และคณะในปี 2008 จีเอพีเค (GAPK) [20] คิดค้นโดย D.H. Wang และX. Li ในปี 2009 ไอจีเอพีเค (iGAPK) [21] คิดค้นโดย D.H. Wang และ X. Li ในปี 2010 อย่างไรก็ตามขั้นตอนวิธีเหล่านี้ยังมีข้อจำกัดในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้าที่ครอบคลุมและแม่นยำ ในกลุ่มสายดีเอ็นเอที่มีคุณลักษณะที่หลากหลาย

ดังนั้น งานวิจัยนี้นำเสนอขั้นตอนวิธี เพื่อการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสที่มีความแม่นยำและถูกต้องครอบคลุมมากขึ้น ประกอบด้วย 2 ขั้นตอนวิธี ขั้นตอนวิธีแรกออกแบบและพัฒนาเพื่อการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า โดยพัฒนาขั้นตอนสานสัมพันธ์ (Nexus) ขึ้นมาใหม่ นำมาเป็นขั้นตอนก่อนการดำเนินการ (Pre-process) เพื่อทำให้สายโมทีฟที่เป็นไปได้ทั้งหมดมีความสัมพันธ์กันอย่างมีนัยสำคัญ พร้อมกับลดขนาดพื้นที่ของปัญหาในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสลง และนำมาประยุกต์ร่วมกับขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค [13] เรียกว่าขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาค (NexusPSO) และขั้นตอนวิธีที่สองออกแบบและพัฒนาเพื่อการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า โดยประยุกต์ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค [13] ร่วมกับแนวคิดระยะทางแฮมมิง [27] เรียกว่าขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคร่วมกับระยะทางแฮมมิง ซึ่งในขั้นตอนก่อนการดำเนินการ เป็นการนำแนวคิดระยะทางแฮมมิงมาคำนวณหาค่าความต่างของสายโมทีฟที่เป็นไปได้ทั้งหมดระหว่างสายข้อมูลที่นำเข้า โดยอาศัยหลักความสัมพันธ์ทางเคมีมาประกอบการคำนวณ [34] เพื่อให้พื้นที่ของปัญหามีความสัมพันธ์กันโดยมีค่าความต่างระหว่างสายโมทีฟประกอบอยู่ และนำแนวคิดเกณฑ์ขั้นต่ำผลติดลบที่ผิดพลาด (Cut-off Minimizing False Negative Rate: minFN) [35] เพื่อตรวจหาสายโมทีฟให้ครบถ้วน

1.2 วัตถุประสงค์ของงานวิจัย

1) สามารถระบุส่วนที่ยึดจับปัจจัยการถอดรหัสในกลุ่มข้อมูลสายดีเอ็นเอ แบบหนึ่งสายโมทีฟต่อสายดีเอ็นเออย่างแม่นยำ โดยประยุกต์ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค

ร่วมกับขั้นตอนก่อนการดำเนินการที่คิดขึ้นใหม่คือขั้นตอนสานสัมพันธ์ (Nexus) เรียกว่า ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาค (NexusPSO)

2) สามารถระบุส่วนที่ยึดจับปัจจัยการถอดรหัสในกลุ่มข้อมูลสายดีเอ็นเอ แบบหลายสายโมทีฟต่อสายดีเอ็นเออย่างครอบคลุมและแม่นยำ โดยประยุกต์ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคร่วมกับแนวคิดระยะทางแฮมมิง (Hamming Distances: HD) เรียกว่า ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคร่วมกับระยะทางแฮมมิง (PSO_HD)
3) ลดปัญหาการติดอยู่ในค่าที่ดีที่สุดเฉพาะที่ (Local Optimums) ของขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสในสายดีเอ็นเอ โดยประยุกต์และออกแบบขั้นตอนก่อนการดำเนินการอย่างเหมาะสมและมีประสิทธิภาพ

1.3 ขอบเขตของงานวิจัย

1) พัฒนาขั้นตอนวิธีโดยใช้ภาษา C# ร่วมกับฐานข้อมูลเชิงสัมพันธ์เพื่อแก้ปัญหาการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสในสายดีเอ็นเอ
2) ทดสอบขั้นตอนวิธีกับกลุ่มข้อมูลสายจีโนม 8 กลุ่ม ซึ่งกลุ่มข้อมูลเหล่านี้คือกลุ่มข้อมูลอักขระ ‘A’, ‘C’, ‘G’ และ ‘T’ พร้อมทั้งมีส่วนที่ยึดจับปัจจัยการถอดรหัสในสายดีเอ็นเอที่หลากหลาย
3) ผลลัพธ์ที่ได้จากขั้นตอนวิธีคือสายโมทีฟที่เป็นส่วนที่ยึดจับปัจจัยการถอดรหัสพร้อมกับลำดับเบสของสายโมทีฟนั้นๆ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถระบุส่วนที่ยึดจับปัจจัยการถอดรหัสในสายดีเอ็นเอที่หลากหลาย ได้อย่างแม่นยำและครอบคลุมด้วยขั้นตอนวิธีทางคอมพิวเตอร์
2) เพิ่มประสิทธิภาพการแก้ปัญหาของขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค ด้วยการจัดเตรียมข้อมูลโดยขั้นตอนก่อนการดำเนินการ (Pre-process)
3) สร้างแนวทางในการใช้ขั้นตอนวิธีแบบสุ่มในปัญหาการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสในสายดีเอ็นเอและปัญหาอื่นๆ ที่เกี่ยวข้อง

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ขั้นตอนวิธีการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบใช้สถิติประกอบด้วย ขั้นตอนวิธีสถิติแบบพื้นฐาน อาทิ ขั้นตอนวิธีค่าคาดหมายสูงสุด (Expectation-maximization: EM) ขั้นตอนวิธีการสุ่มตัวอย่างแบบกิบส์ (Gibbs Sampling) และขั้นตอนวิธีแบบเมตาฮิวริสติก (Metaheuristic) อาทิ ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization: PSO) ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดด้วยระบบอาณาจักรมด (Ant Colony Optimization: ACO) และขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm: GA) เป็นต้น

โดยขั้นตอนวิธีสถิติแบบพื้นฐานคือ การนำสายข้อมูลที่นำเข้ามาคำนวณจำนวนอักขระในตำแหน่งใดๆ หลังจากนั้นเป็นการคำนวณค่าความน่าจะเป็น ซึ่งนำเสนอในรูปแบบค่าน้ำหนักตำแหน่งแบบเมทริกซ์ (Position Weight Matrix: PWM) และใช้ค่าน้ำหนักตำแหน่งแบบเมทริกซ์ประกอบการคำนวณหาสายโมทีฟที่เป็นส่วนยึดจับปัจจัยการถอดรหัส ในขณะที่ขั้นตอนวิธีแบบเมตาฮิวริสติกเป็นการหาคำตอบที่อาศัยการสุ่ม โดยกำหนดให้มีอนุภาค (Particles) ที่สามารถเคลื่อนที่ในพื้นที่ของปัญหา (Problem Space) เพื่อค้นหาวิธีแก้ปัญหา (Solutions) ซึ่งในแต่รอบการทำงานกำหนดให้มีการใช้ข้อมูลที่จำเป็นร่วมกันระหว่างอนุภาค พร้อมกับการใช้ฟิตเนสฟังก์ชัน (Fitness Function) มาคำนวณผลลัพธ์ของแต่ละอนุภาคเพื่อระบุค่าฟิตเนสให้กับแต่ละอนุภาค แล้วนำค่าฟิตเนสทั้งหมดมาเป็นข้อมูลประกอบการเลือกอนุภาคที่ดีที่สุด เมื่อมีการใช้ข้อมูลร่วมกันระหว่างอนุภาคที่ดีที่สุดกับอนุภาคอื่นๆ ส่งผลให้อนุภาคทั้งหมดเคลื่อนที่ไปสู่ผลลัพธ์หรือวิธีการแก้ปัญหาที่ดีขึ้นเรื่อยๆ ในการทำงานแต่ละรอบ

2.1 นิยามของปัญหา

ส่วนที่ยึดจับปัจจัยการถอดรหัสประกอบด้วยสายโมทีฟใดๆ m_{ij} จำนวน k สาย (i และ j คือลำดับสายข้อมูลที่นำเข้าและลำดับของสายย่อยใดๆ ตามลำดับ) ซึ่งกรณีผลลัพธ์ที่มีหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้าแสดงดังรูปที่ 2.1 กำหนดให้

- S_i คือสายข้อมูลที่นำเข้าใดๆ จำนวน n สายและ $n > 0$
- w คือความยาวของสายโมทีฟที่เป็นไปได้ทั้งหมด
- l คือความยาวของสายข้อมูลที่นำเข้าคือ
- pk_{s_i} คือจำนวนสายโมทีฟที่เป็นไปได้ทั้งหมดในหนึ่งสายข้อมูลที่นำเข้า ($pk_{s_i} = l - w + 1$)
- k_{s_i} คือจำนวนสายโมทีฟต่อสายข้อมูลที่นำเข้า เงื่อนไขคือ $k_{s_i} = 1$

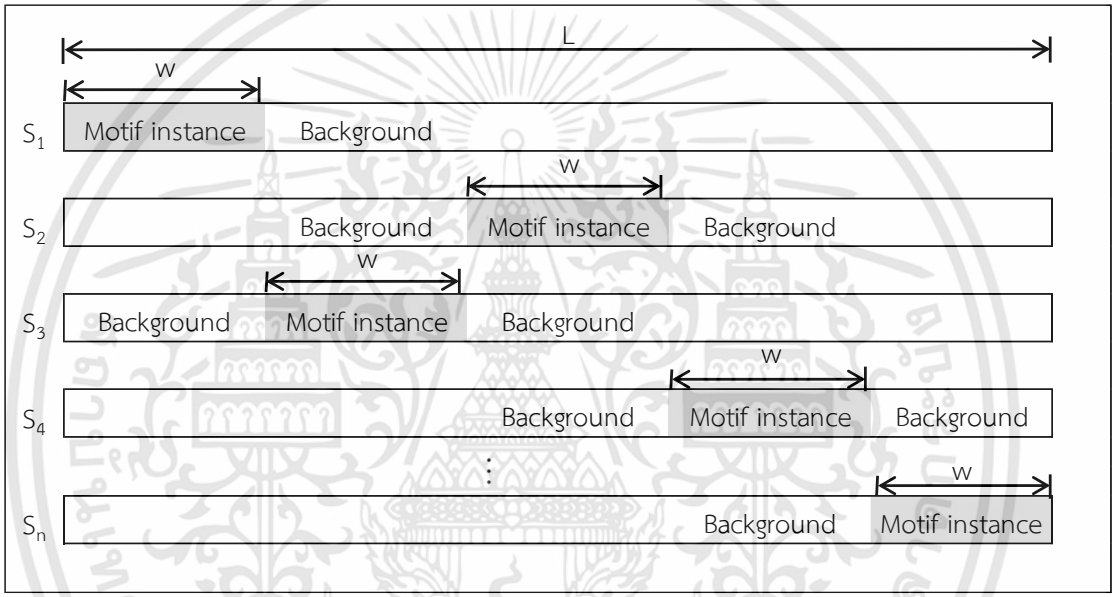
โดยขั้นตอนวิธีที่เกี่ยวข้อง ได้แก่ ขั้นตอนวิธีค่าคาดหมายสูงสุดกล่าวโดยละเอียดในหัวข้อ 2.2 ขั้นตอนวิธีการสุ่มตัวอย่างแบบกิบส์กล่าวโดยละเอียดในหัวข้อ 2.3 ขั้นตอนวิธีเชิงพันธุกรรมกล่าวโดยละเอียดในหัวข้อ 2.4 ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดด้วยระบบอาณาจักรมดกล่าวโดยละเอียดในหัวข้อ 2.5 และขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคกล่าวโดยละเอียดในหัวข้อ 2.6

ในกรณีส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า มีเงื่อนไขแสดงดังรูปที่ 2.2 กำหนดให้

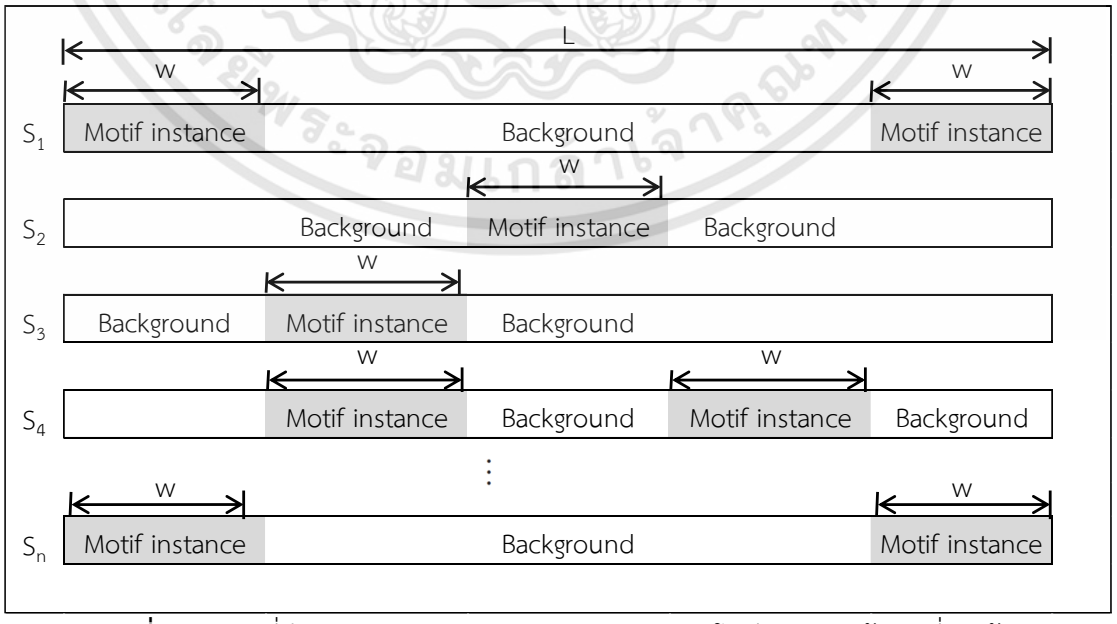
- k_{s_i} คือจำนวนสายโมทีฟต่อสายข้อมูลที่นำเข้า เงื่อนไขคือ $1 <= k_{s_i} < pk_{s_i}$
- pk_{s_i} คือจำนวนสายโมทีฟที่เป็นไปได้ทั้งหมดในหนึ่งสายข้อมูลที่นำเข้า เงื่อนไขคือ

  $pk_{s_i} = l - w + 1$

ซึ่งมีขั้นตอนวิธีที่เกี่ยวข้องได้แก่ ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบประยุกต์กล่าวโดยละเอียดในหัวข้อ 2.7 และขั้นตอนวิธีเชิงพันธุกรรมแบบประยุกต์กล่าวโดยละเอียดในหัวข้อ 2.8



รูปที่ 2.1 ส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า



รูปที่ 2.2 ส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า

2.2 ขั้นตอนวิธีค่าคาดหมายสูงสุด

ขั้นตอนวิธีค่าคาดหมายสูงสุด ประกอบด้วยสองขั้นตอนหลักประกอบด้วย ขั้นตอนการคาดหมาย (Expectation) และขั้นตอนหาค่าสูงสุด (Maximization) โดยขั้นตอนวิธีค่าคาดหมายสูงสุดถูกนำมาประยุกต์ใช้ในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัส [3] มีลำดับการทำงานดังต่อไปนี้

1) สร้างเมทริกซ์ค่าน้ำหนักของสายย่อยที่เป็นไปได้ทั้งหมด โดยคำนวณค่าน้ำหนักของอักขระ ‘A’, ‘C’, ‘G’ และ ‘T’ ของทุกสายย่อยซึ่งมีความยาวเท่ากับ l ในรูปแบบเมทริกซ์ l x 4
2) ดำเนินการในขั้นตอนการคาดหมาย โดยคำนวณค่าความถี่ของสายย่อยใดๆ x_i จากเมทริกซ์ค่าน้ำหนักของสายย่อยที่เป็นไปได้ทั้งหมด
3) ดำเนินการในขั้นตอนหาค่าสูงสุด โดยนำค่าความถี่ของสายย่อยใดๆ x_i ทั้งหมดมาทำการปรับปรุงเมทริกซ์ค่าน้ำหนัก กำหนดให้ค่าความถี่ของสายย่อยเป็นค่าน้ำหนัก (w_i) ซึ่งค่าน้ำหนักของตำแหน่งใดๆ Pos_{iN_j} ในเมทริกซ์คำนวณดังสมการที่ (2.1) และ N_j = {‘A’, ‘C’, ‘G’, ‘T’} ในกรณีที่ค่าน้ำหนักของเมทริกซ์ไม่เปลี่ยนแปลง กำหนดให้ $m_{ij_{best}}$ คือผลลัพธ์

$$Pos_{iN_j} = (w_1x_1 + w_2x_2 + \ldots + w_nx_n) / (w_1 + w_2 + \ldots + w_n) \qquad (2.1)$$

ตัวอย่างเช่น กำหนดให้มีสายข้อมูลที่นำเข้า 2 สาย n = 2 ได้แก่ GATTACA และ ACATTAG โดยความยาวของสายย่อย l = 4 ดังนั้นสายย่อยที่เป็นไปได้ทั้งหมดจึงประกอบด้วยสายย่อย 8 สายดังนี้ {m_{11}(GATT), m_{12}(ATTA), m_{13}(TTAC), m_{14}(TACA), m_{21}(ACAT), m_{22}(CATT), m_{23}(ATTA), m_{24}(TTAG)} มีลำดับการทำงานดังต่อไปนี้

1) สร้างเมทริกซ์ค่าน้ำหนักของสายย่อยที่เป็นไปได้ทั้งหมด โดยตำแหน่งแรกของอักขระ A มี 3 สาย อักขระ C มี 1 สาย อักขระ G มี 1 สายและอักขระ T มี 3 สายในสายย่อยทั้งหมด 8 สาย ดังนั้นค่าน้ำหนักตำแหน่งแรกของสายย่อยใดๆ จึงมีรายละเอียดการคำนวณดังต่อไปนี้ Pos_{1A}=3/8=0.375, Pos_{1C}=1/8=0.125, Pos_{1G}=1/8=0.125, Pos_{1T}=3/8=0.375 เมื่อคำนวณค่าน้ำหนักทุกตำแหน่งของทุกอักขระจึงสามารถแสดงเป็นเมทริกซ์ได้ดังนี้

Pos.	*A*	*C*	*G*	*T*
1	0.375	0.125	0.125	0.375
2	0.375	0.125	0	0.500
3	0.375	0.125	0	0.500
4	0.375	0.125	0.125	0.375

2) ดำเนินการในขั้นตอนการคาดหมาย โดยการคำนวณค่าความถี่ของอักขระในสายย่อยที่เป็นไปได้ทั้งหมด m_{ij} คิดจากสัดส่วนระหว่างความน่าจะเป็นสายโมทีฟ $P_{m_{ij}}$ ต่อความไม่น่าจะเป็นสายโมทีฟ $P_{m_{ij}^{de}}$ ซึ่งค่าความไม่น่าจะเป็นของอักขระใดๆ {‘A’, ‘C’, ‘G’, ‘T’} คือ 0.25 [3]

ดังนั้น ค่าความถี่ของอักขระในสายย่อย $P_{m_{11}(GATT)}$ = (0.125 x 0.375 x 0.5 x 0.375) / (0.25 x 0.25 x 0.25 x 0.25) = 2.25 และค่าความถี่ของสายย่อยอื่นๆ ที่อยู่ในสายข้อมูลที่นำเข้าสายแรกประกอบด้วย $P_{m_{12}}$ = 9.00, $P_{m_{13}}$ = 2.25, $P_{m_{14}}$ = 1.69 และค่าความถี่ของสายย่อยที่อยู่ในสายข้อมูลที่นำเข้าสายที่สองได้แก่ $P_{m_{21}}$ = 1.69, $P_{m_{22}}$ = 2.25, $P_{m_{23}}$ = 9.00, $P_{m_{24}}$ = 2.25

3) ดำเนินการในขั้นตอนหาค่าสูงสุด โดยนำค่าความถี่ของสายย่อย m_{ij} ทั้งหมดมาทำการปรับปรุงเมทริกซ์ค่าน้ำหนัก อาทิ การปรับปรุงเมทริกซ์ค่าน้ำหนักของอักขระ ‘A’ ตำแหน่งแรกของสายย่อยใดๆ จากสายข้อมูลที่นำเข้าทั้งหมด คำนวณได้ดังนี้ Pos_{1A} = ($P_{m_{12}}$+ $P_{m_{21}}$+ $P_{m_{23}}$) / ($P_{m_{11}}$+ $P_{m_{12}}$+ $P_{m_{13}}$+ $P_{m_{14}}$+ $P_{m_{21}}$+ $P_{m_{22}}$+ $P_{m_{23}}$+ $P_{m_{24}}$) = (9.00+ 1.69+ 9) / (2.25+9.00+2.25+1.69+1.69+2.25+9.00+2.25) = 0.648 เมื่อคำนวณตำแหน่งและอักขระอื่นๆ สามารถแสดงเป็นเมทริกซ์ได้ดังนี้

Pos.	*A*	*C*	*G*	*T*
1	0.648	0.074	0.074	0.204
2	0.204	0.056	0	0.741
3	0.204	0.056	0	0.741
4	0.648	0.074	0.074	0.204

2.3 ขั้นตอนวิธีการสุ่มตัวอย่างแบบกิบส์

ขั้นตอนวิธีแบบกิบส์ใช้วิธีการสร้างกลุ่มตัวอย่างจากการสุ่มประชากรทั้งหมด แล้วคำนวณความถี่ของกลุ่มตัวอย่างซึ่งนำเสนอในรูปแบบเมทริกซ์ค่าน้ำหนัก เพื่อเป็นข้อมูลประกอบการคัดเลือกผลลัพธ์ในแต่ละรอบการทำงาน [2] โดยสามารถประยุกต์ขั้นตอนวิธีนี้ในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัส [22][23] ตามขั้นตอนการทำงานดังต่อไปนี้

1) ทำการสุ่มสายย่อยใดๆ m_{ij} ที่มีความยาวเท่าๆ กัน w จากสายข้อมูลที่นำเข้า S_i
2) นำสายย่อยที่ได้จากการสุ่มมาคำนวณความถี่อักขระ ‘A’, ‘C’, ‘G’ และ ‘T’ และนำเสนอในรูปเมทริกซ์ค่าน้ำหนัก w x 4
3) คำนวณหาสายย่อยที่มีความถี่สูงสุด $m_{ij_{best}}$ จากเมทริกซ์ในข้อ 2) (ดำเนินการในเฉพาะสายข้อมูลที่นำเข้า i) ในกรณีที่เมทริกซ์ค่าน้ำหนักไม่มีการเปลี่ยนแปลงกำหนดให้ $m_{ij_{best}}$ คือผลลัพธ์
4) นำ m_{ij} ที่มีความถี่สูงสุดมาปรับปรุงเมทริกซ์ค่าน้ำหนัก
5) เพิ่มค่า i+1 โดย i+1 <= n ในกรณีที่ i+1 > n กำหนดให้ i = 1 แล้วดำเนินการในขั้นตอนที่ 3) จนกระทั่งเมทริกซ์ค่าน้ำหนักไม่มีการเปลี่ยนแปลง

ตัวอย่างเช่น กำหนดให้มีสายข้อมูลที่นำเข้า 4 สาย n = 4 ได้แก่ GATTACA, GATTACA, ACATTAG และ ACATTAG โดยความยาวของสายโมทีฟ w เท่ากับ 4 มีลำดับการทำงานดังต่อไปนี้

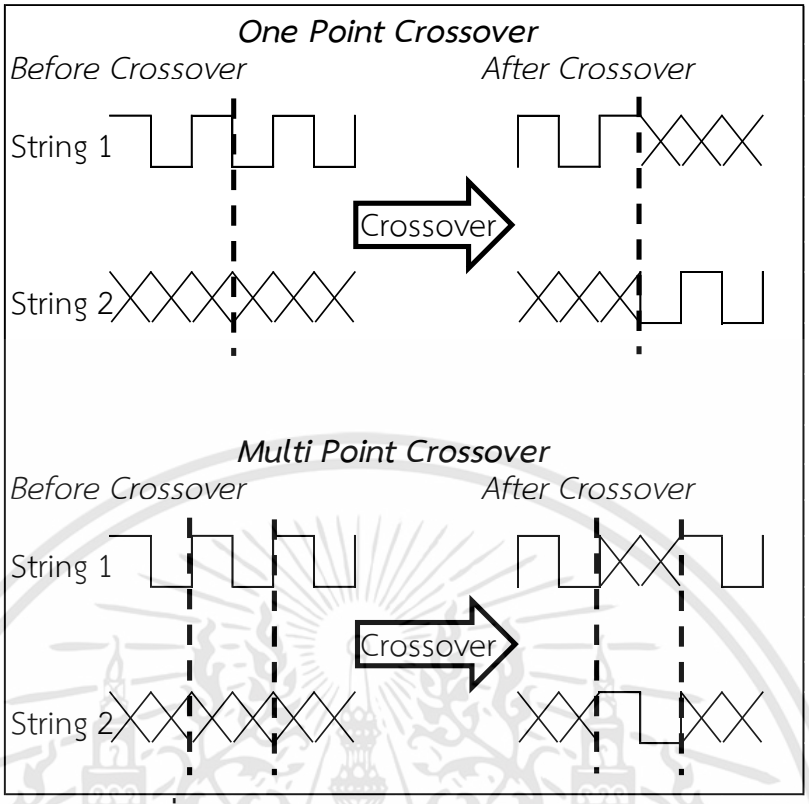
1) ทำการสุ่มสายย่อยใดๆ m_{ij} จากสายข้อมูลที่นำเข้าแต่ละสาย ประกอบด้วย m_{11} = GATT, m_{24} = TACA, m_{33} = ATTA และ m_{42} = CATT และกำหนดค่า i=1
2) สร้างเมทริกซ์ค่าน้ำหนักของสายย่อยจากสายข้อมูลที่นำเข้า i สายที่ 2, 3 และ 4 (ไม่คำนวณสายย่อย i ปัจจุบัน) ประกอบด้วย $TACA_{(m_{24})}$, $ATTA_{(m_{33})}$ และ $CATT_{(m_{42})}$ โดยค่าน้ำหนักของอักขระ 'A' ตำแหน่งแรกของสายย่อยใดๆ คำนวณได้ดังนี้ Pos_{1A} = (1/3) = 0.333, Pos_{1C} = (1/3) =0.333, Pos_{1G} = (0/3) = 0.000 และ Pos_{1T} = (1/3) = 0.333 เมื่อคำนวณทุกตำแหน่งสามารถแสดงผลลัพธ์ได้ดังนี้

Pos.	*A*	*C*	*G*	*T*
1	0.333	0.333	0	0.333
2	0.667	0	0	0.333
3	0	0.333	0	0.667
4	0.667	0	0	0.333

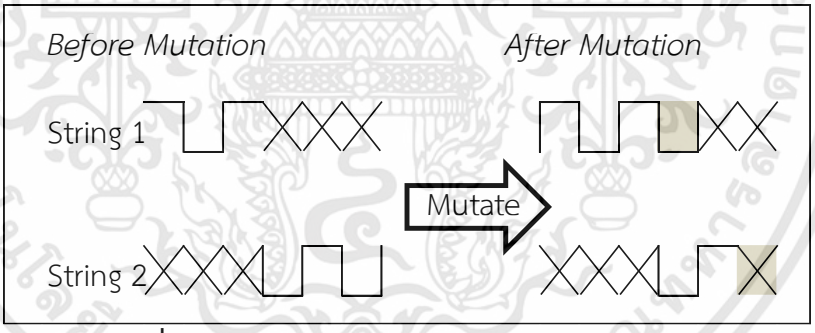
3) คำนวณหาสายย่อยที่ m_{1j} ที่มีความถี่สูงสุดในสายข้อมูลที่นำเข้าสายแรก "GATTACA" อาทิ m_{12} = (0.333 x 0.333 x 0.667 x 0.667) / (0.25 x 0.25 x 0.25 x 0.25) = 12.642 ดังนั้นความถี่ของสายย่อย $m_{i=1j}$ = (0.000, 12.642, 0.000, 12.642)
4) แทนที่สายย่อย GATT ในสายย่อยที่ได้จากการสุ่มลำดับที่ i ด้วยสายย่อยที่มีความถี่สูงสุด $m_{ij_{best}}$, m_{12} = ATTA หรือ m_{14} = TACA
5) ดำเนินการในขั้นตอนที่ 2 อีกครั้ง โดย i = i+1 ถ้า i >= n กำหนดให้ i = 0 ดำเนินการจนกระทั่งเมทริกซ์ค่าน้ำหนักไม่มีการเปลี่ยนแปลง

2.4 ขั้นตอนวิธีเชิงพันธุกรรม

สำหรับการทำงานของขั้นตอนวิธีเชิงพันธุกรรม คิดค้นโดย J.H. Holland ในปี 1975 [24] เริ่มต้นด้วยการสุ่มตัวอย่าง คล้ายกับขั้นตอนวิธีการสุ่มตัวอย่างแบบกิบส์ แต่มีขั้นตอนการดำเนินงานที่แตกต่างกัน โดยขั้นตอนวิธีนี้ใช้การสร้างกลุ่มประชากร (ประชากรเปรียบเหมือนกับวิธีการแก้ปัญหา) ในพื้นที่ของปัญหา ซึ่งการสร้างกลุ่มประชากรแต่ละรุ่น (Generation) กำหนดให้มีการเลือกประชากรชุดเดิมซึ่งการคัดเลือกใช้วิธีการสุ่มเลือกโดยประเมินจากค่าฟิตเนส ต่อไปเป็นการจับคู่ระหว่างประชากรที่ถูกเลือกเพื่อทำการผสมยีน (Crossover) ซึ่งเป็นการนำเอาโครโมโซมมาผสมกันเพื่อให้เกิดลักษณะของประชากรที่แตกต่างในรุ่นถัดมา โดยการผสมยีนมีทั้งแบบจุดเดียว (One Point Crossover) และแบบหลายจุด (Muti-point Crossover) ดังรูปที่ 2.3 หลังจากนั้นเป็นขั้นตอนการกลายพันธุ์ (Mutation) ดังรูปที่ 2.4 ซึ่งเป็นการเลียนแบบกระบวนการวิวัฒนาการตามธรรมชาติ โดยวัตถุประสงค์หลักของขั้นตอนวิธีเชิงพันธุกรรมคือ การสร้างวิธีการแก้ปัญหาที่ดียิ่งขึ้นในแต่ละรอบการทำงาน ซึ่งเกิดจากการเลือกวิธีการแก้ปัญหาที่ดีที่สุดมาผสมรายละเอียดวิธีการแก้ปัญหา เพื่อให้เกิดวิธีการแก้ปัญหาใหม่ที่ดียิ่งขึ้น



รูปที่ 2.3 การผสมยีนแบบจุดเดียวและแบบสองจุด



รูปที่ 2.4 การกลายพันธุ์แบบจุดเดียวและแบบสองจุด

เมื่อนำขั้นตอนวิธีเชิงพันธุกรรมมาประยุกต์เพื่อการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแล้ว [25] สามารถกำหนดตัวแปรได้ดังต่อไปนี้

- m_{ij} คือกลุ่มของข้อมูลสายย่อยที่เป็นได้ทั้งหมดจากแต่ละสายข้อมูลที่นำเข้า
- P_i คือประชากรใดๆ กำหนดให้ $P_i = \{M_{1j}, M_{2j}, \ldots M_{n-1j}, M_{nj}\}$
- w คือความกว้างของสายโมทีฟ
- G คือจำนวนรุ่นหรือจำนวนรอบการทำงานของขั้นตอนวิธี
- S คือค่าการขยับตำแหน่ง (Shift Range)
- x คือจำนวนประชากรทั้งหมดในการทำงานรอบที่แล้ว

เนื่องจากประชากรที่ถูกสร้างขึ้นในแต่ระรอบมีจำนวน 2^*x ด้วยเหตุนี้จึงจำเป็นต้องมีขบวนการที่ช่วยควบคุมจำนวนประชากรทั้งหมด ซึ่งขั้นตอนนี้เรียกว่าขั้นตอนการนำประชากรรุ่นใหม่แทนที่

ประชากรกลุ่มเดิม (Replace Worst Individuals) โดยทำการตัดประชากรกลุ่มเดิมที่มีคุณภาพต่ำ (ค่าฟิตเนสต่ำ) ออกจากกลุ่ม โดยจำนวนประชากรที่ตัดออกในแต่ละรอบการทำงาน คิดเป็นหนึ่งในสามของประชากรทั้งหมด โดยลำดับการดำเนินงานของขั้นตอนวิธีมีดังต่อไปนี้

ขั้นตอนวิธีเชิงพันธุกรรม

1. Initialize parameters; //กำหนดค่าพารามิเตอร์เริ่มต้น
2. Initialize population; //สร้างประชากรจากการสุ่ม
3. EVALUATE each candidate //วิเคราะห์ประชากรแต่ละตัว

   while (-*S* <=*i*<= *S*)

   SET all start positions; //ระบุตำแหน่งเริ่มต้นของสายย่อยที่ได้จากการสุ่ม

   OBTAIN aligned motifs based on start positions; //สร้างตำแหน่ง

   SET fitness value; //คำนวณค่าฟิตเนส

   if (Maximum fitness < current fitness) //เงื่อนไขเปรียบเทียบค่าฟิตเนส

   Maximum fitness = current fitness; //กำหนดค่าฟิตเนสปัจจุบันให้เป็นค่าฟิตเนสสูงสุด

   end if

   end while

   RETURN Maximum fitness; //ส่งตำแหน่งของสายย่อยที่มีค่าฟิตเนสสูงสุดกลับ

   end evaluate
4. while (iteration number = *G*)

   Select parents by roulette wheel selection; //สุ่มเลือกประชากร

   Crossover; //ทำการผสมยีน

   Mutate; //กลายพันธุ์ยีน

   Evaluate new candidates; //วิเคราะห์ประชากรที่ถูกสร้างใหม่ด้วยฟิตเนสฟังก์ชัน

   Replace worst individuals; //นำประชากรรุ่นใหม่แทนที่ประชากรเดิม

   end while
5. Output predicted motifs; //ส่งออกผลลัพธ์ส่วนที่ยึดจับปัจจัยการถอดรหัส

2.5 ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดด้วยระบบอาณาจักรมด

ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดด้วยระบบอาณาจักรมด เป็นขั้นตอนวิธีแบบฮิวริสติกที่มีการแลกเปลี่ยนข้อมูลระหว่างอนุภาคย่อยเพื่อสร้างเส้นทางชุดคำตอบหรือวิธีแก้ปัญหา ที่ส่งผลให้ทุก

อนุภาคสามารถตรวจหาวิธีแก้ปัญหาได้ดีมากขึ้นในแต่ละรอบ พื้นฐานการทำงานของขั้นตอนวิธีนี้พัฒนามาจากการค้นหาอาหารในอาณาจักรของมด และสร้างเส้นทางเพื่อนำอาหารกลับไปที่รัง โดยฝูงมดใช้การทิ้งฟีโรโมน (Pheromone) ตามเส้นทางการเดิน ซึ่งฟีโรโมนบนเส้นทางการเดินสามารถมีความเข้มข้นขึ้นเรื่อยๆ ตามปริมาณและคุณภาพอาหารที่ถูกขนส่งผ่านเส้นทางนั้นๆ ในทางกลับกันความเข้มข้นของฟีโรโมนสามารถลดลงได้ในกรณีที่ปริมาณและคุณภาพอาหารลดลง สามารถวิเคราะห์ได้ดังสมการที่ (2.2) [26] โดยค่าฟีโรโมนสามารถส่งผลต่อการเลือกเส้นทางเดินของอนุภาคใดๆ ได้โดยใช้การคำนวณตามสมการที่ (2.4) [26] ประกอบการพิจารณา

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} \tag{2.2}$$

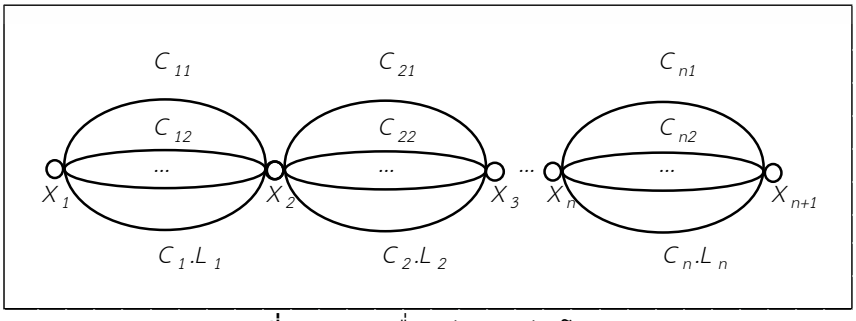
เงื่อนไขคือ $\rho \in (0,1)$ ซึ่งทำให้ค่าฟีโรโมน τ_{ij} ลดลงและ t คือรอบการทำงานแต่ละรอบ โดยการเพิ่มขึ้นของค่าฟีโรโมน $\Delta\tau_{ij}$ คำนวณตามสมการที่ (2.3)

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{2.3}$$

กำหนดให้ $\Delta\tau_{ij}^{k}$ คือค่าข้อมูลเส้นทางการเดินของแต่ละอนุภาค k ในจำนวนอนุภาคทั้งหมด m ซึ่ง i และ j คือเส้นทางการเดิน โดยเงื่อนไขในการกำหนดค่าเส้นทางการเดินคือ เมื่อเส้นทาง i, j ใดถูกเลือกโดยอนุภาค k กำหนดให้เส้นทางนั้นเท่ากับอัตราส่วนของค่าคงที่กับระยะทางการเดินของมดแต่ละตัว ในขณะที่เส้นทางที่ไม่ถูกเลือกกำหนดให้มีค่าเป็น 0

$$P_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{k=1}^{Li}[\tau_{ik}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}} \tag{2.4}$$

สำหรับสมการคำนวณค่าความน่าจะเป็นของเส้นทางการเดิน $P_{ij}(t)$ กำหนดให้ $n_{ij}(t)$ คือฟังก์ชันฮิวริสติก ซึ่งสามารถออกแบบได้ตามลักษณะของการแก้ปัญหา α, β คือค่าที่กำหนดขึ้นมาเพื่อส่งผลต่อข้อมูลเส้นทางการเดิน τ_{ij} และค่าที่ได้จากฟังก์ชันฮิวริสติก n_{ij} เมื่อนำมาประยุกต์กับการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัส กำหนดให้ x_i คือโหนดที่ถูกเชื่อมต่อโดยเส้นเชื่อม C_{ij} (C_{ij} เปรียบเทียบได้กับตำแหน่งเริ่มต้นของสายย่อยใดๆ m_{ij}) ดังรูปที่ 2.5



รูปที่ 2.5 การเชื่อมต่อระหว่างโหนด

ซึ่งจำนวนโหนดทั้งหมดคือ n+1 และกำหนดให้จำนวนเส้นเชื่อมระหว่างโหนด $C_n = l_i$-w+1 (l_i, w คือความยาวของสายข้อมูลที่นำเข้าและความยาวของสายย่อยตามลำดับ) เมื่อขั้นตอนวิธีได้ผลลัพธ์เส้นทางการเดินของอนุภาคใดๆ C_{ij} แล้ว หลังจากนั้นเป็นการตรวจหากลุ่มของสายโมทีฟ $M(J)$ ซึ่งเป็นส่วนที่ยึดจับปัจจัยการถอดรหัส โดยนำ C_{ij} ใดๆ มาเป็นข้อมูลในการตรวจหา โดยลำดับการทำงานของขั้นตอนวิธีเมื่อนำมาประยุกต์กับการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสมีดังต่อไปนี้

ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดด้วยระบบอาณาจักรมด

1. Initialize parameters; //กำหนดค่าพารามิเตอร์เริ่มต้น

   X_s,X_2,..., X_n; คือกลุ่มของสายข้อมูลที่นำเข้า

   maxnum; คือจำนวนรอบการทำงานทั้งหมด

   m; จำนวนของมดที่ใช้

2. Begin

```
for t=1 to maxnum do
        for k=1 to m do
            for i=1 to n do
                ant k selects the edge Cij; //มดเลือกเส้นเชื่อม
            end for i
            get the solution J={j1,j2,...jn}; //ดึงข้อมูลสายโมทีฟ
            IC(M(J)); //คำนวณค่าฟิตเนสโดยสมการ IC กลุ่มสายโมทีฟ
            if IC(M(J)) > IC(Mbest) then
                    Mbest=M(J);
                    Jbest=J;
            end if
        end for k
        Update the pheromone;
 end for t
```

3. Output predicted motifs; //ส่งออกผลลัพธ์ส่วนที่ยึดจับปัจจัยการถอดรหัส

2.6 ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค

ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization: PSO) คือขั้นตอนวิธีแบบเมตาฮิวริสติก ซึ่งประยุกต์จากรูปแบบการเคลื่อนที่ของฝูงนกเพื่อแก้ปัญหาที่ไม่สามารถจัดการได้ในเวลาที่จำกัด [13] โดยขั้นตอนการทำงาน เริ่มต้นจากการสร้างอนุภาค แล้วกำหนดความเร็วให้กับอนุภาคและใช้ฟิตเนสฟังก์ชัน (Fitness Function) เพื่อคำนวณหาอนุภาคที่ดีที่สุด โดยอนุภาคที่ดีที่สุดมีสองประเภทได้แก่ อนุภาคที่ดีที่สุดในการทำงานแต่ละรอบ (Local Best: x_i^{lb}) และอนุภาคที่ดีที่สุดแบบภาพรวม (Global Best: x_i^{nb}) โดยความเร็วของแต่ละอนุภาค $v_i^{(t+1)}$

แปรตามระยะห่างจาก x_i^{lb} และ x_i^{nb} ตามสมการที่ (2.5) ที่มีค่าพารามิเตอร์ปัจจัยความเฉื่อย (Inertia Parameter) α ที่ควบคุมความเร็วของอนุภาคในรอบการทำงานที่ผ่านมา $v_i^{(t)}$ ค่าพารามิเตอร์ความแปรปรวน (Cognitive Parameter) β ที่ควบคุมผลกระทบจาก x_i^{lb} และค่าพารามิเตอร์สังคม (Social Parameter) ที่ควบคุมผลกระทบจาก x_i^{nb} พร้อมกับค่า r_1 และ r_2 ที่เป็นค่าที่ได้จากการสุ่มระหว่าง 0 ถึง 1 โดยตำแหน่งของแต่ละอนุภาคมีการปรับเปลี่ยนไปเรื่อยๆ ในแต่ละรอบของการทำงาน t คำนวณตามสมการที่ (2.6)

$$v_i^{(t+1)} = \alpha v_i^{(t)} + r_1\beta(x_i^{lb} - x_i^{(t)}) + r_2\gamma(x_i^{nb} - x_i^{(t)}) \quad (2.5)$$

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \quad (2.6)$$

การเคลื่อนที่ของฝูงแบบไบนารี (Binary Particle Swarm Optimization: BPSO) ถูกพัฒนาขึ้นมาเพื่อเป็นหลักการสำหรับการเคลื่อนที่ของอนุภาคได้อย่างมีประสิทธิภาพ [27][28][29] ซึ่งมีลำดับการทำงานดังต่อไปนี้

1) คำนวณหาความเร็ว v_{ij} ของอนุภาค P_i ดังสมการที่ (2.5)
2) คำนวณค่า $S(v_{ij}(t+1))$ ตามสมการ (2.7) เพื่อเปรียบเทียบกับ r_3 (r_3 คือค่าที่ได้จากการสุ่มระหว่าง 0 ถึง 1)

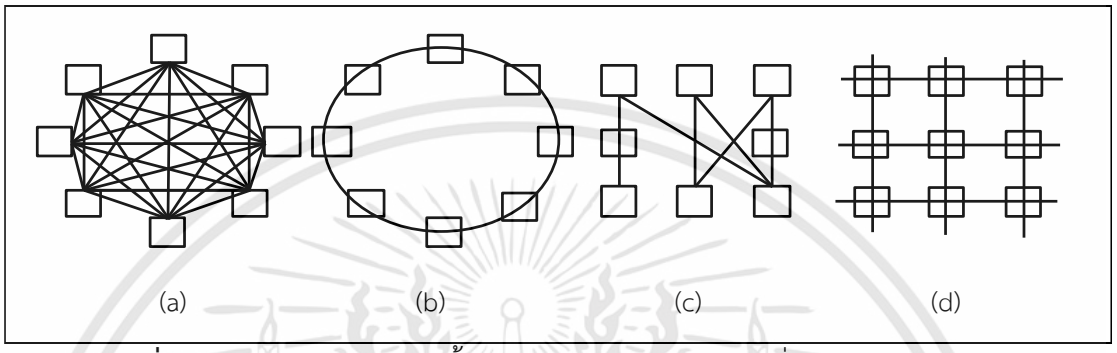
$$S(v_{ij}(t+1)) = \frac{1}{(1+e^{v_{ij}(t+1)})} \quad (2.7)$$

3) เปรียบเทียบ r_3 กับ $S(v_{ij}(t+1))$ ถ้า $r_3 > S(v_{ij}(t+1))$ ให้มีการแทนที่ตำแหน่ง x_{ij} ด้วย x_{bestij} ในกรณีที่ $r_3 <= S(v_{ij}(t+1))$ กำหนดไม่ให้มีการเปลี่ยนแปลงตำแหน่ง x_{ij}

สำหรับวัตถุประสงค์ของการสร้างเครือข่ายในขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคคือ การใช้ข้อมูลร่วมกันระหว่างอนุภาคตามรูปแบบโครงสร้างเครือข่ายของฝูง (Topology) เพื่อให้แต่ละอนุภาคเคลื่อนที่ไปสู่ตำแหน่งที่ดีขึ้น ซึ่งเป็นการใช้ข้อมูลร่วมกันระหว่างอนุภาคที่ดีที่สุด ณ เวลาใดๆ P_{best} กับอนุภาคเพื่อนบ้าน (Neighborhood Particles) P_i โดยรูปแบบเครือข่ายของฝูง [30] มีดังต่อไปนี้

1) GBest: เป็นเครือข่ายแบบอนุภาคเชื่อมโยงหากันทั้งหมด ดังนั้นในแต่ละอนุภาคจึงมีจำนวนอนุภาคเพื่อนบ้าน (Particle's Neighbor) คือ C_P -1 โดย C_P คือจำนวนอนุภาคทั้งหมดในฝูง ดังรูปที่ 2.6(a)
2) Bidirectional Ring: เป็นเครือข่ายแบบวงแหวนโดยอนุภาคเชื่อมโยงกันคล้ายรูปวงแหวนมีอนุภาคเพื่อนบ้าน 2 อนุภาคคืออนุภาค P_i - 1 และ P_i + 1 เมื่อ i คืออนุภาคปัจจุบัน ดังรูปที่ 2.6(b)

3) Random: เป็นเครือข่ายแบบสุ่มโดยอนุภาคเชื่อมโยงแบบไม่เป็นโครงสร้าง เนื่องจากแต่ละอนุภาคเลือกเพื่อนบ้านด้วยการสุ่ม ซึ่งเงื่อนไขของจำนวนเพื่อนบ้านคือ $0 < C_n <= C_p - 1$ เมื่อ C_n คือจำนวนเพื่อนบ้าน ดังรูปที่ 2.6(c)
4) Von Neumann: เป็นเครือข่ายแบบสี่เหลี่ยมโดยอนุภาคเชื่อมโยงเป็นรูปสี่เหลี่ยม (Lattice Structure) มีอนุภาคเพื่อนบ้าน 4 อนุภาคประกอบด้วย อนุภาค P_{i-1} ด้านซ้าย อนุภาค P_{i+1} ด้านขวา อนุภาค P_{si-1} ด้านบน และอนุภาค P_{si+1} ด้านล่าง ดังรูปที่ 2.6(d)



รูปที่ 2.6 เครือข่ายของฝูง (a) ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (b) เครือข่ายแบบวงแหวน (c) เครือข่ายแบบสุ่ม (d) เครือข่ายแบบสี่เหลี่ยม

เมื่อนำขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคมาใช้เพื่อการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัส [16] โดยกำหนดให้แต่ละอนุภาค P_i ประกอบด้วยเวกเตอร์ของตำแหน่ง x_i และความเร็ว v_i พร้อมทั้งกำหนดให้เวกเตอร์ $x_i = \{x_1, x_2,..., x_n\}$ และเวกเตอร์ $v_i = \{v_1, v_2,..., v_n\}$ กำหนดให้ n คือจำนวนสายข้อมูลที่นำเข้าในแต่ละรอบการทำงานของอนุภาค P_i มีการเปลี่ยนแปลงตำแหน่งของสายย่อย x_i ในพื้นที่ของปัญหา ซึ่งมีขั้นตอนการทำงานดังต่อไปนี้

ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค

```
k = 0;
Begin
while (k < numMotifs) //ดำเนินการทีละสายย่อย
  Initialize parameters;
  Initialize connections;
  while (l < numIteration) //ดำเนินการจนครบจำนวนรอบการทำงานที่กำหนด
     Update Vi; //ปรับความเร็วของแต่ละอนุภาค
     Update Xi; //ปรับตำแหน่งของอนุภาค
     Update Particle's fitness; //ปรับค่าฟิตเนสของอนุภาค
     Select best particle (Local best, Global best); //เลือกอนุภาคที่ดีที่สุด
```

$l = l + 1$;

end while

$k = k + 1$;

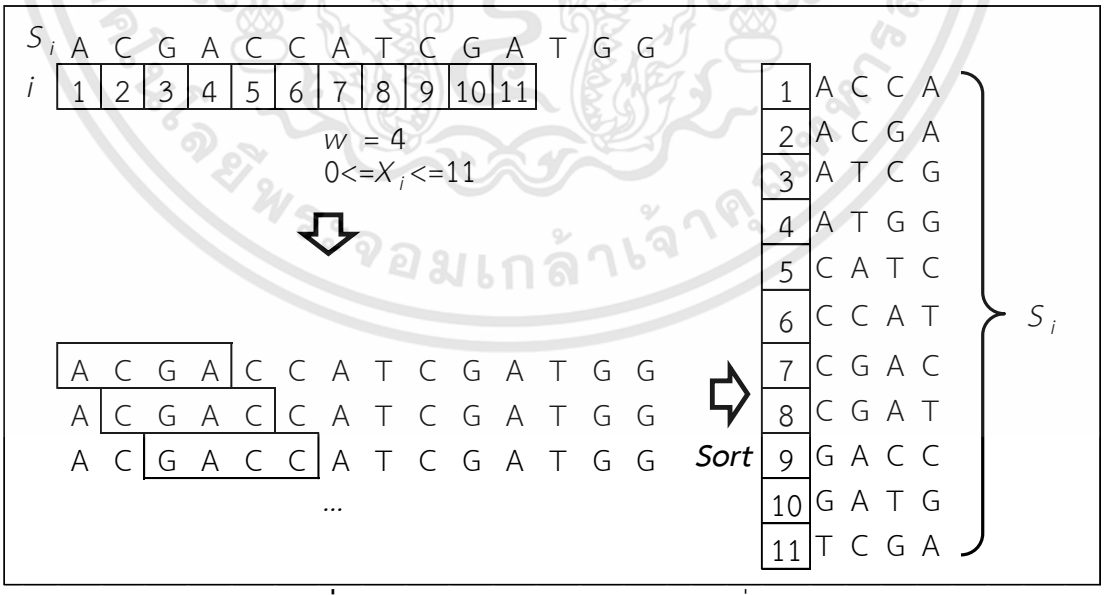
end while k

Output predicted motifs; //ส่งออกผลลัพธ์ส่วนที่ยึดจับปัจจัยการถอดรหัส

2.7 ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบประยุกต์

ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคถูกนำมาประยุกต์ร่วมกับขั้นตอนก่อนการดำเนินการ (Pre-process) และหลังการดำเนินการ (Post-process) [17] เพื่อการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า

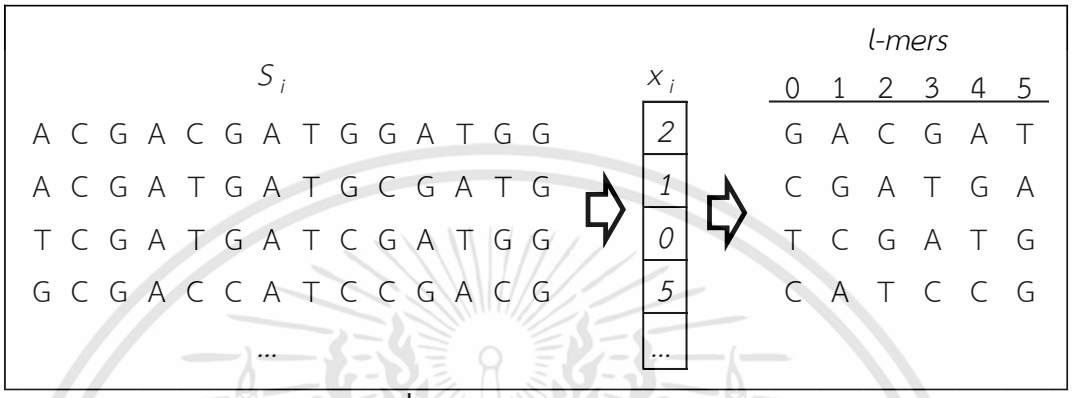
สำหรับความไม่ต่อเนื่องในสายข้อมูลที่นำเข้า ส่งผลให้การเคลื่อนที่ของอนุภาคขาดความต่อเนื่องด้วย อาทิ สายข้อมูลที่นำเข้า S_i = ACGACCATCGATGG และความยาวสายโมทีฟ w = 4 โดยจำนวนสายย่อยที่เป็นไปได้ทั้งหมดคือ 11 สาย ในกรณีที่ตำแหน่งปัจจุบันของอนุภาคอยู่ในลำดับที่ 7 ($m_{ij=7}$ สายย่อยคือ ATCG) ต้องการปรับตำแหน่งให้ไปอยู่ในลำดับที่ 8 ($m_{ij=8}$ สายย่อยคือ TCGA) นั้นสามารถสังเกตได้ว่าสายย่อยทั้งสองสายนี้ไม่มีคุณลักษณะที่ใกล้เคียงกัน (Common Pattern) ดังนั้น จึงได้มีการพัฒนาขั้นตอนก่อนการดำเนินการ เป็นการจัดการพื้นที่ของปัญหาในสายข้อมูลที่นำเข้าเพื่อให้มีความต่อเนื่อง โดยการปรับคุณลักษณะสายข้อมูลที่นำเข้า S_i ให้กลายเป็นพื้นที่ของปัญหาที่มีความลาดชันอย่างต่อเนื่อง (Gradient Distribution) ด้วยวิธีการเรียงลำดับ (Sorting) ตัวอักษรของสายย่อยที่เป็นไปได้ทั้งหมด ดังรูปที่ 2.7



รูปที่ 2.7 การปรับคุณลักษณะสายข้อมูลที่นำเข้า

ดังนั้นอนุภาค P_i จึงประกอบด้วย

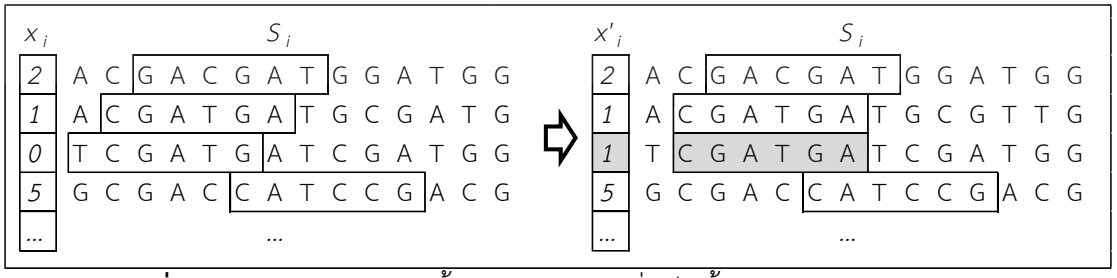
- *l-mers* คือสายย่อยใหม่แสดงดังรูปที่ 2.8 ซึ่งเกิดจากการสุ่มข้อมูลจากสายข้อมูลที่นำเข้า S_i
- P_i ประกอบด้วยเวกเตอร์ x_i และ v_i เพื่อควบคุมตำแหน่งและความเร็วของอนุภาค
- x_i คือเวกเตอร์ของตำแหน่งของอนุภาค
- v_i คือเวกเตอร์ของความเร็วของอนุภาค



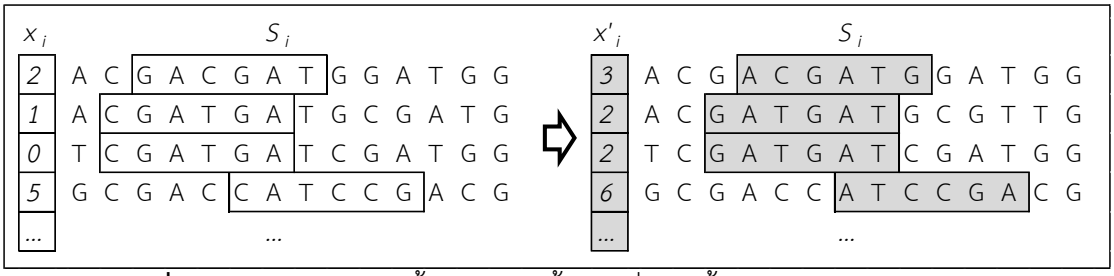
รูปที่ 2.8 การสร้างสายย่อยใหม่

เมื่อดำเนินการไปเรื่อยๆ จนกระทั่งตำแหน่งของทุกอนุภาคอยู่ในตำแหน่งเดียวกัน สามารถระบุได้ว่าตำแหน่งนั้นคือผลลัพธ์ ในกรณีที่ครบรอบการทำงานที่กำหนดแล้ว แต่ทุกอนุภาคไม่อยู่ในตำแหน่งเดียวกัน ขั้นตอนวิธีกำหนดให้ตำแหน่งของอนุภาคที่มีค่าฟิสเนตสูงสุดเป็นตำแหน่งผลลัพธ์ พร้อมกับอีกหนึ่งขบวนการที่สำคัญคือ กรณีที่อนุภาคที่ดีที่สุดเป็นอนุภาคเดิมเกิน 10 รอบ ขั้นตอนวิธีนี้กำหนดให้มีการเปลี่ยนรูปแบบสายย่อยในแต่ละอนุภาค (Perturb Particles)

อย่างไรก็ตาม ปัญหาการติดอยู่ในค่าที่ดีที่สุดเฉพาะที่ยังคงเกิดขึ้น ขั้นตอนวิธีนี้จึงออกแบบขั้นตอนหลังการดำเนินการเพื่อพัฒนาผลลัพธ์ให้มีความแม่นยำมากขึ้น ซึ่งในขั้นตอนหลังการดำเนินการประกอบด้วยสองขั้นตอนหลัก คือ ขั้นตอนจัดทำใหม่ (Re-alignment) และขั้นตอนปรับตำแหน่งทั้งหมด (Simultaneous) สำหรับขั้นตอนจัดทำใหม่คือการปรับตำแหน่งของสายโมทีฟไปด้านซ้ายหรือขวา j ตำแหน่ง โดยที่ $0 <= j <= L_i\text{-}w+1$ (L_i, w คือความยาวสายข้อมูลที่เข้าและความยาวของสายโมทีฟตามลำดับ) ดังตัวอย่างในรูปที่ 2.9 โดยทำการคัดเลือกสายโมทีฟที่ให้ค่าฟิตเนสของอนุภาคสูงสุดเป็นผลลัพธ์สุดท้าย ในขณะที่ขั้นตอนปรับตำแหน่งทั้งหมดคือการปรับสายโมทีฟทั้งหมดไปด้านซ้ายหรือขวาดังตัวอย่างในรูปที่ 2.10



รูปที่ 2.9 การปรับผลลัพธ์ในขั้นตอนจัดทำใหม่ซึ่งเป็นขั้นตอนการหลังดำเนินการ



รูปที่ 2.10 ปรับผลลัพธ์ในขั้นตอนปรับทั้งหมดซึ่งเป็นขั้นตอนการหลังดำเนินการ

โดยเงื่อนไขในการปรับตำแหน่งมีดังต่อไปนี้

- ในกรณีที่ X_{best} ตัวปัจจุบันมีค่ามากกว่า X_{best}^{right} หรือ X_{best}^{left} กำหนด X_{best} เป็นผลลัพธ์ของขั้นตอนวิธี
- กรณีที่ X_{best}^{right} หรือ X_{best}^{left} มีค่ามากกว่า X_{best} ปัจจุบัน กำหนดให้แทนที่ X_{best}^{right} หรือ X_{best}^{left} ที่มีค่าสูงสุดเป็น X_{best} และดำเนินการไปเรื่อยๆ จนกระทั่งค่าฟิตเนสของอนุภาคไม่สูงขึ้น
- $X_{best}^{right} = \{x_1+1, x_2+1, x_3+1,... x_{n-1}+1, x_n+1\}$
- $x_{best}^{left} = \{x_1-1, x_2-1, x_3-1,... x_{n-1}-1, x_n-1\}$

ซึ่งขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบประยุกต์มีลำดับการทำงาน (Pseudocode) ดังต่อไปนี้

ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบประยุกต์

```
Initialize parameters //กำหนดค่าพารามิเตอร์
    Load data and transform data S_i into S_i^c ; //แปลงข้อมูลให้ต่อเนื่อง
    k = 0;
    while (k < numMotifs)
        Initialize particles; //สร้างหน่วยย่อย
        Initialize connections; //สร้างความสัมพันธ์ระหว่างหน่วยย่อย
        i = 0;
        while (i < maxIterations OR not converged)
            Update particles' velocities; //ปรับความเร็วของอนุภาค
            Update particles' positions; //ปรับตำแหน่งของอนุภาค
            Update particles' fitnesses; //ปรับค่าฟิตเนสให้กับอนุภาค
            Select local best particle; // เลือกหน่วยย่อยที่ดีที่สุด
        If (best fitness stagnate for 10 itertions)
```

Perturb particles; //การเปลี่ยนรูปแบบสายย่อยในแต่ละอนุภาค

Initialize connections; //สร้างความสัมพันธ์ระหว่างหน่วยย่อยอีกครั้ง

end if

i = *i*+1;

end while *i*

Post-process; //ขั้นตอนหลังดำเนินการ

Add best particle to the output list; //จัดเก็บผลลัพธ์ที่ดีที่สุดไว้ในรายการเอาท์พุต

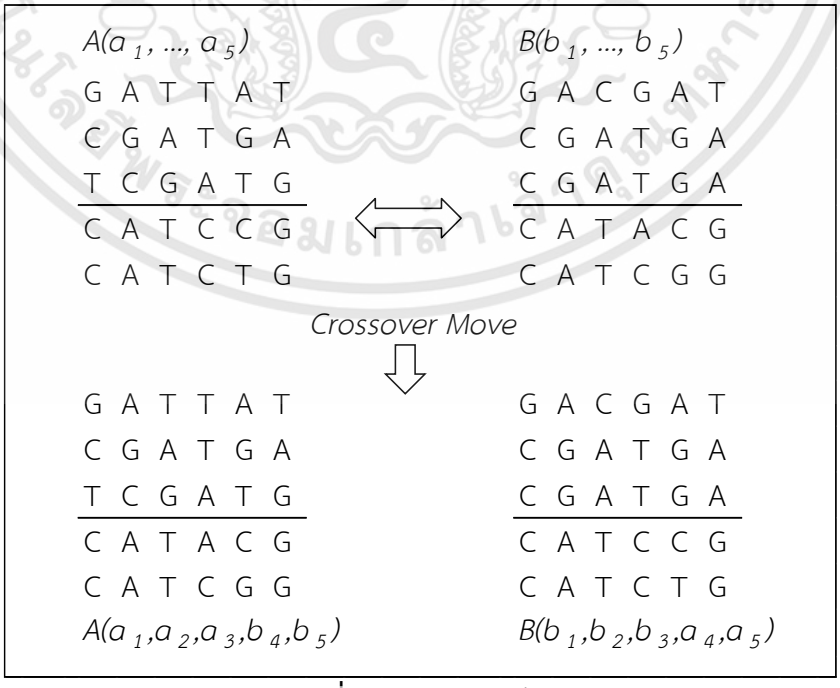
k = *k* + 1;

end while *k*

Output predicted motifs; //ส่งออกผลลัพธ์ส่วนที่ยึดจับปัจจัยการถอดรหัส

2.8 ขั้นตอนวิธีเชิงพันธุกรรมแบบประยุกต์

สำหรับการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้าได้มีการพยายามพัฒนาความแม่นยำและครอบคลุมในการตรวจหา โดยประยุกต์ขั้นตอนวิธีเชิงพันธุกรรมเพื่อช่วยในการตรวจหา [18] โดย Z. Wei และ S.T. Jensen ในปี 2006 ซึ่งมีขั้นตอนสำคัญหลักๆ อยู่สองขั้นตอนประกอบด้วย ขั้นตอนการผสมยีน (Crossover) และขั้นตอนการกลายพันธุ์ (Mutation) แสดงดังรูปที่ 2.11



รูปที่ 2.11 การผสมยีน

ขั้นตอนวิธีเชิงพันธุกรรมแบบประยุกต์กำหนดให้หน่วยย่อยใดๆ A ประกอบด้วยเวกเตอร์ $A(a_1, a_2, ..., a_{m-1}, a_m)$ เมื่อทำการผสมยีนระหว่าง $A(a_1, a_2, ..., a_{m-1}, a_m)$ และ $B(b_1, b_2, ..., b_{m-1}, b_m)$ โดยผลลัพธ์กำหนดให้มีการสุ่มตัวเลขตัวแหน่งที่ต้องการสลับที่ c ซึ่ง $0< c <= m$ โดยผลลัพธ์ที่ได้จากขั้นตอนการผสมยีนคือ $A'(a_1, a_2, ..., a_c, b_{c-1},..., b_{m-1}, b_m)$ และ $B'(b_1, b_2,... , b_c, a_{c-1},..., a_{m-1}, a_m)$ สำหรับขั้นตอนการกลายพันธุ์คือการเปลี่ยนแปลงบางตำแหน่งในสายโมทีฟ ที่เป็นผลลัพธ์ดังตัวอย่างในรูปที่ 2.12 โดยขั้นตอนวิธีดำเนินการไปเรื่อยๆ จนกระทั่งครบจำนวนรอบที่กำหนดหรือค่าฟิตเนสของหน่วยย่อยมีค่าคงที่

รูปที่ 2.12 การกลายพันธุ์

สำหรับรายละเอียดการทำงานของขั้นตอนวิธีมีดังต่อไปนี้

ขั้นตอนวิธีเชิงพันธุกรรมแบบประยุกต์

Initialization: i = 0

Setting parameters: //กำหนดค่าพารามิเตอร์

population size N = 500;

mutation rate r = 0.001;

maximum generation G = 3000;

Generating initial population P_0;

Repeat: $i = i + 1$;

Mutate individuals; //ดำเนินการขั้นตอนกลายพันธุ์

Crossover individuals; //ดำเนินการขั้นตอนการผสมยีน

Selection of individuals; //เลือกอนุภาคที่ดีที่สุด

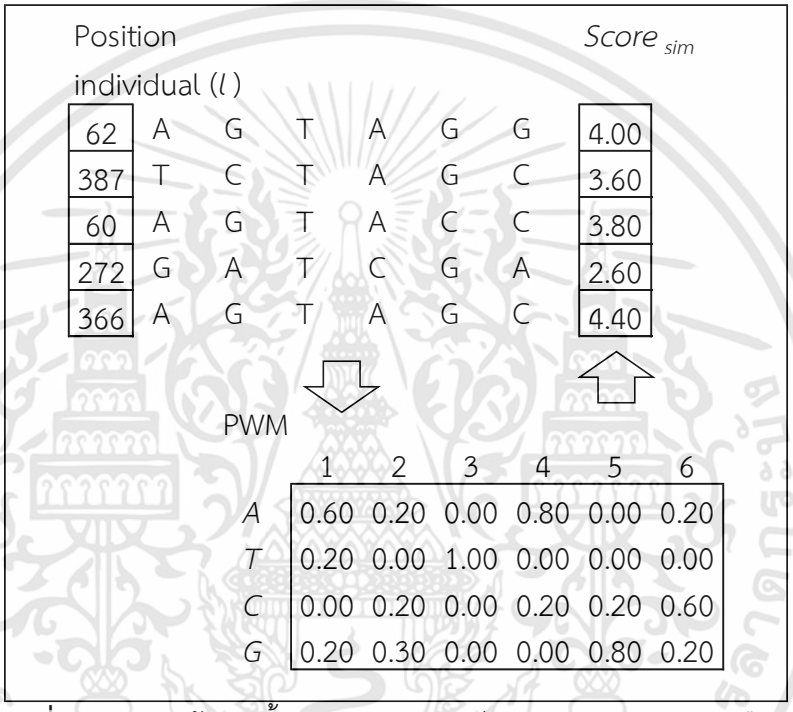
Until ($i >= G$ or convergence)

Choose best individual A_{opt}; //เลือกผลลัพธ์ที่ดีที่สุด

Output predicted motifs; //ส่งออกผลลัพธ์ส่วนที่ยึดจับปัจจัยการถอดรหัส

*หมายเหตุ: กำหนดให้ A_{opt} กลายเป็นผลเมื่อผลลัพธ์ไม่ดีขึ้นใน 50 รอบการทำงาน

ขั้นตอนก่อนการดำเนินการ (Pre-process) ถูกนำมาประยุกต์ในขั้นตอนวิธีเชิงพันธุกรรม อาทิ แนวคิดการกรองข้อมูล (Filtering) [19] และแนวคิดการประยุกต์นำข้อมูลองค์ความรู้เบื้องต้นของสายดีเอ็นเอมาประยุกต์ใช้ (Prior Knowledge) [20] สำหรับแนวคิดการกรองข้อมูลมีวัตถุประสงค์เพื่อการตัดข้อมูลผลบวกที่ผิดพลาดออก (False Positive) โดยอาศัยการคำนวณค่าความเหมือน ($Score_{sim}$) เพื่อเป็นข้อมูลประกอบการพิจารณาตัดผลบวกที่ผิดพลาด อันดับแรกคือการสร้างอนุภาคใดๆ (Individual: *I*) ที่ประกอบด้วยสายย่อยใดๆ p_i ที่ได้จากการสุ่ม $I = \{p_1, p_2, ..., p_N\}$ โดยที่ N คือจำนวนสายย่อยทั้งหมดที่ได้จากการสุ่ม นำมาสร้างค่าน้ำหนักแบบเมทริกซ์ แล้วนำค่าน้ำหนักในเมทริกซ์มาคำนวณค่าความเหมือนแสดงดังรูปที่ 2.13



รูปที่ 2.13 การสร้างค่าน้ำหนักแบบเมทริกซ์และคำนวณค่าความเหมือน

หลังจากนั้นทำการตรวจหาสายย่อยใหม่ที่ทำให้ค่าความเหมือนของสายย่อยในอนุภาค *I* สูงขึ้น $p_{Rnk(k)}$ (กำหนดให้สามารถตรวจหาสายย่อยใหม่ได้ในเฉพาะสายข้อมูลที่นำเข้าเดียวกับ S_i เท่านั้น) ซึ่งลำดับการทำงานอย่างละเอียดของแนวคิดการกรองข้อมูลมีดังต่อไปนี้

การกรองข้อมูล

Input: Individual $I = \{p_1, p_2, ..., p_N\}$ //สร้างหน่วยย่อยจากการสุ่มสายย่อย

Filtering (*I*)

Sort all the instances of *I* by $Score_{sim}$; //คำนวณค่าความเหมือน

Rnk(1), *Rnk*(2), ..., *Rnk*(*N*); //นำสายย่อยมาเรียงลำดับตามค่าความเหมือนจากมากไปน้อย

For ($k = N$; k<=2; *I*--)

Scan sequence R*nk*(*k*) to get $q_{Rnk(k)}$ with best $Score'_{sim}$; //ตรวจหาสายย่อยที่ทำให้ค่าความเหมือนสูงขึ้น

$p_{Rnk(k)} = q_{Rnk(k)}$;
If ($Score_{Sim}(p_{Rnk(k)}) <= Score_{sim}(p_{Rnk(k-1)})$) //เงื่อนไขในการหยุดทำงาน
Return the new *I*;
Output: new *I*;

สำหรับขั้นตอนก่อนการดำเนินการอีกประเภทหนึ่งคือ การนำข้อมูลองค์ความรู้เบื้องต้นของสายดีเอ็นเอที่ต้องการทดลอง มาประยุกต์ใช้เพื่อการลดข้อมูลที่รบกวน (Background Noise) ให้น้อยลง โดยเอาข้อมูลเบื้องต้นที่ถูกเตรียมไว้แล้ว ซึ่งนำเสนอในรูปแบบเมทริกซ์ความถี่ (Position Frequency Matrix: PFM) ที่สามารถบอกความถี่ของอักขระต่างๆ อย่างมีนัยสำคัญ [31][32][33] โดยแนวคิดที่เกี่ยวข้องกับการใช้ข้อมูลองค์ความรู้เบื้องต้นประกอบด้วย การสร้าง *k-mer* (*k-mer* คือสายย่อยที่มีขนาดความยาว *k*) อาทิ $T_1, T_2, ..., T_k$ เงื่อนไขคือ $T_j \in \sum$ = {A, C, G, T} และ j = 1, 2, ..., k โดยนำเสนอในรูปแบบเมทริกซ์ที่เข้ารหัสแล้ว e(k-mer) = $[a_{ij}]_{4xk}$, a_{ij} = 1 ถ้า $T_j = V_i$ นอกนั้นกำหนดให้เป็น 0 เงื่อนไขคือ (V_1, V_2, V_3, V_4) = (A, C, G, T) อาทิเช่น 7-mer ของสายข้อมูล AGCGTGT สามารถเข้ารหัสได้ดังต่อไปนี้

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

และการสร้างเมทริกซ์ความถี่ ใช้รูปแบบในการสร้างดังสมการที่ (2.8)

$$M = \frac{1}{|S|} \sum_{K_p \in S} e(K_p) \tag{2.8}$$

กำหนดให้ $|S|$ คือค่าที่ถูกกำหนดไว้ในกลุ่มข้อมูล เป็นค่าที่มีความสัมพันธ์กับส่วนที่ยึดจับปัจจัยการถอดรหัส K_p คือกลุ่มข้อมูลสายย่อย สำหรับกระบวนการจัดการข้อมูลที่รบกวนประกอบด้วยสองขั้นตอน ขั้นตอนแรกคือการหาค่ารัศมี (Maximum Radius) ซึ่งคำนวณจากสมการระยะทางแฮมมิง (d) ดังสมการที่ (2.9)

$$\delta = \max_{K_p \in S} d(K_p, \frac{1}{|S|} \sum_{K_p \in S} K_p) \tag{2.9}$$

และขั้นตอนที่สองคือการตัดข้อมูลที่รบกวน (Rule-based Filter) โดยเงื่อนไขมีดังสมการที่ (2.10)

$$d(K, \frac{1}{|S|} \sum_{K_p \in S} K_p) > \delta \tag{2.10}$$

ถ้าเป็นจริงตามเงื่อนไขนี้กำหนดให้ไม่ต้องพิจารณาสายย่อย K (ตัดสายย่อย K ออก)

ขั้นตอนหลังดำเนินการ (Post-process) เป็นขั้นตอนที่สำคัญสำหรับการพัฒนาผลลัพธ์ของขั้นตอนวิธีเชิงพันธุกรรมให้มีความแม่นยำและครอบคลุมมากขึ้น โดยเฉพาะอย่างยิ่งในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสในสายดีเอ็นเอที่มีความหลากหลายของคุณลักษณะ ซึ่งมีการประยุกต์ปรับปรุงขั้นตอนหลังดำเนินการในหลายรูปแบบ อาทิ การปรับลำดับสายโมทีฟในผลลัพธ์ (Adjust) และการเลื่อนตำแหน่ง (Shift) ที่ประยุกต์ในขั้นตอนวิธีจีเอเอ็มอี (GAME) [18] การลบและเพิ่มผลลัพธ์ (I_{best}) ที่ประยุกต์ในขั้นตอนวิธีจีเอแอลเอฟพี (GALF-P) [19] สำหรับการปรับลำดับสายโมทีฟในผลลัพธ์คือการเลื่อนตำแหน่งสายย่อย a_i ในสายข้อมูลที่นำเข้า S_i เงื่อนไขการเลื่อนตำแหน่งของสายย่อยคือ $1 <= a_i <= l_i - w + 1$ โดยลำดับการทำงานของการปรับลำดับสายโมทีฟในผลลัพธ์มีรายละเอียดการทำงานดังต่อไปนี้

การปรับลำดับสายโมทีฟในผลลัพธ์

Adjust: $A(a_1,...,a_i,...,a_m)$; // กลุ่มข้อมูลสายย่อยที่ต้องการปรับตำแหน่ง

$i = 1$;

DO

a'_i = argmax $f(a_1,...,a_i,...,a_m)$, $1 <= a_i <= l_i-w+1$ //ขยับสายย่อยทีละสายเพื่อหาผลลัพธ์สูงสุด

$a_i = a'_i$; // เก็บสายย่อยที่ให้ค่าฟิตเนสสูงสุดเป็นผลลัพธ์ปัจจุบัน

$i = i+1$;

if $i > m$ then $i = 1$

UNTIL no further improvements obtained //ดำเนินการจนกระทั่งผลลัพธ์ไม่มีการเปลี่ยนแปลง

ในขณะที่การเลื่อนตำแหน่งของผลลัพธ์เป็นการเลื่อนตำแหน่งของสายย่อย a_i ทั้งหมด ซึ่งมีรายละเอียดการทำงานดังต่อไปนี้

การเลื่อนตำแหน่ง

Shift: ($A(a_1,...,a_i,...,a_m)$)

k'= argmax $f(a_1 + k,...,a_m + k)$, $-w <= k <= w$; //ขยับสายย่อยทั้งหมดเพื่อหาผลลัพธ์สูงสุด

$A = A\ (a + k',...,a_m + k')$; //เก็บผลลัพธ์ที่ให้ค่าฟิตเนสสูงสุดเป็นผลลัพธ์ปัจจุบัน

Note 1: if $a_i = 0$ then $a_i + k = 0$ for all k (no added sites) //ถ้าสายย่อย a_i =0 กำหนดให้ไม่มีการเคลื่อนที่

Note 2: If $a_i + k < 0$ or $a_i + k > l_i - w + 1$, then set $a_i + k = 0$ //ถ้าสายย่อย $a_i + k < 0$ หรือมากกว่า $li - w + 1$ กำหนดให้ $a_i = 0$

สำหรับการเพิ่มผลลัพธ์และตัดจากผลลัพธ์ของขั้นตอนวิธี $I_{best} = \{m_1, m_2,..., m_N\}$ เป็นการตรวจหาสายโมทีฟที่เป็นผลบวกที่ผิดพลาด (False Positive) เพิ่มเติมพร้อมกับตัดสายโมทีฟที่ไม่มีนัยสำคัญออกไป การเพิ่มผลลัพธ์เป็นการตรวจหาสายโมทีฟ *M'* ใดๆ โดยเงื่อนไขคือ สายโมทีฟ *M'* นั้นๆ ต้องทำผลลัพธ์ให้มีค่าฟิตเนสสูงขึ้นตามที่กำหนด ในขณะที่การตัดสายย่อยคือการลบสายย่อย m_i ออกเพื่อทำให้ผลลัพธ์มีค่าฟิตเนสสูงขึ้นตามที่กำหนด โดยลำดับการทำงานอย่างละเอียดมีดังต่อไปนี้

การเพิ่ม/การตัดผลลัพธ์

//Adding: ขั้นตอนการตรวจหาสายย่อยเพิ่ม

$I_{best} = \{m_1, m_2, ..., m_N\}$; //ผลลัพธ์จากขั้นตอนวิธี

Calculate $IC'_{I_{best}}$ with peudo-counts; //คำนวนค่าฟิตเนส

$\epsilon_0 = \beta * w$;

$\delta = -\epsilon_0$; //กำหนดค่ามาตรฐาน

$DIF = \max_{m_{i,k}} (IC'_{m_{i,k}} - IC'_{I_{best}})$; //คำนวณหาค่าความต่างสูงสุดระหว่างค่าฟิตเนสของสายย่อยอื่นๆ $m_{i,k}$ และค่าฟิตเนสของผลลัพธ์ I_{best}

If ($DIF <= \epsilon_0$)

$r = 0$;

Return $I_{best}+ = I_{best}$; //กำหนดให้ไม่จำเป็นต้องเพิ่มสายย่อยใดๆ เพื่อเป็นผลลัพธ์

end if

while ($\delta < \epsilon_0$)

$M' = \{m_{i,k} \mid m_{i,k} != m_i, IC'_{m_{i,k}} > IC'_{I_{best}} + \delta\}$; //คำนวณหา $m_{i,k}$ ที่ให้ค่าความต่างระหว่างค่าฟิตเนสของผลลัพธ์ I_{best} กับค่าฟิตเนสของผลลัพธ์ที่รวมสายย่อยอื่นๆ แล้ว

$\delta = \text{avg}_{m_{i,k} \in M'} (IC'_{m_{i,k}} - IC'_{I_{best}})$; //คำนวณความต่างระหว่างฟิตเนสของ $m_{i,k} \in M'$ และ I_{best}

end while

$r = |M'|$;

Return $I_{best}+ = I_{best} \cup M'$; //สร้างผลลัพธ์โดยนำ I_{best} ยูเนียนกับ *M'*

//Removing Stage: ขั้นตอนการตัดสายย่อยออกจากผลลัพธ์

$I_{best}- = I_{best}+$; //นำผลลัพธ์จากขั้นตอนการตรวจหาสายย่อยเพิ่ม $I_{best}+$ เก็บในตัวแปร $I_{best}-$

$\epsilon'_0 = \max(\delta, \beta * w * r, \beta * w)$; //หาค่ามาตรฐานสูงสุด

$\delta' = \epsilon'_0$; //กำหนดค่ามาตรฐานสูงสุดไว้ที่ตัวแปร δ'

while (not converged) //ดำเนินการจนกระทั่งผลลัพธ์ไม่มีการเปลี่ยนแปลง

Calculate $IC'_{I_{best}-}$ of $I_{best}-$ with peudo - counts; //คำนวณค่าฟิตเนสของ $I_{best}-$

M' = {$m_{i,j}$| $m_{i,j} \in I_{best}^-$, $IC'_{-mi,j} > IC'_{I_{best}^-} + \delta'$}; //ตรวจหาสายย่อย $-m_{i,j}$ ที่ถูกตัดออกแล้วทำให้ค่าฟิตเนสของ I_{best}^- เพิ่มขึ้น

If (M' = null)

Return I_{best}^-; //กำหนดให้ I_{best}^- เป็นผลลัพธ์ในกรณีที่ M' เป็นค่าว่าง

end if

DIF' = $\max_{m_{i,j} \in M'}$ ($IC'_{-m_{i,j}} - IC'_{I_{best}^-}$); //คำนวณหาความต่างที่มากที่สุดระหว่างค่าฟิตเนสของ $-m_{i,j}$ และ I_{best}

I_{best}^- = I_{best}^- – {the instance corresponding to DIF'}; //กำหนดให้ผลลัพธ์คือ I_{best}^- ลบสายย่อย $m_{i,j}$ ที่ทำให้เกิดค่า DIF'

δ' = (ϵ'_0 + DIF')/2; //ปรับค่ามาตรฐานสูงสุด

end while

2.9 ระยะทางแฮมมิง

ระยะทางแฮมมิง (Hamming Distance: *HD*) คือระยะห่างระหว่างสองไบนารีเวกเตอร์ (Vectors of Binary) ซึ่งมีการนำมาประยุกต์เพื่อการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัส คิดค้นโดย M. Stine และคณะในปี 2003 [27] เป็นการคำนวณด้วยการเปรียบเทียบไบนารีที่อยู่ในตำแหน่งเดียวกันระหว่างสองเวกเตอร์ แล้วทำการนับจำนวนไบนารีที่แตกต่างกัน หลักการของระยะทางแฮมมิงคือการนำเวกเตอร์ X และ เวกเตอร์ Y มาหาผลต่าง $\sum|X_i - Y_i|$ กำหนดให้ X_i, Y_i คือไบนารีใดๆ ของเวกเตอร์ X และ Y ตามลำดับ และความกว้างของเวกเตอร์ X และ Y มีขนาดเท่ากันคือ L ดังนั้น $HD(X, Y) > 0$ ในกรณี X ไม่เท่ากับ Y และ $HD(X, Y) = 0$ ในกรณี X เท่ากับ Y

ตัวอย่างเช่น ถ้าต้องการหาระยะทางแฮมมิงที่สั้นที่สุด (Minimum Distance) ระหว่างเวกเตอร์ {*a*, *b*, *c*, *d*}

กำหนดให้แต่ละเวกเตอร์มีค่าดังต่อไปนี้

a = (11111)

b = (01001)

c = (10100)

d = (00010)

$HD(a, b)$ =3, $HD(a, c)$ =3, $HD(a, d)$ =4, $HD(b, c)$ =4, $HD(b, d)$ =3, $HD(c, d)$ =3

ดังนั้นระยะทางแฮมมิงที่สั้นที่สุดของเวกเตอร์ {*a*, *b*, *c*, *d*} คือ HD = 3

สำหรับการประยุกต์ระยะทางแฮมมิงกับปัญหาการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัส กำหนดให้ความยาวของสายย่อย w คือความกว้างของเวกเตอร์ L โดยแต่ละนิวคลีโอไทด์ (Nucleotides) A, C, G และ T ถูกแปลงให้มีขนาดสองบิต A = 00, T = 01, G = 10, T = 11 ตาม

เงื่อนไขความสัมพันธ์ทางเคมี (Chemical Relationship) [34] ซึ่งระยะทางแฮมมิงระหว่างอักขระแสดงในตารางที่ 2.1 สำหรับตัวอย่างการแปลงไบนารีเวกเตอร์ *V* = <1110010010> ให้กลายเป็นอักขระโดยพิจารณาจากตารางที่ 2.1 ผลลัพธ์การแปลงไบนารีของเวกเตอร์ *V* คือ <CGTAG>

ตารางที่ 2.1 การแปลงรหัสให้เป็นสองบิตเพื่อคำนวณระยะทางแฮมมิง

Rows		Columns			
		A	T	G	C
		00	01	10	11
A	00	0	1	1	2
T	01	1	0	2	1
G	10	1	2	0	1
C	11	2	1	1	0

2.10 คะแนนความคล้ายแบบเมทริกซ์

คะแนนความคล้ายแบบเมทริกซ์ (Matrix Similarity Score: *mSS*) คือคะแนนที่วัดความเหมือนระหว่างสายย่อยที่อยู่ในเมทริกซ์ [35] ซึ่งเป็นการคำนวณทุกหลักที่อยู่ในเมทริกซ์ โดยมีวิธีการคำนวณดังสมการที่ (2.11)

$$mSS = \frac{Current - Min}{Max - Min} \tag{2.11}$$

กำหนดให้

- $Current$ คือค่าเมทริกซ์ปัจจุบัน
- Min คือค่าความถี่ต่ำสุดของอักขระในตำแหน่ง i
- Max คือค่าความถี่สูงสุดของอักขระในตำแหน่ง i

สำหรับค่าเมทริกซ์ปัจจุบัน ($Current$) เกิดจากการนำอไลเมนที่เป็นผลลัพธ์จากกลุ่มสายโมทีฟนำมาสร้างเมทริกซ์ แล้วทำการคำนวณดังสมการที่ (2.12)

$$Current : \sum_{i=1}^{L} I(i) f_{i,b_i} \tag{2.12}$$

กำหนดให้

- L คือความยาวของสายโมทีฟ
- f_{i,b_i} คือจำนวนอักขระ b_i ที่ปรากฏในตำแหน่ง i โดยที่ $b \in$ {‘A’, ‘C’, ‘G’, ‘T’}
- $I(i)$ คือค่าข้อมูลเวกเตอร์ซึ่งคำนวณดังสมการที่ (2.15)

ค่าความถี่ต่ำสุด (Min) เกิดจากการคำนวณความถี่ของอักขระที่มีจำนวนน้อยที่สุดในแต่ละตำแหน่ง i ในสายโมทีฟ มีวิธีการคำนวณดังสมการที่ (2.13)

$$Min: \sum_{i=1}^{L} I(i) f_i^{\min} \tag{2.13}$$

กำหนดให้

- f_i^{min} คือจำนวนที่น้อยที่สุดของอักขระในตำแหน่ง i

ค่าความถี่สูงสุด (Max) เกิดจากการคำนวณความถี่ของอักขระที่มีจำนวนมากที่สุดในแต่ละตำแหน่ง i ในสายโมทีฟ มีวิธีการคำนวณดังสมการที่ (2.14)

$$Max: \sum_{i=1}^{L} I(i) f_i^{\max} \tag{2.14}$$

กำหนดให้

- f_i^{max} คือจำนวนที่มากที่สุดของอักขระในตำแหน่ง i

ค่าข้อมูลเวกเตอร์ ($I(i)$) เกิดจากการคำนวณความถี่ของทุกอักขระที่ไม่อยู่ในเมทริกซ์ปัจจุบันมีวิธีการคำนวณดังสมการที่ (2.14)

$$I(i) = \sum_{B \in \{A,T,G,C\}} f_{i,B} \ln(4 f_{i,B}), \quad i = 1, 2, \ldots, L \tag{2.15}$$

กำหนดให้

- $f_{i,B}$ คือจำนวนของอักขระ ‘A’, ‘C’,‘G’, ‘T’ ที่ไม่อยู่ในเมทริกซ์หรืออไลเมนปัจจุบัน

2.11 ฟิตเนสฟังก์ชัน (Fitness Function)

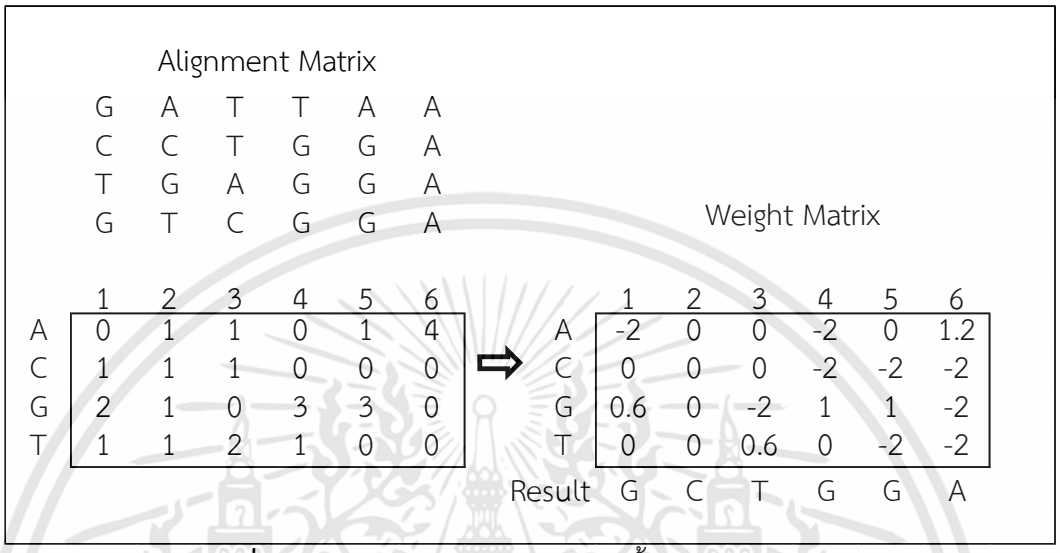
ขั้นตอนวิธีในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัส มีกระบวนการหนึ่งที่สำคัญคือการเลือกอนุภาคที่ดีที่สุดหรือเหมาะสมที่สุด ซึ่งใช้การวิเคราะห์จากค่าฟิตเนส (Fitness Value) ที่คำนวณจากฟิตเนสฟังก์ชัน (Fitness Function) สำหรับขั้นตอนวิธีเชิงพันธุกรรม [25] ใช้ฟิตเนสฟังก์ชันคำนวณหาค่าความน่าจะเป็นของผลลัพธ์ แล้วนำค่าความน่าจะเป็นเหล่านั้นมาเป็นข้อมูลในวงล้อรูเล็ต (Roulette Wheel) เพื่อการสุ่มเลือกพาเรนต์ (Parents) และมีการประยุกต์ใช้ฟิตเนสฟังก์ชันสำหรับคำนวณหาเส้นทางที่สั้นที่สุด ซึ่งฟิตเนสฟังก์ชันที่นำมาประยุกต์ใช้กับปัญหาการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบใช้สถิติ นิยมคำนวณในรูปแบบอไลเมน อาทิ

1) อไลเมนเมทริกซ์ (Alignment Matrix) คือการคำนวณค่าน้ำหนักแบบเมทริกซ์ (Weight Matrix) ของอักขระในอไลนเมนแต่ละตำแหน่งด้วยสมการที่ (2.16) แล้วทำการเลือกค่าอักขระที่มีน้ำหนักสูงที่สุดในหลักเดียวกันเป็นผลลัพธ์ดังรูปที่ 2.14

$$weight\ matrix = ln \frac{(n_i, j + n_i, j)/(N+1)}{pi} \tag{2.16}$$

กำหนดให้

- N คือจำนวนของสายข้อมูลที่นำเข้าทั้งหมด
- $n_{i,j}$ คือจำนวนอักขระในตำแหน่ง i, j
- p_i คือค่าความน่าจะเป็นก่อนการทดลอง (Priori Probability)



รูปที่ 2.14 อไลเมนและการคำนวณค่าน้ำหนักแบบเมทริกซ์

2) คอนเซนซัส (Consensus Scoring: CS) คือการคำนวณเพื่อวัดระดับความเหมือนระหว่างสายย่อยในอไลเมนดังสมการที่ (2.17) [4]

$$CS = 2 - (1/W)\sum_{i=1}^{W}\sum_{b=\{A,C,G,T\}} p_{bi}log_2(p_{bi}) \quad (2.17)$$

กำหนดให้

- b คืออักขระที่เป็นไปได้ทั้งหมดประกอบด้วย ‘A’, ‘C’, ‘G’ และ ‘T’
- w คือความยาวของสายย่อย
- p_{bi} คือความถี่ของตัวอักขระ b ในอไลเมน

3) อินฟอร์เมชั่นคอนเทนท์ (Information Content: *IC*) คือการคำนวณความเหมือนของรูปแบบอักขระในอไลเมน โดยพิจารณาข้อมูลอักขระที่ไม่ได้อยู่ในอไลเมน (Background) [36] ดังสมการที่ (2.18) ซึ่งอไลเมนที่มีค่าอินฟอร์เมชั่นคอนเทนท์สูง แสดงให้เห็นได้ว่ารูปแบบของโมทีฟที่อยู่ในอไลเมนมีความคล้ายคลึงกันมาก กล่าวคือโมทีฟที่อยู่ในอไลเมนมีโอกาสเป็นส่วนที่ยึดจับปัจจัยการถอดรหัสสูง

$$IC = \sum f_b log_2(f_b/p_b) \quad (2.18)$$

กำหนดให้

- f_b คือความถี่ของอักขระที่อยู่ในอไลเมนคำนวณได้จากสมการที่ (2.19)
- p_b คือความถี่ของอักขระที่ไม่ได้อยู่ในอไลเมนคำนวณได้จากสมการที่ (2.20)

$$f'_b = \frac{C_b + d_b}{N - 1 + D} \tag{2.19}$$

$$p'_b = \frac{C_{0b} + d_b}{S + D} \tag{2.20}$$

กำหนดให้

- c_b คือจำนวนอักขระที่ปรากฏในอไลเมน
- c_{0b} คือจำนวนอักขระที่ไม่ปรากฏในอไลเมน
- N คือจำนวนสายข้อมูลที่นำเข้าทั้งหมด
- S คือผลรวมของอักขระทั้งหมดที่ไม่ได้อยู่ในอไลเมน
- d_b คือค่าจำลองเพื่อป้องกันผลลัพธ์ที่เป็น 0 (Pseudo Counts) [2]
- D คือผลรวมของค่าจำลอง (Sum of Pseudo Counts)

4) เกณฑ์การให้คะแนนแบบเบย์ (Bayesian Scoring) ถูกนำมาประยุกต์เพื่อการคำนวณหาค่าความถี่ของโมทีฟในอไลเมน ดังสมการที่ (2.21) [37] ซึ่งเป็นฟังก์ชันในการคำนวณความถี่ของอักขระในอไลเมนที่ไม่จำกัดจำนวนสายโมทีฟในอไลเมน

$$(A) = |A| \left(\log \left(\frac{\hat{p}_0}{1 - \hat{p}_0} \right) - 1 + \prod_{j=1}^{w} \coprod_{k=1}^{4} \hat{\theta}_{jk} \log \left(\frac{\hat{\theta}_{jk}}{\hat{\theta}_{0k}} \right) \right) \tag{2.21}$$

กำหนดให้

- $|A|$ คือจำนวนของโมทีฟที่อยู่ในอไลเมน
- $\hat{p}_0$ คือค่าอัตราส่วนระหว่างข้อมูลในอไลเมนกับข้อมูลในสายข้อมูลที่นำเข้าทั้งหมด
- $\hat{\theta}_{jk}$ คือค่าความถี่ของอักขระ k ในคอลัมน์ j
- $\hat{\theta}_{0k}$ คือค่าความถี่ของอักขระ k ที่ไม่อยู่ในอไลเมน (Background Frequency)

5) ค่าเอฟสกอร์ (*F*-score) ถูกนำมาประยุกต์เพื่อการวัดประสิทธิภาพความแม่นยำและความถูกต้องครอบคลุม [18] ของผลลัพธ์ในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า คำนวณตามสมการ (2.22) ซึ่งค่าเอฟสกอร์เป็นค่าที่พิจารณาทั้งความแม่นยำ (Precision) และความถูกต้องครอบคลุม (Recall) โดยค่าความแม่นยำและค่าความถูกต้องครอบคลุมคำนวณตามสมการ (2.23) และ (2.24) ตามลำดับ

$$F = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.22}$$

$$Precision = \frac{\#_c}{\#_p} \tag{2.23}$$

$$Recall = \frac{\#_c}{\#_t} \tag{2.24}$$

กำหนดให้

- $\#c$ คือผลลัพธ์สายโมทีฟที่ถูกต้อง (True Positive) ซึ่งใช้การเปรียบเทียบผลลัพธ์ที่ได้จากขั้นตอนวิธีกับผลเฉลย เงื่อนไขของผลลัพธ์ที่ถูกต้องคือ ตำแหน่งของผลลัพธ์สายโมทีฟต้องอยู่ห่างจากผลเฉลยไม่เกิน $0.25*w$ [20] เมื่อ w คือความยาวของสายโมทีฟ
- $\#p$ คือจำนวนผลลัพธ์ทั้งหมด
- $\#t$ คือจำนวนผลลัพธ์ที่ถูกต้องทั้งหมด

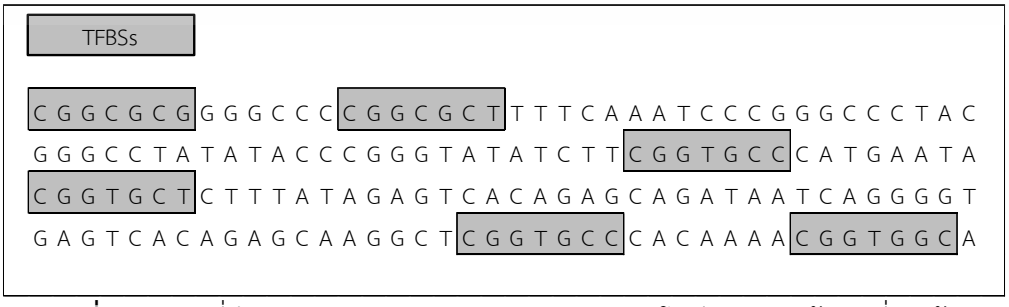
บทที่ 3
วิธีการดำเนินงานวิจัย

แนวคิดที่นำเสนอคือการประยุกต์และพัฒนาขั้นตอนวิธีในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า และแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า พร้อมทั้งพัฒนาขั้นตอนก่อนการดำเนินการสำหรับเตรียมข้อมูล (เตรียมพื้นที่ปัญหา) เพื่อให้การดำเนินการตรวจหาของขั้นตอนวิธีเป็นไปอย่างมีประสิทธิภาพ

3.1 นิยามของปัญหาการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัส

สำหรับงานวิจัยนี้กำหนดให้สายข้อมูลที่นำเข้าทั้งหมดคือ $S = \{S_1, S_2, ..., S_n\}$ และ n คือจำนวนสายข้อมูลที่นำเข้า โดยแต่ละสายข้อมูล S_i มีขนาดความยาว L ที่เท่าๆ กัน ซึ่งสายย่อย (Subsequences) M_{ij} ในสายข้อมูลที่นำเข้า S มีขนาดความยาวที่เท่ากันคือ w โดยกำหนดให้ลำดับสายข้อมูลที่นำเข้าและลำดับสายย่อยคือ i และ j ตามลำดับ โดยข้อมูลสายย่อยที่เป็นไปได้ทั้งหมดในแต่ละสายข้อมูลที่นำเข้า $S_i = \{M_{i1}, M_{i2}, M_{i3},..., M_{iL-w}, M_{iL-w+1}\}$ และข้อมูลอักขระหรือนิวคลีโอไทด์ (Nucleotides) $N_s = \{$'A', 'C', 'G', 'T'$\}$ ดังนั้นสายโมทีฟที่เป็นส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า $M = \{(M_{S_1}, M_{S_2},..., M_{S_{n-1}}, M_{S_n})\}$ แสดงดังรูปที่ 3.1 โดยจำนวนสายโมทีฟของ $M = n$ และสายโมทีฟที่เป็นส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า $MulM \in \{(M_{11}, M_{12}, ..., M_{1L-w}, M_{1L-w+1}), (M_{21}, M_{22}, ..., M_{2L-w}, M_{2L-w+1}), ..., (M_{n1}, M_{n2}, ..., M_{nL-w}, M_{nL-w+1})\}$ แสดงดังรูปที่ 3.2 โดยจำนวนสายโมทีฟของ $MulM <= n*(L-w+1)$

รูปที่ 3.1 ส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า



รูปที่ 3.2 ส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า

3.2 การตรวจหาแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า

ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาค (Nexus Particle Swarm Optimization: NexusPSO) คือแนวคิดที่นำเสนอเพื่อตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสในสายจีโนมแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า ซึ่งเป็นการประยุกต์ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค [13] คิดค้นโดย J. Kennedy และ R. Eberhart ในปี 1995 ร่วมกับขั้นตอนสานสัมพันธ์ (Nexus) ซึ่งเป็นขั้นตอนก่อนการดำเนินการที่คิดขึ้นใหม่ โดยขั้นตอนสานสัมพันธ์ถูกออกแบบเพื่อช่วยจัดการพื้นที่ของปัญหาสำหรับการตรวจหาโดยการสุ่มของขั้นตอนวิธีให้มีประสิทธิภาพมากขึ้น ขั้นตอนสานสัมพันธ์คือขั้นตอนก่อนการดำเนินการ (Pre-process) ที่ประกอบด้วยขั้นตอนย่อยที่สำคัญได้แก่ การจัดกลุ่ม (Grouping) กล่าวโดยละเอียดในส่วนที่ 3.2.1 การสร้างเครือข่ายระหว่างอนุภาค (Connection) อธิบายโดยละเอียดในส่วนที่ 3.2.2 และการเลือกสายความสัมพันธ์ (Selection) กล่าวโดยละเอียดในส่วนที่ 3.2.3 โดยวัตถุประสงค์หลักของขั้นตอนสานสัมพันธ์ คือลดขนาดพื้นที่ของปัญหา (Problem Space) ลง ด้วยการลดจำนวนของผลลัพธ์ที่เป็นไปได้ทั้งหมดแต่ยังคงรักษาผลลัพธ์ที่ถูกต้องไว้ (Pruning) พร้อมกับจัดการพื้นที่ของปัญหาเพื่อทำให้อนุภาคสามารถเคลื่อนที่ในพื้นที่ของปัญหาได้อย่างครอบคลุม ซึ่งทำให้โอกาสการติดอยู่ในค่าที่ดีที่สุดเฉพาะที่ (Local Optimums) น้อยลง

3.2.1 การจัดกลุ่ม (Grouping)

การจัดกลุ่มคือการจัดสายย่อยใดๆ ที่มีรูปแบบที่คล้ายกันให้อยู่ในกลุ่มเดียว กำหนดให้มีจำนวนกลุ่มทั้งหมด 4 กลุ่มตามจำนวนสมาชิกของ N_s ประกอบด้วยกลุ่ม ‘A’ กลุ่ม ‘C’ กลุ่ม ‘G’ และกลุ่ม ‘T’ โดยวิธีการวัดความคล้ายกันระหว่างสายย่อยคือ การคำนวณความถี่ของอักขระในแต่ละสายย่อยที่เป็นไปได้ทั้งหมด MS_{ij} โดยใช้การนับ กำหนดให้ i และ j คือลำดับสายข้อมูลที่นำเข้าและลำดับสายย่อยตามลำดับ

ตัวอย่างเช่น เมื่อ MS_{ij} = “CGGTAAA” ความถี่ของแต่ละอักขระจึงมีรายละเอียดดังนี้
‘A’ = 3, ‘C’ = 1, ‘G’ = 2 และ ‘T’ = 1 เป็นต้น

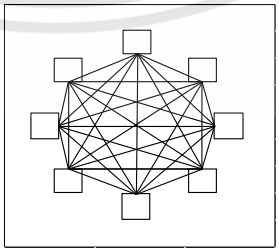
เมื่อ MS_{ij} มีความถี่ของอักขระ b สูงสุด MAX(b) โดย b = {‘A’, ‘C’, ‘G’, ‘T’} ดังนั้น MS_{ij} จึงถูกจัดให้อยู่ในกลุ่ม Group(b) ในกรณีที่สายย่อย MS_{ij} ใดๆ มีความถี่ของอักขระ b สูงสุด > 1 สามารถกำหนดให้สายย่อย MS_{ij} นั้นอยู่ได้มากกว่าหนึ่งกลุ่มตามจำนวนของอักขระ b สูงสุด ดังตัวอย่างในตารางที่ 3.1 ซึ่งเป็นการจัดกลุ่มสายย่อยในสายข้อมูลที่นำเข้าทั้งหมด 4 สาย (S_n=4) โดยมีจำนวนสายย่อย MS_{ij} ที่เป็นไปได้ทั้งหมดต่อสายข้อมูลนำเข้า 8 สาย สังเกตได้ว่าในสายที่ 1 (S_1) มีสายย่อยที่ถูกกำหนดให้อยู่มากกว่าหนึ่งกลุ่ม เนื่องจากมีความถี่ของอักขระสูงสุดมากกว่าหนึ่ง อาทิ สายย่อย MS_{11}= “CAAATCC” จัดอยู่ในกลุ่ม A และกลุ่ม C สายย่อยลำดับที่ MS_{13}= “AATCCGG” จัดอยู่ในกลุ่ม A กลุ่ม C และกลุ่ม G ในขณะที่สายย่อยลำดับที่ MS_{25} = “CTATATA” จัดอยู่ในกลุ่ม A และกลุ่ม T เป็นต้น

ตารางที่ 3.1 ตัวอย่างกลุ่มสายย่อยในสายข้อมูลที่นำเข้า 4 สาย

Ms_{ij}	Sequence	Max(b)	Group	Sequence	Max(b)	Group
	$S_{i=1}$			$S_{i=2}$		
j=1	C A A A T C C	*A,C=3*	*AC*	G G G C C T A	*G=3*	*G*
j=2	A A A T C C G	*A=3*	*A*	G G C C T A T	*C,G,T=2*	*CGT*
j=3	A A T C C G G	*A,C,G=2*	*ACG*	G C C T A T A	*A,C,T=2*	*ACT*
j=4	A T C C G G G	*G=3*	*G*	C C T A T A T	*T=3*	*T*
j=5	T C C G G G C	*C,G=3*	*CG*	C T A T A T A	*A,T=3*	*AT*
j=6	C C G G G C C	*C=4*	*C*	T A T A T A C	*A,T=3*	*AT*
j=7	C G G G C C C	*C=4*	*C*	A T A T A C C	*A=3*	*A*
j=8	G G G C C C C	*C=4*	*C*	T A T A C C C	*C=3*	*C*
	$S_{i=3}$			$S_{i=4}$		
j=1	C G G T G C T	*G=3*	*G*	G A G T C A C	*A,C,G=2*	*ACG*
j=2	G G T G C T C	*G=3*	*G*	A G T C A C A	*A=3*	*A*
j=3	G T G C T C T	*T=3*	*T*	G T C A C A G	*A,C,G=2*	*ACG*
j=4	T G C T C T T	*T=4*	*T*	T C A C A G A	*A=3*	*A*
j=5	G C T C T T T	*T=4*	*T*	C A C A G A G	*A=3*	*A*
j=6	C T C T T T A	*T=4*	*T*	A C A G A G C	*A=3*	*A*
j=7	T C T T T A T	*T=5*	*T*	C A G A G C A	*A=3*	*A*
j=8	C T T T A T A	*T=4*	*T*	A G A G C A A	*A=4*	*A*

3.2.2 การสร้างเครือข่าย (Connection)

เครือข่ายแบบอนุภาคเชื่อมโยงหากันทั้งหมด (Gbest) ถูกนำมาใช้ในการเชื่อมโยงอนุภาคย่อยเพื่อสร้างสายความสัมพันธ์ ซึ่งเป็นการใช้ข้อมูลร่วมกันระหว่างอนุภาคทุกอนุภาค ดังรูปที่ 3.3 การสร้างเครือข่ายถือเป็นขั้นตอนที่สำคัญเนื่องจากเป็นการเชื่อมโยงสายย่อยที่เป็นไปได้ทั้งหมดให้เป็นเครือข่าย ซึ่งส่งผลต่อการเลือกสายความสัมพันธ์ (กล่าวโดยละเอียดใน 3.2.3)



รูปที่ 3.3 เครือข่ายแบบอนุภาคเชื่อมโยงหากันทั้งหมด (Gbest)

การสร้างสายสัมพันธ์เริ่มจากนำสายย่อยที่เป็นไปได้ทั้งหมดในสายข้อมูลที่นำเข้า *i* สร้างความสัมพันธ์กับสายย่อยที่เป็นไปได้ทั้งหมดในสายข้อมูลที่นำเข้า *i*+1 ซึ่งกำหนดให้พิจารณาเฉพาะสายย่อยที่อยู่ในกลุ่มเดียวกันเท่านั้น (1 <= *i* <= *n*-1 เมื่อ *n* คือจำนวนเส้นยีนทั้งหมด) ดังนั้นรูปแบบความสัมพันธ์ที่เกิดขึ้นระหว่างเส้นยีนที่ *i* และเส้นยีนที่ *i*+1 มีดังนี้

$([M(A)_{ij} \bowtie M(A)_{i+1j}], [M(C)_{ij} \bowtie M(C)_{i+1j}], [M(G)_{ij} \bowtie M(G)_{i+1j}], [M(T)_{ij} \bowtie M(T)_{i+1j}])$

กำหนดให้ $\bowtie$ คือสายความสัมพันธ์ (Related Pairs) ของสายย่อยที่อยู่ในสายข้อมูลที่นำเข้า S_i และสายข้อมูลที่นำเข้า S_{i+1} ซึ่งสายความสัมพันธ์ระหว่างหน่วยย่อยแต่ละเส้นถูกกำหนดค่าคอนเซนซัสดังตัวอย่างในตารางที่ 3.2

ตารางที่ 3.2 ตัวอย่างค่าสายความสัมพันธ์ระหว่างสายย่อยจากสายข้อมูลที่นำเข้า S_i และ S_{i+1}

$S_i = 1$		$S_i = 2$		$S_i = 3$	
S_{i+1}	CS	S_{i+1}	CS	S_{i+1}	CS
3	0.25	3	0.8	3	0.25
5	0.4	5	0.4	5	0.4
6	0.4	6	0.2	6	0.4
7	0.3	7	0.3	7	0.3
32	0.6	32	0.6	32	0.6
33	0.7	33	0.7	33	0.8
34	0.7	34	0.7	34	0.8

...

3.2.3 การเลือกสายความสัมพันธ์ (Selection)

การเลือกสายความสัมพันธ์เป็นกระบวนการสุดท้ายของขั้นตอนสานสัมพันธ์ ซึ่งผลลัพธ์ที่ได้จากขั้นตอนนี้ คือกลุ่มข้อมูลสายย่อยที่มีการเชื่อมโยงกันระหว่างสายข้อมูลที่นำเข้าทั้งหมด โดยกำหนดให้เลือกสายความสัมพันธ์ระหว่างสายข้อมูลที่นำเข้าที่มีค่าคอนเซนซัสสูงสุดสองอันดับแรกในเครือข่าย (Connection) เท่านั้น

$[\text{Top2}\{M(A)_{ij} \bowtie M(A)_{i+1j}\}, \text{Top2}\{M(C)_{ij} \bowtie M(C)_{i+1j}\},$

$\text{Top2}\{M(G)_{ij} \bowtie M(G)_{i+1j}\}, \text{Top2}\{M(T)_{ij} \bowtie M(T)_{i+1j}\}]$

ตัวอย่างในตารางที่ 3.3 คือสายความสัมพันธ์ที่ถูกเลือกจากสายข้อมูลที่นำเข้า S_i และ S_{i+1} ที่อยู่ในกลุ่มเดียวกันซึ่งมีค่าคอนเซนซัสสูงสุดสองอันดับแรก โดยข้อมูลในตารางนี้เลือกมาจากข้อมูลในตารางที่ 3.2 และคาดการณ์ว่าผลลัพธ์จากขั้นตอนนี้สามารถช่วยลดปัญหาการติดอยู่

ในค่าที่ดีที่สุดเฉพาะที่ (Local Optimums) ของการสุ่มในขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคได้

ตารางที่ 3.3 ตัวอย่างสายความสัมพันธ์ที่ถูกเลือกจากสายข้อมูลที่นำเข้า S_i และ S_{i+1}

$S_i = 1$		$S_i = 2$		$S_i = 3$		
S_{i+1}	CS	S_{i+1}	CS	S_{i+1}	CS	
32	0.6	3	0.8	32	0.6	. . .
33	0.7	33	0.7	33	0.8	
34	0.7	34	0.7	34	0.8	

3.2.4 อนุภาคตั้งต้น (Particles Initialization)

การกำหนดอนุภาคของขั้นตอนวิธีเพื่อใช้ในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัส คือการสร้างอนุภาคใดๆ P_i ในฝูง (Swarm) ซึ่งกำหนดให้แต่ละอนุภาค P_i ประกอบด้วยสายย่อย M_{ij} (กำหนดให้ i และ j คือสายข้อมูลที่นำเข้าใดๆ และสายย่อยใดๆ ตามลำดับ) ซึ่งแต่ละสายข้อมูลที่นำเข้า S_i มีสายย่อย M_{ij} เพียงหนึ่งสาย งานวิจัยนี้กำหนดให้สายข้อมูลที่นำเข้าสายแรก S_1 เป็นสายข้อมูลสำหรับโมทีฟแรกของแต่ละอนุภาค โดย $S_1 = \{M_{11}, M_{12}, ..., M_{1L-w-1}, M_{1L-w}, M_{1L-w+1}\}$ และสายย่อยสายแรกของอนุภาคใดๆ $P_{i(M1)} = M_{1j}$ กำหนดให้ $i = j$ ดังนั้นจำนวนอนุภาคทั้งหมดในฝูงจึงเท่ากับ L-w+1 โดยแต่ละอนุภาคใดๆ P_i ทำการเลือกสายย่อยจากสายข้อมูลที่นำเข้าสายที่สองเป็นต้นไปจนถึงสายข้อมูลสายสุดท้ายตามลำดับ ซึ่งเลือกสายย่อยจากสายความสัมพันธ์ที่มีค่าคอนเซนซัสสูงสุด ส่งผลให้อนุภาค $P_i = (P_{i(M1)}, P_{i(M2)}, ..., P_{i(Mn-1)}, P_{i(Mn)})$ โดย n คือจำนวนสายข้อมูลที่นำเข้าทั้งหมด ดังนั้นรูปแบบของอนุภาค P_i ใน แต่ละกลุ่มมีรูปแบบการสร้างสายสัมพันธ์กันดังต่อไปนี้

$P(A)$ i อนุภาคใดๆ ในกลุ่ม ‘A’

$[\text{Top1}\{ M(A)_{1j} \bowtie (M(A)_{2j_{Top1}}, M(A)_{2j_{Top2}}) \}$

$\bowtie \text{Top1}\{ M(A)_{2j_{Op}} \bowtie (M(A)_{3j_{Top1}}, M(A)_{3j_{Top2}}) \}$

.
.
.

$\bowtie \text{Top1}\{ M(A)_{n-1j_{Op}} \bowtie (M(A)_{nj_{Top1}}, M(A)_{nj_{Top2}}) \}]$

$P(C)$ i อนุภาคใดๆ ในกลุ่ม ‘C’

$[\text{Top1}\{ M(C)_{1j} \bowtie (M(C)_{2j_{Top1}}, M(C)_{2j_{Top2}}) \}$

$\bowtie \text{Top1}\{ M(C)_{2j_{Op}} \bowtie (M(C)_{3j_{Top1}}, M(C)_{3j_{Top2}}) \}$

.
.
.

⋈ Top1{ $M(C)_{n\text{-}1j_{Op}}$ ⋈ ($M(C)_{nj_{Top1}}$, $M(C)_{nj_{Top2}}$) }]

P(*G*) *i* อนุภาคใดๆ ในกลุ่ม ‘G’

[Top1{ $M(G)_{1j}$ ⋈ ($M(G)_{2j_{Top1}}$, $M(G)_{2j_{Top2}}$) }

⋈ Top1{ $M(G)_{2j_{Op}}$ ⋈ ($M(G)_{3j_{Top1}}$, $M(G)_{3j_{Top2}}$) }

.
.
.

⋈ Top1{ $M(G)_{n\text{-}1j_{Op}}$ ⋈ ($M(G)_{nj_{Top1}}$, $M(G)_{nj_{Top2}}$) }]

P(*T*) *i* อนุภาคใดๆ ในกลุ่ม ‘T’

[Top1{ $M(T)_{1j}$ ⋈ ($M(T)_{2j_{Top1}}$, $M(T)_{2j_{Top2}}$) }

⋈ Top1{ $M(T)_{2j_{Op}}$ ⋈ ($M(T)_{3j_{Top1}}$, $M(T)_{3j_{Top2}}$) }

.
.
.

⋈ Top1{ $M(T)_{n\text{-}1j_{Op}}$ ⋈ ($M(T)_{nj_{Top1}}$, $M(T)_{nj_{Top2}}$) }]

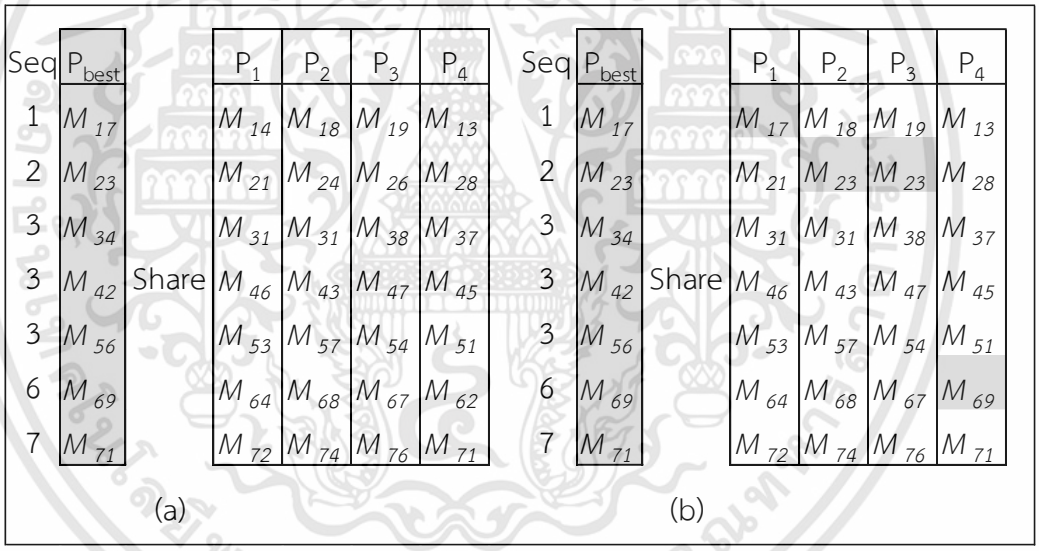
โดยความหมายของสัญลักษณ์และตัวแปรมีดังต่อไปนี้

- ⋈ คือสายความสัมพันธ์ของสายย่อยที่อยู่ระหว่างสายข้อมูลที่นำเข้า S_i และ S_{i+1}
- $M(b)_{ij}$ คือสายย่อยใดๆ ในสายจีโนม กำหนดให้ *i* และ *j* คือลำดับสายข้อมูลที่นำเข้าใดๆ และสายย่อยใดๆ ตามลำดับ โดยสมาชิกของ *b* คือ {‘A’,‘C’,‘G’,‘T’}
- $M(b)_{ij_{Top1}}$ คือสายย่อยใดๆ ที่เชื่อมโยงอยู่กับสายย่อย $M(b)_{i\text{-}1j}$ โดยมีค่าสายความสัมพันธ์สูงสุดอันดับหนึ่ง
- $M(b)_{ij_{Top2}}$ คือสายย่อยใดๆ ที่เชื่อมโยงอยู่กับสายย่อย $M(b)_{i\text{-}1j}$ โดยมีค่าสายความสัมพันธ์สูงสุดอันดับสอง
- *n* คือจำนวนสายข้อมูลที่นำเข้าทั้งหมด
- $M(b)_{ij_{op}}$ คือผลลัพธ์สายย่อยที่เหมาะสมที่สุด

3.2.5 การปรับตำแหน่งอนุภาค (Particle's Movement)

ในรอบแรกของการทำงาน กำหนดให้ความเร็วเริ่มต้นของทุกอนุภาคเป็น 0 และใช้ฟิตเนสฟังก์ชันจากสมการที่ (2.18) ซึ่งกล่าวโดยละเอียดในหัวข้อ 2.11 มาคำนวณค่าฟิตเนสของแต่ละอนุภาค หลังจากนั้นทำการเปรียบเทียบค่าฟิตเนสของแต่ละอนุภาคเพื่อระบุอนุภาคที่เหมาะสมที่สุด (P_{best}) ดังตัวอย่างในรูปที่ 3.4(a) แล้วทำการปรับตำแหน่งอนุภาคเพื่อนบ้าน (Particle's Neighbors) โดยการนำเอาข้อมูลสายย่อย M_{ij} ของอนุภาคที่เหมาะสมที่สุด P_{best} แทนที่สายย่อยของอนุภาคเพื่อนบ้าน P_i ด้วยการสุ่มดังรูปที่ 3.4(b) การปรับตำแหน่งอนุภาคในแต่ละรอบการทำงานส่งผลให้อนุภาคในฝูงเกิดการเคลื่อนที่ไปเรื่อยๆ จนกระทั่งสิ้นสุดการทำงาน เงื่อนไขสำหรับจบการทำงานของขั้นตอนวิธีมีสองกรณีประกอบด้วย

1) กรณีทุกอนุภาคอยู่ในตำแหน่งสายย่อยเดียวกันภายในจำนวนรอบการทำงานที่กำหนดไว้ โดยถือว่าตำแหน่งเหล่านั้นคือผลลัพธ์ของขั้นตอนวิธี
2) กรณีที่ครบจำนวนรอบการทำงานที่กำหนดไว้แต่อนุภาคยังไม่อยู่ในตำแหน่งเดียวกันซึ่งขั้นตอนวิธีใช้การพิจารณาเลือกอนุภาคที่มีค่าฟิตเนสสูงสุดเป็นผลลัพธ์



รูปที่ 3.4 ตัวอย่างการปรับตำแหน่งของอนุภาค (a) แสดงอนุภาค 5 อนุภาคในสายข้อมูลที่นำเข้า 7 เส้น (b) แสดงการแทนที่ระหว่างหน่วยย่อย

3.2.6 การระบุค่าฟิตเนสให้กับอนุภาค (Fitness Function)

มาตรวัดสำหรับวัดอนุภาคที่เหมาะสมที่สุด P_{best} ณ เวลาใดๆ t_i คือฟิตเนสฟังก์ชัน โดยขั้นตอนวิธีที่นำเสนอนำสมการที่ (2.18) มาใช้เป็นฟิตเนสฟังก์ชัน ซึ่งเป็นสมการสำหรับคำนวณหาค่าอินฟอร์เมชั่นคอนเทนท์ กล่าวโดยละเอียดในหัวข้อ 2.11 ของส่วนที่ยึดจับปัจจัยการถอดรหัส กำหนดให้ความยาวของสายย่อยคือ W เงื่อนไขคือ $0 < W <= L\text{-}1$ และ L คือความยาวของสายข้อมูลที่นำเข้า สำหรับอักขระที่เป็นไปได้ทั้งหมดคือ b = {'A', 'C', 'G', 'T'} ค่าความถี่ของอักขระ b ที่ปรากฏอยู่ในผลลัพธ์ของอนุภาคคือ f_b คำนวณได้จากสมการที่

(2.19) และค่าความถี่อักขระ b ที่ไม่อยู่ในผลลัพธ์ของอนุภาคคือ p'_b คำนวณได้จากสมการที่ (2.20) ซึ่งกล่าวโดยละเอียดในหัวข้อ 2.11

3.2.7 การจัดเก็บข้อมูลที่นำเข้าและขั้นตอนวิธีของแนวคิดที่นำเสนอ

งานวิจัยนี้ออกแบบฐานข้อมูลเพื่อรองรับการทำงานของขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาค ซึ่งประกอบด้วยตารางความสัมพันธ์ดังต่อไปนี้ ตารางข้อมูลสายข้อมูลที่นำเข้า ตารางสายสัมพันธ์ระหว่างสายย่อยที่เป็นไปได้ทั้งหมดและตารางข้อมูลอนุภาค สำหรับขั้นตอนหลักในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสประกอบด้วยขั้นตอนการเตรียมข้อมูล (เตรียมพื้นที่ของปัญหา) ซึ่งเป็นขั้นตอนก่อนการดำเนินการ และขั้นตอนการตรวจหาผลลัพธ์ โดยลำดับการทำงานหรือซูโดโค้ด (Pseudocode) ของขั้นตอนวิธีที่นำเสนอมีดังต่อไปนี้

ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาค

Input: w = the length of subsequence;
Maximum = number of iterations;
N = number of input sequences;
L = length of input sequences, b = {'A', 'C', 'G', 'T'};

1. Nexus Process (Pre-process)

```
for i = 1 to N do
  for j = 1 to L_i-w+1 do
    grouping M[i][j];
    connection: M[i][j] and M[i+1][j];
  end for j
end for i
```

2. PSO Process

```
Initialize particles from best connection pair, start from first of sequences;
    for k = 1 to Maximum OR not converged do
      select local best particle;
      update velocity of particles;
```

update position of particles;

if k = 1 or local best > global best then

update global best from local best;

end if

end for k

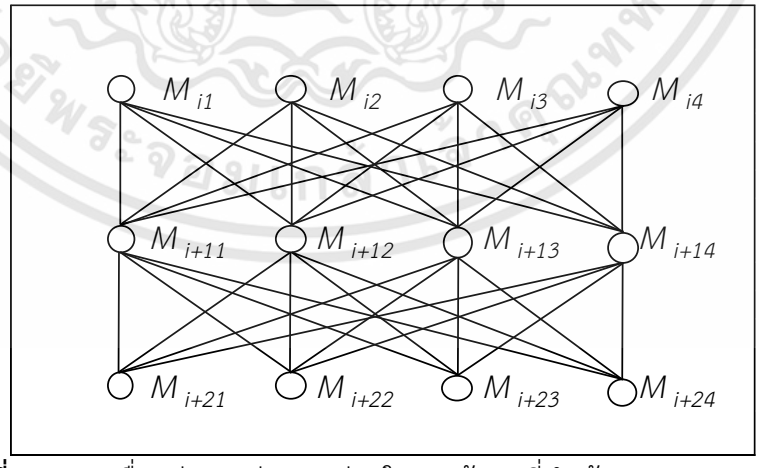
Output: The set of subsequences M;

3.3 การการตรวจหาแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า

สำหรับปัญหาการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า งานวิจัยนี้นำเสนอขั้นตอนวิธีในการแก้ปัญหา โดยประยุกต์ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค [13] ร่วมกับระยะทางแฮมมิงแบบประยุกต์กับนิวคลีโอไทด์ในสายดีเอ็นเอ [27] ซึ่งคิดค้นโดย M. Stine และคณะในปี 2003 เรียกขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคร่วมกับระยะทางแฮมมิง (PSO_HD) เพื่อเพิ่มประสิทธิภาพการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสอย่างแม่นยำ พร้อมกับประยุกต์แนวคิดเกณฑ์ขั้นต่ำผลลบที่ผิดพลาด (Cut-off Minimizing False Negative Rate: *minFN*) [35] เพื่อการตรวจหาสายโมทีฟคงเหลือสำหรับนำมาสร้างผลลัพธ์ปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า

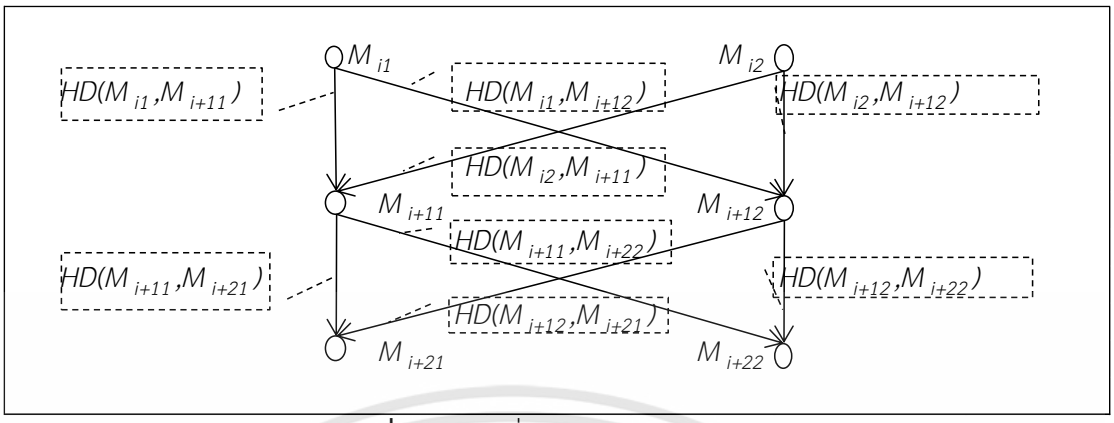
3.3.1 ขั้นตอนก่อนการดำเนินการ (Pre-process)

เพื่อการเตรียมพื้นที่ของปัญหาที่มีความต่อเนื่องและมีนัยสำคัญ แนวคิดที่นำเสนอทำการสร้างสายความสัมพันธ์ ระหว่างสายย่อยจากสายข้อมูลที่นำเข้า i และ i+1 (1< i+1 <= n) ดังรูปที่ 3.5 เพื่อให้พื้นที่ของปัญหามีความต่อเนื่อง



รูปที่ 3.5 การเชื่อมต่อระหว่างสายย่อยในสายข้อมูลที่นำเข้า i และ i+1

โดยแต่ละสายความสัมพันธ์มีค่าน้ำหนัก ซึ่งคำนวณด้วยระยะทางแฮมมิงที่ใช้หลักการจับคู่ของสายดีเอ็นเอและคลาสของนิวคลีโอไทด์เพื่อแปลงเป็นสองบิตดังรูปที่ 3.6



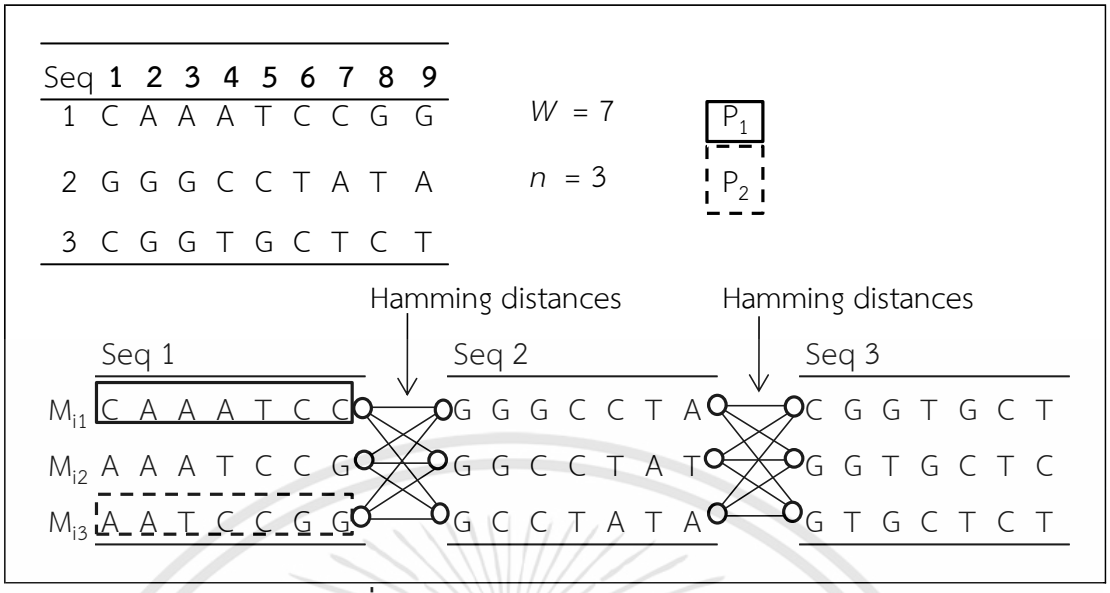
รูปที่ 3.6 การเชื่อมต่อระหว่างสายย่อย

ผลที่เกิดขึ้นหลังจากดำเนินการในขั้นตอนก่อนการดำเนินการคือ กลุ่มของสายย่อย M_{ij} จากสายข้อมูลที่นำเข้า S_i ซึ่งเป็นการเชื่อมต่อแบบทั้งหมด (Cross Join) ที่มีค่าน้ำหนัก ($\bowtie_{(HD)}$) กำกับแต่ละสายความสัมพันธ์ {($M_{ij} \bowtie_{(HD)} M_{i+1j}$), ($M_{i+1j} \bowtie_{(HD)} M_{i+2j}$), ($M_{i+2j} \bowtie_{(HD)} M_{i+3j}$) ...,($M_{n-2j} \bowtie_{(HD)} M_{n-1j}$), ($M_{n-1j} \bowtie_{(HD)} M_{nj}$) } มีรายละเอียดดังนี้

{($M_{ij} \bowtie_{(HD)} M_{i+1j}$, $M_{ij} \bowtie_{(HD)} M_{i+1j+1}$, $M_{ij} \bowtie_{(HD)} M_{i+1j+2}$, ..., $M_{ij} \bowtie_{(HD)} M_{iL-w}$, $M_{ij} \bowtie_{(HD)} M_{iL-w+1}$),
($M_{i+1j} \bowtie_{(HD)} M_{i+2j}$, $M_{i+1j} \bowtie_{(HD)} M_{i+2j+1}$, $M_{i+1j} \bowtie_{(HD)} M_{i+2j+2}$, ..., $M_{i+1j} \bowtie_{(HD)} M_{i+2L-w}$, $M_{i+1j} \bowtie_{(HD)} M_{i+2L-w+1}$),
($M_{i+2j} \bowtie_{(HD)} M_{i+3j}$, $M_{i+2j} \bowtie_{(HD)} M_{i+3j+1}$, $M_{i+2j} \bowtie_{(HD)} M_{i+3j+2}$, ..., $M_{i+2j} \bowtie_{(HD)} M_{i+3L-w}$, $M_{i+2j} \bowtie_{(HD)} M_{i+3L-w+1}$),
...,
($M_{n-2j} \bowtie_{(HD)} M_{n-1j}$, $M_{n-2j} \bowtie_{(HD)} M_{n-1j+1}$, $M_{n-2j} \bowtie_{(HD)} M_{n-1j+2}$, ..., $M_{n-2j} \bowtie_{(HD)} M_{n-1L-w}$, $M_{n-2j} \bowtie_{(HD)} M_{n-1L-w+1}$),
($M_{n-1j} \bowtie_{(HD)} M_{nj}$, $M_{n-1j} \bowtie_{(HD)} M_{nj+1}$, $M_{n-1j} \bowtie_{(HD)} M_{nj+2}$, ..., $M_{n-1j} \bowtie_{(HD)} M_{nL-w}$, $M_{n-1j} \bowtie_{(HD)} M_{nL-w+1}$)}

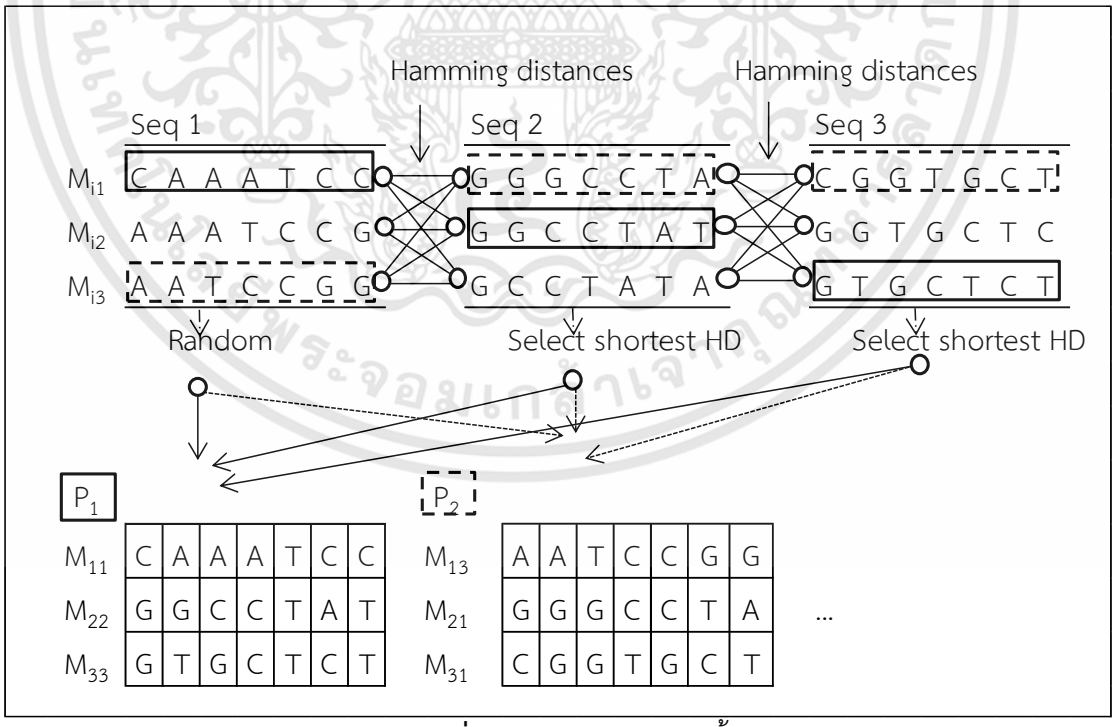
3.3.2 อนุภาคตั้งต้น (Particles Initialization)

อนุภาคใดๆ P_i สำหรับการตรวจหาลำดับเบสจำเพาะและอนุรักษ์จำนวน k อนุภาคมีข้อมูลสายย่อย M_{ij} จำนวน n สาย ซึ่งจำนวนอนุภาค $k <= L\text{-}w+1$ ในแต่ละอนุภาค P_i กำหนดให้ทำการเลือกสายย่อย M_{1j} จากสายข้อมูลที่นำเข้าสายแรก S_1 ด้วยการสุ่ม จากตัวอย่างในรูปที่ 3.7 ซึ่งเป็นการสุ่มเลือกสายย่อยจากสายข้อมูลที่นำเข้าสายแรก (Seq 1) จาก 2 อนุภาค (P1 คือเส้นทึบ P2 คือเส้นประ) โดยความยาวของสายข้อมูลที่นำเข้า L และความยาวของสายย่อย W กำหนดให้เป็น 9 และ 7 ตามลำดับ ในจำนวนสายข้อมูลที่นำเข้าทั้งหมด 3 สาย จากนั้นทำการเลือกสายย่อยในสายข้อมูลที่นำเข้าถัดไปที่มีระยะทางแฮมมิงที่สั้นที่สุดในรูปที่ 3.7



รูปที่ 3.7 การสร้างความสัมพันธ์ระหว่างสายย่อย

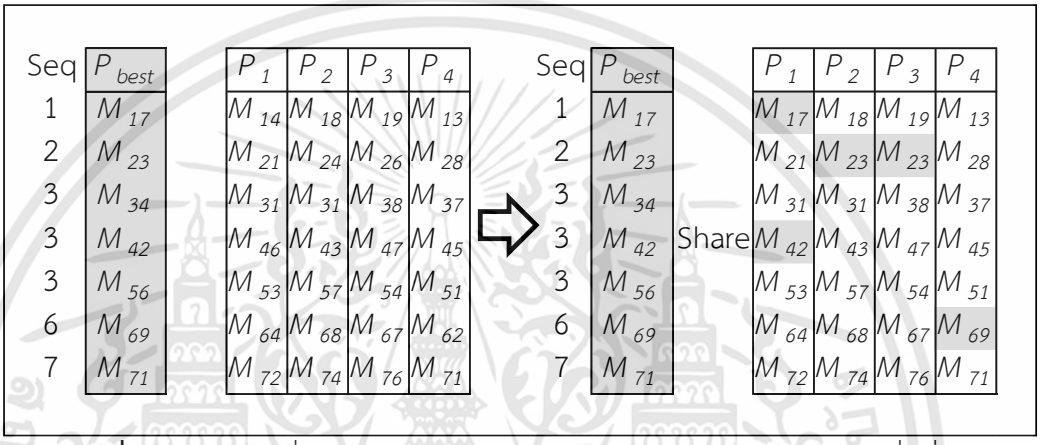
ในกรณีที่จำนวนสายย่อยมีระยะทางแฮมมิงสั้นที่สุดมากกว่าหนึ่งสาย $HD_{best}(M_{ij}, M_{i+1j}) > 1$ กำหนดให้ทำการสุ่มเลือกสายย่อยใน M_{i+1j} ที่มีระยะทางแฮมมิงสั้นที่สุดและอยู่ในสายความสัมพันธ์ M_{ij} โดยกำหนดให้สุ่มเลือกเพียงหนึ่งสาย ดังนั้นข้อมูลสายย่อยในอนุภาคใดๆ $P_i = \{M_{1j}, M_{2j}, ..., M_{n-1j}, M_{nj}\}$ ทำให้สมาชิกของแต่ละอนุภาคกระจายในพื้นที่ของปัญหาอย่างมีนัยสำคัญ โดยผลลัพธ์จากการสร้างอนุภาคตั้งต้นแสดงดังรูปที่ 3.8



รูปที่ 3.8 การสร้างอนุภาคตั้งต้น

3.3.3 การเคลื่อนที่ของอนุภาค (Particle's Movement)

วิธีการเคลื่อนที่ของฝูงของขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคร่วมกับระยะทางแฮมมิง คือการนำอนุภาคมาสร้างเครือข่ายเพื่อใช้ข้อมูลร่วมกัน ภายใต้รูปแบบเชื่อมโยงทุกอนุภาค (GBest) แล้วทำการหาอนุภาคที่ดีที่สุด ณ เวลาใดๆ (P_{best}) โดยการคัดเลือกอนุภาคที่มีค่าฟิตเนสสูงสุด ซึ่งแต่ละอนุภาคคืออไลเมนในรูปแบบเมทริกซ์ที่มีขนาด n แถว w หลัก โดยที่ n คือจำนวนสายข้อมูลที่นำเข้าและ w คือความยาวของสายย่อย หลังจากนั้นทำการเคลื่อนที่อนุภาคเพื่อนบ้านตามอนุภาคที่ดีที่สุดด้วยการแทนที่ตำแหน่ง M_{ij} ด้วย $M_{best_{ij}}$ ดังรูปที่ 3.9

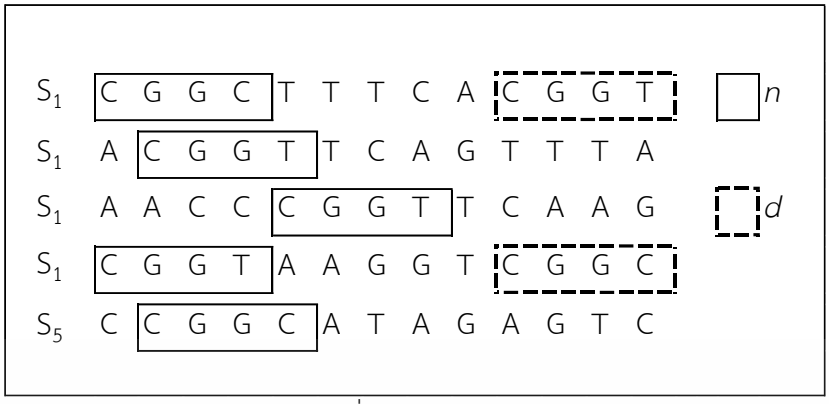


รูปที่ 3.9 การแทนที่ตำแหน่งสายย่อยของอนุภาคด้วยสายย่อยของอนุภาคที่ดีที่สุด

โดยประยุกต์แนวคิดการเคลื่อนที่ของฝูงแบบไบนารี เพื่อเป็นเงื่อนไขในการเคลื่อนที่ของอนุภาคในฝูง [27]-[29] ซึ่งมีหลักการทำงานดังนี้ กำหนดให้แต่ละอนุภาค P_i ประกอบด้วยสายย่อยที่ได้จากการสุ่มจากสายข้อมูลที่นำเข้า โดยเวกเตอร์ X และ V คือเวกเตอร์ของตำแหน่งและความเร็วของอนุภาค P_i ตามลำดับ โดยสมาชิกเวกเตอร์ $X = \{x_{1j}, x_{2j}, ..., x_{n-1j}, x_{nj}\}$ และสมาชิกของเวกเตอร์ $V = \{v_{1j}, v_{2j}, ..., v_{n-1j}, v_{nj}\}$ โดย x_{ij} คือตำแหน่งแต่ละสายย่อยและ v_{ij} คือความเร็วของแต่ละสายย่อย สำหรับความเร็วที่เปลี่ยนแปลงไปในแต่ละรอบการทำงานขึ้นอยู่กับระยะห่างระหว่างอนุภาค P_i และอนุภาคที่ดีที่สุด ณ เวลาใดๆ P_{best} สำหรับการเคลื่อนที่ของอนุภาคขั้นตอนวิธีที่นำเสนอใช้การเคลื่อนที่ของฝูงแบบไบนารี ซึ่งเป็นการแทนที่ตำแหน่ง x_{ij} ของอนุภาค P_i ด้วยตำแหน่ง $x_{best_{ij}}$ ของอนุภาค P_{best} ในแต่ละรอบการทำงานดังรายละเอียดในหัวข้อ 2.6 กำหนดให้ดำเนินการจนกระทั่งทุกอนุภาคอยู่ในตำแหน่งที่เป็นผลลัพธ์เดียวกันหรือดำเนินการครบจำนวนรอบตามที่กำหนดแล้วทำการเลือกอนุภาคที่มีค่าฟิตเนสสูงสุดเป็นผลลัพธ์ โดยผลลัพธ์จากขั้นตอนนี้คือ อไลเมน (Alignment: A)

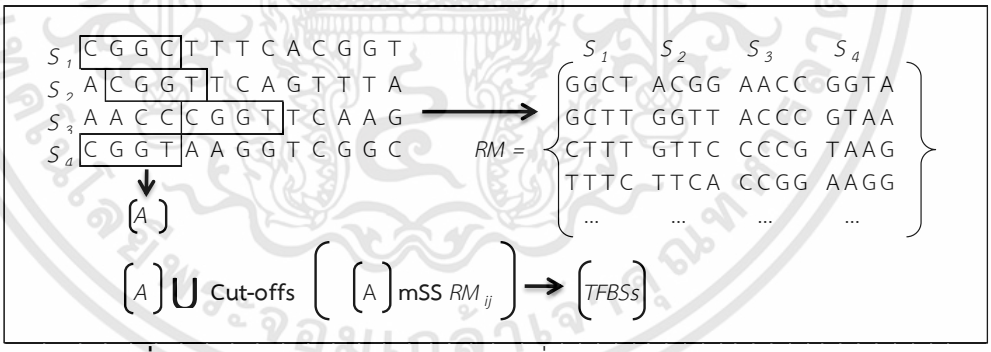
3.3.4 การตรวจหาสายโมทีฟคงเหลือ (Detection of Remaining Motifs)

สำหรับการตรวจหาสายโมทีฟคงเหลือจำนวน d สาย เพื่อนำมาสร้างผลลัพธ์ของปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้าดังรูปที่ 3.10 ใช้วิธีการนำผลลัพธ์อไลเมน A (อไลเมน A มีจำนวนโมทีฟ n สาย) มาทำการคำนวณค่าความเหมือนแบบเมทริกซ์



รูปที่ 3.10 อไลเมนการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสและสายโมทีฟคงเหลือ

กับสายย่อยคงเหลือทั้งหมด [35] คิดค้นโดย A.E. Kel และคณะในปี 2003 มีวิธีการคำนวณดังสมการที่ (2.11) โดยนำข้อมูลจากอไลเมน *A* มาสร้างเป็นเมทริกซ์ปัจจุบันแล้วคำนวณดังสมการที่ (2.12) แล้วนำสายย่อยใดๆ RM_{ij} มายูเนียนกับ อไลเมน *A* ทีละสาย (สายย่อย *RM* คือสายย่อยที่ไม่ได้อยู่ในอไลเมน *A*) เพื่อคำนวณหาค่าความถี่ต่ำสุดของอักขระดังสมการที่ (2.13) และคำนวณหาค่าความถี่สูงสุดของอักขระดังสมการที่ (2.14) กำหนดให้ f_i^{min} คือจำนวนที่น้อยที่สุดของอักขระในตำแหน่ง *i* และ f_i^{max} คือจำนวนที่มากที่สุดของอักขระในตำแหน่ง *i* กล่าวโดยละเอียดในหัวข้อ 2.10 หลังจากนั้นเป็นการคัดเลือกโมทีฟคงเหลือโดยตัดผลลัพธ์ที่ต่ำกว่าเกณฑ์ที่ตั้งไว้ออก [35] แล้วนำผลลัพธ์ที่สูงกว่าอัตราที่ตั้งไว้มายูเนียนกับผลลัพธ์อไลเมนจากขั้นตอน 3.3.3 แสดงดังรูปที่ 3.11 เพื่อเป็นผลลัพธ์ปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า



รูปที่ 3.11 ตรวจหาสายโมทีฟคงเหลือเพื่อสร้างผลลัพธ์ปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า

3.3.5 การดำเนินการของขั้นตอนวิธี

การดำเนินการเพื่อการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้าของขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคร่วมกับระยะทางแฮมมิง มี 4 ส่วนหลักประกอบด้วย

1) การกำหนดพารามิเตอร์

2) การสร้างสายความสัมพันธ์

3) การเคลื่อนที่ของฝูง

4) การตัดสายย่อยที่ต่ำกว่าเกณฑ์

โดยผลลัพธ์ของของขั้นตอนวิธีคือปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า ซึ่งลำดับการทำงานหรือซูโดโค้ด (Pseudocode) ของขั้นตอนวิธีที่นำเสนอมีดังต่อไปนี้

ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคร่วมกับระยะทางแฮมมิง

Input:
w = length of subsequence;
Maximum = number of iterations;
N = number of input sequences;
L = length of input sequences;
b = {*'A', 'C', 'G', 'T'*};

Begin:
Set:
HD[ ][ ][ ] = Null;
EM = null;
remainM[ ][ ] = null;
Plb[ ] = null;
Pnb[ ] = null;
TFBSs as DataTable = null;

//Hamming distance
for *i*=1 to *N*
  for *j*=1 to *w*-*l*+1
    EM = Encrypt (*M*[*i*][*j*]);
    for *k*=1 to *w*-*l*+1 // Motif in second row
      HD[*i*][*j*][*k*] = Hamming distance (*EM*, Encrypt(*M*[*i*+1][*k*]));
    end for *k*
  end for *j*
end for *i*

//Swarm movement
Initialize particle;
for *i* = 1 to *Maximum* **OR** *not converged* **do**
    select local best particle;
    update velocity of particles;

update position of particles;
if *i* = 1 **or** local best > global best **then**
update global best from local best;
end if
end for *i*

//Cut-offs minimising false negative (*minFN*)
for *i*=1 to *N*
for *j*=1 to *w*-*l*+1 //Motif in first row
if *M*[*i*][*j*] not exist(global best) **then**
calculate M_{ij} by mSS;
if error rate mss(M_{ij}) **then**
remainM[*i*][*j*] = minSS(M_{ij});
end if
end if
end for *j*
end for *i*

Output:
TFBSs = concatenate (*remainM*[*i*][*j*], global best);

3.4 การดำเนินการ

สำหรับการดำเนินการในขั้นตอนวิธีที่นำเสนอ ประกอบด้วยส่วนของฮาร์ดแวร์ ซึ่งหน่วยประมวลผลกลาง (CPU) ความเร็วอยู่ที่ 2.70 กิกะเฮริตซ์ (GHz) และหน่วยความจำหลัก (RAM) อยู่ที่ 8 กิกะไบต์ (GB) ในขณะที่ส่วนของโปรแกรมพัฒนาโดยภาษา C# เวอร์ชั่น 5.0 และส่วนฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ใช้ SQL Server 2012 เป็นซอฟต์แวร์ในการจัดการ สำหรับตารางข้อมูลในฐานข้อมูลเชิงสัมพันธ์ประกอบด้วย ตารางข้อมูลอนุภาค ตารางสายความสัมพันธ์ และตารางสายข้อมูลที่นำเข้าทั้งหมด

3.4.1 กลุ่มข้อมูลที่นำมาทดลองและการตั้งค่าพารามิเตอร์

สำหรับการทดสอบประสิทธิภาพของขั้นตอนวิธีที่นำเสนอ ในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า และแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า โดยแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้าทดสอบกับฐานข้อมูล SCPD [39] 5 กลุ่ม ซึ่งเป็นกลุ่มข้อมูลของยีสต์ (*Saccaromyces Cerevisiae*) ได้แก่ กลุ่มข้อมูล GAL4, RAP1, REB1, MCB และ PDR3 ดังตารางที่ 3.4 และจากฐานข้อมูล Genbank [40] 5 กลุ่ม ซึ่งเป็นกลุ่มข้อมูลของมนุษย์ (*Homo Sapiens*) ได้แก่กลุ่มข้อมูล ELK4, E2F1, FOXD1, USF1 และ RELA ดังตารางที่ 3.5 สำหรับแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้าทดสอบกับกลุ่มข้อมูล 7 กลุ่มประกอบด้วย กลุ่มข้อมูล CREB, E2F, MEF2 และ SRF จากฐานข้อมูล

JASPAR [40] แสดงดังตารางที่ 3.6 และกลุ่มข้อมูล ERE, MYOD และ TBP จากฐานข้อมูล TRANSFAC [41] แสดงดังตารางที่ 3.7 โดยกลุ่มข้อมูลเหล่านี้เป็นกลุ่มข้อมูลที่มีส่วนที่ยึดจับปัจจัยการถอดรหัสที่มีความหลากหลาย ซึ่งนิยมนำมาทดสอบประสิทธิภาพของหลายขั้นตอนวิธี [17-20]

ตารางที่ 3.4 คุณสมบัติของกลุ่มข้อมูลจากฐานข้อมูล SPCD

Datasets	Size	Motif Width	Consensus Sequence
GAL4	6	17	CGGNNNNNNNNNNCCG
RAP1	16	7	RMACCCA
REB1	14	7	YYACCCG
MCB	6	6	WCGCGW
PDR3	7	8	TCCGYGGA

ตารางที่ 3.5 คุณสมบัติของกลุ่มข้อมูลจากฐานข้อมูล Genbank

Datasets	Size	Motif Width	Consensus Sequence
ELK4	20	9	ACCGGAAGT
E2F1	10	8	TTTGGCGC
FOXD1	20	8	GTAAACAT
USF1	30	7	CACGTGG
RELA	18	10	GGGAATTTCC

ตารางที่ 3.6 คุณสมบัติของกลุ่มข้อมูลจากฐานข้อมูล JASPAR

Datasets	Size	Motif Width	TFBSs Embedded	Length
CREB	17	8	19	200
E2F	25	11	27	200
MEF2	17	7	17	200
SRF	20	10	36	200

ตารางที่ 3.7 คุณสมบัติของกลุ่มข้อมูลจากฐานข้อมูล TRANSFAC

Datasets	Size	Motif Width	TFBSs Embedded	Length
ERE	25	13	25	200
MYOD	17	6	21	200
TBP	95	6	95	200

โดยคุณสมบัติของกลุ่มข้อมูลที่นำมาทดลองในตารางที่ 3.4 ถึงตารางที่ 3.7 ประกอบด้วยข้อมูล
- Size คือจำนวนของสายข้อมูลที่นำเข้า
- Motif Width คือ ความยาวของสายโมทีฟ
- Consensus Sequence คือสายคอนเซนชัสเป็นผลเฉลย
- TFBSs Embedded คือจำนวนสายโมทีฟที่เป็นส่วนที่ยึดจับปัจจัยการถอดรหัส
- Length คือความยาวของสายข้อมูลที่นำเข้า

สำหรับกลุ่มข้อมูล CRP จากฐานข้อมูล RegulonDB [42] คือกลุ่มข้อมูลสายจีโนมของอีโคไล (*Escherichia Coli*: *E.Coli*) โดยกลุ่มข้อมูลนี้ใช้เพื่อทดสอบประสิทธิภาพของขั้นตอนวิธีในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัส ทั้งแบบหนึ่งสายและแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า เนื่องจากสายจีโนมนี้มีสายโมทีฟอย่างน้อยหนึ่งเส้นในแต่ละสายดีเอ็นเอ พร้อมกับรูปแบบข้อมูลนิวคลีโอไทด์ที่หลากหลาย จึงนิยมนำมาทดสอบประสิทธิภาพของขั้นตอนวิธีต่างๆ [3][4][15][17][22][25] สำหรับข้อมูลคุณสมบัติในตารางที่ 3.8 คือกลุ่มข้อมูล CRP ซึ่งสายข้อมูลที่นำเข้าแต่ละสายมีความยาว 105 นิวคลีโอไทด์ (105 อักขระ) สายโมทีฟมีความยาว 22 นิวคลีโอไทด์ [43] และจำนวนสายโมทีฟที่เป็นส่วนที่ยึดจับปัจจัยการถอดรหัสทั้งหมด 23 สาย

ตารางที่ 3.8 คุณสมบัติของกลุ่มข้อมูลอีโคไล (*Escherichia Coli: E.Coli*)

No.	Names	Motif 1	Motif 2
1	CE1CG	17	61
2	ECOARABOP	17	55
3	ECOBGLR1	76	-
4	ECOCRP	63	-
5	ECOYA	50	-
6	ECODEOP2	7	60
7	ECOGALE	42	-
8	ECOILVBPR	39	-
9	ECOLAC	9	80
10	ECOMALBA	14	-
11	ECOMALBA2	61	-
12	ECOMALT	41	-
13	ECOOMPA	48	-
14	ECOTNAA	71	-
15	ECOXUL	17	-
16	PBR-P4	53	-
17	TRN9CAT	1	84
18	TDC	78	-

การกำหนดค่าพารามิเตอร์เพื่อการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้าและแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า ของขั้นตอนวิธีที่นำเสนอประกอบด้วยค่าพารามิเตอร์ปัจจัยความเฉื่อย (Inertia Weight) ค่าพารามิเตอร์ความแปรปรวน (Cognitive) ค่าพารามิเตอร์สังคม (Social) และจำนวนรอบเพื่อหยุดการทำงาน (Number of Iterations) ดังรายละเอียดในตารางที่ 3.9 ซึ่งกำหนดตามคุณสมบัติของกลุ่มข้อมูลที่นำมาทดลองอย่างเหมาะสม [17] โดยจำนวนอนุภาคสำหรับขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาคและขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคร่วมกับระยะทางแฮมมิง มีรายละเอียดดังตารางที่ 3.10

ตารางที่ 3.9 ค่าพารามิเตอร์สำหรับอนุภาค

Description	Parameters	Size
Inertia Weight	α	0.4
Cognitive	β	0.8
Social	γ	0.8
Number of Iterations	*Maximum*	3000

ตารางที่ 3.10 จำนวนอนุภาคที่กำหนดสำหรับขั้นตอนวิธีที่นำเสนอ

Algorithms	Number of Particles	Description
NexusPSO	$l - w + 1$	*l = length of an input seqeunce* *w = width of a motif seqeunce*
PSO_HD	100	The number of particles are properly defined with typical attributes for testing datasets.

บทที่ 4
ผลการวิจัยและการอภิปรายผล

สำหรับผลการวิจัยนี้เป็นการนำเสนอผลลัพธ์ที่ได้จากการทดลองมาวัดประสิทธิภาพการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสเพื่อการอภิปราย ซึ่งแบ่งการทดลองออกเป็น 2 กลุ่ม โดยกลุ่มแรกเป็นการทดลองตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า ซึ่งงานวิจัยนี้นำเสนอขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาค (NexusPSO) โดยกระบวนการที่นำมาใช้ในการประเมินผลประกอบด้วย

- กระบวนการสร้างสายคอนเซนซัสจากผลลัพธ์ (Consensus Sequence) [44] แล้วนำไปเปรียบเทียบกับสายคอนเซนซัสที่เป็นผลเฉลยจากกลุ่มข้อมูล
- กระบวนการนำผลลัพธ์ที่ได้เปรียบเทียบกับผลลัพธ์ดีเอ็นเอฟุตพรินติ้ง (DNA Footprinting) [36] ซึ่งเป็นตำแหน่งผลเฉลยจากกระบวนการทางชีววิทยา
- กระบวนการคำนวณหาค่าอินฟอร์เมชั่นคอนเทนท์ (Information Content) ของผลลัพธ์ [33] เพื่อการวัดประสิทธิภาพความแม่นยำของผลลัพธ์จากขั้นตอนวิธี
- กระบวนการใช้ค่าสถิติทดสอบที (*T*-values) เพื่อวัดการกระจายของข้อมูลผลลัพธ์จากการทดลองในแต่ละรอบ [45]

และกลุ่มที่สองเป็นการทดลองตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบอนุภาคร่วมกับระยะทางแฮมมิง (PSO_HD) โดยกระบวนการที่นำมาใช้ในการประเมินผลประกอบด้วย

- กระบวนการหาค่าความแม่นยำ (Precision) ของผลลัพธ์โดยสมการค่าความแม่นยำ [46]
- กระบวนการหาค่าความถูกต้อง (Recall) ของผลลัพธ์โดยสมการค่าความถูกต้อง [46]
- กระบวนการวัดประสิทธิภาพของผลลัพธ์เรื่องความแม่นยำและถูกต้องครอบคลุม โดยหาค่าเอฟสกอร์ (*F*-score) ของผลลัพธ์ [46]

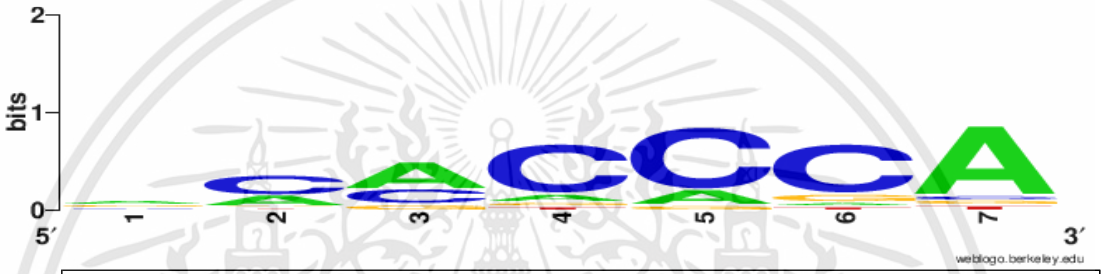
งานวิจัยนี้ได้ทำการตรวจสอบเวลาที่ใช้ในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสของขั้นตอนวิธีที่นำเสนอ ทั้งขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาคและขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบอนุภาคร่วมกับระยะทางแฮมมิง

4.1 การประเมินผลลัพธ์แบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้า

สายคอนเซนซัสจากผลลัพธ์ของขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาคเป็นผลลัพธ์จากการตรวจหาสายโมทีฟในสายจีโนมของกลุ่มข้อมูลสายจีโนมของยีสต์ (*Saccharomyces Cerevisiae*) ได้แก่ สายดีเอ็นเอ GAL4, RAP1, REB1, MCB และ PDR3 แสดงในรูปที่ 4.1 ถึงรูปที่ 4.5 ซึ่งวิเคราะห์แล้วพบว่าสายคอนเซนซัสจากผลลัพธ์ของขั้นตอนวิธีที่นำเสนอมีความใกล้เคียงกับสายคอนเซนซัสที่เป็นผลเฉลยของกลุ่มข้อมูล ดังตารางที่ 3.4 กล่าวโดยละเอียดในหัวข้อ 3.4.1

NexusPSO result					
1	CGGAGCACTGTTGAGCG	3	CGGAAGACTCTCCTCCG	5	CGGCGCACTCTCGCCCG
2	CGGAGCAGTGCGGCGCG	4	CGGGGCGGATCACTCCG	6	CGGGGCAGACTATTCCG

รูปที่ 4.1 ผลลัพธ์สายคอนเซนซัสจากผลลัพธ์ในกลุ่มสายดีเอ็นเอ GAL4



NexusPSO result					
1	ACACCCA	7	ACCCAGT	13	AAACCCA
2	GAACCCA	8	GAACCCA	14	CCGTGCA
3	ACACCCA	9	GCACCCA	15	GCACCCA
4	ACCCATG	10	TGCACCA	16	ACACCCA
5	GACACGC	11	CACCCAA		
6	CTGGACA	12	ACACCCA		

รูปที่ 4.2 ผลลัพธ์สายคอนเซนซัสจากผลลัพธ์ในกลุ่มสายดีเอ็นเอ RAP1

NexusPSO result					
1	TTACCCT	6	CCGGGCA	11	CTACCCG
2	TAACCCG	7	GGGTAGC	12	GTACGGG
3	TTACCCG	8	TCACCCG	13	CGGGTAA
4	TCACCCG	9	TTACCCG	14	TTACCCG
5	TTACCCG	10	CGGGTTA		

รูปที่ 4.3 ผลลัพธ์สายคอนเซนซัสจากผลลัพธ์ในกลุ่มสายดีเอ็นเอ REB1

NexusPSO result					
1	ACGCGT	3	ACGCGT	5	ACGCGT
2	ACGCGA	4	TCGCGA	6	ACGCGT

รูปที่ 4.4 ผลลัพธ์สายคอนเซนซัสจากผลลัพธ์ในกลุ่มสายดีเอ็นเอ MCB

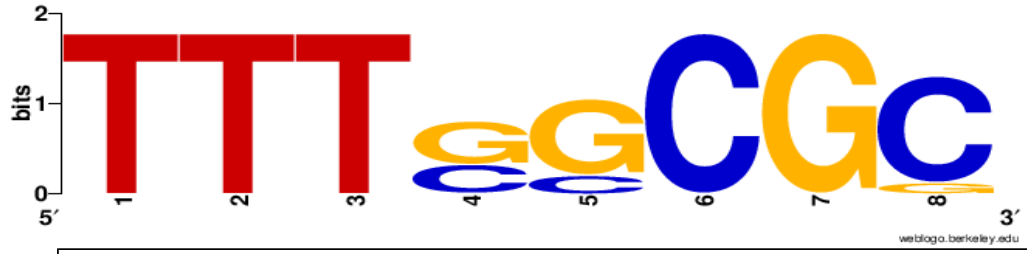
NexusPSO result					
1	TCCGCGGA	4	TCCGTGGA	6	TCCGTGGA
2	TCCGCGGA	5	TCCGTGGA	7	TCCGCGGA
3	TCCGCGGA				

รูปที่ 4.5 ผลลัพธ์สายคอนเซนซัสจากผลลัพธ์ในกลุ่มสายดีเอ็นเอ PDR3

และกลุ่มข้อมูลสายจีโนมของมนุษย์ (*Homo Sapiens*) ได้แก่สายดีเอ็นเอ ELK4, E2F1, FOXD1, USF1 และ RELA แสดงในรูปที่ 4.6 ถึงรูปที่ 4.10 ซึ่งวิเคราะห์แล้วพบว่าสายคอนเซนซัสจากผลลัพธ์ของขั้นตอนวิธีที่นำเสนอมีความใกล้เคียงกับสายคอนเซนซัสที่เป็นผลเฉลยของกลุ่มข้อมูล ดังตารางที่ 3.5 กล่าวโดยละเอียดในหัวข้อ 3.4.1

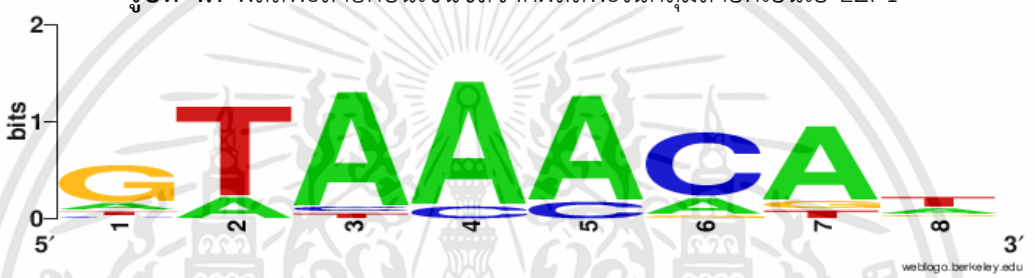
NexusPSO result							
1	ACCGGAAGT	6	ACCGGAAGT	11	ACCGGAAGC	16	GGATATAGC
2	ACCGGAAGT	7	ACCGGATGT	12	ACAACCGGA	17	ACCGGAAAC
3	ACCGGAAGT	8	ACCGGATGT	13	GCCGGAAGT	18	TCCGGAAGC
4	ACCGGAAGT	9	acACCGGAA	14	CCCGGAAGT	19	GCCGGAAAT
5	ACCGGAAGT	10	ACCGGAAGC	15	ACCGGATGC	20	ACCGGAACA

รูปที่ 4.6 ผลลัพธ์สายคอนเซนซัสจากผลลัพธ์ในกลุ่มสายดีเอ็นเอ ELK4



NexusPSO result	
1 TTTGGCGC	6 TTTCGCGC
2 TTTGGCGC	7 TTTCGCGC
3 TTTGGCGC	8 TTTGCCGC
4 TTTGGCGC	9 TTTCCCGC
5 TTTCGCGC	10 TTTGGCGG

รูปที่ 4.7 ผลลัพธ์สายคอนเซนซัสจากผลลัพธ์ในกลุ่มสายดีเอ็นเอ E2F1

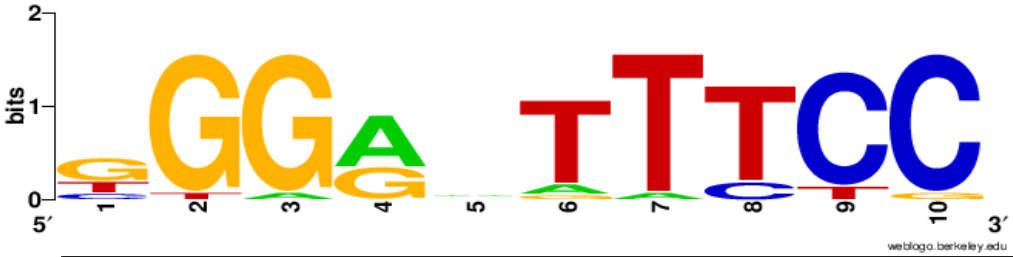


NexusPSO result			
1 GTAAACAT	6 GTAAACAT	11 TAAACAAT	16 ATAAACAA
2 GTAAACAT	7 GTAAACAT	12 GTAAACAA	17 CTAAACAG
3 GTAAACAT	8 GTAAACAA	13 AAACACGT	18 GTCAACAG
4 GTAAACAT	9 AAACAATG	14 GTAAACAC	19 GTAACAAT
5 GTAAACAT	10 GTAAACAA	15 TAAACAGA	20 gTTAAGTA

รูปที่ 4.8 ผลลัพธ์สายคอนเซนซัสจากผลลัพธ์ในกลุ่มสายดีเอ็นเอ FOXD1

NexusPSO result			
1 CACGTGG	9 CACGTGG	17 CACGTGA	25 CACGTGC
2 CACGTGG	10 CACGTGG	18 CACGTGA	26 TGTGGGA
3 CACGTGG	11 CACGTGG	19 CGTGTAA	27 CATGTGA
4 CACGTGG	12 CACGTGG	20 CACGTGT	28 CACATGA
5 CACGTGG	13 CACGTGA	21 CACGTGT	29 CACGCGG
6 CACGTGG	14 CACGTGA	22 CACGTGT	30 CACGGGA
7 CACGTGG	15 CACGTGA	23 CACGTGT	
8 CACGTGG	16 CACGTGA	24 CACGTGC	

รูปที่ 4.9 ผลลัพธ์สายคอนเซนซัสจากผลลัพธ์ในกลุ่มสายดีเอ็นเอ USF1



NexusPSO result			
1 GGGAATTTCC	6 GGAATTTCCG	11 CGGACTTTCC	16 TGGGGTTTTC
2 GGGAATTTCC	7 GGGGATTTCC	12 CGGACTTTCC	17 GGGGGATTCC
3 GGGAATTTCC	8 GGGACTTTCC	13 GGGGAATTCC	18 TGGGTTTCCc
4 TGGAATTTCC	9 CGGAGTTTCC	14 TGGGGTTTCC	
5 TGGAATTTCC	10 GGGAATTCCC	15 GTGGGGATTC	

รูปที่ 4.10 ผลลัพธ์สายคอนเซนซัสจากผลลัพธ์ในกลุ่มสายดีเอ็นเอ RELA

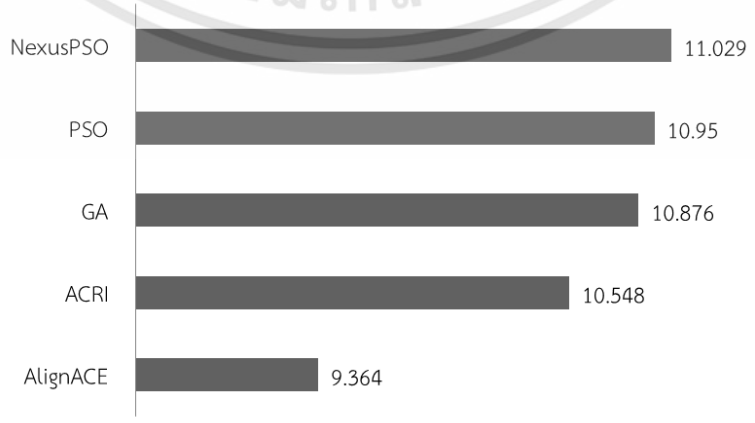
ตารางที่ 4.1 การเปรียบเทียบผลลัพธ์ขั้นตอนวิธีแบบเดิม ขั้นตอนวิธีที่เกี่ยวข้องและขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาค

No.	BS	Name	Traditional Algorithms						Related Works						NexusPSO	diff
			Gibbs Sampler	diff	Align ACE	diff	MEME	diff	GA	diff	PSO	diff	ACRI	diff		
1	17,61	CE1CG	59	2	63	2	61	0	62	1	61	0	63	2	61	0
2	17,55	ECOARABOP	53	2	57	2	55	0	56	1	55	0	57	2	55	0
3	76	ECOBGLR1	74	2	48	28	76	0	77	1	76	0	78	2	76	0
4	63	ECOCRP	59	4	65	2	63	0	64	1	63	0	65	2	63	0
5	50	ECOYA	11	39	52	2	13	37	51	1	50	0	52	2	50	0
6	7,60	ECODEOP2	5	2	9	2	7	0	8	1	7	0	9	2	7	0
7	42	ECOGALE	40	2	26	16	42	0	43	1	24	18	44	2	42	0
8	39	ECOILVBPR	37	2	41	2	39	0	40	1	39	0	41	2	39	0
9	9,80	ECOLAC	7	2	11	2	9	0	10	1	9	0	11	2	9	0
10	14	ECOMALBA	12	2	16	2	14	0	15	1	14	0	16	2	14	0
11	61	ECOMALBA2	59	2	63	2	35	26	62	1	61	0	63	2	61	0
12	41	ECOMALT	47	6	43	2	34	7	42	1	41	0	43	2	41	0
13	48	ECOOMPA	46	2	50	2	48	0	49	1	48	0	50	2	48	0
14	71	ECOTNAA	69	2	73	2	71	0	72	1	71	0	73	2	71	0
15	17	ECOXUL	15	2	19	2	75	58	18	1	17	0	19	2	17	0
16	53	PBR-P4	49	4	55	2	6	47	54	1	53	0	55	2	53	0
17	1,84	TRN9CAT	25	24	68	16	27	26	56	28	5	4	95	11	5	4
18	78	TDC	74	4	80	2	16	62	77	1	76	2	78	0	76	2

เมื่อทำการทดสอบขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาคกับกลุ่มข้อมูลอีโคไล (*Escherichia Coli*: *E.Coli*) [42] ทั้งหมด 18 ครั้งแล้วนำผลลัพธ์ที่อยู่ในระดับค่าเฉลี่ยไปเปรียบเทียบกับผลลัพธ์ขั้นตอนวิธีแบบดั้งเดิม (Traditional Algorithms) ได้แก่ขั้นตอนวิธีการสุ่มตัวอย่างแบบกิบส์ (Gibbs Sampler) [22] ขั้นตอนวิธีอไลเอซีอี (AlignACE) [5] และขั้นตอนวิธีค่าคาดหมายสูงสุด (MEME) [3] โดยนำตำแหน่งเริ่มต้นของโมทีฟจากแต่ละขั้นตอนวิธี มาเปรียบเทียบกับตำแหน่งของผลเฉลย (ผลเฉลยได้จากวิธีการดีเอ็นเอฟุตพรินติ้ง) ทำให้ได้ผลการทดลองดังตารางที่ 4.1 โดยที่ข้อมูล BS คือตำแหน่งสายโมทีฟที่เป็นผลเฉลย และ diff คือความแตกต่างระหว่างตำแหน่งที่เป็นผลลัพธ์ของขั้นตอนวิธีกับตำแหน่งของผลเฉลย ซึ่งขั้นตอนวิธีการสุ่มตัวอย่างแบบกิบส์มีผลลัพธ์ที่ไกลจากผลเฉลยถึง 39 ตำแหน่งในสายดีเอ็นเอลำดับที่ 5 (ECOYA) และไกลจากผลเฉลย 24 ตำแหน่งในสายดีเอ็นเอลำดับที่ 17 (TRN9CAT) ส่วนขั้นตอนวิธีอไลเอซีอีมีผลลัพธ์ที่ไกลจากผลเฉลย 16 ตำแหน่งในสายดีเอ็นเอลำดับที่ 7 (ECOGALE) และสายดีเอ็นเอลำดับที่ 17 (TRN9CAT) ซึ่งทั้งขั้นตอนวิธีการสุ่มตัวอย่างแบบกิบส์และขั้นตอนวิธีอไลเอซีอี ไม่มีผลลัพธ์สายโมทีฟใดที่ตรงกับผลเฉลย ในขณะที่ขั้นตอนวิธีค่าคาดหมายสูงสุดมีผลลัพธ์ที่ไกลจากผลเฉลย 37 ตำแหน่งในสายดีเอ็นเอลำดับที่ 5 (ECOYA) ไกลจากผลเฉลย 26 ตำแหน่งในสายดีเอ็นเอลำดับที่ 11 (ECOMALBA2) และห่างจากผลเฉลย 58, 47, 26 และ 62 ตำแหน่ง ในสายดีเอ็นเอลำดับที่ 15 (ECOXUL) ลำดับที่ 16 (PBR-P4) ลำดับที่ 17 (TRN9CAT) และลำดับที่ 18 (TDC) ตามลำดับ ในขณะที่มีสายโมทีฟถึง 11 สายที่ตรงผลเฉลย

สำหรับผลลัพธ์สายโมทีฟของขั้นตอนวิธีเชิงพันธุกรรม (GA) ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (PSO) และขั้นตอนวิธีเอซีอาร์ไอ (ACRI) ในตารางที่ 4.1 ของสายดีเอ็นเอลำดับที่ 17 (TRN9CAT) ยังคงห่างจากผลเฉลย 28, 4 และ 11 ตำแหน่งตามลำดับ ในขณะที่ผลลัพธ์ตำแหน่งสายโมทีฟในสายดีเอ็นเออื่นๆ ใกล้เคียงกับผลเฉลย จึงระบุได้ว่าขั้นตอนวิธีเหล่านี้ยังไม่สามารถตรวจหาสายโมทีฟในสายดีเอ็นเอลำดับที่ 17 (TRN9CAT) ได้อย่างแม่นยำนัก

สำหรับขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาคซึ่งเป็นขั้นตอนวิธีที่นำเสนอ สามารถตรวจหาสายโมทีฟในสายดีเอ็นเอลำดับที่ 17 (TRN9CAT) ได้ใกล้เคียงที่สุด โดยผิดพลาดจากผลเฉลยไปเพียง 4 ตำแหน่ง และสามารถตรวจหาสายโมทีฟที่ตรงกับผลเฉลยได้ทั้งหมด 16 สายจากสายข้อมูลที่นำเข้าทั้งหมด 18 สาย ส่งผลให้ค่าอินฟอร์เมชั่นคอนเทนท์ของผลลัพธ์จากขั้นตอนวิธีที่นำเสนอมีค่าสูงสุดตามรูปที่ 4.11



รูปที่ 4.11 กราฟเปรียบเทียบค่าอินฟอร์เมชั่นคอนเทนท์

โดยข้อมูลในตารางที่ 4.2 แสดงค่าอินฟอร์เมชั่นคอนเทนท์ในการทดลอง 18 ครั้ง ระหว่างขั้นตอนวิธีค่าคาดหมายสูงสุด (MEME) ขั้นตอนวิธีอไลเอซีอี (AlignACE) ขั้นตอนวิธีเอซีอาร์ไอ (ACRI) และขั้นตอนวิธีที่นำเสนอ (NexusPSO) พร้อมกับค่าเฉลี่ย (Average) ของแต่ละขั้นตอนวิธีในการทดลอง 18 ครั้ง ซึ่งขอบเขตเวลาการดำเนินการของขั้นตอนวิธีที่นำเสนอในการทดลอง 18 ครั้งอยู่ระหว่าง 980 ถึง 1650 มิลลิวินาที

ตารางที่ 4.2 เปรียบเทียบค่าอินฟอร์เมชั่นคอนเทนท์ในการทดลอง 18 ครั้ง

No.	MEME	AlignACE	ACRI	NexusPSO
1	10.032	9.651	10.01	11.045
2	9.075	9.887	10.28	11.804
3	10.02	9.576	9.987	11.018
4	10.05	9.624	10.403	10.946
5	9.117	10.235	10.457	10.354
6	9.892	9.71	10.184	11.934
7	9.554	9.01	9.895	11.005
8	10.124	9.934	10.258	11.112
9	9.646	9.807	10.354	10.124
10	9.439	9.853	10.421	11.053
11	9.121	10.12	10.53	10.984
12	9.16	9.399	10.415	10.852
13	9.684	9.976	10.38	11.074
14	9.773	9.825	10.286	11.04
15	9.024	9.769	10.179	11.206
16	9.008	10.314	10.3	11.704
17	9.105	9.011	10.14	11.029
18	9.32	9.835	10.431	10.254

สำหรับการตรวจสอบการกระจายของข้อมูลผลลัพธ์จากการทดลองในแต่ละรอบของขั้นตอนวิธีที่นำเสนอ งานวิจัยนี้ใช้ค่าสถิติทดสอบที (*T*-values) ซึ่งเป็นการเปรียบเทียบการกระจายตัวของผลลัพธ์จากการทดลองหลายๆ ครั้ง ระหว่างขั้นตอนวิธีที่นำเสนอและขั้นตอนวิธีอื่นๆ ในกรณีที่ค่าสถิติทดสอบทีสูงแสดงให้เห็นว่าผลลัพธ์จากการทดลองหลายๆ ครั้ง ระหว่างขั้นตอนวิธีมีนัยสำคัญต่อกัน ซึ่งผลลัพธ์ค่าสถิติทดสอบทีแสดงในตารางที่ 4.3 กำหนดให้ค่าสถิติทดสอบทีจากการทดลอง 18 ครั้ง (18 Samples) มีค่าความอิสระ(The Degree of Freedom) อยู่ที่ 18+18-2 = 34 (n+n-2 เป็นสมการมาตรฐานในการคำนวณค่าอิสระ) และกำหนดค่าระดับความมีนัยสำคัญ (Significance Level: **α**) เป็น **α** = 0.05 (**α** = 0.05 หมายถึงการกระจายตัวของข้อมูลอยู่ในระดับความน่าเชื่อถือที่

95 เปอร์เซ็น) ซึ่งผลลัพธ์ $t0.95(34) = 1.691$ จากวิเคราะห์ค่าสถิติทดสอบทีในตารางที่ 4.3 พบว่าค่าสถิติทดสอบทีของขั้นตอนวิธีที่นำเสนอมีค่ามากกว่าขั้นตอนวิธีที่เกี่ยวข้อง ที่ $t0.95(34)$ แสดงให้เห็นว่าค่าเฉลี่ยของผลลัพธ์ของขั้นตอนวิธีที่นำเสนอ มีการกระจายตัวอย่างมีนัยสำคัญกับขั้นตอนวิธีที่เกี่ยวข้อง

ตารางที่ 4.3 ค่าสถิติทดสอบทีระหว่างขั้นตอนวิธีที่นำเสนอกับขั้นตอนวิธีที่เกี่ยวข้อง

t -value	MEME	AlignACE	ACRI
compared with NexusPSO	10.354	9.181	6.34

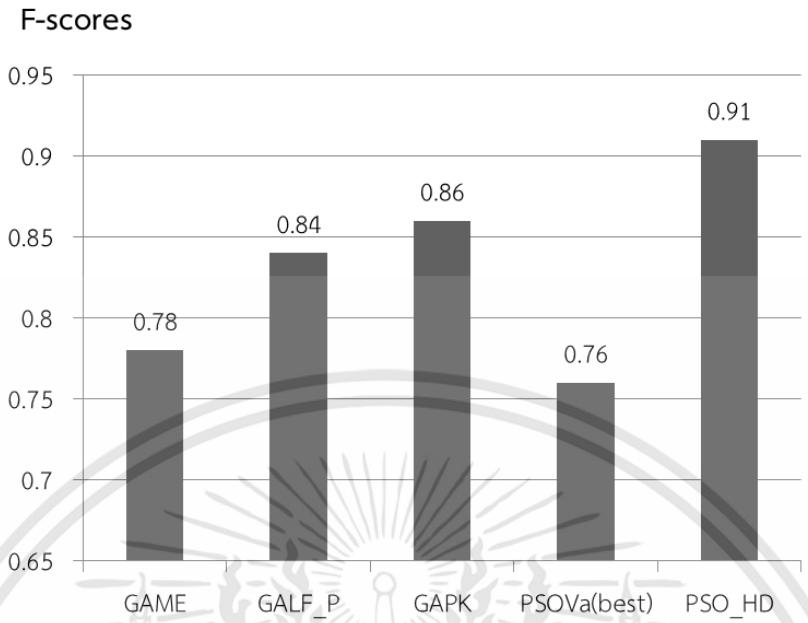
4.2 การประเมินผลลัพธ์แบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า

งานวิจัยนี้ใช้ค่าเอฟสกอร์ ซึ่งนิยมนำมาใช้วัดประสิทธิภาพความแม่นยำและความถูกต้องครอบคลุมของผลลัพธ์ ในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า [18][19][20][21][27] คำนวณตามสมการ (2.22) ซึ่งค่าเอฟสกอร์เป็นค่าที่พิจารณาทั้งความแม่นยำ (Precision) และความถูกต้องครอบคลุม (Recall) กล่าวโดยละเอียดในหัวข้อ 2.11 และผลการวิจัยนำค่าเหล่านี้มาใช้ในการวัดประสิทธิภาพระหว่าง ผลลัพธ์ของขั้นตอนวิธีที่นำเสนอและขั้นตอนวิธีที่เกี่ยวข้อง

งานวิจัยนี้ทำการทดลองขั้นตอนวิธีทั้งหมด 20 ครั้ง ตามรูปแบบการทดลองของงานที่เกี่ยวข้อง [18][19][20][21] แล้วนำค่าเอฟสกอร์สูงสุดจากการทดลอง 20 ครั้งของขั้นตอนวิธีที่นำเสนอ มาเปรียบเทียบค่าที่ดีที่สุดของขั้นตอนวิธีเชิงพันธุกรรม (GA) ทดสอบโดย D. Wang และ X. Li [20] และค่าที่ดีที่สุดของขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบประยุกต์ (PSOVa) ทดสอบโดย M. Karabulut และ T. Ibrikci [17] พบว่าขั้นตอนวิธีที่นำเสนอมีค่าเอฟสกอร์สูงสุดในกลุ่มข้อมูล 6 กลุ่มจาก 8 กลุ่ม ในขณะที่กลุ่มข้อมูลอีก 2 กลุ่ม ได้แก่ ERE และ MEF มีค่าเอฟสกอร์อยู่ที่ 0.85 และ 0.98 ตามลำดับซึ่งสูงสุดเป็นอันดับที่สองดังรายละเอียดในตารางที่ 4.4 อย่างไรก็ตามเมื่อนำค่าเอฟสกอร์สูงสุดในทุกกลุ่มข้อมูลมาหาค่าเฉลี่ย พบว่าขั้นตอนวิธีที่นำเสนอมีค่าเฉลี่ยสูงสุดดังรูปที่ 4.12

ตารางที่ 4.4 ค่าเอฟสกอร์ของขั้นตอนวิธีที่นำเสนอเปรียบเทียบกับขั้นตอนวิธีที่เกี่ยวข้อง

Algorithm	CREB	CRP	ERE	E2F	MEF	MYOD	SRF	TBP	Average
PSO_HD	**0.90**	**0.92**	0.85	**0.92**	0.98	**0.91**	**0.91**	**0.89**	0.91
PSOVa(best)	0.72	0.86	0.82	0.67	0.91	0.43	0.79	0.84	0.76
GAPK	0.73	0.91	**0.92**	0.82	**1.00**	0.83	0.74	0.88	0.86
GALF_P	0.73	0.91	0.79	0.85	0.97	0.71	0.85	0.84	0.84
GAME	0.71	0.88	0.76	0.79	0.97	0.44	0.83	0.77	0.78



รูปที่ 4.12 กราฟแท่งเปรียบเทียบค่าเอฟสกอร์เฉลี่ยในกลุ่มข้อมูลทั้ง 8 กลุ่ม

เมื่อนำรายละเอียดการทดลองทั้ง 20 ครั้งในกลุ่มข้อมูล 8 กลุ่ม มาเปรียบเทียบค่าความแม่นยำ ค่าความถูกต้องครอบคลุมและค่าเอฟสกอร์ ระหว่างขั้นตอนวิธีที่นำเสนอและขั้นตอนวิธีเชิงพันธุกรรมที่ทดสอบโดย D. Wang และ X. Li [21] ปรากฏว่าผลลัพธ์ของขั้นตอนวิธีที่นำเสนอให้ค่าเอฟสกอร์เฉลี่ยสูงสุดอยู่ที่ 0.89 ดังตารางที่ 4.5 ซึ่งมี 7 กลุ่มข้อมูลที่ขั้นตอนวิธีที่นำเสนอมีค่าเอฟสกอร์สูงสุด ในขณะที่กลุ่มข้อมูล CRP ค่าเอฟสกอร์อยู่ในลำดับที่สองรองจากขั้นตอนวิธี GALF-P [19] และค่าเบี่ยงเบนมาตรฐานของค่าเอฟสกอร์ของขั้นตอนวิธีที่นำเสนออยู่ที่ 0.02 ถึง 0.08

ตารางที่ 4.5 เปรียบเทียบค่าเอฟสกอร์ (*F*-score) และค่าเบี่ยงเบนมาตรฐาน

Algorithm		CREB	CRP	ERE	E2F	MEF	MYOD	SRF	TBP	Average
PSO_HD	*F* -score	**0.90**	0.89	**0.79**	**0.89**	**0.97**	**0.88**	**0.91**	**0.89**	0.89
	±	0.05	0.03	0.07	0.08	0.02	0.06	0.02	0.02	
iGAPK	*F* -score	0.66	0.87	0.79	0.75	0.89	0.87	0.78	0.77	0.80
	±	0.06	0.02	0.11	0.03	0.06	0.05	0.03	0.06	
GAPK	*F* -score	0.66	0.86	0.78	0.77	0.87	0.74	0.67	0.83	0.77
	±	0.04	0.04	0.10	0.02	0.03	0.09	0.04	0.06	
GALF-P	*F* -score	0.53	**0.91**	0.72	0.78	0.89	0.36	0.76	0.80	0.72
	±	0.26	0.04	0.10	0.07	0.11	0.32	0.09	0.09	
GAME	*F* -score	0.43	0.88	0.72	0.72	0.93	0.24	0.78	0.62	0.67
	±	0.32	0.03	0.06	0.06	0.04	0.16	0.06	0.25	

โดยข้อมูลตารางที่ 4.6 แสดงค่าความแม่นยำของขั้นตอนวิธีที่นำเสนอสูงสุด 6 กลุ่ม ประกอบด้วย CREB, ERE, E2F, MYOD, SRF และ TBP แต่ข้อมูลตารางที่ 4.7 แสดงค่าความถูกต้องครอบคลุมของแนวคิดที่นำเสนอสูงสุดเพียง 3 กลุ่มประกอบด้วย CREB, MEF และ TBP อย่างไรก็ตามเมื่อ

เปรียบเทียบค่าเฉลี่ยของค่าความแม่นยำและค่าความถูกต้องจากกลุ่มข้อมูลทั้ง 8 กลุ่ม พบว่าขั้นตอนวิธีที่นำเสนอให้ค่าเฉลี่ยของค่าความแม่นยำและค่าความถูกต้องครอบคลุมสูงสุด อยู่ที่ 0.90 และ 0.88 ตามลำดับ พร้อมกับแสดงค่าเบี่ยงเบนมาตรฐานของค่าความแม่นยำอยู่ที่ 0.03 ถึง 0.08 และค่าเบี่ยงเบนมาตรฐานของค่าความถูกต้องครอบคลุมอยู่ที่ 0.02 ถึง 0.09 ตามลำดับ ดังตารางที่ 4.6 และ 4.7

ตารางที่ 4.6 เปรียบเทียบค่าความแม่นยำ (Precision) และค่าเบี่ยงเบนมาตรฐาน

Algorithm		CREB	CRP	ERE	E2F	MEF	MYOD	SRF	TBP	Average
PSO_HD	Precision	**0.87**	0.94	**0.79**	**0.90**	0.95	**0.92**	**0.96**	**0.88**	0.90
	±	0.04	0.05	0.08	0.08	0.03	0.06	0.04	0.03	
iGAPK	Precision	0.68	0.90	0.73	0.69	0.87	0.83	0.75	0.73	0.77
	±	0.06	0.05	0.15	0.02	0.10	0.06	0.04	0.10	
GAPK	Precision	0.65	**0.96**	0.71	0.66	**1.00**	0.68	0.63	0.83	0.77
	±	0.04	0.06	0.15	0.02	0.00	0.14	0.05	0.06	
GALF-P	Precision	0.47	0.95	0.65	0.67	0.85	0.28	0.68	0.74	0.66
	±	0.24	0.02	0.15	0.08	0.16	0.24	0.12	0.12	
GAME	Precision	0.44	0.93	0.63	0.62	0.90	0.24	0.67	0.67	0.64
	±	0.31	0.05	0.07	0.05	0.05	0.17	0.06	0.28	

ตารางที่ 4.7 เปรียบเทียบค่าความถูกต้องครอบคลุม (Recall) และค่าเบี่ยงเบนมาตรฐาน

Algorithm		CREB	CRP	ERE	E2F	MEF	MYOD	SRF	TBP	Average
PSO_HD	Recall	**0.94**	0.85	0.80	0.87	**0.99**	0.86	0.86	**0.90**	0.88
	±	0.06	0.04	0.07	0.09	0.02	0.08	0.02	0.02	
iGAPK	Recall	0.65	0.84	**0.88**	0.83	0.92	**0.92**	0.81	0.83	0.84
	±	0.06	0.03	0.03	0.06	0.04	0.08	0.05	0.04	
GAPK	Recall	0.68	0.80	0.89	0.92	0.77	0.82	0.74	0.83	0.81
	±	0.07	0.04	0.06	0.05	0.05	0.06	0.05	0.06	
GALF-P	Recall	0.60	**0.88**	0.84	**0.93**	0.94	0.51	0.88	0.86	0.81
	±	0.29	0.05	0.04	0.05	0.06	0.45	0.06	0.02	
GAME	Recall	0.43	0.84	0.84	0.86	0.96	0.24	**0.92**	0.58	0.71
	±	0.30	0.03	0.06	0.09	0.06	0.16	0.06	0.24	

เมื่อพิจารณาการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้าของขั้นตอนวิธีที่นำเสนอ จากการทดลอง 20 ครั้งในกลุ่มข้อมูล 8 กลุ่ม พบว่าเวลาในการดำเนินการของขั้นตอนวิธีที่นำเสนอใช้เวลาในกลุ่มข้อมูล MEF2 940 ถึง 1023 มิลลิวินาที ใช้เวลาในกลุ่มข้อมูล MYOD 1000 ถึง 1110 มิลลิวินาที ใช้เวลาในกลุ่มข้อมูล E2F 1150 ถึง 1297 มิลลิวินาที ใช้เวลาในกลุ่มข้อมูล ERE 1180 ถึง 1378 มิลลิวินาที ใช้เวลาในกลุ่มข้อมูล CREB 1250 ถึง 1462 มิลลิวินาที ใช้เวลาในกลุ่มข้อมูล SRF 1300 ถึง 1400 มิลลิวินาที ใช้เวลาในกลุ่มข้อมูล TBP 1500 ถึง 1698 มิลลิวินาที และใช้เวลาในกลุ่มข้อมูล CRP 1650 ถึง 2000 มิลลิวินาที ดังตารางที่ 4.8 ซึ่ง

สังเกตได้ว่าเวลาที่ใช้ในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสในกลุ่มข้อมูล CRP มากที่สุด เนื่องจากรูปแบบข้อมูลมีความหลากหลายและสายโมทีฟมีขนาดยาว

ตารางที่ 4.8 เปรียบเทียบเวลาในการทดลอง 20 ครั้งในกลุ่มข้อมูลทั้ง 8 กลุ่ม

Time	MEF2	MYOD	E2F	ERE	CREB	SRF	TBP	CRP
Mininum Time (Millisecond)	940	1000	1150	1180	1250	1300	1500	1650
Maximum Time (Millisecond)	1023	1110	1297	1378	1462	1400	1698	2000

4.3 อภิปรายผลการทดลองของขั้นตอนวิธีที่นำเสนอ

สำหรับผลลัพธ์สายคอนเซนซัสของขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาคในกลุ่มข้อมูลทั้ง 10 กลุ่ม สังเกตได้ว่าให้ผลลัพธ์ของขั้นตอนวิธีที่นำเสนอตรงกับสายคอนเซนซัสที่เป็นผลเฉลยดังรูปที่ 4.1 ถึง 4.10 พร้อมกับความแม่นยำของขั้นตอนวิธีที่นำเสนอที่สูงกว่าขั้นตอนวิธีที่เกี่ยวข้อง โดยเฉพาะอย่างยิ่งสายดีเอ็นเอลำดับที่ 17 ในกลุ่มข้อมูลอีโคไล แนวคิดที่นำเสนอสามารถหาผลลัพธ์ได้ใกล้เคียงกับตำแหน่งผลเฉลยมากที่สุดดังตารางที่ 4.1 และค่าสถิติทดสอบที เพื่อวัดการกระจายของผลลัพธ์จากการทดลองทั้งหมด 18 ครั้ง ระหว่างขั้นตอนวิธีที่นำเสนอกับขั้นวิธีที่เกี่ยวข้อง จากผลการเปรียบเทียบแสดงให้เห็นว่าขั้นตอนวิธีที่นำเสนอมีการกระจายของข้อมูลที่มีนัยสำคัญต่อขั้นตอนวิธีที่เกี่ยวข้อง

สำหรับการเปรียบเทียบค่าเอฟสกอร์ของขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบอนุภาคร่วมกับระยะทางแฮมมิงและขั้นตอนวิธีที่เกี่ยวข้องในกลุ่มข้อมูลทั้ง 8 กลุ่ม พบว่าขั้นตอนวิธีที่นำเสนอมีค่าเอฟสกอร์ต่ำกว่าขั้นตอนวิธี GALF-P ในกลุ่มข้อมูล CRP เพียงกลุ่มเดียว แต่มีค่าเอฟสกอร์สูงสุดในกลุ่มข้อมูลอีก 7 กลุ่มที่เหลือได้แก่ CREB, ERE, E2F, MEF, MYOD, SRF และ TBP อย่างไรก็ตามขั้นตอนวิธีที่นำเสนอยังคงมีค่าเฉลี่ยเอฟสกอร์สูงสุดดังตารางที่ 4.5 แสดงให้เห็นว่าขั้นตอนวิธีที่นำเสนอสามารถพัฒนาการตราวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสได้แม่นยำและครอบคลุมมากขึ้น

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้ได้เสนอขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาค (NexusPSO) ซึ่งเป็นการนำขั้นตอนสานสัมพันธ์ที่คิดขึ้นใหม่มาประยุกต์เข้ากับขั้นตอนวิธีการหาหน่วยย่อยที่เหมาะสมแบบกลุ่มอนุภาค (Particle Swarm Optimization: PSO) เพื่อตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหนึ่งสายโมทีฟต่อสายข้อมูลที่นำเข้าอย่างมีประสิทธิภาพและแม่นยำขึ้น โดยขั้นตอนสานสัมพันธ์สามารถจัดการพื้นที่ของปัญหา (Problem Space) ให้เล็กลงแต่ยังคงข้อมูลที่ถูกต้องไว้ จึงช่วยให้ขั้นตอนการสุ่มของขั้นตอนวิธีสามารถหลีกเลี่ยงผลลัพธ์ที่ดีแต่ไม่ถูกต้องได้ (Local Optimums) ซึ่งจากผลการทดลองแสดงให้เห็นว่าขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบสานสัมพันธ์ในกลุ่มอนุภาค มีความแม่นยำในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสสูงกว่าขั้นตอนวิธีอื่นๆ ที่เกี่ยวข้อง พิจารณาจากค่าอินฟอร์เมชั่นคอนเทนท์ (Information Content : IC) ของขั้นตอนวิธีที่นำเสนออยู่ที่ 11.029 ที่สูงกว่าขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค [17] ขั้นตอนวิธีเชิงพันธุกรรม [25] และขั้นตอนวิธีการหาค่าเหมาะสมที่สุดด้วยระบบอาณาจักรมด [15] ซึ่งมีค่าอินฟอร์เมชั่นคอนเทนท์ 10.95, 10.876 และ 10.548 ตามลำดับ เมื่อพิจารณาค่าสถิติทดสอบที (*T*-test) พบว่าขั้นตอนวิธีที่นำเสนอมีค่าที (*T*-value) จากการทดสอบ 18 ครั้ง สูงกว่าขั้นตอนวิธีที่เกี่ยวข้องอย่างมีนัยสำคัญ โดยมีค่านัยสำคัญอยู่ที่ 95 เปอร์เซ็น และมีการนำสายคอนเซนซัสที่เป็นผลลัพธ์จากขั้นตอนวิธีที่นำเสนอไปเปรียบเทียบกับสายคอนเซนซัสที่เป็นผลลัพธ์จากวิธีการดีเอ็นเอฟุตพรินติ้ง (DNA Footprinting) ซึ่งเป็นวิธีการทางด้านชีววิทยาที่เชื่อถือได้ จากการเปรียบเทียบพบว่าผลลัพธ์สายคอนเซนซัสของขั้นตอนวิธีที่นำเสนอตรงกลับสายคอนเซนซัสที่เป็นผลลัพธ์จากวิธีการดีเอ็นเอฟุตพรินติ้ง ซึ่งแสดงให้เห็นว่าผลลัพธ์จากขั้นตอนวิธีที่นำเสนอมีประสิทธิภาพดี

สำหรับการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้า งานวิจัยนี้ได้นำเสนอขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบอนุภาคร่วมกับระยะทางแฮมมิง (PSO_HD) โดยประยุกต์ขั้นตอนวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคร่วมกับแนวคิดระยะทางแฮมมิง ซึ่งมีวัตถุประสงค์หลักคือ พัฒนาการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสแบบหลายสายโมทีฟต่อสายข้อมูลที่นำเข้าที่มีความแม่นยำและครอบคลุมในกลุ่มข้อมูลที่หลากหลาย ซึ่งจากการใช้ค่าความแม่นยำ (Precision) ค่าความถูกต้องครอบคลุม (Recall) และค่าเอฟสกอร์ (*F*-score) เพื่อวัดประสิทธิภาพของผลลัพธ์จากตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสของขั้นตอนวิธีที่นำเสนอ ในกลุ่มข้อมูล 8 กลุ่ม พบว่าขั้นตอนวิธีที่นำเสนอให้ค่าเอฟสกอร์สูงสุด 7 กลุ่มข้อมูล ยกเว้นผลลัพธ์ในกลุ่มข้อมูล CRP อย่างไรก็ตามค่าเอฟสกอร์เฉลี่ยในทุกกลุ่มข้อมูลของขั้นตอนวิธีที่นำเสนอยังคงมีค่าสูงสุด จึงสามารถสรุปได้ว่าขั้นตอนวิธีที่นำเสนอสามารถพัฒนาประสิทธิภาพการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสที่แม่นยำและถูกต้องครอบคลุมมากขึ้น

5.2 ข้อเสนอแนะ

ภายใต้ทรัพยากรเครื่องคอมพิวเตอร์ที่กำหนดไว้ ขั้นตอนวิธีที่นำเสนอใช้เวลาดำเนินการได้ไม่ดีนักเมื่อเทียบกับขั้นตอนวิธีอื่นๆ ดังนั้นการเพิ่มความรวดเร็วในการตรวจหาส่วนที่ยึดจับปัจจัยการถอดรหัสของขั้นตอนวิธียังคงจำเป็นต่อการศึกษาและพัฒนาต่อไป โดยเฉพาะอย่างยิ่งในกลุ่มของสิ่งมีชีวิตที่มีสายจีโนมขนาดยาว ซึ่งงานวิจัยในอนาคตคือการประยุกต์การประมวลผลแบบคู่ขนานเพื่อเพิ่มความรวดเร็วในการดำเนินการมากขึ้น

เอกสารอ้างอิง

[1] L. Elnitski, V.X. Jin, P.J. Farnham, S.J.M. Jones, “Locating mammalian transcription factor binding sites: a survey of computational and experimental techniques,” *Genome Research*, vol.16, pp. 1455–1464, 2006.

[2] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, J.C. Wootton, “Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignmzent,” *Science*, vol. 262, pp. 208-214, 1993.

[3] T.L. Bailey, C. Elkan, “Fitting a mixture model by expectation maximization to discover motifs in biopolymers,” *Proceedings of the International Conference on Intelligent System for Molecular Biology*, vol. 15, pp. 28-36, 1994.

[4] G.Z. Hertz, G.D. Stormo, “Identifying DNA and protein patterns with statistically significant alignments of multiple sequences,” *Bioinformatics*, vol. 15, no. 7, pp. 563–577, 1999.

[5] J.D. Hughes, P.W. Estep, S. Tavazoie, G.M. Church, “Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae,” *Journal of Molecular Biology*, vol. 296, pp. 1205–1214, 2000.

[6] X. Liu, D.L. Brutlag, J.S. Liu, “BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes,” *Pacific Symposium on Biocomputing*, vol. 6, pp. 127–138, 2001.

[7] X.S. Liu, D.L. Brutlag, J.S. Liu, “An algorithm for finding protein–DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments,” *Nature Biotechnology*, vol. 20, pp. 835–839, 2002.

[8] K. Socha, M. Dorigo, “Ant colony optimization for continuous domains,” *European Journal Operation Research*, vol. 185, no. 3, pp. 1155–1173, 2008.

[9] D.D. Duc, H.Q. Dinh, H.H. Xuan, “On the pheromone update rules of ant colony optimization approaches for the job shop scheduling problem,” *in: Proceedings of the 11th Pacific Rim International Conference on Multi-Agents*, *Intelligent Agents and Multi-Agent Systems*, vol. 5357, pp. 153–160, 2008.

[10] V. Maniezzo, A. Carbonaro, “An ANTS heuristic for the frequency assignment problem,” *Future Generation Computer Syst*ems, vol. 16, no. 8, pp. 927–935, 2000.

[11] S.J. Shyu, C.Y. Tsai, “Finding the longest common subsequence for multiple biological sequences by ant colony optimization,” *Computer Operation Research*, vol.36, no. 1, pp. 73-91, 2009.

[12] R. Poli, “Analysis of the publications on the applications of particle swarm optimisation,” *Journal of Artificial Evaluation and Applications*, pp. 1–10, 2008.

[13] J. Kennedy, R. Eberhart, "Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks," *IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[14] M. Dorigo, V. Maniezzo and A. Colomi, "Ant system: optimization by a colony of coorperating agents," *IEEE Transaction on System and Cybernetics Part B*, vol. 26, no. 1, pp. 29–41, 1996.

[15] W. Liu, H. Chen, L. Chen, "ACRI: an ant colony optimization based algorithm for identifying gene regulatory elements," *Computer in Biology and Medicine*, vol. 43, pp. 922-932, 2013.

[16] W. Zhou, C. Zhou, Guixia Liu and Yanxin Huang, "Identification of Transcription Factor Binding Sites Using Particle Swarm Optimization," *The Tenth International Conference on Rough Sets and Granular Computing 2005*, pp. 438–445, 2005.

[17] M. Karabulut, T. Ibrikci, "A Bayesian Scoring Scheme based Particle Swarm Optimization algorithm to identify transcription factor binding sites," *Applied Soft Computing*, vol. 12, pp. 2846-2855, 2012.

[18] Z. Wei and S.T. Jensen, "GAME: detecting cis-regulatory elements using a genetic algorithm," *Bioinformatics*, vol. 22, no. 13, pp. 1577–1584, 2006.

[19] T.M. Chan, K.S. Leung, and K.-H. Lee, "TFBS identification based on genetic algorithm with combined representations and adaptive post-processing," *Bioinformatics*, vol. 24, no. 3, pp. 341–349, 2008.

[20] D.H. Wang and X. Li, "GAPK: Genetic algorithms with prior knowledge for motif discovery in DNA sequences," *IEEE Congress on Evolutionary Computation 2009*, pp. 277–284, 2009.

[21] D.H. Wang and X. Li, "iGAPK: Improved GAPK algorithm for regulatory DNA motif discovery," *Neural Information Processing*, pp. 217–225, 2010.

[22] A.F. Neuwald, J.S. Liu, C.E. Lawrence, "Gibbs motif sampling: detection of bacterial outer membrane protein repeats," *Protein Science*, vol. 4, no. 8, pp. 1618–1632, 2004.

[23] J.D. Hughes, P.W. Estep, S. Tavazoie and G.M. Church, "Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae," *Journal of Molecular Biology*, vol. 296, pp. 1205–1214, 2000.

[24] J.H. Holland, B. Blocks, "Cohort Genetic Algorithms and Hyperplane-Defined Functions" *Evolutionary Computation*, pp. 373–391, 2000

[25] D. Che, Y. Song and K. Rasheed, "MDGA: motif discovery using a genetic algorithm," *Genetic and Evolutionary Computation (GECCO 2005)*, pp. 447–452, 2005.

[26] W. Liu, H. Chen, L. Chen, "ACRI: an ant colony optimization based algorithm for identifying gene regulatory elements," *Computer in Biology and Medicine*, vol. 43, pp. 922–932, 2013.

[27] M. Stine, D. Dasgupta and S. Mukatira., "Motif discovery in upstream sequences of coordinately expressed genes," *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003*, pp.1596–1603, 2003.

[28] M. S. Mohamad, S. Omatu, S. Deris, M. Yoshioka, A. Abdullah, and Z. Ibrahim, "An enhancement of binary particle swarm optimization for gene selection in classifying cancer classes," *Algorithms for Molecular Biology*, vol. 8, no. 4, pp.1584–1618, 2013.

[29] N. Jin and Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multi-objective implementations," *IEEE Transaction Antennas Propagation*, vol. 55, no. 3, pp. 556–567, 2007.

[30] J. Kennedy, R. Mendes, "Population structure and particle swarm performance, in: Proceedings of the Evolutionary Computation on 2002. CEC '02, Proceedings of the 2002 Congress – Vol. 02," *IEEE Computer Society*, vol. 02, pp. 1671–1676, 2002.

[31] S. Rahmann, T. Buller, and M. Vingron, "On the power of profiles for transcription factor binding site detection," *Statistical Applications in Genetics and Molecular Biology*, vol. 2, no. 1, pp. 1544–6115, 2003.

[32] J.M. Claverie and S. Audic, "The statistical significance of nucleotide position-weight matrix matches," *Computer Applications in the Biosciences*, vol. 12, no. 5, pp. 431-439, 1996.

[33] G.D. Stormo and D. S. Fields, "Specificity, free energy and information content in protein-DNA interactions," *Trends in Biochemical Sciences*, vol. 23, no. 3, pp. 109-113, 1998.

[34] J.B. Reece, L.A. Cain and M.L. Wasserman "Building a structural molecule of DNA," *Campbell biology*, pp. 316–318, 2011.

[35] A.E. Kel, E. Go¨ßling, I. Reuter, E. Cheremushkin, O.V. Kel-Margoulis and E. Wingender "MATCHTM: a tool for searching transcription factor binding sites in DNA sequences Nucleic Acids Research," *Nucleic Acids Res.*, vol. 31, no. 13, pp. 3576–3579, 2003.

[36] G.D. Storm, "Computer methods for analyzing sequence recognition of nucleic acids," *Annual Review of Biochemistry*, vol. 17, pp. 241–263, 1988.

[37] S. Jensen, S. Liu, Q. Zhou and J. Liu "Computational discovery of gene regulatory binding motifs: a Bayesian perspective," *Statistical Science*, no. 19, pp. 188–204, 2004.

[38] M. Mandal, J. Mondal, and A. Mukhopadhyay, “A PSO-Based approach for pathway maker Identification from gene expression data,” *IEEE Transactions on Nanobioscience*, vol. 14, no. 6, pp. 591–597, 2015.

[39] J. Zhu, M.Q. Zhang, “SCPD: a promoter database of the yeast Saccharomyces cerevisiae,” *Bioinformatics*, vol. 15, no. 7–8, pp. 607–611, 1999.

[40] J.C. Bryne, E. Valen, et al., “JASPAR: the open access database of transcription factor-binding profiles: new content and tools in the 2008 update,” *Nucleic Acids Res.*, vol. 36, pp. 102–106, 2008.

[41] V. Matys, O.V. Kel-Margoulis, E. Fricke, et al., “TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes,” *Nucleic Acids Research*, vol. 34, pp. D108–D110, 2006.

[42] A.M. Huerta, H. Salgado, D. Thieffry and J. Collado-Vides, “RegulonDB: a database on transcriptional regulation in Escherichia coli,” *Nucleic Acids Research*, vol. 26, no. 1, pp. 55–59, 1998.

[43] G.D. Stormo, G.W. Hartzell, “Identifying protein-binding sites from unaligned DNA fragments,” *Proceedings of the National Academy of the Science of United States of America*, vol. 86, no. 4, pp. 1183–1187, 1989.

[44] G.E. Crooks, G. Hon, J. Chandonia, and S.E. Brenner, “WebLogo: A Seqeunce Logo Generator,” *Genome Research*, pp. 1188–1190, 2004.

[45] J.H. MacDoNald, “Handbook of Biological Statistics”, *Sparky House Publishing, Baltimore*, pp. 1–18, 2008.

[46] W.M. Shaw, R. Buring, P. Howell, “Performance standards and evaluations in IR test collections: cluster-based retrieval models”, *Information Processing & Management*, vol. 33, no. 1, pp. 1–14, 1997.

ภาคผนวก

ภาคผนวก ก
ผลงานวิจัยที่ได้รับการตีพิมพ์

1. Sarawoot Som-in and Warangkhana Kimpan, “NexusPSO: A Novel Algorithm to Detect Transcription Factor Binding Sites,” IAENG International Journal of Computer Science, Vol. 45, no. 3, pp. 478-487, 2018.
2. Sarawoot Som-in and Warangkhana Kimpan, “Enhanching of Particle Swarm Optimization Based Method for Multiple Motifs Detection in DNA Sequences Collections,” IEEE/ACM Transactions on Computational Biology and Bioinformatics, ISSN: 1545-5963.

NexusPSO: A Novel Algorithm to Detect Transcription Factor Binding Sites

Sarawoot Som-in, *Member*, *IAENG*, Warangkhana Kimpan

***Abstract*—The detection of transcription factor binding sites is a major problem in research in Biology. Methods and computer algorithms can be applied to reduce time complexity and cost of detecting transcription factor binding sites in laboratory experiments. One of the well-known methods commonly used is swarm intelligence. However, errors in detection of transcription factor binding sites can be caused by different binding sites in the same genome sequence. The purpose of this research is to improve the effectiveness and accuracy in the detection of transcription factor binding sites by applying the newly developed pre-processing procedure, Nexus, to Particle Swarm Optimization algorithm (NexusPSO). The accuracy of the NexusPSO algorithm was measured in comparison with other algorithms, using information content (IC) as an indicator, with *Escherichia coli* data. This study found that NexusPSO is the most accurate method being tested. NexusPSO was then tested using consensus sequences on *Saccharomyces cerevisiae* and *Homo sapiens*. NexusPSO showed nearly identical results when compared to DNA footprinting methods.**

***Index Terms*—Particle swarm optimization, Transcription factor binding site (TFBSs), Motif detection.**

I. Introduction

Among major DNA sequences component, there is conserved sequences fragment called Transcription factor binding sites (TFBSs). TFBSs are an integral part of the gene transcription process leading to protein synthesis. The TFBSs consist of subsequences known as motif sequences consisting of the same nucleotides: A, T, C and G. TFBSs assist the biological researchers in knowing the location of gene transcription which leads to protein synthesis. This information benefits researchers by reducing the cost, time and resources used in detecting TFBSs in the laboratory setting. TFBSs can be detected by employing rigorous labor using expensive laboratory equipment [1] resulting in high cost of experiments. Therefore, a computer application was developed to reduce the cost of detection by applying the Gibb Sampling algorithm, developed by Charles E. Lawrence et al [2]. Later, the Gibb Sampling algorithm was developed to detect TFBSs via online computing programs, including: AlignACE [3] and BioProspector [4]. Gibb Sampling algorithm consists of two main processes. The first process is the sampling step where random DNA sequences are sampled an analyzed for possible TFBSs. The data is input into a Position Weight Matrix (PWM). The PWM showed the probability of each alphabet ('A', 'C', 'G', 'T') in every position of the motif sequence. The second process is the predictive update step, where the full sequence of DNA is sampled, and the PWM is optimized and selects the most suitable motifs.

Gibb Sampling was further developed to detect TFBSs more effectively using software such as MEME [5], Weeder [6] and MDScan [7]. The Gibb sampling algorithm was then applied with the Bayesian probability model by Gibbs sampler [8]. Gibb Sampling is an algorithm classified as a type of searching or detecting method using statistical optimization. This was the most suitable technique of stochastic optimization suitable for searching in long sequences. However, the Gibb Sampling algorithm had limitation in terms of efficiency of time and accuracy.

The Genetic Algorithm (GA) was applied by Falcon F.M Liu et al. [9] to increase the efficiency of detecting motifs through a program called FMGA. This method can be applied to TFBSs. GA used a crossover technique to randomly process motif sequences for speed, and the mutation technique to generate quality PWM indicators in detection using SAGA [10], MDGA [11] algorithms.

When analyzing detection patterns of TFBSs, it can be considered a NP-Hard problem similar to the Traveling Salesman Problem (TSP) [12], Job-shop Scheduling Problem (JSP) [13], Flow Shop Scheduling Problem (FSP) [14], Longest Common Subsequence problem (LCS) [15], etc. [16]. Researchers have developed algorithms to solve NP-Hard problems such as Particle Swarm Optimization (PSO) algorithm [17] by J.Kennedy and R.Eberhart in 1995, the Ant Colony Optimization (ACO) algorithm by Dorigo et al. in 1996 [18], and Memetic algorithm by J. Yan and M. Li in 2015 [19]. However, such algorithms are still need to be improved as the problem of local optimums. These algorithms can be applied to detect TFBSs using hybrid concepts to avoid the problem of local optimums and/or to reduce time consumption of the algorithm process. Therefore, the algorithms were developed and applied for

Manuscript received April 12, 2018; revised June 9, 2018. This work was supported in part by King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand.

S. Som-in is with the Department of Computer Science, Faculty of Science, KMITL (Corresponding auther, phone: +662-329-8400, e-mail: fender.stad@gmail.com)

W. Kimpan is with the Department of Computer Science, Faculty of Science, KMITL (e-mail: warangkhana.ki@kmitl.ac.th)

exceeding these limitations such as time improvement in solving LCS problem using Simple Polynomial Time Algorithm [20] and improvement in both time and quality in detecting TFBSs using Ant Colony Regulatory Identification (ACRI) by Wei Liu et al. [21] and Particle Swarm Optimization Variants (PSO Variants) by Mustafa Karabulut and Turkay Ibrikci [22]. Both algorithms achieved admirable results, while ACRI can improve the speed of result, PSO can increase the accuracy. However, detection accuracy of TFBSs is still limited when detecting motif sequences containing different characteristics.

This paper proposes applying the PSO algorithm [17] and the newly developed Nexus procedure, called NexusPSO algorithm to yield more accurate results and avoid the problem of local optimums in detection of TFBSs. Nexus functions by creating custom subsequences in the genome sequence. Following the characterization of each subsequence, relationships are created within the subsequences. The quality of each relationship between subsequences is evaluated, and weak relationships pruned.
The remaining parts of this research are presented as follows: Section II discusses the problem domain and related work; Section III describes the proposed approach; the data set and experiments are explained in Section IV; and Section V is the conclusions.

II. Background and Related Theories

A. Background and Signification of the Research Problem

Detection of TFBSs can be considered a NP-Hard Problem. The variables of the problem can be defined as follows: The DNA sequences can be defined from the input sequence which is S_i where i is the sequence of any input sequence. While n is the total number of input sequences. The length of input sequence S_i is L_{S_i} and the length of motif sequences is w. The number of total motif sequences (number of M_{s_i}) in the input sequence S_i is number of $M_{S_i} = L_{S_i} - w + 1$ where $w < L_{S_i}$. The total number of input sequences are defined as $S = \{S_1, S_2,...,S_n\}$and the group of motif sequences in each input sequence is $S_i = \{M_1, M_2,..., M_{L-w-1}, M_{L-w}, M_{L-w+1}\}$. The group of total alphabet data possible in the genome sequences is $b = \{'A', 'C', 'G', 'T'\}$.

If the detection of TFBSs independently allowed motif abundance, each sequence will be varied and the complexity would be $O((2^{l_i-w})^n$ [23],[24]. Therefore, restricting the number of motif in each particular sequence is preferred in this experiment.

B. Definition of Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) has been developed from the principles of swarm intelligence initiated from the research on the behaviors in movement in schools of birds or fish. While traveling, these groups vary group leaders to have the most effective leader at each iteration. Therefore, swarm intelligence has been developed by J.Kennedy and R.Eberhart in 1995 [17] as an algorithm for solving the NP-hard problems. This algorithm requires each bird or fish to be the considered a particle, with each particle selecting a different solution for each problem. Leaders are selected by running a fitness function and selecting the particle or particles with the highest calculated score.

One of the main principles of PSO is the definition of the particles. Then, topology is set, selecting the best particle at each iteration, including adjustments for speed and positions of each particle. The operation is repeated until each particle obtains the most optimal solution or the operation has reached the maximum iteration. There is also a research [25] which approaches the adjustment of particle speeds using Swap Sequence (SS) to achieve the better solutions.

The topology and connection among the particles within the PSO algorithm allow particles to share data according to the topography pattern. This causes each particle to move to a more suitable position by employing the data together among the best particles at each iteration within the neighborhood particles. The topologies [26] are as follows:

1. GBest: is the topology of total relative particles. Therefore, each particle has the number neighbors each particle has which is $C_P - 1$, having C_P as the total number of particles as shown in Fig. 1(a).
2. Bidirectional Ring: is the topology of a ring with each particle having two neighboring particles: P_{i-1} and P_{i+1} when i is the current particle as shown in Fig. 1(b).
3. Random: is the random topology of non-structured relative particles as each particle chooses the neighbors by random and defines the number of neighbors C_n and $0 < C_n <= C_P - 1$; particle as shown in Fig. 1(c).
4. Von Neumann: is the squared topology having the relative particles in a lattice structure. Each particle has four neighboring particles, consisting of: left P_{i-1} particle, right P_{i+1} particle, above $P_{s_{i-1}}$ particle, and below $P_{s_{i+1}}$ particle, as shown in Fig 1(d).

C. Fitness Function for Accuracy Measurement

The fitness function is run to consider and find the appropriate subsequences (appropriate motif sequences) that have the strongest solution. The factors used to calculate the fitness score of the particles or results of the motif consist of: equation (1) Consensus scoring (CS) [21] and equation (2) Information content (IC) [27]. CS is used to calculate the frequency of alphabetic patterns 'A', 'C', 'G' and 'T' in the results. This variable will not consider the frequency of other alphabets not involved in the motif sequences (background) as shown in Fig. 2. It is possible that high scores from CS can be attributed to background levels that are not accounted for in the score.

$$CS = 2 - (1/W)\sum_{i=1}^{W}\sum_{b=\{A,C,G,T\}} p_{bi} log_2(p_{bi}) \quad (1)$$

- b refers to all possible alphabets.
- w is the length of motif sequence.

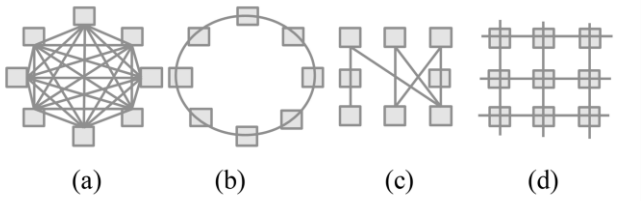


Fig. 1. Network pattern of Particle Swarm Optimization (PSO), Fig. (a) GBest. (b) Bidirectional Ring (c) Random and (d) Von Neumann.

- p_{bi} is the frequency of alphabet b.

Equation (2) Information Content (IC) is the variable used in calculating the similarity value of the alphabetic patterns between the results of each motif sequence. This variable will consider the frequency of other alphabets not involved in the results of motif sequences (background) as well.

$$IC = \sum f_b log_2 (f_b/p_b) \quad (2)$$

- b refers to all possible alphabets.
- f_b is the frequency of alphabet b in any motif sequence.
- p_b is the frequency of alphabet b which is not in the results of motif sequences (background).

The best particle from all iterations is p_{best} and g_{best} is the best particle in the neighborhood from each iteration. p_{best} and g_{best} are the center in which the particle's neighborhood are required to move along, at different speeds depending on the distance of each particle relative to p_{best} and g_{best}. Considering equation (3), as the positions of particle p_i which is distance from particle p_{best} and particle g_{best} increase, particle speed will increase. On the contrary, p_i speed decreases the more near it draws to particles p_{best} and g_{best}.

$$v_{i+1} = w_i.v_i + c_1 y_i (x_{Pbest_i} - x_{Pi}) + c_2 z_i (x_{Gbest_i} - x_i) \quad (3)$$

$$x_{i+1} = x_i + v_{i+1} \quad (4)$$

The variables in the equations (3), (4) are as follows:

- w_i is the internal factor influencing the speed of particle p_i in the next generation v_{i+1}.
- c_1 ,c_2 is the value gained at random being from 0 to 1.
- x_{Pbest_i} is the best position from the previous functional round.
- x_{Gbest_i} is the best position from the group at each iteration with the definition as follows:

$$x_{Gbest_i} = arg\ min\ f(x*) = \{x* \in P : f(x*) <= f(x), \forall x \in I\}$$

- y_i and z_i is the parameter influencing the speed of particle p_i.

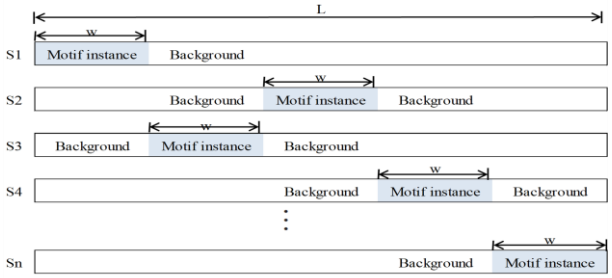


Fig. 2. Motif and background.

- v_i is the velocity at each iteration.
- x_{Pi} is the position of particle p_i at each iteration.

III. Proposed Principles and Concepts

The Nexus algorithm, which is a pre-process newly invented, can be applied to the PSO algorithm to increase the effectiveness in detecting TFBSs by reducing the chance of adhering to local optimums. The Nexus algorithm is able to reduce the problem space, which reduces the number of all possible subsequences, while still maintaining accurate results.

The Nexus algorithm consists of: grouping which will be stated descriptively in Section B; connection between the particles which will be stated descriptively in Section C; and the selection which will be stated descriptively in Section D.

A. Indication of Variables

In this research, all input sequences are defined as $S = \{S_1, S_2, ..., S_n\}$ and n is the number of input sequences. Each sequence of S_i has the equal length L. Each subsequence in the input sequences S has equal length w. Any non-selected area is called background as shown in Fig. 2. The members of motif sequences, which are TFBSs $CoM = \{M_{S_1}, M_{S_2}, ..., M_{S_{n-1}}, M_{S_n}\}$. The members of alphabet or nucleotides $b = \{$'A', 'C', 'G', 'T'$\}$. All subsequences in each input sequence $S_i = \{M_1, M_2, ..., M_{L-w-1}, M_{L-w}, M_{L-w+1}\}$. Therefore, the total number of subsequences in the genome sequence is $(L-w+1)*S$

B. Grouping

This method uses a grouping procedure, which is the arrangement of subsequence into 4 groups following the number of members of N_s consisting of Group A, Group C, Group G, and Group T. Measuring counts the number of alphabets in each subsequence $M_{S_{ij}}$ from the total number of subsequences when any $M_{S_{ij}}$ has the maximum frequency of alphabet b $MAX(b)$ having $b \in N_s$. Therefore $M_{S_{ij}}$ is classified into $Group(b)$. In the case that any $M_{S_{ij}}$ subsequence has the maximum frequency of alphabet $b > 1$, the subsequence $M_{S_{ij}}$ can be grouped into more than one group according to the maximum number of alphabet b as shown Table I. Table I shows grouping of subsequences with a total of 4 input

TABLE I
EXAMPLE OF INPUT SEQUENCE RESULTS FROM GROUPING

Ms_{ij}	Sequence	Max(b)	Group	Sequence	Max(b)	Group
	S_1			S_2		
1	C A A A T C C	*A,C=3*	*AC*	G G G C C T A	*G=3*	*G*
2	A A A T C C G	*A=3*	*A*	G G C C T A T	*C,G,T=2*	*CGT*
3	A A T C C G G	*A,C,G=2*	*ACG*	G C C T A T A	*A,C,T=2*	*ACT*
4	A T C C G G G	*G=3*	*G*	C C T A T A T	*T=3*	*T*
5	T C C G G G C	*C,G=3*	*CG*	C T A T A T A	*A,T=3*	*AT*
6	C C G G G C C	*C=4*	*C*	T A T A T A C	*A,T=3*	*AT*
7	C G G G C C C	*C=4*	*C*	A T A T A C C	*A=3*	*A*
8	G G G C C C C	*C=4*	*C*	T A T A C C C	*C=3*	*C*
	S_3			S_4		
1	C G G T G C T	*G=3*	*G*	G A G T C A C	*A,C,G=2*	*ACG*
2	G G T G C T C	*G=3*	*G*	A G T C A C A	*A=3*	*A*
3	G T G C T C T	*T=3*	*T*	G T C A C A G	*A,C,G=2*	*ACG*
4	T G C T C T T	*T=4*	*T*	T C A C A G A	*A=3*	*A*
5	G C T C T T T	*T=4*	*T*	C A C A G A G	*A=3*	*A*
6	C T C T T T A	*T=4*	*T*	A C A G A G C	*A=3*	*A*
7	T C T T T A T	*T=5*	*T*	C A G A G C A	*A=3*	*A*
8	C T T T A T A	*T=4*	*T*	A G A G C A A	*A=4*	*A*

sequences ($S_{n=4}$) having a total of possible $M_{S_{ij}}$ subsequences in the 8 input sequences. It can be noted the 1st sequence (S_1) is defined for more than one group because the maximum frequency of that subsequence matches more than one alphabet. Other examples include: the 1st subsequence ($M_{S_{11}}$), the 3rd subsequence ($M_{S_{13}}$), and the 5th subsequence ($M_{S_{15}}$), etc.

C. Connection

The creation of relations starts by taking all possible subsequences M_{ij} in the input sequence i to create relations with possible subsequences in the input sequence $i+1$ considering only subsequences in the same group ($1 <= i <= n$-1 where n is the total number of input sequences). Therefore, the occurring pattern of relations between input sequence i and input sequence i+1 is as follows:

$$([M(A)_{ij} \bowtie M(A)_{i+1j}],[M(C)_{ij} \bowtie M(C)_{i+1j}],[M(G)_{ij} \bowtie M(G)_{i+1j}],[M(T)_{ij} \bowtie M(T)_{i+1j}])$$

$\bowtie$ is defined to be the related pairs of the subsequences in the input sequence S_i and S_{i+1}. The related pairs of the subsequences in the input sequence S_i and each input sequence S_{i+1} will define the CS value. The data in Table II shows an example of related pairs calculated as equation of CS as shown in equation (1).

D. Selection

The selection of related pairs is the last process of the Nexus algorithm, where the best related pairs created in the connection process are selected. To select related pairs, the two input sequences with the highest CS value are selected.

$$[\text{Top2}\{M(A)_{ij} \bowtie M(A)_{i+1j}\},\text{Top2}\{M(C)_{ij} \bowtie M(C)_{i+1j}\},$$
$$\text{Top2}\{M(G)_{ij} \bowtie M(G)_{i+1j}\},\text{Top2}\{M(T)_{ij} \bowtie M(T)_{i+1j}\}]$$

The example in Table III shows the related pairs being selected from the subsequence M_{ij} and subsequence M_{i+1j} being in the same group. The data in this table is selected from the data in Table II. This process is intended to reduce the problem of local optimums from the random PSO process.

TABLE II
EXAMPLE OF RELATED PAIRS BETWEEN SUBSEQUENCES IN THE INPUT SEQUENCE S_i AND S_{i+1}

$S_i = 1$		$S_i = 2$		$S_i = 3$		
S_{i+1}	CS	S_{i+1}	CS	S_{i+1}	CS	
3	0.25	3	0.8	3	0.25	
5	0.4	5	0.4	5	0.4	
6	0.4	6	0.2	6	0.4	
7	0.3	7	0.3	7	0.3	• • •
32	0.6	32	0.6	32	0.6	
33	0.7	33	0.7	33	0.8	
34	0.7	34	0.7	34	0.8	

TABLE III
EXAMPLE OF RELATED PAIRS SELECTED FROM THE INPUT SEQUENCE S_i AND S_{i+1}

$S_i = 1$		$S_i = 2$		$S_i = 3$		
S_{i+1}	CS	S_{i+1}	CS	S_{i+1}	CS	
32	0.6	3	0.8	32	0.6	
33	0.7	33	0.7	33	0.8	• • •
34	0.7	34	0.7	34	0.8	

E. Particles Initialization

The process of defining particles in the NexusPSO is through the creation of particle P_i in the swarm. Each particle P_i consists of subsequence M_{ij} (defining i and j as any input sequence and subsequence, respectively) from each input sequence S_i. The condition allows one subsequence per input sequence. This research defines the first input sequence S_1 to be the data sequence defining the first motif of each particle having $S_1 = \{ M_{11}, M_{12}, \ldots, M_{1L-w-1}, M_{1L-w}, M_{1L-w+1}\}$ where particle $P_{i(M_1)} = M_{1j}$. $P_{i(M_1)}$ is the first subsequence of the particle (initial subsequence) defining each particle P_i to select the subsequence from the next input sequence until the last data sequence is determined. Subsequences with the highest CS score are selected from the related pairs resulting in $P_i = (P_{i(M_1)}, P_{i(M_2)}.., P_{i(M_{n-1})}, P_{i(M_n)})$, where n is the total number of input data. The patterns of particle P_i in each group have created the related pairs as follows:

$P(A)_i$ any particle in group 'A'

$$[\text{Top1}\{M(A)_{1j} \bowtie (M(A)_{2j_{Top1}}, M(A)_{2j_{Top2}})\}$$
$$\bowtie \text{Top1}\{M(A)_{2j_{Op}} \bowtie (M(A)_{3j_{Top1}}, M(A)_{3j_{Top2}})\}$$
$$\vdots$$
$$\bowtie \text{Top1}\{M(A)_{n\text{-}1j_{Op}} \bowtie (M(A)_{nj_{Top1}}, M(A)_{nj_{Top2}})\}]$$

$P(C)_i$ any particle in group 'C'

$$[\text{Top1}\{M(C)_{1j} \bowtie (M(C)_{2j_{Top1}}, M(C)_{2j_{Top2}})\}$$
$$\bowtie \text{Top1}\{M(C)_{2j_{Op}} \bowtie (M(C)_{3j_{Top1}}, M(C)_{3j_{Top2}})\}$$
$$\vdots$$
$$\bowtie \text{Top1}\{M(C)_{n\text{-}1j_{Op}} \bowtie (M(C)_{nj_{Top1}}, M(C)_{nj_{Top2}})\}]$$

$P(G)_i$ any particle in group 'G'

$[\text{Top1}\{M(G)_{1j} \bowtie (M(G)_{2j_{Top1}}, M(G)_{2j_{Top2}})\}$

$\bowtie \text{Top1}\{M(G)_{2j_{Op}} \bowtie (M(G)_{3j_{Top1}}, M(G)_{3j_{Top2}})\}$

$\vdots$

$\bowtie \text{Top1}\{M(G)_{n\text{-}1j_{Op}} \bowtie (M(G)_{nj_{Top1}}, M(G)_{nj_{Top2}})\}]$

$P(T)_i$ any particle in group 'T'

$[\text{Top1}\{M(T)_{1j} \bowtie (M(T)_{2j_{Top1}}, M(T)_{2j_{Top2}})\}$

$\bowtie \text{Top1}\{M(T)_{2j_{Op}} \bowtie (M(T)_{3j_{Top1}}, M(T)_{3j_{Top2}})\}$

$\vdots$

$\bowtie \text{Top1}\{M(T)_{n\text{-}1j_{Op}} \bowtie (M(T)_{nj_{Top1}}, M(T)_{nj_{Top2}})\}]$

The meanings of symbols and variables are as follows:

- $\bowtie$ is the relation of pairs in the sequences between the input sequence S_i with the input sequence S_{i+1}.
- $M(b)_{ij}$ is a subsequence in the genome defining i and j to be any input sequence and any subsequence, respectively. The set of b is {'A','C','G','T'}.
- $M(b)_{ij_{Top1}}$ is a subsequence with the highest CS value in relation to subsequence $M(b)_{i\text{-}1j}$.
- $M(b)_{ij_{Top2}}$ is the subsequence with the second highest CS value in relation to subsequence $M(b)_{i\text{-}1j}$.
- n is the total number of input sequences.
- $M(b)_{ij_{op}}$ is the optimal result of subsequences.

The particles of NexusPSO algorithm are defined to have the number of particles equal to the total possible subsequences of each input sequence L_i-w+1 with a size of $w < L_i$.

F. Particle's Movement

The initial position of the particles is defined in the process of initializing particles, as described in Section E. The NexusPSO defines the initial velocity of all particles as 0 and uses the fitness value from equation (5), which is discussed in Section G. This is used to calculate the fitness value of each particle. The fitness values from every particle are then compared to indicate the most suitable particle P_{best} as shown in Fig. 3(a). The comparison will be conducted by Gbest topology, with the topology using data shared among all particles, as shown in Fig. 1(a).

After, the position of each particle within the neighborhood is adjusted by applying the data of subsequence M_{ij} from the best particle P_{best} to replace the subsequences of particle's neighborhood P_i, as shown in Fig. 3(b). Adjusting the position of particles in each iteration, results in the particles having continuous movement, until each particle obtains the most optimal solution or the operation has reached the maximum iteration. If the process ceases because the total number of iterations was reached, the algorithm will select the particle with the highest fitness score from the last iteration. The results of the NexusPSO algorithm indicate the position of TFBSs in the genome sequences.

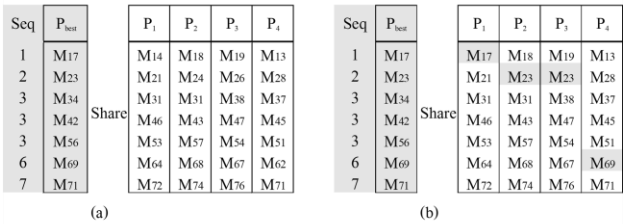


Fig. 3. Example of adjusting the particle position. (a) represents the 5 particles in the input sequences. (b) shows the replacements within the subsequences.

G. Fitness Function

The scale measuring the particles optimal P_{best} at each iteration t_i is the fitness function. The NexusPSO algorithm uses equation (5) as the fitness function. Equation (5) calculates the Information Content (IC) of the TFBSs as shown in Equation (2).

Equation (5) defines the length of subsequence W. The condition is $0 < W <= L\text{-}1$ and L is the length of the input sequence. The possible alphabets are b = {'A', 'C', 'G', 'T'}. The frequency of alphabet b appearing in the result of the particle is f'_b calculated from equation (6) and the frequency of alphabet b not being in the results of particles is p'_b calculated from equation (7).

$$fitness = \sum_{i=1}^{W} IC \tag{5}$$

$$f'_b = \frac{c_b + d_b}{N - 1 + D} \tag{6}$$

$$p'_b = \frac{c_{0b} + db}{S + D} \tag{7}$$

The symbols and variables are described below:

- c_b is the number of times any alphabet b appears in the subsequences within each column.
- c_{0b} is the number of times any alphabet b appears outside the selected subsequences (background).
- N is the total number of input sequences.
- S is the total number of alphabets not selected within the chosen subsequences.
- d_b is the pseudo counts [2].
- D is the sum of pseudo counts.

H. Input data Collection and NexusPSO Algorithm

The Nexus algorithm is the pre-process consisting of: the grouping of subsequences (grouping), creation of connections between the subsequences (initializing), and the process of selecting the most suitable related pairs in the first two ranks (selection). This research collects relation tables, which consist of: table of input sequences, table of

total possible subsequences, and table of particle data. PSO randomly selects subsequences from the Nexus procedure. The Pseudocode of the NexusPSO algorithm is as follow:

Algorithm NexusPSO

Input: w = the length of subsequence, *Maximum* = number of iterations, N = number of input sequences, L = length of input sequences, b = {'A', 'C', 'G', 'T'}.
Output: the set of subsequences *CoM*
1: Nexus process (pre-process)
1.1: **for** i = 1 to N **do**
1.2: **for** j = 1 to L_i-w+1 **do**
1.3: grouping *M[i][j]*;
1.4: connection: *M[i][j] and M[i+1][j]*;
end for i
end for i

2. PSO process
2.1. Initialize particles from best connection pair, start from first of sequences.
2.2. Particle movement
2.3 **for** k = 1 to *Maximum* **OR** *not converged* **do**
2.4 select local best particle;
2.5 update velocity of particles;
2.6 update position of particles;
2.7 **if** k = 1 or local best > global best **then**
update global best from local best;
end if
end for k

IV. Experiment

A. Dataset and Parameter Settings

The dataset of genome sequences to be tested for efficiency and accuracy of NexusPSO algorithm consists of 3 groups as follows:

- *Saccharomyces Cerevisiae* [28] from the database SCPD. The length of input DNA sequences is 550 nucleotide pairs (550 alphabets) with other properties as shown in Table IV.
- *Homo sapiens* [29] from the database JASPAR. The length of input DNA sequences is 600 nucleotide pairs (600 alphabets) with other properties as shown in Table V.
- *Escherichia coli: E.Coli* [27] from the dataset of cyclic-AMP receptor protein (CRP) with properties as shown in Table VI. The length of each input DNA sequence is 105 nucleotide pairs (105 alphabets). The length of motif is defined to be 22 nucleotides [27]. This genome sequence has at least one TFBS sequence in each input DNA sequence. Also, these sequences have varied nucleotide patterns, which make them a popular data set to test the efficiency of detection algorithms [3, 5, 8, 11, 18, 22].

The parameter settings of particles in Nexus PSO algorithm are shown in Table VII which are proper data for the tested dataset [22].

B. Operation

This research developed the NexusPSO algorithm using the C# language, version 5.0 in the Windows operating system. This research also used the SQL Server 2012 database management system as the design-related database in order to store the data of: DNA sequences, data of relations between all possible motifs, and the particle data. This research employs Weblogo (https://weblogo.berkeley.edu/logo.cgi) to generate consensus sequences, which were used to analyze the efficiency of results gained from the NexusPSO algorithm.

TABLE IV
Properties of the Genome Sequences of *Saccharomyces Cerevisiae*

TF	Size	Length	Consensus Sequence
GAL4	6	17	CGGNNNNNNNNNNCCG
RAP1	16	7	RMACCCA
REB1	14	7	YYACCCG
MCB	6	6	WCGCGW
PDR3	7	8	TCCGYGGA

TABLE V
Properties of the Genome Sequences of *Homo Sapiens*

TF	Size	Length	Consensus Sequence
ELK4	20	9	ACCGGAAGT
E2F1	10	8	TTTGGCGC
FOXD1	20	8	GTAAACAT
USF1	30	7	CACGTGG
RELA	18	10	GGGAATTTCC

TABLE VI
Data of TFBSs of the Dataset of *Escherichia Coli*

No.	Names	Motif 1	Motif 2	No.	Names	Motif 1	Motif 2
1	CE1CG	17	61	10	ECOMALBA	14	
2	ECOARABOP	17	55	11	ECOMALBA2	61	
3	ECOBGLR1	76		12	ECOMALT	41	
4	ECOCRP	63		13	ECOOMPA	48	
5	ECOCYA	50		14	ECOTNAA	71	
6	ECODEOP2	7	60	15	ECOXUL	17	
7	ECOGALE	42		16	PBR-P4	53	
8	ECOILVBPR	39		17	TRN0CAT	1	84
9	ECOLAC	9	80	18	TDC	78	

TABLE VII
Properties of the Parameters for Particles

Description	Parameters	Size
inertia weight	α	0.4
cognitive	β	0.8
social	γ	0.8
number of interations	*Maximum*	3000

NexusPSO result					
1	CGGAGCACTGTTGAGCG	3	CGGAAGACTCTCCTCCG	5	CGGCGCACTCTCGCCCG
2	CGGAGCAGTGCGGCGCG	4	CGGGGCGGATCACTCCG	6	CGGGGCAGACTATTCCG

Fig. 4. Results of CS from the group of DNA sequences GAL*4*

NexusPSO result					
1	ACACCCA	7	ACCCAGT	13	AAACCCA
2	GAACCCA	8	GAACCCA	14	CCGTGCA
3	ACACCCA	9	GCACCCA	15	GCACCCA
4	ACCCATG	10	TGCACCA	16	ACACCCA
5	GACACGC	11	CACCCAA		
6	CTGGACA	12	ACACCCA		

Fig. 5. Results of CS from the group of DNA sequences RAP1

NexusPSO result					
1	TTACCCT	6	CCGGGCA	11	CTACCCG
2	TAACCCG	7	GGGTAGC	12	GTACGGG
3	TTACCCG	8	TCACCCG	13	CGGGTAA
4	TCACCCG	9	TTACCCG	14	TTACCCG
5	TTACCCG	10	CGGGTTA		

Fig. 6. Results of CS from the group of DNA sequences REB1

NexusPSO result					
1	ACGCGT	3	ACGCGT	5	ACGCGT
2	ACGCGA	4	TCGCGA	6	ACGCGT

Fig. 7. Results of CS from the group of DNA sequences MCB

NexusPSO result					
1	TCCGCGGA	4	TCCGTGGA	6	TCCGTGGA
2	TCCGCGGA	5	TCCGTGGA	7	TCCGCGGA
3	TCCGCGGA				

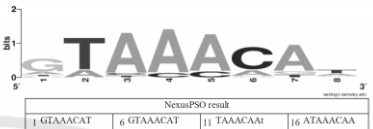
Fig. 8. Results of CS from the group of DNA sequences PDR3

NexusPSO result			
1 ACCGGAAGT	6 ACCGGAAGT	11 ACCGGAAGC	16 GGATATagc
2 ACCGGAAGT	7 ACCGGATGT	12 acaACCGGA	17 ACCGGAAAC
3 ACCGGAAGT	8 ACCGGATGT	13 GCCGGAAGT	18 TCCGGAAGC
4 ACCGGAAGT	9 acACCGGAA	14 CCCGGAAGT	19 GCCGGAAAT
5 ACCGGAAGT	10 ACCGGAAGC	15 ACCGGATGC	20 ACCGGAACA

Fig. 9. Results of CS from the group of DNA sequences ELK4

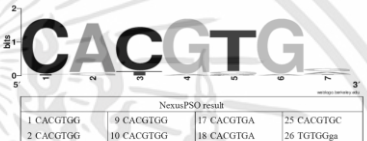
NexusPSO result			
1	TTTGGCGC	6	TTTCGCGC
2	TTTGGCGC	7	TTTCGCGC
3	TTTGGCGC	8	TTTGCCGC
4	TTTGGCGC	9	TTTCCCGC
5	TTTCGCGC	10	TTTGGCGG

Fig. 10. Results of CS from the group of DNA sequences E2F1



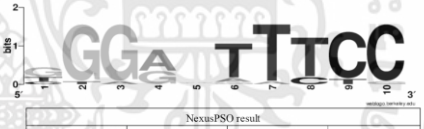
NexusPSO result			
1 GTAAACAT	6 GTAAACAT	11 TAAACAAt	16 ATAAACAA
2 GTAAACAT	7 GTAAACAT	12 GTAAACAA	17 CTAAACAG
3 GTAAACAT	8 GTAAACAA	13 AAACACgt	18 GTCAACAG
4 GTAAACAT	9 AAACAAtg	14 GTAAACAC	19 GTAACAAT
5 GTAAACAT	10 GTAAACAA	15 TAAACAGa	20 gTTAAGTA

Fig. 11. Results of CS from the group of DNA sequences FOXD1



NexusPSO result			
1 CACGTGG	9 CACGTGG	17 CACGTGA	25 CACGTGC
2 CACGTGG	10 CACGTGG	18 CACGTGA	26 TGTGGga
3 CACGTGG	11 CACGTGG	19 CGTGTaa	27 CATGTGA
4 CACGTGG	12 CACGTGG	20 CACGTGT	28 CACATGA
5 CACGTGG	13 CACGTGA	21 CACGTGT	29 CACGCGG
6 CACGTGG	14 CACGTGA	22 CACGTGT	30 CACGGGA
7 CACGTGG	15 CACGTGA	23 CACGTGT	
8 CACGTGG	16 CACGTGA	24 CACGTGC	

Fig. 12. Results of CS from the group of DNA sequences USF1



NexusPSO result				
1 GGGAATTTCC	6 GGAATTTCCg	11 CGGACTTTCC	16 tGGGGTTTTC
2 GGGAATTTCC	7 GGGGATTTCC	12 CGGACTTTCC	17 GGGGGATTCC
3 GGGAATTTCC	8 GGGACTTTCC	13 GGGGAATTCC	18 TGGGTTTCCc
4 TGGAATTTCC	9 CGGAGTTTCC	14 TGGGGTTTCC	
5 TGGAATTTCC	10 GGGAATTCCC	15 gtGGGGATTC	

Fig. 13. Results of CS from the group of DNA sequences RELA

C. Experimental Results for Anal yzing the Efficiency

The results from consensus sequences using the NexusPSO algorithm to detect the motif sequences in the genome sequences of *Saccharomyces cerevisiae* dataset are shown in Figures 4 to 8. The consensus sequences results for *Homo sapiens* are shown in Figures 9 to 13. Table IV and Table V show the consensus sequences of NexusPSO algorithm are identical to the consensus sequences from DNA footprinting methods.

Table VIII shows the representative sequences result of NexusPSO, selected by average IC value from all 18 runs, compared to the results from the traditional algorithms consisting of AlignACE [3], MEME [5], and Gibbs sampler [8] to detect the motifs in the genome sequences of *Escherichia coli*. Also, Table VIII compares the positions of motif sequences obtained from each algorithm with the positions of TFBSs. The Gibb Sampler results reveal 2 motif sequences with results more than 20 positions from TFBSs, the 5th DNA sequences (ECOYA) and the 17th DNA sequences (TRN9CAT). The AlignACE results show there

are 2 motif sequences more than 15 positions from the TFBSs, the 7th DNA sequence (ECOGALE) and the 17th DNA sequence (TRN9CAT). Both algorithms do not have any resulting motif sequences match TFBSs. Results from the MEME algorithm show there are 4 motif sequences more than 20 positions from TFBSs and 1 motif sequence 16 positions from the TFBS, the 5th DNA sequences (ECOCYA), the 15th (ECOXUL), the 16th (PBR-P4), the 17th (TRN9CAT), and the 11th (ECOMALBA2), respectively, while 11 motif sequences match the TFBSs.

TABLE VIII
COMPARISON ON THE RESULTS OF TRADITIONAL ALGORITHMS, RELEVANT ALGORITHMS, AND NEXUSPSO ALGORITHM

		Traditional Algorithm						Related Work							
No.	BS	Gibbs Sampler	diff	Align ACE	diff	MEME	diff	GA	diff	PSO	diff	ACRI	diff	NexusPSO	diff
1	17,61	59	2	63	2	61	0	62	1	61	0	63	2	61	0
2	17,55	53	2	57	2	55	0	56	1	55	0	57	2	55	0
3	76	74	2	48	2	76	0	77	1	76	0	78	2	76	0
4	63	59	4	65	2	63	0	64	1	63	0	65	2	63	0
5	50	11	39	52	2	13	37	51	1	50	0	52	2	50	0
6	7,6	5	2	9	2	7	0	8	1	7	0	9	2	7	0
7	42	40	2	26	16	42	0	43	1	24	18	44	2	42	0
8	39	3	2	41	2	39	0	40	1	39	0	41	2	39	0
9	9,80	7	2	11	2	9	0	10	1	9	0	11	2	9	0
10	14	12	2	16	2	14	0	15	1	14	0	16	2	14	0
11	61	59	2	63	2	35	16	62	1	61	0	63	2	61	0
12	41	47	6	43	2	34	7	42	1	41	0	43	2	41	0
13	48	46	2	50	2	48	0	49	1	48	0	50	2	48	0
14	71	69	2	73	2	71	0	72	1	71	0	73	2	71	0
15	17	15	2	19	2	75	58	18	1	17	0	19	2	17	0
16	53	49	4	55	2	6	47	54	1	53	0	55	2	53	0
17	1,84	25	24	68	16	27	26	56	28	5	4	95	11	5	4
18	78	74	4	80	2	16	2	77	1	76	2	78	0	76	2

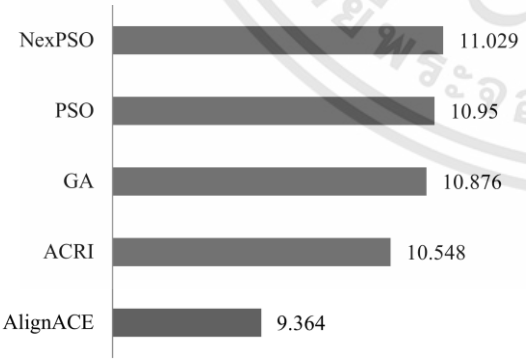


Fig. 14. Comparison results of IC value among AlignACE, GA, PSO, ACRI, and NexusPSO

According to the data in Table VIII, the motif sequence results of the GA [11] and ACRI [21] algorithms for the 17th DNA sequence (TRN9CAT) are shifted from the TFBS by 28 and 11 positions respectively. The result of the PSO [30] algorithm for the 7th sequence is also shifted from TFBS by 18 positions. This shows these algorithms cannot detect the motif sequences of the 17th and 7th DNA sequence (TRN9CAT, ECOGALE) accurately. The NexusPSO algorithm had the most accurate detection of the motif sequence in the 17th DNA sequence (TRN9CAT) with a deviation from the TFBS of only 4 positions. Also, NexusPSO detected the motif sequences by completely

TABLE IX
AVERAGE IC VALUES FROM 18 RUNS AMONG THE DIFFERENT ALGORITHMS

MEME	AlignACE	ACRI	NexusPSO
9.508	9.752	10.273	11.030

TABLE X
IC VALUES FROM 18 RUNS AMONG THE DIFFERENT ALGORITHMS

No.	MEME	AlignACE	ACRI	NexusPSO
1	10.032	9.651	10.01	11.045
2	9.075	9.887	10.28	11.804
3	10.02	9.576	9.987	11.018
4	10.05	9.624	10.403	10.946
5	9.117	10.235	10.457	10.354
6	9.892	9.71	10.184	11.934
7	9.554	9.01	9.895	11.005
8	10.124	9.934	10.258	11.112
9	9.646	9.807	10.354	10.124
10	9.439	9.853	10.421	11.053
11	9.121	10.12	10.53	10.984
12	9.16	9.399	10.415	10.852
13	9.684	9.976	10.38	11.074
14	9.773	9.825	10.286	11.04
15	9.024	9.769	10.179	11.206
16	9.008	10.314	10.3	11.704
17	9.105	9.011	10.14	11.029
18	9.32	9.835	10.431	10.254

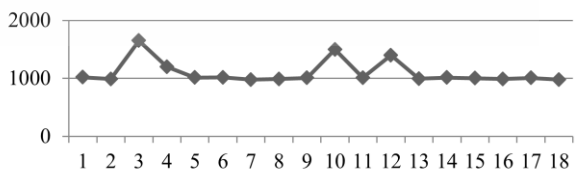


Fig. 15. The process times of NexusPSO for 18 runs

TABLE XI
t-VALUES OF NEXUSPSO COMPARED WITH OTHER ALGORITHMS

t-value	MEME	AlignACE	ACRI
compared with NexusPSO	10.354	9.181	6.34

matching the TFBSs for 16 sequences, resulting in the NexusPSO algorithm having the highest IC value, as shown in Fig. 14. Table IX shows the average IC values from 18 runs among the different algorithms including MEME, AlignACE, ACRI and NexusPSO. Table X shows the IC values of each run. The computational times are between 980 and 1650 milliseconds as shown in Fig. 15. The comparison of *t*-values among the relevant algorithms including NexusPSO is shown in Table XI. Considering *t*-test from 18 samples, the degree of freedom is 18+18-2 = 34 and let the significance level is $\alpha = 0.05$ (confidence level is 95%), so that $t_{0.95}(34) = 1.691$. Comparing to *t*-value of NexusPSO from Table XI, we found that *t*-value of NexusPSO is higher than $t_{0.95}(34)$.

V. CONCLUSIONS

There are many algorithms available for detecting TFBSs, many of which were tested in this study. The Nexus procedure is designed to manage the problem space to become smaller, helping the random process of the algorithm avoid local optimums results.

The data from this study shows that NexusPSO can detect TFBSs more efficiently and accurately than other available methods. According to the samples in this study, NexusPSO have the highest IC at 11.029 scoring better than previously recorded results for PSO [30], GA [11] and ACRI [21] which had IC values of 10.95, 10.876, and 10.548, respectively. Considering *t*-test, it indicates that there are different significances between the information content by NexusPSO and other algorithms. Furthermore, the results of consensus sequences of NexusPSO show efficient results when compared to the results from DNA footprinting method.

However, the NexusPSO algorithm still needs to develop the competence to detect TFBSs with multiple motifs in each input sequence.

REFERENCES

[1] L. Elnitski, V.X. Jin, P.J. Farnham, S.J.M. Jones, "Locating mammalian transcription factor binding sites: a survey of computational and experimental techniques," *Genome Research*, vol.16, pp. 1455–1464, 2006.

[2] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, J.C. Wootton, "Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignmzent," *Science*, vol. 262, pp. 208-214, 1993.

[3] J.D. Hughes, P.W. Estep, S. Tavazoie, G.M. Church, "Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae," *Journal of Molecular Biology*, vol. 296, pp. 1205–1214, 2000.

[4] X. Liu, D.L. Brutlag, J.S. Liu, "BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes," *Pacific Symposium on Biocomputing*, 2001, pp. 127–138.

[5] T.L. Bailey, N. Williams, C. Misleh, W.W. Li, "MEME: discovering and analyzing DNA and protein sequence motifs," *Nucleic Acids Research*, vol. 34, pp. 369–373, 2006.

[6] G. Pavesi, P. Mereghetti, F. Zambelli, M. Stefani, G. Mauri, G. Pesole, "MoD Tools: regulatory motif discovery in nucleotide sequences from co-regulated or homologous genes," *Nucleic Acids Research*, vol. 34, pp. 566–570, 2006.

[7] X.S. Liu, D.L. Brutlag, J.S. Liu, "An algorithm for finding protein–DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments," *Nature Biotechnology*, vol. 20, pp. 835–839, 2002.

[8] A.F. Neuwald, J.S. Liu, C.E. Lawrence, "Gibbs motif sampling: detection of bacterial outer membrane protein repeats," *Protein Sci.*, vol. 4, no. 8, pp. 1618–1632, 2004.

[9] F.F.M. Liu, J.J.P. Tsai, R.M. Chen, S.N. Chen and S.H. Shih, "FMGA: finding motifs by genetic algorithm," *IEEE Fourth Symposium on Bioinformatics and Bioengineering (BIBE 2004)*, pp. 459–466, 2004.

[10] C. Notredame, D.G Higgins, "SAGA: Sequence alignment by genetic algorithm," *Nucleic Acids Res.*, vol. 24, no. 8, pp. 1515–1524, 1996.

[11] D. Che, Y. Song, K. Rasheed, "MDGA: motif discovery using a genetic algorithm," *Genetic and Evolutionary Computation (GECCO 2005)*, pp. 447–452, 2005.

[12] K. Socha, M. Dorigo, "Ant colony optimization for continuous domains," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1155–1173, 2008.

[13] D.D. Duc, H.Q. Dinh, H.H. Xuan, "On the pheromone update rules of ant colony optimization approaches for the job shop scheduling problem," *in: Proceedings of the 11th Pacific Rim International Conference on Multi-Agents*, *Intelligent Agents and Multi-Agent Systems*, vol. 5357, pp. 153–160, 2008.

[14] V. Maniezzo, A. Carbonaro, "An ANTS heuristic for the frequency assignment problem," *Future Gener. Comput. Syst*., vol. 16, no. 8, pp. 927–935, 2000.

[15] S.J. Shyu, C.Y. Tsai, "Finding the longest common subsequence for multiple biological sequences by ant colony optimization," Comput Oper Res, vol.36, no. 1, pp. 73-91, 2009.

[16] R. Poli, "Analysis of the publications on the applications of particle swarm optimisation," *Journal of Artificial Evaluation and Applications 2008*, 2008, pp. 1–10.

[17] J. Kennedy, R. Eberhart, "Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks," *IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[18] M. Dorigo, V. Maniezzo and A. Colomi, "Ant system: optimization by a colony of coorperating agents," *IEEE Trans. Syst. Man Cybern.—Part B*, vol. 26, no. 1, pp. 29–41, 1996.

[19] J. Yan, M. Li, and J. Xu, "An Adaptive Strategy Applied to Memetic Algorithms," *IAENG International Journal of Computer Science*, vol. 42, no. 2, pp. 73-84, 2015.

[20] D. Zhu, L. Wang, J. Tian and X. Wang, "A Simple Polynomial Time Algorithm for the Generalized LCS Problem with Multiple Substring Exclusive Constraints," *IAENG International Journal of Computer Science*, vol. 42, no. 2, pp. 214-220, 2015.

[21] W. Liu, H. Chen, L. Chen, "ACRI: an ant colony optimization based algorithm for identifying gene regulatory elements," *Computer in Biology and Medicine*, vol. 43, pp. 922-932, 2013.

[22] M. Karabulut, T. Ibrikci, "PSO-variants: a Bayesian Scoring Scheme based Particle Swarm Optimization algorithm to identify transcription factor binding sites," *Applied Soft Computing*, vol. 12, pp. 2846-2855, 2012.

[23] Z. Wei, S.T. Jensen, "GAME: detecting cis-regulatory elements using a genetic algorithm," *Bioinformatics*, vol. 22, pp. 1577–1584, 2006.

[24] T.M. Chan, K.S. Leung, K.H. Lee, "TFBS identification based on genetic algorithm with combined representations and adaptive post-processing," *Bioinformatics*, vol. 24, pp. 341–349, 2008.

[25] M. A. H. Akhand, S. Akter, M. A. Rashid and S.B. Yaakob, "Velocity Tentative PSO: An Optimal Velocity Implementation based Particle Swarm Optimization to Solve Traveling Salesman Problem," *IAENG International Journal of Computer Science*, vol. 42, no. 2, pp. 221-232, 2015.

[26] J. Kennedy, R. Mendes, "Population structure and particle swarm performance, in: Proceedings of the Evolutionary Computation on 2002. CEC '02, Proceedings of the 2002 Congress – Vol. 02," *IEEE Computer Society*, vol. 02, pp. 1671–1676, 2002.

[27] G.D. Stormo, G.W. Hartzell, "Identifying protein-binding sites from unaligned DNA fragments," *Proc. Natl Acad. Sci.USA*, vol. 86, no. 4, pp. 1183–1187, 1989.
[28] J. Zhu, M.Q. Zhang, "SCPD: a promoter database of the yeast Saccharomyces cerevisiae," *Bioinformatics*, vol. 15, no. 7–8, pp. 607–611, 1999.
[29] J.C. Bryne, E. Valen, et al., "JASPAR:the open access database of transcription factor-binding profiles: new content and tools in the 2008 update," *Nucleic Acids Res.*, vol. 36, pp. 102–106, 2008.
[30] H. Ge, L. Sun, Y. Yao and J. Yu, "An automatic motif regnition algorithm in DNA sequences based on particle swarm optimization and random projection," *Bioinformatics and Biomedicine (BIBM)*, 2017, pp. 2241-2243.

Sarawoot Som-in received his B.S. in the Department of Computer Science from Huachiew Chalermprakiet University, Bangkok, Thailand in 2005. He received the Master degree in the Department of Computer Science from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand in 2010, where he is currently working toward the Ph.D. degree. His research interests include design and analysis of algorithm and bioinformatics.

Warangkhana Kimpan received her Ph.D. degree in System Information Engineering from Kagoshima University, Japan. She is currently an assistant professor in Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. Her main research interests are Swarm Intelligence, Biomedical Engineering, Big Data, Data Science and analytics, Cloud Computing, and Internet of Things.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2872978, IEEE/ACM Transactions on Computational Biology and Bioinformatics

Enhancing of Particle Swarm Optimization Based Method for Multiple Motifs Detection in DNA Sequences Collections

Sarawoot Som-in and Warangkhana Kimpan, *Member, IEEE*

Abstract—Genome sequence data consists of DNA sequences or input sequences. Each one includes nucleotides with chemical structures presented as characters: 'A', 'C', 'G', and 'T', and groups of motif sequences, called Transcription Factor Binding Sites (TFBSs), which are subsequences of DNA that lead to protein-synthesis. The detection of TFBSs is an important problem for bioinformatics research. With the similar patterns of motif sequences in TFBSs, computational algorithms for TFBSs detection have been improved to reduce resources used in laboratory setting. The metaheuristic algorithm is the important issue that has been continually improved to detect TFBSs with greater precision and recall. This paper proposes PSO_HD by applying Particle Swarm Optimization (PSO) as a pre-process and using Hamming distance to improve the efficiency of detecting TFBSs with more precision and recall. In order to measure its efficiency, the paper compares the TFBSs detection using PSO_HD algorithm with relevant algorithms in 8 datasets. *F*-score is used as a measurement unit and compared to the related algorithms. The experimental results show that PSO_HD algorithm gives the highest average *F*-score, which can be indicated that the PSO_HD algorithm can improve the efficiency of detecting TFBSs with more precision and recall.

Index Terms—particle swarm optimization, transcription factor binding sites, hamming distance, matrix similarity score

1 Introduction

MOTIF Discovery Problem (MDP) is an important problem for bioinformatics, defined as a NP-Hard problem by Maier in 1978 [1], and Riviere et al. in 2008 [2]. One of the problems is the detection of Transcription Factor Binding Sites (TFBSs), the data that specify regions of DNA leading to protein synthesis. Generally, the tested genome sequence from various DNA sequences has a maximal identity in sequence content [3]. Due to the high cost of TFBSs detection in laboratory settings, many computer algorithms have been developed to help reduce costs [4]-[8]. Developments have been made in the detection of TFBSs using metaheuristic algorithms, for example: Genetic algorithm [9]-[11] and swarm intelligence (SI) improve detection efficiency. Ant Colony Optimization [12] is an algorithm that can be used to improve the efficiency in detection time, and Bacterial Foraging Optimization (BFO) can be applied with Tabu Search (TS) [13] to detect TFBSs with more precision.

The numbers of motif sequence results from many algorithms attempting to detect TFBSs are defined by the number of input sequences, however in general, the number of motif sequences which are TFBSs are not equal to the number of input sequences (DNA sequences). Therefore, detecting TFBSs with precision and recall are still a major problem in bioinformatics.

Genetic Algorithm was developed as a method to detect TFBSs without pre-determining the number of TFBS motif sequences. One notable example is the GAME [14] algorithm by Wei and Jensen (2008), which applies Genetic Algorithm (GA) and Bayesian-based statistics. There are several other algorithms that use GA to improve detection with more precision, including: GALF_P [15], GAPK [16], and iGAPK [17].

Research in recent years has developed processes which support and improve the detection of TFBSs, for example, Planted (l, d) Motif Discovery Problem, where l is the length of motif sequences and d is a number of mismatched characters in motif sequences. MCES [18] is the algorithm which applies suffix array (SA) and longest common prefix array (LCP) to determine motif length. Also, AMDILM [19] is the algorithm which determines the optimal motif length to detect TFBSs by applying GA, including Mutation, Addition and Deletion. Also, the determination of initial l-mers has been recently improved by algorithms such as: kmerGA [20], which uses Position Frequency Matrix (PFM) to detect TFBSs from Protein Binding Microarray data; and MOTOMATA [21], which can quickly detect TFBSs by using parallel processing, a non-deterministic finite automata (NFA) designed to work in micron automata processor (AP).

Particle Swarm Optimization (PSO) is an optimization procedure that allows the processing of different candidate solutions to communicate and respond to one another as a means of reaching the optimum result. However, the PSO algorithm can still be limited in effectiveness by becoming trapped in local optimums, where the algorithm converges on the local optimums rather than the global optimum. M. Karabulut and T. Ibrikci created a Particle Swarm Optimization Variant (PSOVa) [22] with a pre-process adjusting the pattern of input sequences into gradient distribution, so the search space could be continuously searched reducing the percentage of results

S. Som-in and W. Kimpan are with the Department of Computer Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand.
Email: sarawoot.somin@gmail.com, warangkhana.ki@kmitl.ac.th

xxxx-xxxx/0x$xx.00 © 200x IEEE

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2872978, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TABLE 1
THE ENCODING INTO BINARIES FOR COMPUTING THE HAMMING DISTANCES

Rows		Columns			
		A	T	G	C
		00	01	10	11
A	00	0	1	1	2
T	01	1	0	2	1
G	10	1	2	0	1
C	11	2	1	1	0

trapped in local optimums. However, there are precision limitations in the results of PSOVa in detecting some motif sequences which have different characteristics from others.

The purpose of this research is to improve the efficiency of TFBSs detection with more precision and recall, by applying PSO with Hamming Distance (HD) [23] as a preprocess, called PSO_HD. HD was applied to calculate the weight of the related pairs between subsequences in the pre-process. The calculation of Hamming distance is to count the differences between subsequences, for each subsequence was converted from character into binary by using Genetic Analysis [24] to continually and significantly relate subsequences between input sequences. The particles are created in the algorithm by randomly selecting subsequences from the first input sequence. Next, each particle chooses its next subsequence according to its related pairs with the least Hamming distance, until it reaches the last input sequence. This happens in the movement of particles so the search space could be continuously and heuristically searched, thus the results trapped in local optimums are reduced. Particles continue to move until they reach alignment with one motif sequence per input sequence. The algorithm is defined to detect the remaining motif sequences by using cut-offs minimizing false negative rate [25]. Then, to create the TFBSs result, merge the result that satisfies the criteria with alignment which is the result of particle movement process. *F*-score is used to measure precision and recall, where the True Positive (TP) result as an overlapping threshold at 25% measured with the footprint experiments.

The remaining parts of this paper are presented as follows: Section II background and related theories; Section III describes the proposed approach; the data set and experiments and discussions are explained in Section IV; and Section V is the conclusion.

2 BACKGROUND AND RELATED THEORIES

2.1 Definition and Background

The TFBSs consist of k number of motif sequences (m_i) located in input sequence S_i. This is the criteria of number of motif sequences k_{si} per input sequence, $0 < k_{si} <= pk_{si}$. pk_{si} is the number of all possible subsequences in each input sequence. Suppose the width of a motif sequence is w and l is the length of an input sequence, therefore $pk_{si} = l\text{-}w+1$. By comparing between the total possible subsequences from each input sequence, the detection takes time complexity of $O((2^{Li\text{-}w})^n)$ [14],[15]. Generally, each genome has many input sequences with long lengths, making the problem of detecting TFBSs a NP-Hard problem.

2.2 Hamming Distance

Hamming distance *HD* is the distance between two vectors of binary which is calculated by comparing binaries in the same position between two vectors, followed by counting the distinguished binaries [23]. Assume the Hamming distance of vector X and vector Y is $\sum |X_i - Y_i|$, where X_i, Y_i are a binary of vectors X and Y, respectively. Define the width of vector X and Y as the same length which is L. Therefore, $HD(X,Y) > 0$ when $X \neq Y$ and $HD(X,Y) = 0$ when $X = Y$; for finding the minimum distance of Hamming distance between vectors $\{a, b, c, d\}$ where a = (11111), b = (01001), c = (10100), d = (00010), $HD(a, b)$ =3, $HD(a, c)$ =3, $HD(a, d)$ =4, $HD(b, c)$ =4, $HD(b, d)$ =3, $HD(c, d)$ =3; therefore, the minimum Hamming distance of the vector $\{a, b, c, d\}$ is $HD = 3$.

In order to apply the Hamming distance to detect TFBSs, the length of sub sequence w is defined as the length of vector L. Each nucleotide 'A', 'C', 'G', and 'T' are converted into binaries, such as A = 00, T = 01, G = 10, T = 11, by the condition of chemical relationship [24]. The pattern of binary conversion is due to the relationship of the nucleotides. A (Adenine) and G (Guanine) are classified as purine nucleotides, while, T (Thymine) and C (Cytosine) are in pyrimidine class. Also, the base pairing rule is that A pairs with T and C pairs with G. The Hamming distance between the characters converted into binaries are shown in Table 1. For example, the Hamming distance between subsequence ATCCGA <000111111000> and AAGGCC <000010101111> is equal to 6.

2.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a metaheuristic algorithm, classified as a stochastic search, which is applied from the movement of a swarm, becoming an algorithm to solve a NP-hard problem [26]. The process of PSO begins with creating particles. Then PSO sets their velocities, using a fitness function to determine the best particle in each iteration. There are two types of best particles; a local best particle (x_i^{lb}) and a global best particle (x_i^{nb}). The velocity of each particle $v_i^{(t)}$ is varied by its distances to x_i^{lb} and x_i^{nb} as shown in equation (1). Where a is the inertia parameter which controls the last iteration velocity $v_i^{(t+1)}$, β is the cognitive parameter which controls the effects from x_i^{lb}, γ is the social parameter which controls the effects from x_i^{nb} ,and r_1, r_2 are the random values between 0 and 1. The position of each particle is adjusted continually at each iteration t, calculated in the equation (2).

$$v_{i+1} = a v_i^{(t)} + r_1\beta(x_i^{lb} - x^{(t)}) + r_2\gamma(x_i^{nb} - x_i^{(t)}) \quad (1)$$

$$x_{i+1} = x_i + v_{i+1} \quad (2)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2872978, IEEE/ACM Transactions on Computational Biology and Bioinformatics

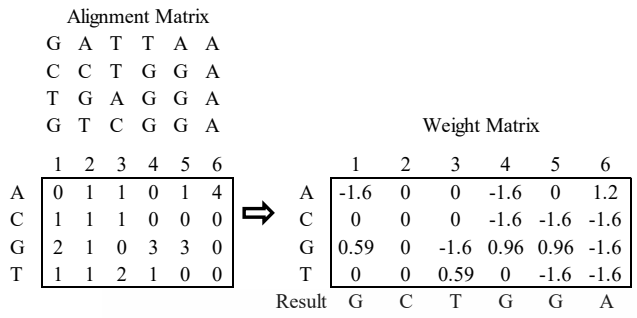


Fig. 1. Example of alignments and weight matrix

The informants among particles depend on the pattern of the topology consisting of many patterns of topology [27]. For example, Gbest is the topology of total relative particles; Bidirectional Ring is the topology of a ring with each particle having two neighboring particles: P_{i-1} and P_{i+1} when i is the current particle; and Random is the random topology of non-structured relative particles as each particle chooses the neighbors by random and defines the number of neighbors as C_n, and the number of total particles as C_P, where $0<C_n<=C_{P-1}$. As the process of PSOVa [22] in detecting TFBSs is to randomly choose a subsequence from each input sequence, the positions of particles could not be heuristically distributed among the problem space. As a result, the detection of TFBSs needs to be improved in precision and recall.

2.4 Fitness Function

Algorithms for detecting TFBSs provide an important process to select the best or the most fit particle where fitness values of each particle is calculated by the fitness function. When detecting TFBSs with genetic algorithm (GA) [11], the fitness function is used to calculate the probability of individual particles becoming parent particles in the next iteration, in other words, individual can become a parent by fitness proportionate selection. The fitness function is applied to calculate the shortest distance of data for selecting the optimal solutions of Ant Colony Optimization (ACRI) algorithm [28] and it is used to calculate the fitness value of particles in Particle Swarm Optimization (PSOVa) algorithm [22]. When applied to detect TFBSs, the fitness function is usually calculated as an alignment as follows:

1. Alignment matrix is used to calculate the weight matrix of each character in alignment with equation (3) and select the character using the highest weight as a result shown in Fig. 1.

$$weight\ matrix = ln\frac{(n_{i,j}+p_i)/(N+1)}{p_i} \quad (3)$$

where

- N is the number of total input sequences.
- $n_{i,j}$ is the number of characters in position (i,j).
- p_i is priori probability.

2. Consensus scoring (CS) is the calculation which measures a similarity between subsequences in an alignment as shown in equation (4) [24].

$$CS = 2 - \left(\frac{1}{W}\right)\sum_{i=1}^{W}\sum_{b=\{A,C,G,T\}} p_{bi}log_2(p_{bi}) \quad (4)$$

where

- b is the total possible characters consisting of 'A', 'C', 'G' and 'T'.
- w is the length of the subsequence.
- p_{bi} is priori probability of the character b in alignment.

3. Information Content (IC) is the calculation of similarity in character patterns for alignment by considering the character data which is not in the alignment [29], called background, as shown in the equation (5). The alignment with a high IC value shows the patterns of motifs in the alignment with more similarity. Therefore, the motifs in alignment are likely to be TFBSs.

$$IC = \sum f'_b log_2\left(\frac{f'_b}{p'_b}\right) \quad (5)$$

where

- f'_b is the frequency of characters in an alignment, calculated by the equation (6).
- p'_b is the frequency of characters in the background, calculated by the equation (7).

$$f'b = \frac{C_b + d_b}{N - 1 + D} \quad (6)$$

$$p'b = \frac{C_{0b} + d_b}{S + D} \quad (7)$$

where

- c_b is the number of characters which appear in an alignment.
- c_{0b} is the number of characters in the back ground.
- N is the number of total input sequences.
- S is the summation of the total characters in the background.
- d_b is the pseudo counts [30].
- D is the summation of pseudo counts.

4. Bayesian scoring is applied to calculate the motifs frequency in an alignment by Jensen et al. [31], shown in the equation (8). Equation (8) is the func-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2872978, IEEE/ACM Transactions on Computational Biology and Bioinformatics

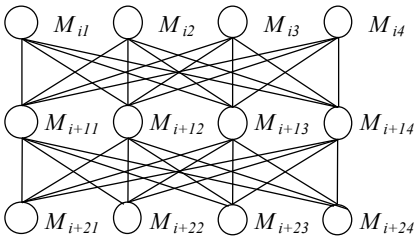


Fig. 2. The connection between subsequences in form of Cross product

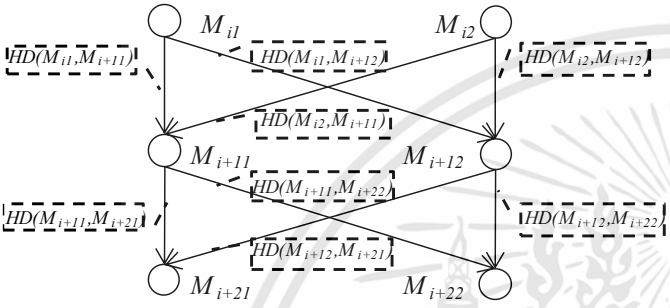


Fig. 3. The Hamming distance between subsequences

tion used to calculate the frequency of characters in the alignment with no motif sequence number limitation in an alignment.

$$\Psi(A) = |A|\left(\log(p_0/(1-p_0))\text{-}1 + \sum_{j=1}^{w}\sum_{k=1}^{4} q_{jk}\log(q_{jk}/q_{ok})\right) \quad (8)$$

where

- $|A|$ is the number of motifs in an alignment.
- p_0 is the ratio of the data in an alignment and the data in the total input sequence.
- q_{jk} is the frequency of character k in column j.
- q_{0k} is the background frequency of character k.

3 Proposed Principles and Concepts

The detection of TFBSs with multiple motif sequences per an input sequence (*MultiM*) using PSO_HD algorithm is the application of Hamming distance and PSO, including applying the cut-off minimizing false negative rate (*minFN*) to select the remaining motifs.

3.1 The Definition of TFBSs Detection Problem

Each genome sequence consists of DNA sequences or input sequences of n number, having character data in the genome sequence of N = {'A', 'C', 'G', 'T'}. Suppose each input sequence S_i has a length L and a length of each subsequence M_{ij} is w. i and j are any input sequence and subsequence, respectively. Therefore, the data of all possible subsequences in each input sequence S_i = {M_{i1}, M_{i2}, M_{i3},…, $M_{iL\text{-}w}$, $M_{iL\text{-}w+1}$}. The number of all motif of *MultiM* TFBSs in an input sequence S_i is m and the members of *MultiM* ϵ {(M_{11}, M_{12},…, $M_{1L\text{-}w}$, $M_{1L\text{-}w+1}$), (M_{21}, M_{22},…, $M_{2L\text{-}w}$, $M_{2L\text{-}w+1}$),…, (M_{n1}, M_{n2},…, $M_{nL\text{-}w}$, $M_{nL\text{-}w+1}$)}, where m <= n*(L-w+1)

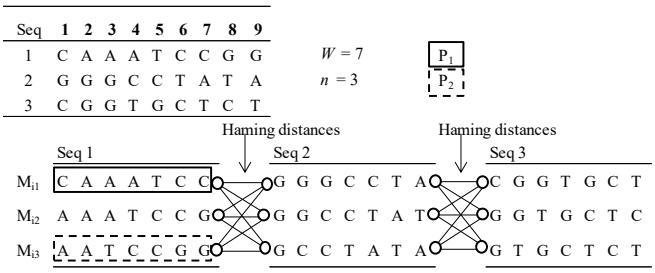


Fig. 4. Creation of the particles

3.2 Pre-Processing

To prepare a problem space which has continuation and signification, the proposed approach created related pairs between subsequences from an input sequence S_i and the next input sequence S_{i+1}, as shown in Fig. 2. The related pairs are order-connected according to the cross product, to continuously connect within the problem space. Each pair has a weight computed by Hamming distance, using base-pairing rules to convert characters into binaries, as shown in Fig. 3. The result of the pre-process is a group of subsequences M_{ij} from an input sequence S_i which are Cross joined. A weight between each input sequence is equal to HD ($\times_{HD}$). {(M_{ij} $\times_{HD}$ M_{i+1j}), (M_{i+1j} $\times_{HD}$ M_{i+2j}), (M_{i+2j} $\times_{HD}$ M_{i+3j}) …,(M_{n-2j} $\times_{HD}$ M_{n-1j}), (M_{n-1j} $\times_{HD}$ M_{nj}) }

The details are shown as follows:
{(M_{ij} *join* M_{i+1j}, M_{ij} *join* M_{i+1j+1}, M_{ij} *join* M_{i+1j+2}, …, M_{ij} *join* $M_{i+1L\text{-}w}$, M_{ij} *join* $M_{i+1L\text{-}w+1}$),

(M_{i+1j} *join* M_{i+2j}, M_{i+1j} *join* M_{i+2j+1}, M_{i+1j} *join* M_{i+2j+2}, …, M_{i+1j} *join* $M_{i+2L\text{-}w}$, M_{i+1j} *join* $M_{i+2L\text{-}w+1}$),

(M_{i+2j} *join* M_{i+3j}, M_{i+2j} *join* M_{i+3j+1}, M_{i+2j} *join* M_{i+3j+2}, …, M_{i+2j} *join* $M_{i+3L\text{-}w}$, M_{i+2j} *join* $M_{i+3L\text{-}w+1}$),
...,
(M_{n-2j} *join* M_{n-1j}, M_{n-2j} *join* M_{n-1j+1}, M_{n-2j} *join* M_{n-1j+2}, …, M_{n-2j} *join* $M_{n\text{-}1L\text{-}w}$, M_{n-2j} *join* $M_{n-1L\text{-}w+1}$),

(M_{n-1j} *join* M_{nj}, M_{n-1j} *join* M_{nj+1}, M_{n-1j} *join* M_{nj+2}, …, M_{n-1j} *join* $M_{nL\text{-}w}$, M_{n-1j} *join* $M_{nL\text{-}w+1}$)}

3.3 The Creation of Particles in the Swarm

The purpose of the pre-process is to create relations between subsequences in input sequences S_i and S_{i+1} which are all connected as shown in Fig. 4, defining i+1 <= n and using Hamming distance as weighted values of relations Any particles P_i, in TFBSs detection for k number, provide n number of subsequences data M_{ij}, where the number of particles k <= L-w+1. In each particle P_i, suppose a subsequence M_{1j} from the first input sequence S_1 has been chosen by randomization. Then, select a subsequence from the next input sequences which has the shortest Ham-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2872978, IEEE/ACM Transactions on Computational Biology and Bioinformatics

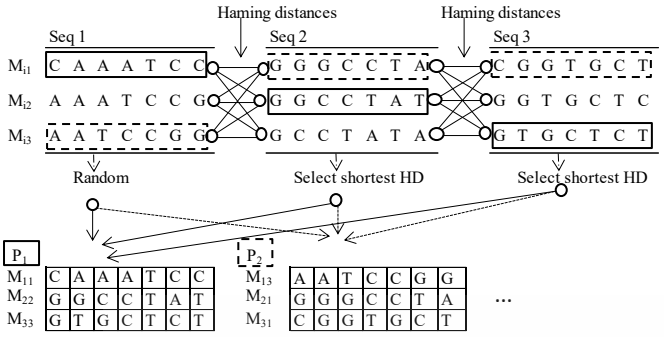


Fig. 5. Creation of the particles

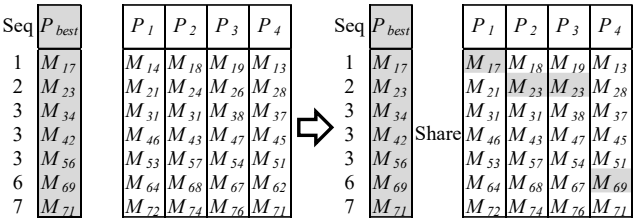


Fig. 6. The subsequence position replacement of particles with the best particle

ming distance as shown in Fig. 5. In case of more than one sequence share the shortest Hamming distance, one sequence will be randomly chosen. Therefore, the subsequence data in any particle is $P_i = \{M_{1j}, M_{2j}, \ldots, M_{n-1j}, M_{nj}\}$

3.4 The Movement of Particles in PSO_HD Algorithm

The movement pattern of PSO_HD is controlled by Gbest topology. The best particle at each iteration (P_{best}) is selected by the highest fitness value, calculated from equation IC (5). Each particle is an alignment in a matrix pattern of n rows and w columns. Where n is the number of input sequences and w is the length of subsequence.

Next, neighbor particles move towards the best particle by replacing position M_{ij} with $M_{best^{ij}}$ as shown in Fig. 6. This application of the swarm movement principle with binaries (Binary Particle Swarm Optimization: BPSO) is to create a condition of swarm movement with the particles [32], [33], [34]. Suppose that each particle P_i has an initial velocity of 0, where vector X and V is the vector of position and velocity of particle P_i, respectively. The members of vector $X = \{x_{1j}, x_{2j}, \ldots, x_{n-1j}, x_{nj}\}$ and the members of vector $V = \{v_{1j}, v_{2j}, \ldots, v_{n-1j}, v_{nj}\}$, where x_{ij} is a position of each subsequence and v_{ij} is a velocity of each subsequence. As for the movement of particles, the position x_{ij} of the particle P_i has to be replaced with the position x_{ijbest} of P_{best}, the procedure and criteria are as follows:

Step 1: Calculate the velocity v_{ij} of particle P_i using equation (1).
Step 2: Calculate $S(v_{ij}^{(t+1)})$ using equation (9) to compare with r_3 (r_3 is a random value between 0 to 1).

$$S(v_{ij}^{(t+1)}) = 1/(1+e^{v_{ij}^{(t+1)}}) \quad (9)$$

Step 3: Compare r_3 with $S(v_{ij}^{(t+1)})$. If $r_3 > S(v_{ij}^{(t+1)})$, replace x_{ij} with x_{ijbest}. If $r_3 <= S(v_{ij}^{(t+1)})$, there is no x_{ij} replacement.

PSO_HD algorithm uses these 3 steps as the criteria of movement of the swarm at each iteration until every particle is at the same position, or the algorithm has completed the maximum number of iterations. In cases of the maximum iteration limit is reached, the particle with the highest fitness value is chosen as the result, which is an alignment A.

3.5 Detection of Remaining Motifs

In the detection of d number of remaining motifs as shown in Fig. 7, used for creating *MultiM* TFBSs, we calculate the Matrix Similarity Score (mSS) [25] between n number of alignment results A from the procedure of movement of particles (the movement of particles in PSO_HD algorithm) and the remaining motif (RM), as follows:

$$mSS = \frac{(Current - Min)}{Max - Min} \quad (10)$$

$$Current: \sum_{i=1}^{L} I(i) f_{j,bi} \quad (11)$$

Define $f_{i,bi}$ as the number of character B appeared at a position i, $B \in \{'A', 'C', 'G', 'T'\}$, f_i^{min} is the minimum number of the characters at a position i according to equation (12), f_i^{max} is the maximum number of the characters at a position i according to equation (13) and $I(i)$ is the information vector accourding to equation (14).

$$Min: \sum_{i=1}^{L} I(i) f_i^{min} \quad (12)$$

$$Max: \sum_{i=1}^{L} I(i) f_i^{max} \quad (13)$$

$$I(i) = \sum_{i \in \{A,C,G,T\}} f_i ln(4 f_{i,B}), \quad i = 1,2,\ldots,L \quad (14)$$

The next procedure is to select the remaining motifs using cut-offs minimizing false negative rate (10%) [25]. The TFBSs result of PSO_HD algorithm is to merge the results from this procedure with the alignment A as shown in Fig. 8.

3.6 The Algorithm of PSO_HD

The process of TFBS detection with the PSO_HD algorithm consists of 4 main parts: parameters setting, related pairs creation, the swarm movement, and cut-offs minimizing false negative rate. The results of this process are TFBSs. As mentioned above, the pseudo-code is as follows:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2872978, IEEE/ACM Transactions on Computational Biology and Bioinformatics

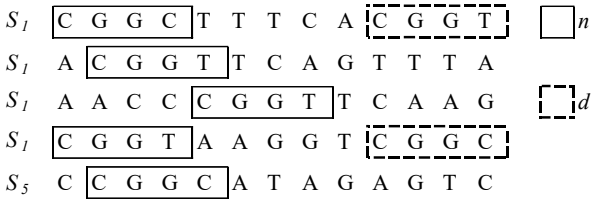


Fig. 7. An alignment and the remaining motif

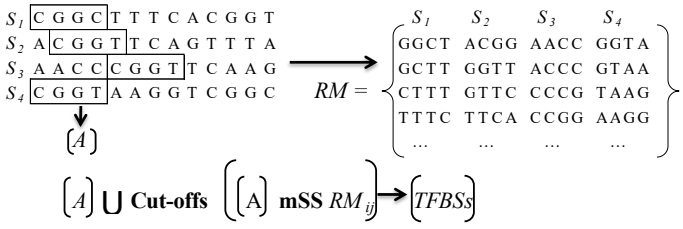


Fig. 8. The detection of remaining motifs for creating TFBSs results

Algorithm PSO_HD

Input:
w = the length of subsequence;
Maximum = maximum number of iterations;
N = number of input sequences;
l = length of input sequences;
b = {'A', 'C', 'G', 'T'};

Set:
HD[][][] = Null;
EM = null;
RM[][] = null;
Plb[] = null;
Pnb[] = null;
TFBSs as DataTable = null;
//Hamming distance
for *i*=1 to *N*
 for *j*=1 to w-*l*+1
  EM = Encrypt (*M*[*i*][*j*]);
  for *k*=1 to *w*-*l*+1
   HD[*i*][*j*][*k*] = Cal HD (*EM*, Encrypt(*M*[*i*+1][*k*]));
  end for *k*
 end for *j*
end for *i*
//Swarm movement
Initialize particle;
for i = 1 to *Maximum* **OR** not converged do
  select local best particle;
  update velocity of particles;
  update position of particles;
  if *i* = 1 or local best > global best **then**
   update global best from local best;
  end if
end for *i*
//Cut-offs minimizing false negative (minFN)
for *i*=1 to *N*
 for *j*=1 to *w*-*l*+1
  if *M*[*i*][*j*] not exist(global best) then
   if error rate of mSS(*M*[*i*][*j*]) > 10% then
    RM[*i*][*j*] = mSS(*M*[*i*][*j*]);
   end if
  end if
 end for *j*
end for *i*

Output:
TFBSs = concatenate(*RM*[*i*][*j*],global best);

4 Experimental Results and Discussions

4.1 Operation

The procedure of PSO_HD is to cooperate among other programs, developed with C# language 5.0. For the relational database, SQL Server 2012 was used for software management. The data tables in the relational database include the particle data table, the related pair table, and the total input sequences table.

4.2 Data sets and Parameters Setting

This proposed approach tests the efficiency of algorithms in detecting TFBSs for 8 genome datasets including CRP from RegulonDB database [35], ERE, MYOD and TBP from TRANSFAC database [36], CREB, E2F, MEF2 and SRF from JASPAR database [37]. These datasets have non-redundant known binding sites and are often used in efficiency testing of several algorithms [14]-[16], [22]. The details of the testing datasets are shown in Table 2, including: size, the number of input sequences, TFBSs embedded, the number of motif sequences in TFBSs, motif length and width, the length of input sequences, and the length of motif sequences.

The details of the parameters setting and the number of particles of PSO_HD algorithm for detecting TFBSs are shown in Table 3. The datasets are properly defined with typical attributes for testing datasets [22].

4.3 Discussions

The efficiency testing of relevant algorithms in detecting TFBSs uses *F*-score [14], [22], [38] as shown in equation (15). The *F*-score is the value that considers both precision and recall, which are calculated from equation (16) and (17), respectively. N_{pc} is defined as the number of the correct motif sequence result (True Positive), N_{pt} as the number of total results, and N_t as the number of total correct result. N_{pc} can be counted by comparing the results of algorithms with the results from DNA footprinting method under the rational condition of overlapping at 0.25**w* [16], as *w* is the length of the motif sequence result.

$$F\text{-score} = 2*PS*RC \,/\, PS+RC \tag{15}$$

$$PS = N_{pc}/\, N_{pt} \tag{16}$$

$$RC = N_{pc}/N_t \tag{17}$$

According to the 20-run experiment of PSO_HD algorithm, *F*-scores were compared with the best results of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2872978, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TABLE 2
THE ATRIBUTES OF THE TESTING DATASETS

Data sets	Size	TFBSs embedded	Length (bp)	Motif width
CREB	17	19	200	8
CRP	18	23	105	22
ERE	25	25	200	13
E2F	25	27	200	11
MEF2	17	17	200	7
MYOD	17	21	200	6
SRF	20	36	200	10
TBP	95	95	200	6

TABLE 3
THE PARAMETERS FOR THE PARTICLES

Parameters	Description	Size
α	inertia weight	0.4
β	cognitive	0.8
γ	social	0.8
Maximum	number of iterations	3000
Number of particles		100

TABLE 4
THE DETAILS OF *F*-SCORE OF ALGORITHMS

Algorithm	CREB	CRP	ERE	E2F	MEF	MYOD	SRF	TBP	Average
PSO_HD	**0.90**	**0.92**	0.85	**0.92**	0.98	**0.91**	**0.91**	**0.89**	0.91
PSOVa(best)	0.72	0.86	0.82	0.67	0.91	0.43	0.79	0.84	0.76
GAPK	0.73	0.91	**0.92**	0.82	**1.00**	0.83	0.74	0.88	0.86
GALF_P	0.73	0.91	0.79	0.85	0.97	0.71	0.85	0.84	0.84
GAME	0.71	0.88	0.76	0.79	0.97	0.44	0.83	0.77	0.78

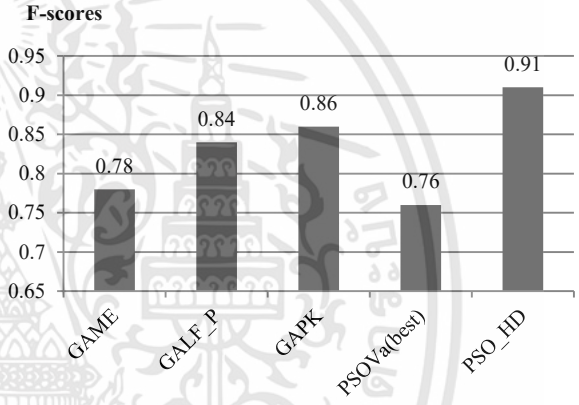


Fig. 9. Average of PSO_HD and relevant algorithm on the 8 datasets in term of *F*-scores

GA algorithm, tested by D. Wang and X. Li [16] and the best result of PSOVa algorithm, tested by M. Karabulut and T. Ibrikci [22]. Results show PSO_HD gives the highest *F*-scores in 6 of the 8 datasets which show in Table 4. In another 2 datasets, ERE and MEF, PSO_HD gives *F*-scores at 0.85 and 0.98 respectively. Overall, the proposed PSO_HD algorithm yields the highest average *F*-score as shown in Fig. 9.

By comparing *F*-score, precision and recall between each result from the 20-run experiment from 8 datasets and the results from experiments using GA by D.Wang and X.Li [17], it shows that PSO_HD gives the highest average *F*-score of 0.89 shown in Table 5, with 7 datasets of highest *F*-score and the 2nd highest after GALF_P [15] in the CRP dataset. PSO_HD gives the highest precision from 6 datasets including CREB, ERE, E2F, MYOD, SRF, and TBP, while the highest recalls appear only from 3 datasets of CREB, MEF, and TBP. However, the total average precision and recall of PSO_HD are the highest among the relevant algorithms at 0.90 and 0.88, respectively. Meanwhile, the average precision of GAME, GALF-P, GAPK, and iGAPK algorithm are at 0.64, 0.66, 0.77 and 0.77, respectively. The average recall of GAME, GALF-P, GAPK, and iGAPK algorithm are at 0.71, 0.81, 0.81 and 0.84, respectively. It also shows from PSO_HD the standard deviations of precision are between 0.03 and 0.08, the standard deviations of recall are between 0.02 and 0.09, and the standard deviations of the *F*-score are between 0.02 and 0.08.

In addition to the main purpose of this approach, using PSO_HD to improve *F*-score, precision and recall in the TFBSs detection, the algorithm can perform in a reasonable amount of time under the computational environment. The processing time of PSO_HD algorithm from 20-run experiment with 8 datasets, as shown in Table 6, are 940-1110 milliseconds in MEF2 and MYOD dataset, 1200-1462 milliseconds in CREB, ERE, E2F, and SRF dataset while spend 1560-2000 milliseconds in CRP and TBP dataset. The longest time spent in detecting TFBSs is at CRP dataset, which has diverse data patterns and has a long motif length.

5 CONCLUSION

Applying the metaheuristic algorithm to detect TFBSs is an important issue and has been continually improved in various studies. However, there are some limitations in detecting TFBSs from various datasets. This approach proposed to use PSO_HD algorithm applied with Particle Swarm Optimization (PSO) and Hamming distance (HD) to improve the quality of results. From the efficiency test

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2872978, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TABLE 5
COMPARISON OF PSO_HD AND RELEVANT ALGORITHMS ON THE 8 DATASETS FOR 20 RUNS

Algorithm		CREB	CRP	ERE	E2F	MEF	MYOD	SRF	TBP	Average
PSO_HD	Precision	**0.87**	0.94	**0.79**	**0.90**	0.95	**0.92**	**0.96**	**0.88**	0.90
	±	0.04	0.05	0.08	0.08	0.03	0.06	0.04	0.03	
	Recall	**0.94**	0.85	0.80	0.87	**0.99**	0.86	0.86	**0.90**	0.88
	±	0.06	0.04	0.07	0.09	0.02	0.08	0.02	0.02	
	F-score	**0.90**	0.89	**0.79**	**0.89**	**0.97**	**0.88**	**0.91**	**0.89**	0.89
	±	0.05	0.03	0.07	0.08	0.02	0.06	0.02	0.02	
iGAPK	Precision	0.68	0.90	0.73	0.69	0.87	0.83	0.75	0.73	0.77
	±	0.06	0.05	0.15	0.02	0.10	0.06	0.04	0.10	
	Recall	0.65	0.84	**0.88**	0.83	0.92	**0.92**	0.81	0.83	0.84
	±	0.06	0.03	0.03	0.06	0.04	0.08	0.05	0.04	
	F-score	0.66	0.87	0.79	0.75	0.89	0.87	0.78	0.77	0.80
	±	0.06	0.02	0.11	0.03	0.06	0.05	0.03	0.06	
GAPK	Precision	0.65	**0.96**	0.71	0.66	**1.00**	0.68	0.63	0.83	0.77
	±	0.04	0.06	0.15	0.02	0.00	0.14	0.05	0.06	
	Recall	0.68	0.80	0.89	0.92	0.77	0.82	0.74	0.83	0.81
	±	0.07	0.04	0.06	0.05	0.05	0.06	0.05	0.06	
	F-score	0.66	0.86	0.78	0.77	0.87	0.74	0.67	0.83	0.79
	±	0.04	0.04	0.10	0.02	0.03	0.09	0.04	0.06	
GALF-P	Precision	0.47	0.95	0.65	0.67	0.85	0.28	0.68	0.74	0.66
	±	0.24	0.02	0.15	0.08	0.16	0.24	0.12	0.12	
	Recall	0.60	**0.88**	0.84	**0.93**	0.94	0.51	0.88	0.86	0.81
	±	0.29	0.05	0.04	0.05	0.06	0.45	0.06	0.02	
	F-score	0.53	**0.91**	0.72	0.78	0.89	0.36	0.76	0.80	0.73
	±	0.26	0.04	0.10	0.07	0.11	0.32	0.09	0.09	
GAME	Precision	0.44	0.93	0.63	0.62	0.90	0.24	0.67	0.67	0.64
	±	0.31	0.05	0.07	0.05	0.05	0.17	0.06	0.28	
	Recall	0.43	0.84	0.84	0.86	0.96	0.24	**0.92**	0.58	0.71
	±	0.30	0.03	0.06	0.09	0.06	0.16	0.06	0.24	
	F-score	0.43	0.88	0.72	0.72	0.93	0.24	0.78	0.62	0.67
	±	0.32	0.03	0.06	0.06	0.04	0.16	0.06	0.25	

TABLE 6
THE PROCESSING TIME COMPARISON OF THE 20-RUN EXPERIMENT BETWEEEN EACH DATASET

	MEF2	MYOD	CREB	ERE	E2F	SRF	CRP	TBP
Mininum Time (ms)	940	1031	1312	1239	1200	1320	1745	1560
Maximum Time (ms)	1023	1110	1462	1378	1297	1400	2000	1698

in precision and recall with 8 datasets, it shows that the result of PSO_HD algorithm outperforms the other tested methods. While, the results of using PSO_HD algorithm with ERE and MEF dataset give the second highest *F*-scores, their values are still higher than the other 5 algorithms. Furthermore, when we compare the average *F*-score of the experiments for 20 runs to GA, it shows that PSO_HD algorithm gives the highest Fscore in all 7 datasets, except for CRP dataset. However, PSO_HD algorithm still has the best average *F*-score of all 8 datasets. Therefore, results indicate PSO_HD algorithm can improve the efficiency in detecting TFBSs with higher precision and recall. The algorithm can perform in a reasonable amount of time under the computational environment. The computation time of TFBSs can be improved in the future study, by applying parallel processing [39], especially, in livings which have long genome sequences.

REFERENCES

[1] D. Maier, "The complexity of some problems on subsequences and supersequences," *J. ACM*, vol. 25, no. 2, pp. 322–336, 1978.

[2] R. Riviere, D. Barth, J. Cohen, and A. Denise, "Shuffling biological sequences with motif constraints," *J. Discrete Algorithms*, vol. 6, no. 2, pp. 192–204, 2008.

[3] J. van Helden, B. Ander, and L. Collado-Vides, "Extracting regulatory sites from upstream region of yeast genes by computational analysis of oligonucleotide frequencies," *J. Mol. Biol.*, vol. 281, no. 5, pp. 827–842, 1998.

[4] T.L. Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers," *in: Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, vol. 2, pp. 28–36, 1994.

[5] F.P. Roth, J.D. Hughes, P.W. Estep, and G. M. Church, "Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation," *Nat. Biotechnol.*, vol. 16, no. 10, pp. 939–945, 1998.

[6] X.S. Liu, D.L. Brutlag and J.S. Liu, "BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes," *in: Proceedings of the Pacific Symposium on Biocomputing*, vol. 6, pp. 127–138, 2001.

[7] G.Z. Hertz and G.D. Stormo, "Identifying DNA and protein patterns with statistically significant alignments of multiple sequences," *Bioinformatics*, vol. 15, no. 7, pp. 563–577, 1999.

[8] X.S. Liu, D.L. Brutlag and J.S. Liu, "An algorithm for finding protein–DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments," *Nat. Biotechnol.*, vol. 20, no. 8, pp. 835–839, 2002.

[9] C. Notredame and D.G. Higgins, "SAGA: Sequence alignment by genetic algorithm," *Nucleic Acids Res.*, vol. 24, no. 8, pp. 1515–1524, 1996.

[10] F.F.M. Liu, J.J.P. Tsai, R.M. Chen, S.N. Chen , and S.H. Shih, "FMGA: finding motifs by genetic algorithm," *IEEE Fourth Symposium on Bioinformatics and Bioengineering (BIBE 2004)*,

2004.

[11] D. Che, Y. Song, and K. Rasheed, "MDGA: motif discovery using a genetic algorithm," *Genetic and Evolutionary Computation (GECCO 2005)*, pp. 447–452, 2005.

[12] M. Dorigo and V. Maniezzo, "A. Colomi, Ant system: optimization by a colony of coorperating agents," *IEEE Trans. Syst. Man Cybern. – Part B*, vol. 26, no. 1, pp. 29–41, 1996.

[13] L. Shao and Y.Chen, "Bacterial foraging optimization algorithm integrating tabu search for motif discovery," *In: IEEE International Conference on Bioinformatics and Biomedicine (BIBM'09)*, 2009.

[14] Z. Wei and S.T. Jensen, "GAME: detecting cis-regulatory elements using a genetic algorithm," *Bioinformatics*, vol. 22, no. 13, pp. 1577–1584, 2006.

[15] T.-M. Chan, K.-S. Leung, and K.-H. Lee, "TFBS identification based on genetic algorithm with combined representations and adaptive post-processing," *Bioinformatics*, vol. 24, no. 3, pp. 341–349, 2008.

[16] D.H. Wang and X. Li, "GAPK: Genetic algorithms with prior knowledge for motif discovery in DNA sequences," *In: CEC 2009: IEEE Congress on Evolutionary Computation 2009*, 2009.

[17] D. Wang and X. Li, "iGAPK: Improved GAPK algorithm for regulatory DNA motif discovery," *In: Neural Information Processing*, 2010.

[18] Q. Yu, H. Huo, X. Chen, H. Guo and J. Scott, "An Efficient Algorithm for Discovering Motifs in Large DNA Data Sets," *IEEE Trans. Nanobioscience,* vol. 14, no. 5, 2015

[19] Y. Fan, W. Wu, J. Yang, W. Yang and R. Liu, "An Algorithm for Motif Discovery with Iteration on Lengths of Motifs," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 12, no. 1, pp. 136-141, 2015

[20] K. Wong, C. Peng and Y. Li, "Evolving Transcription Factor Binding Site Models From Protein Binding Microarray Data," *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 415-424, 2017

[21] I. Roy and S. Aluru, "Discovering Motifs in Biological Sequences Using the Micron Automata Processor," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 13, no. 1, 2016.

[22] M. Karabulut and T. Ibrikci, "A Bayesian Scoring Scheme based Particle Swarm Optimization algorithm to identify transcription factor binding sites," *Applied Soft Computing*, vol. 12, pp. 2846–2855, 2012.

[23] M. Stine, D. Dasgupta and S. Mukatira., "Motif discovery in upstream sequences of coordinately expressed genes," *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003 (Conference proceedings)*, 2003.

[24] A. J. F., Griffiths, H. Miller, D. T. Suzuki, R. C. Lewontin and W. M. Gelbart, "An Introduction to Genetic Analysis.," *New York: W. H. Freeman, seventh edition*, 2000.

[25] A.E. Kel, E. Go¨ßling, I. Reuter, E. Cheremushkin, O.V. Kel-Margoulis and E. Wingender "MATCHTM: a tool for searching transcription factor binding sites in DNA sequences Nucleic Acids Research," *Nucleic Acids Res.*, vol. 31, no. 13, 2003.

[26] J. Kennedy and R. Eberhart, "Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks," *IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[27] J. Kennedy and R. Mendes, "Population structure and particle swarm performance, in: Proceedings of the Evolutionary Computation on 2002. CEC '02," *IEEE Computer Society*, vol. 02, pp. 1671–1676, 2002.

[28] W. Liu, H. Chen and L. Chen, "An ant colony optimization based algorithm for identifying gene regulatory elements," *Computers in Biology and Medicine*, vol.43, pp. 922–932, 2013.

[29] G.D. Storm, "Computer methods for analyzing sequence recognition of nucleic acids," *Annu. Rev. BioChem.*, vol. 17, pp. 241–263, 1988.

[30] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald and Wooton, J.C. "Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment," *Science*, vol. 262, no. 1531, pp. 208–214, 1993.

[31] S. Jensen, S. Liu, Q. Zhou and J. Liu "Computational discovery of gene regulatory binding motifs: a Bayesian perspective," *Statistical Science*, no. 19, pp. 188–204, 2004.

[32] M. S. Mohamad, S. Omatu, S. Deris, M. Yoshioka, A. Abdullah, and Z. Ibrahim, "An enhancement of binary particle swarm optimization for gene selection in classifying cancer classes," *Algorithms Mol. Biol.*, vol. 8, no. 15, 2013.

[33] N. Jin and Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multi-objective implementations," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 556–567, 2007.

[34] M. Mandal, J. Mondal, and A. Mukhopadhyay, "A PSO-Based Approach for Pathway Maker Identification From Gene Expression Data," *IEEE Trans. Nanobioscience*, vol. 14, no. 6, pp. 591–597, 2015

[35] A.M. Huerta, H. Salgado, D. Thieffry and J. Collado-Vides, "RegulonDB: a database on transcriptional regulation in Escherichia coli," *Nucleic Acids Res.*, vol. 26, no. 1, pp. 55-59, 1998.

[36] V. Matys, O.V. Kel-Margoulis, E. Fricke, et al., "TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes," *Nucleic Acids Res.*, vol. 34, pp. D108–D110, 2006.

[37] J.C. Bryne, E. Valen, et al., "JASPAR: the open access database of transcription factor-binding profiles: new content and tools in the 2008 update," *Nucleic Acids Res.*, vol. 36, pp. 102–106, 2008.

[38] W.M. Shaw, R. Burgin and P. Howell, "Performance standards and evaluations in IR test collections: cluster-based retrieval models," *Information Processing & Management*, vol. 33, 1–14, 1997.

[39] L. Dagum and R. Menon, "OpenMP: An industry standard API for shared memory programming," *IEEE Comput. Sci. Eng.*, vol. 5, no. 1, pp. 46-55, 1998.

Sarawoot Som-in received his Master degree in the Department of Computer Science from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand in 2010, where he is currently working toward the Ph.D. degree. His research interests include design and analysis of algorithm and bioinformatics.

Warangkhana Kimpan received her Ph.D. degree in System Information Engineering from Kagoshima University, Japan. She is currently an assistant professor in Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. Her main research interests are Swarm Intelligence, Biomedical Engineering, Big Data, Data Science and analytics, Cloud Computing, and Internet of Things.

ประวัติผู้เขียน

ชื่อ	นายศราวุธ โสมอินทร์
วัน เดือน ปีเกิด	7 มกราคม 2526
ที่อยู่ปัจจุบัน	2288/22 ชั้น 5 อาคารเดอะรูม สุขุมวิท 62 ถนน สุขุมวิท แขวงบางจาก เขตพระโขนง กรุงเทพมหานคร 10260
ประวัติการศึกษา	พ.ศ. 2548 วิทยาศาสตรบัณฑิต สาขา วิทยาการคอมพิวเตอร์ เกรดเฉลี่ย 2.17 มหาวิทยาลัยหัวเฉียวเฉลิมพระเกียรติ พ.ศ. 2554 วิทยาศาสตรมหาบัณฑิต สาขา วิทยาการคอมพิวเตอร์ เกรดเฉลี่ย 3.22 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ทุนการศึกษาที่ได้รับ	-
ผลงานทางวิชาการ	1. Sarawoot Som-in and Warangkhana Kimpan, “NexusPSO: A Novel Algorithm to Detect Transcription Factor Binding Sites,” IAENG International Journal of Computer Science, Vol. 45, no. 3, pp. 478-487, 2018. 2. Sarawoot Som-in and Warangkhana Kimpan, “Enhanching of Particle Swarm Optimization Based Method for Multiple Motifs Detection in DNA Sequences Collections,” IEEE/ACM Transactions on Computational Biology and Bioinformatics, ISSN: 1545-5963.