ขั้นตอนวิธีเชิงวิวัฒน์ดัดแปลงสำหรับการจัดเรียงลำดับเบสหลายลำดับ

A Modified Evolutionary Algorithm for Multiple Sequence Alignment

พสุตม์ สีเหลืองสวัสดิ์ และ ประภาส จงสถิตย์วัฒนา
Pasut Seeluangsawat and Prabhas Chongstitvatana
Department of Computer Engineering
Chulalongkorn University, Bangkok 10300, Thailand
e-mail: g46psl@cp.eng.chula.ac.th , prabhas@chula.ac.th

บทคัดย่อ

ปัญหาการจัดเรียงลำดับเบสหลายลำดับเป็นปัญหาทางด้านชีวสารสนเทศศาสตร์(Bioinformatics) ที่มีมานาน และมีการค้นคว้าปัญหานี้อย่างกว้างขวาง ซึ่งได้มีเครื่องมือที่ได้รับความนิยมมากคือ โปรแกรมครัสตอล(Clustal)แต่เนื่องจากคำตอบที่ได้จากครัสตอลนั้นยังไม่ใช่คำตอบที่ดีที่สุด ดังนั้นจึงสามารถพัฒนาคำตอบเพื่อที่จะให้ได้คำตอบที่ดีขึ้นต่อไป

งานวิจัยนี้นำเอาขั้นตอนวิธีเชิงวิวัฒน์มาพัฒนาคำตอบต่อจากครัสตอลโดยใช้ตัวดำเนินการปรับปรุงสามชนิด ซึ่งจะสามารถทำให้ได้คำตอบที่ดีขึ้นในส่วนของข้อมูลลำดับเบสนั้นได้มาจากฐานข้อมูลบาลิเบส(BaliBASE) และการทดลองนี้ได้ใช้การวัดค่าประสิทธิภาพด้วยวิธีผลรวมคู่เบส(Sum of Pair)

Abstract

Multiple sequence alignment (MSA) is a problem in bioinformatics. It is well studied and there is one popular tool to solve this problem, Clustal. However, the solution obtained from Clustal is not optimal, it can be improved.

This work proposed using evolutionary algorithms to improve the solution obtained from Clustal. Three improvement operators are introduced. The experiment data is from BAliBASE. The measurement of effectiveness of the proposed method is sum-of-pair scoring.

Key-Words: multiple sequence alignment, Clustal, evolutionary algorithms, BaliBASE

1.ความเป็นมาและความสำคัญของปัญหา

การจัดลำดับเบสหลายลำดับนั้นเป็นปัญหาพื้นฐานทางด้านชีวสารสนเทศศาสตร์ และเป็นพื้นฐานของชีววิทยาระดับโมเลกุล (molecular biology) ลำดับเบสแต่ละลำดับที่มีความใกล้เคียงกันทั้งทางโครงสร้างและองค์ประกอบ จะแสดงถึงความใกล้เคียงกันทางสายพันธุ์ที่มี นำไปสู่การสร้างต้นไม้วิวัฒนาการ (phylogenetic tree)ได้ และการเปรียบเทียบกันระหว่างลำดับเบสนั้นยังสามารถทำให้ได้ข้อมูลโครงสร้างในขั้นที่สอง และขั้นที่สาม(secondary and tertiary structure) [1], [2]ได้ด้วย

การจัดเรียงลำดับเบสหลายลำดับเป็นปัญหาที่แก้ได้ยาก เพราะลำดับที่ต้องการเปรียบเทียบกันนั้นไม่ได้เหมือนกันทั้งหมด เนื่องจากลำดับเบสนั้นมี การลบ(deletion) และการแทรก (insertion)อยู่อย่างสม่ำเสมอตั้งแต่ในอดีต โดยขบวนการแทรกนั้นเป็นตัวที่ทำให้เบสในตำแหน่งนั้นเปลี่ยนไปจากเดิม

วิธีที่ใช้ในการแก้ปัญหาการจัดเรียงลำดับเบสหลายลำดับที่นิยมนั้นใช้กำหนดการพลวัต(dynamic programming) แต่วิธีนี้มีข้อเสียคือ ขนาดของปัญหาจะใหญ่ขึ้นตามจำนวนของความยาว และจำนวนของลำดับเบส ปัญหาการจัดเรียงลำดับเบสหลายลำดับจึงถูกจัดอยู่ในปัญหาที่แก้ได้ยาก(NP-hard) ทำให้มีการนำวิธีการศึกษาสำนึก(heuristic)มาประยุกต์ใช้ เช่นวิธีการจัดเรียงแบบก้าวหน้า(progressive alignment algorithm) วิธีนี้จะค่อยๆทำการจัดเรียง โดยขั้นแรกมีการประมาณระยะทางของการวิวัฒนาการ(evolutionary distance)ระหว่างทุกๆลำดับ

เบส ต่อมาจึงนำลำดับที่ได้มาจัดเรียงตามลำดับความใกล้เคียงกันของวิวัฒนาการ โปรแกรมที่ใช้วิธีนี้และเป็นที่รู้จักกันดีคือ คร้สตอลเอ็กซ์ คร้สตอลดับเบิลยู และคร้สตอลวี(ClustalX, ClustalW and ClaustalV) [3], [4] อย่างไรก็ตามวิธีขั้นตอนการจัดเรียงแบบก้าวหน้านั้นอาจไม่ได้คำตอบที่ดีที่สุด(local optima) อีกวิธีหนึ่งคือวิธีการทำซ้ำ(iterative) ที่มีทั้งวิธีการจำลองการหลอมละลาย(simulate annealing) และขั้นตอนวิธีเชิงวิวัฒน์(Evolutionary Algorithms) [5], [6], [7], [8], [9], [10], [11], [12], [13] ทั้งวิธีการจำลองการหลอมละลาย และวิธีขั้นตอนเชิงวิวัฒน์นั้นมีประสิทธิภาพในการหาคำตอบของปัญหาขนาดใหญ่ได้ดี แต่วิธีการจำลองการหลอมละลาย กับการแก้ปัญหาการจัดเรียงลำดับเบสหลายลำดับนั้นอาจเกิดการลู่เข้าหาคำตอบที่ไม่ดีได้ ซึ่งขั้นตอนวิธีเชิงวิวัฒน์มีวิธีในการที่หาคำตอบได้ดีกว่า [8] อย่างไรก็ตามขั้นตอนวิธีเชิงวิวัฒน์นั้นยังมีปัญหาบางอย่าง เช่นการที่ต้องใช้เวลามากในการหาคำตอบจากขั้นตอนต่างๆ เช่น การสร้างกลุ่มของประชากรในตอนเริ่มต้น และต้องใช้เวลาในการที่จะค่อยๆปรับปรุงคำตอบเพื่อที่จะให้ได้คำตอบที่ดีที่สุด เป็นต้น

2.วิธีในการแก้ปัญหาการจัดเรียงลำดับเบสหลายลำดับ

2.1.การเข้ารหัสปัญหา

การแก้ปัญหาการจัดเรียงลำดับเบสหลายลำดับใช้ตารางของตัวอักษรในการจัดเก็บข้อมูล โดยตัวอักษรของลำดับโปรตีนจะมีทั้งหมดเท่ากับ 21 ตัวอักษร ดังนี้{C, S, T, P, A, G, N, D, E, Q, H, R, K, M, I, L, V, F, Y, W, X} โดย ‘X’ คือโปรตีนที่ยังไม่รู้จัก และจะมีแก็ป ‘-‘ เพิ่มขึ้นมาอีกหนึ่งตัวอักษรเพื่อแสดงถึงการแทรก หรือการลบที่เกิดขึ้นจากการวิวัฒนาการในสิ่งมีชีวิต ตารางนี้จะมีความกว้างเท่ากับจำนวนลำดับที่มีการจัดเรียง และมีความยาวเท่ากับความยาวของลำดับที่มีลำดับโปรตีนที่ยาวที่สุดคูณกับ 1.2 คือมีการเพิ่มความยาวให้กับผลเฉลยตั้งต้นอีก 20%

2.2.การสร้างผลเฉลยตั้งต้น

นำเอาคำตอบของโปรแกรมคร้สตอลเอ็กซ์มาเป็นผลเฉลยตั้งต้น โดยเพิ่มแก็ปต่อท้ายลำดับเบสจนเต็มความกว้างของตาราง

การนำเอาคำตอบของโปรแกรมคร้สตอลเอ็กซ์มาเป็นผลเฉลยตั้งต้นเป็นการลดเวลาในการหาคำตอบ เพราะโปรแกรมคร้สตอลเอ็กซ์สามารถหาคำตอบที่ดีได้ในเวลาสั้น

2.3.ตัวปฏิบัติการในการปรับปรุงให้กับผลเฉลย

ในขั้นตอนวิวัฒนาการ ผลเฉลยทุกตัวจะถูกปรับปรุงโดยตัวปฏิบัติการแตกต่างกันตามโอกาส โปรแกรมนี้มีตัวปฏิบัติการทั้งหมด 3 ตัวคือ DirectedLocalShuffle [8], การสร้างหลักตัวอักษร และการจัดเรียงบางส่วนแบบนีดเดิลแมน-วูนซ์ โดยตัวปฏิบัติการทั้ง 3 ตัวนี้มีขั้นตอนการทำงานดังนี้

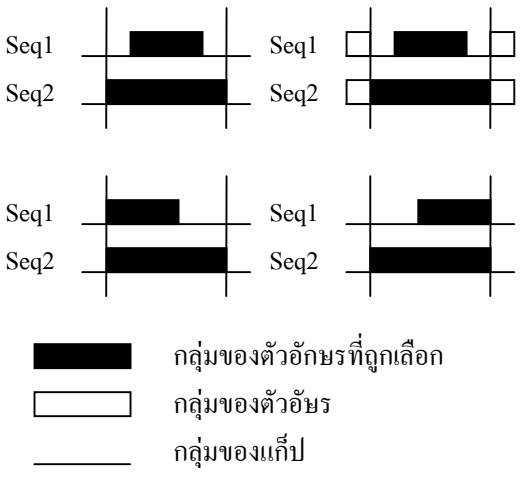
ตัวปฏิบัติการ DirectedLocalShuffle จะทำการสุ่มเลือกแถวและตัวอักษร โดยตัวอักษรที่ถูกเลือกนั้นจะต้องมีแก็ปอยู่ติดกัน จากนั้นจะทำการเคลื่อนตัวอักษรที่ถูกเลือกนั้นไปทางแก็ปที่ติดกันอยู่ และจะไปหยุดที่ตำแหน่งที่ให้คะแนนมากที่สุด

ตัวปฏิบัติการ การสร้างหลักตัวอักษร จะตรวจว่าหลักที่ i และ i+1 สามารถทำให้เกิดหลักที่มีตัวอักษรที่เหมือนกันได้หรือไม่โดยการแทรกแก็ป ซึ่งถ้าสามารถทำให้เกิดหลักที่มีตัวอักษรเหมือนกันได้ จากนั้นจะทำการตรวจว่าหลักที่ i หรือ i+1 นั้นแถวไหนเป็นแถวที่มีตัวอักษรที่ทำให้เกิดหลักที่มีตัวอักษรเหมือนกันได้มากกว่ากัน จากนั้นทำการเลือกแถวที่มีตัวอักษรมากกว่า และทำการใส่แก็ปดังรูปที่1 ซึ่งขั้นตอนนี้จะมีการทำทุกๆหลักในผลเฉลย

i	i+1		i	i+1	i+2			i	i+1		i	i+1	i+2
A	**G**		**A**	**G**	**-**			**G**	**A**		**-**	**G**	**A**
G	**B**	**>**	**-**	**G**	**B**			**B**	**G**	**>**	**B**	**G**	**-**
C	**G**		**C**	**G**	**-**			**G**	**C**		**-**	**G**	**C**

รูปที่ 1. วิธีการเพิ่มแก็ปของตัวปฏิบัติการสร้างหลักตัวอักษร

ตัวปฏิบัติการจัดเรียงบางส่วนแบบนีดเดิลแมน-วูนซ์ จะทำการสุ่มแถว และกลุ่มของตัวอักษร จากนั้นจะทำการสุ่มหาอีกแถว โดยอีกแถวจะต้องอยู่ในเงื่อนไขที่ว่าต้องครอบคลุมแถวแรก ยกตัวอย่างเช่นในรูปที่2



รูปที่ 2. ตัวอย่างการเลือกแถวที่สองที่ใช้ในการคำนวณของตัวปฏิบัติการการจัดเรียงบางส่วนแบบนีดเดิลแมน-วูนซ์

ต่อจากนั้นนำตำแหน่งในแถวที่หนึ่ง และแถวที่สองที่ถูกเลือกมาเข้าขั้นตอนวิธีนีดเดิลแมน-วูนซ์(Needleman-Wunsch algorithm) [2] ซึ่งเป็นกำหนดการพลวัตชนิดหนึ่ง เมื่อได้คำตอบแล้วนำคำตอบที่ได้ไปใส่ในแถวและช่วงเดิม ถ้าคำตอบมีแก็ปในแถวที่สองเพิ่มจะมีการเพิ่มแก็ปทั้งหลักนั้น

ในขั้นตอนวิธีนีดเดิลแมน-วูนซ์มีการกำหนดค่าทำโทษไว้เท่ากับ -8 และใช้ตารางบลอซัม45 (BLOSUM45) ในการคำนวณ

หลังจากได้ผลเฉลยจากตัวปฏิบัติการต่างๆแล้วอาจเกิดหลักที่เป็นแก็ปทั้งหลักขึ้น ต้องย้ายหลักที่เป็นแก็ปทั้งหมดนั้นไปยังตำแหน่งด้านขวาสุดของผลเฉลยเสมอ

2.4.การวัดค่าประสิทธิภาพของผลเฉลย

การวัดค่าประสิทธิภาพ(fitness) ใช้การให้รางวัล และการทำโทษ โดยการให้รางวัลจะใช้วิธีผลรวมคู่เบส(sum of pair) และการทำโทษนั้นจะใช้ค่าคงที่ในการทำโทษ โดยมีสมการดังนี้

$$fitness = SymbolScore - GapPenaltyScore \quad (1)$$

SymbolScore เป็นค่าผลรวมของคู่ลำดับ ค่านี้ขึ้นอยู่กับตารางแต่ละชนิดเช่น แพม(PAM) [15] หรือ บลอซัม(BLOSUM) [14] เป็นต้น ซึ่งตารางแต่ละชนิดนั้นบอกความเป็นไปได้ในการที่จะเกิดหลักที่เหมือนกัน หรือหลักที่ไม่เหมือนกัน

$$SymbolScore = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} BLOSUM45(l_i, l_j) \quad (2)$$

และ GapPenaltyScore ใช้เจาะจงในการทำโทษสำหรับทุกแก็ปที่อยู่ในทุกๆลำดับ โดยความสัมพันธ์ของค่าในการทำโทษแก็ปคือ

$$GapPenaltyScore = Constants \times GAPS \quad (3)$$

GAPS คือจำนวนของแก็ปทั้งหมด และ Constants คือค่าคงที่ในการทำโทษสำหรับแก็ป

2.5.ขั้นตอนการวิวัฒนาการ

ขั้นตอนวิธีที่นำเสนอนี้มีชื่อว่าอีเอ (EA) มีขั้นตอนการวิวัฒนาการโดยอธิบายเป็นรหัสเทียมดังรูปที่ 3

```
procedure EA
  t = 0
  สร้างประชากรตั้งต้น P(t) ด้วยโปรแกรมครัสตอล
  หาค่าประสิทธิภาพของประชากร P(t)
  while (not เงื่อนไขจบการวิวัฒนาการ)
    การกลายพันธุ์ผลเฉลยในประชากร P(t)
    หาค่าประสิทธิภาพของประชากร P(t)
    คัดเลือกประชากร P(t+1) จากประชากร P(t)
    t = t+1
end
```

รูปที่ 3. ขั้นตอนวิธีอีเอ

ขั้นตอนวิธีอีเอใช้ขั้นตอนวิธีเชิงวิวัฒน์ในการแก้ปัญหา โดยการพัฒนาผลเฉลยไปเรื่อยๆจนกว่าจะถึงจำนวนรุ่นที่กำหนด ระหว่างการวิวัฒนาการนั้นประชากรรุ่นลูกจะเกิดจากการสุ่มเลือกประชากรในรุ่นของพ่อแม่ โดยประชากรรุ่นลูกจะถูกปรับปรุงผลเฉลยด้วยตัวดำเนินการทั้งสามชนิดตามอัตราส่วนที่กำหนด ในแต่ละรุ่นที่มีการปรับปรุงผลเฉลยเมื่อได้ประชากรรุ่นลูกครบถ้วนแล้วจะทำการคัดเลือกเพื่อหาประชากรรุ่นพ่อแม่ของรุ่นใหม่ต่อไป

การคัดเลือกประชากรพ่อแม่รุ่นใหม่ใช้การคัดเลือกแบบมิวบวกแลมดา ((λ+μ)-selection) โดยประชากรรุ่นพ่อแม่ของรุ่นใหม่นั้นจะแบ่งเป็นสองส่วนคือ ส่วนแรกจะคัดเลือกจากจากประชากรรุ่นลูก และรุ่นพ่อแม่ที่มีค่าประสิทธิภาพมากที่สุดเรียงตามลำดับมา ส่วนที่สองจากประชากรรุ่นลูก และรุ่นพ่อแม่ที่มีจำนวนแถวที่มีอักษรเดียวกันมากที่สุด โดยที่แถวที่มีอักษรเดียวกันมากที่สุดนั้นจะต้องมีค่าประสิทธิภาพมากที่สุดด้วย ประชากรรุ่นพ่อแม่ของรุ่นใหม่ทั้งสองส่วนนั้นจะแบ่งในอัตราส่วนที่แตกต่างกัน ซึ่งในการทดลองนี้ได้ทำการแบ่งให้ส่วนแรก 20% และส่วนที่สอง 80% ทางด้านผลเฉลยที่เหลือจะทิ้งไป

2.6.ค่าที่กำหนดในพารามิเตอร์สำหรับการทดลอง

โปรแกรมอีเอกำหนดค่าพารามิเตอร์ดังนี้ จำนวนรุ่นในการวิวัฒนาการเท่ากับ 2000 รุ่น จำนวนประชากรรุ่นพ่อแม่เท่ากับ 50 จำนวนประชากรรุ่นลูกเท่ากับ 30 ค่าความน่าจะเป็นในการเลือกตัวดำเนินการ Directed LocalShuffle เท่ากับ 0.6 ค่าความน่าจะเป็นในการเลือกตัวดำเนินการสร้างหลักตัวอักษร เท่ากับ 0.2 ค่าความน่าจะเป็นในการเลือกตัวดำเนินการจัดเรียงบางส่วนแบบนีดเดิลแมน-วูนซ์ เท่ากับ 0.2 ตารางที่ใช้หาค่าผลรวมคู่เบสคือ ตารางบลอซัม45 และค่าคงที่ในการทำโทษสำหรับแก็ปเท่ากับ 10

2.7.ข้อมูลที่ใช้ในการทดลอง

ตารางที่1 เป็นตารางข้อมูลที่ได้จากฐานข้อมูลบาลิเบส [16] ซึ่งได้เลือกมา 8 ข้อมูลในการทดลอง

ตารางที่ 1. ข้อมูลที่ใช้ในการทดลองจากฐานข้อมูลบาลิเบส โดย NSEQ คือจำนวนลำดับ และ LSEQ คือความยาวของลำดับ

DATA SET	NSEQ	LSEQ (min,max,avg)
1taq	5	(806,928,865.2)
1pii	4	(247,259,251.5)
1pfc	5	(108,117,112)
1hfh	5	(116,135,121.2)
451c	5	(70,87,80)
kinase	5	(263,276,270.2)
1aboA	5	(49,80,63.6)
1tvxA	4	(54,70,61.75)

3.ผลการทดลอง

ตารางที่ 2 เปรียบเทียบค่าประสิทธิภาพระหว่างครัสตอลเอ็กซ์กับอีเอ ข้อมูลจากอีเอ แสดงค่าดีที่สุด และค่าเฉลี่ยทุกชุดข้อมูลค่าประสิทธิภาพของอีเอดีกว่า

ตารางที่ 3 เปรียบเทียบจำนวนหลักที่มีอักษรเหมือนกันระหว่าง ครัสตอลเอ็กซ์, MSAEA, และอีเอ ทุกชุดข้อมูล ค่าจากอีเอสูงที่สุด

ผลที่ได้เป็นผลที่ดีที่สุดจากการทำการทดลองทั้งหมด 20 ครั้ง และใช้จำนวนรุ่นในการทดลองทั้งหมด 2000 รุ่น เมื่อสังเกตดูค่าประสิทธิภาพจากในรูปแล้วอาจเป็นไปได้อีกที่จะสามารถหาค่าประสิทธิภาพได้มากขึ้นเมื่อเพิ่มจำนวนรุ่น หรือจำนวนประชากรให้มากกว่าเดิม

การทดลองทั้งหมดได้ใช้เวลาในการประมวลผล 20-110 วินาที ใช้เครื่องที่มีความเร็วนาฬิกา 1.67 GHz หน่วยความจำ 256 MB และใช้ระบบปฏิบัติการ WindowXP เวลาในการประมวลผลนั้นขึ้นอยู่กับจำนวนความยาว และจำนวนลำดับของผลเฉลย

ผลการทดลองที่ได้มานั้นเป็นผลที่ได้โดยจากการใช้พารามิเตอร์ชุดเดียวสำหรับข้อมูลทุกชนิด ซึ่งถ้าต้องการคำตอบที่ดีมากขึ้นก็สามารถทำการปรับเปลี่ยนพารามิเตอร์ได้ เพื่อความเหมาะสมของแต่ละชุดข้อมูล

ตารางที่ 2. แสดงค่าประสิทธิภาพเทียบกับครัสตอลเอ็กซ์

Data Set	ClustalX	EA	
	*	Best	Mean
1taq	15312	16075	15954
1pii	1306	1555	1478
1pfc	1452	1653	1620
1hfh	1341	1717	1668
kinase	149	594	532
451c	381	653	605
1aboA	-505	-262	-300
1tvxA	-345	-249	-281

ตารางที่ 3. แสดงจำนวนหลักที่มีอักษรเหมือนกันเทียบกับครัสตอลเอ็กซ์ และ MSAEA

Data Set	ClustalX	MSAEA	EA
1taq	164	165	176
1pii	31	34	37
1pfc	11	12	16
1hfh	12	13	14
kinase	14	15	20
451c	5	5	7
1aboA	2	2	3
1tvxA	1	1	1

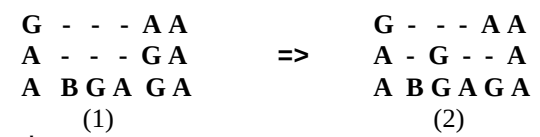
4.วิจารณ์ผลการทดลอง

โปรแกรมประเภทขั้นตอนวิธีเชิงวิวัฒน์นั้นใช้เวลาในการคำนวณเป็นเวลานานมาก เมื่อเทียบกับโปรแกรมครัสตอล แต่คำตอบที่ได้จากโปรแกรมครัสตอล นั้นถ้าลำดับที่นำมาจัดเรียงมีความแตกต่างกันมากแล้ว คำตอบที่ได้อาจไม่ดีที่สุด วิธีที่เสนอจึงนำคำตอบจากโปรแกรมครัสตอล มาเป็นผลเฉลยตั้งต้นแล้วนำเอาขั้นตอนเชิงวิวัฒน์มาปรับปรุงคำตอบ

โปรแกรมที่ใช้ขั้นตอนวิธีเชิงวิวัฒน์ที่ใช้ผลเฉลยตั้งต้นจากโปรแกรมประเภทครัสตอล นั้นมีโปรแกรม MSAEA ซึ่งพัฒนาโดย Rene Thomsen , Gary B. Fogel และ Thiemo Krink [8] โปรแกรม MSAEA นั้นพยายามหาหลักที่มีตัวอักษรเหมือนกันให้ได้มากที่สุด แต่ก็ยังคงมีบางหลักที่ยังไม่สามารถจัดเรียงกันให้เป็นตัวอักษรเดียวกันได้

การเลือกใช้การวัดค่าประสิทธิภาพ โดยใช้เพียงค่าคงที่ในการทำโทษเพียงค่าเดียวนั้นมีสาเหตุมาจาก ถ้าใช้ค่าการทำโทษแก็ปที่อยู่ติดกันร่วมกับ ค่าการทำโทษแก็ปแรก ทำให้เกิดการปรับปรุงคำตอบได้ยาก แต่ถ้าใช้ค่าคงที่เพียงอย่างเดียวในการทำโทษแล้วถึงแม้จะเกิดการปรับปรุงผลเฉลยไปแล้ว แต่ผลเฉลยที่โดนปรับปรุงอาจจะถูกคัดเลือกไปใช้ในรุ่นต่อไปได้ ยกตัวอย่างเช่น

จากรูปที่ 4 รูป (1) เปลี่ยนไปเป็นรูป (2) เมื่อค่าการทำโทษแก็ปที่อยู่ติดกันร่วมกับค่าการทำโทษแก็ปแรก จะทำให้ค่าประสิทธิภาพลดลง แต่ถ้าทำโทษโดยใช้ค่าคงที่เพียงอย่างเดียว ค่าประสิทธิภาพที่ได้จะมีค่าเท่าเดิม



รูปที่ 4. แสดงค่าเปรียบเทียบระหว่างการทำโทษที่ใช้ค่าการทำโทษแก็ปที่อยู่ติดกันร่วมกับค่าการทำโทษเริ่มต้น กับค่าคงที่เพียงอย่างเดียว

โปรแกรมนี้พัฒนาตัวดำเนินการขึ้นมาใหม่สองตัว ซึ่งสามารถพัฒนาผลเฉลยตั้งต้นให้มีค่าประสิทธิภาพให้มากขึ้น หรือมีจำนวนหลักที่เป็นอักษรเดียวกันให้มากขึ้น

จากผลการทดลอง ผลที่ได้จากตัวดำเนินการสร้างหลักตัวอักษร เป็นการเพิ่มหลักของตัวอักษรที่เหมือนกัน โดยการเพิ่มแก็ปเข้าไปในลำดับโปรตีน อาจกล่าวได้ว่าเมื่อมีการเพิ่มแก็ปเข้าไปในตำแหน่งที่เหมาะสมแล้ว จะสามารถทำให้เกิดหลักที่เป็นตัวอักษรที่เหมือนกันได้ ดังนั้นควรพัฒนาตัวดำเนินการซึ่งทำการใส่แก็ปลงในลำดับของโปรตีนให้น้อยที่สุดแล้วสามารถ ทำให้หาผลเฉลยที่ดีขึ้นได้

จากการทดลองกับปัญหาทำให้สังเกตได้ว่า ค่าประสิทธิภาพอาจจะมีค่าน้อยลงจากรุ่นที่ผ่านมา เนื่องจากมีการเพิ่มแก็ปลงไปในลำดับโปรตีนเพื่อทำการเพิ่มหลักที่มีตัวอักษรเหมือนกันให้เพิ่มขึ้น จึงกล่าวได้ว่าบางทีค่าประสิทธิภาพที่ได้จากวิธีผลรวมคู่เบสนั้นอาจจะไม่ใช้วิธีที่ดีที่สุด ที่จะบอกว่าคำตอบที่ได้เป็นคำตอบที่ดีที่สุดก็เป็นได้

5.รายการอ้างอิง

[1] C. Gibas and P. Jambeck, "Developing Bioingormatics Computer Skills," O'Reilly, 2001.

[2] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, "Biological sequence analysis Probabilistic models of proteins and nucleic acids," Cambridge University Press, 2001.

[3] J. Thompson, D. Higgins, and T. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice," Nucleic Acids Research, vol. 22, pp. 4673-4680, 1994.

[4] J. Thompson, T. Gibson, F. Plewniak, F. Jeanmougin, and D. Higgins, "The ClustalX windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools," Nucleic Acids Research, vol. 24, pp. 4876–4882, 1997.

[5] C. Zhang and A. Wong, "A technique of genetic algorithm and sequence synthesis for multiple molecular sequence alignment,".;Systems, Man, and Cybernetics, vol. 3, pp. 2442 - 2447, 1998.

[6] C. Zhang and A. Wong, "Toward efficient multiple molecular sequence alignment: a system of genetic algorithm and dynamic programming,".;Systems, Man and Cybernetics, vol. 27, pp. 918 - 932, 1997.

[7] C. Zhang, "A genetic algorithm for molecular sequence comparison," Man, and Cybernetics, vol. 2, pp. 1926 - 1931, 1994.

[8] R. Thomsen, G. Fogel, and T. Krink, "Improvement of Clustal-Derived Sequence Alignments with Evolutionary Algorithms.," Proceedings of the Fifth Congress on Evolutionary Computation, 2003.

[9] R. Thomsen, G. Fogel, and T. Krink, "A Clustal Alignment Improver using Evolutionary Algorithms.," Proceedings of the Fourth Congress on Evolutionary Computation, 2002.

[10] C. Notredame and D. Higgins, "SAGA: sequence alignment by genetic algorithm," Nuc. Acids Res, vol. 24(8), pp. 1515-1524, 1996.

[11] K. Karadimitriou and D. Kraft, "Genetic Algorithms and the Multiple Sequence Alignment problem in Biology," Proceedings of Annual Molecular Biology and Biotechnology Conference, 1996.

[12] M. Isokawa, M. Wayama, and T. Shimizu, "Multiple sequence alignment using a genetic algorithm," Genome Informatics, vol. 7, pp. 176-177, 1996.

[13] K. Chellapilla and G. Fogel, "Multiple sequence alignment using evolutionary programming," Congress on Evolutionary Computation, pp. 445-452, 1999.

[14] S. Henikoff and J. Henikoff, "Amino acid substitution matrices from protein blocks.," Proc. Natl. Acad. Sci., vol. 89, pp. 10915–10919, 1992.

[15] S. Gupta, J. Kececioglu, and A. Schaffer, "Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment.," J. Comp. Bio., vol. 2, pp. 459–472, 1995.

[16] F. Plewniak, J. Thompson, and O. Poch, "BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs," Bioinformatics, vol. 15, pp. 87–88, 1999.