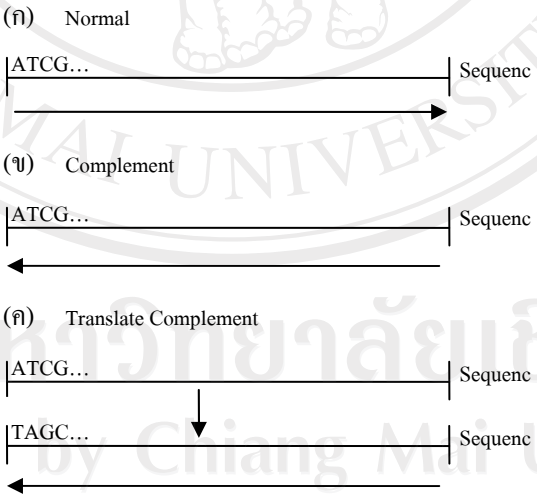
บทที่ 4
วิธีดำเนินการวิจัย

งานวิจัยชิ้นนี้ใช้สายลำดับดีเอ็นเอจีโนมของแบคทีเรียเป็นข้อมูลในการทดลอง ซึ่งแบ่งเป็นสายลำดับเบส (sequence) ย่อย ๆ จำนวน 385 ไฟล์ และแบ่งการทดลองเป็น 3 ขั้นตอนดังนี้

ขั้นที่ 1 พัฒนาโปรแกรมขึ้นใหม่โดยใช้หลักการของ Fickett J.W. (1982) โดยเครื่องมือที่ใช้พัฒนาโปรแกรมคือภาษา R version 2.2.0 โดยมีตัวแปรสำคัญ 3 ตัวแปรดังนี้

ตัวแปรแรกคือวิธีการอ่านสายลำดับเบสทดสอบมาเป็นข้อมูลเข้า (input) แบ่งเป็น 3 วิธีคือ วิธี Normal เป็นการอ่านสายลำดับเบสจากเบสแรกเป็นตำแหน่งที่ 1 ไปยังเบสสุดท้ายเป็นตำแหน่งสุดท้ายตามปกติดังรูป 4.1(ก) วิธี Complement จะอ่านสายลำดับเบสจากเบสสุดท้ายเป็นตำแหน่งที่ 1 ไปยังลำดับเบสแรกเป็นตำแหน่งสุดท้าย ดังรูป 4.1(ข) และอีกวิธีหนึ่งคือวิธี Translate Complement จะทำการแปลงเบสในสายลำดับเบส จาก T เป็น A, A เป็น T, C เป็น G และ G เป็น C หลังจากนั้นอ่านสายลำดับเบสเช่นเดียวกับวิธี Complement ดังรูป 4.1(ค)

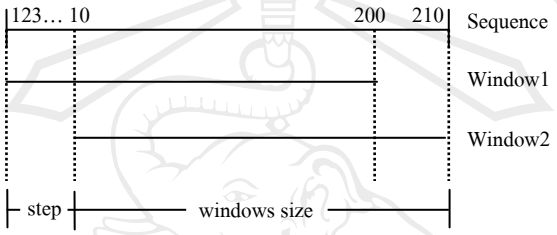


รูป 4.1 วิธีการอ่านสายลำดับเบส 3 วิธี

ตัวแปรที่สองคือการกำหนดค่าขนาดของ window (windows size) ในการตัดสายลำดับเบสออกเป็น window ย่อย ๆ เพื่อนำมาทำนาย ทั้งนี้เพราะว่าทฤษฎีของ Fickett J.W. (1982) ตอบคำถามได้เพียงว่า สายลำดับเบสที่นำมาทดสอบเป็น coding region หรือไม่ จึงต้องแบ่งสายลำดับเบส

ออกเป็น window ย่อย ๆ เพื่อสามารถที่จะระบุตำแหน่งของส่วนที่เป็น coding region ได้ดังรูป 4.2 ซึ่งในผลการทดลองของ Fickett J.W. (1982) ได้บอกไว้ว่าผลการทำนายจะมีความถูกต้องสูง เมื่อขนาดของสายลำดับเบสที่นำมาทดสอบมีความยาวมากกว่า 200 เบสจึงทำการกำหนดขนาด windows size เป็น 200 และ 500

ตัวแปลสุดท้ายคือ ระยะห่างของจุดเริ่มต้นระหว่างแต่ละ window (step) ดังรูป 4.2 โดยกำหนดให้เท่ากับ 10 เหตุผลที่ไม่ขยับทีละ 1 เพราะทำให้เวลาในการคำนวณเพิ่มขึ้น โดยผลการทำนายที่ได้แทบไม่แตกต่างกัน



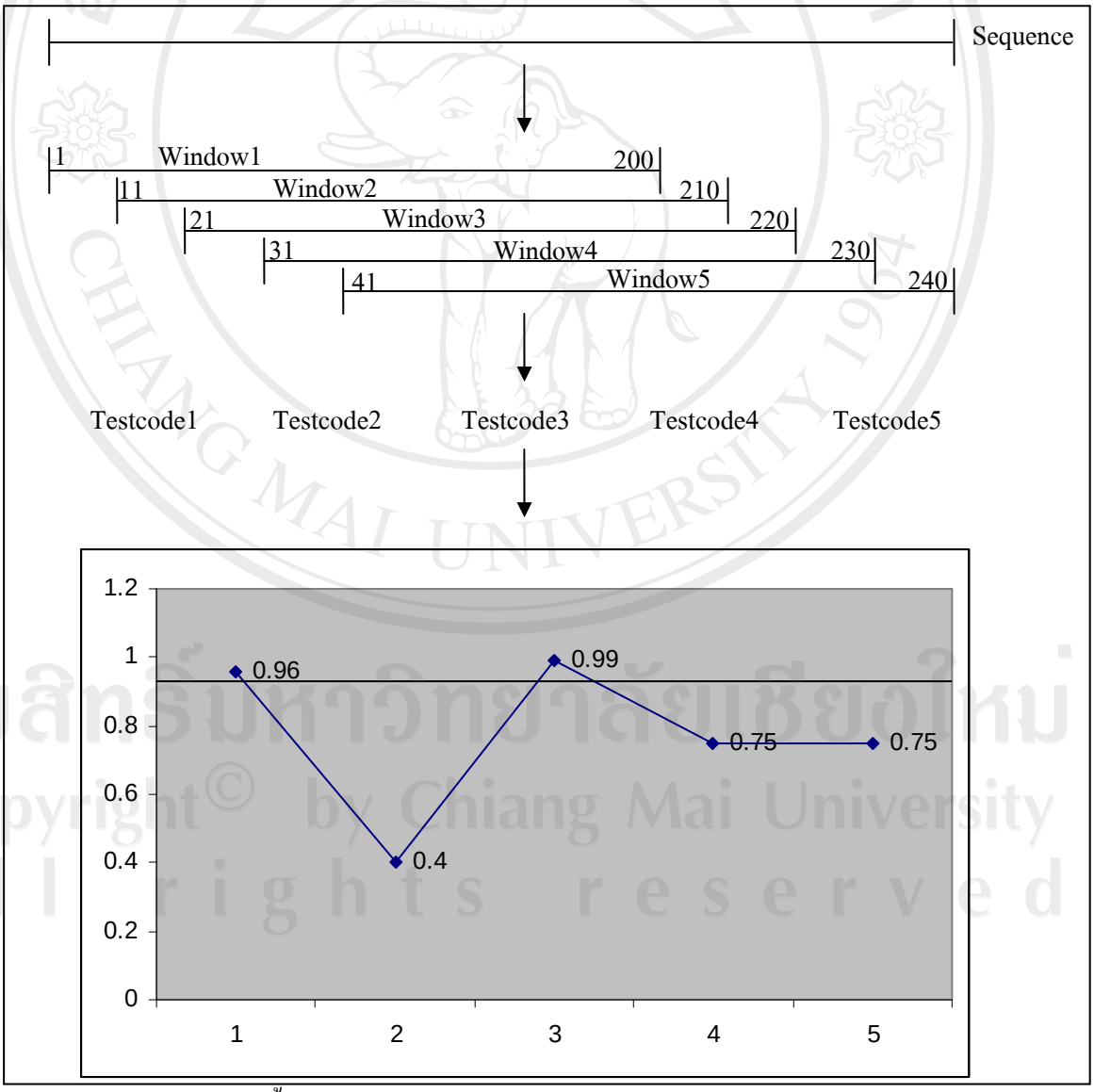
รูป 4.2 อธิบายตัวแปร step และ windows size

อัลกอริทึมในการทำงานของโปรแกรมมีดังนี้

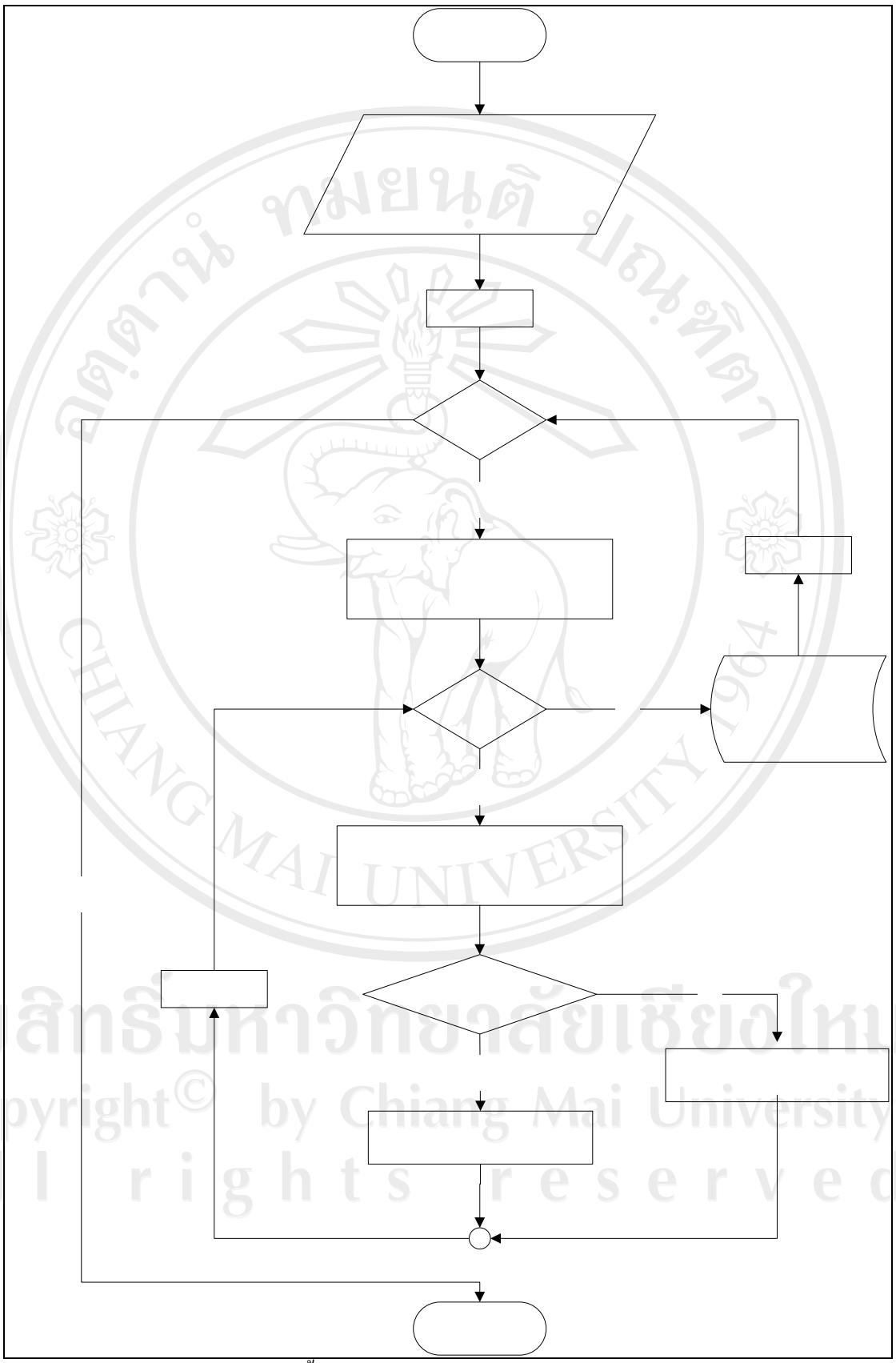
```
1.    read all sequences to parameter input
2.    for i <- 1 to 385
3.    do  {calculate n <- number of window's input[i]
4.        for j <- 1 to n
5.        do  {calculate TESTCODE from Fickett's method
6.            answer window[j] is coding or not
7.            }
8.        Find coding region of input[i]
9.        Check overlap of result
10.       Save result of input[i] to file
11.       }
```

การทำงานของโปรแกรมเริ่มต้นโดยการอ่านข้อมูลลำดับเบสจาก text ไฟล์ หลังจากนั้นในบรรทัดที่ 3 แบ่งลำดับเบสออกเป็น windows ย่อย ๆ จำนวน n window ตาม windows size ที่กำหนด บรรทัดที่ 4, 5 และ 6 เป็นการวนลูปคำนวณค่า Testcode ของแต่ละ window ตามวิธีการ

ของ Fickett เพื่อหาว่า window ใดบ้างเป็น coding ต่อมาในบรรทัดที่ 8 เป็นการค้นหาว่าข้อมูลลำดับเบสนั้น ๆ มี coding region อยู่ที่ตำแหน่งใดบ้าง โดยหากค่า Testcode มากกว่าหรือเท่ากับ 0.95 จะสรุปว่า window นั้น ๆ เป็น coding region หลังจากนั้นบรรทัดที่ 9 ทำการตรวจสอบความเหลื่อมทับกันของผลการทำนาย โดยหากตำแหน่งของลำดับเบสที่เป็น coding region เหลื่อมทับกันก็จะสรุปผลเป็นตำแหน่งเริ่มต้นและสิ้นสุดของ coding region เพียงชุดเดียว ดังตัวอย่างในรูป 4.3 ผลการทำนายคือ window1 ลำดับเบสที่ 1 ถึง 200 และ window3 ลำดับเบสที่ 21 ถึง 220 เป็น coding region แต่ผลการทำนายมีความเหลื่อมทับกัน จึงสรุปผลการทำนายได้ว่า ลำดับเบสที่ 1 ถึง 220 เป็น coding region สุดท้ายบรรทัดที่ 10 เก็บผลการทำนายของ input[i] ลง text ไฟล์ แล้ววนลูปทำนายลำดับเบสไฟล์ต่อๆ ไปจนครบ 385 ไฟล์



รูป 4.3 ขั้นตอนการทำงานของโปรแกรมทำนายยีนด้วยวิธี Ficket

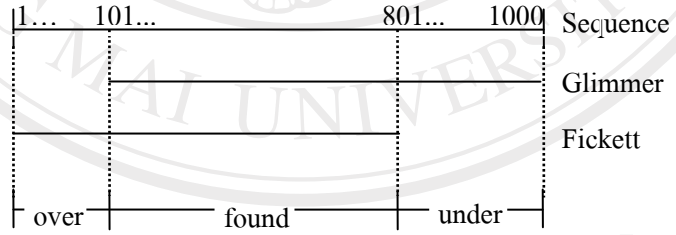


รูป 4.4 Flowchart แสดงขั้นตอนการทำงานของโปรแกรมทำนายยีนด้วยวิธี Ficket

ขั้นที่ 2 เป็นการทดลองนำข้อมูลทั้งหมดมาทำนายหาว่าช่วงใดบ้างเป็น coding region ด้วยโปรแกรมที่ได้จากขั้นที่ 1 โดยแบ่งทดลองเป็น 6 กลุ่มจากวิธีการอ่านสายลำดับเบสและการกำหนดค่า windows size ดังนี้

1) อ่านสายลำดับเบสวิธี Normal และกำหนด windows size = 200
2) อ่านสายลำดับเบสวิธี Normal และกำหนด windows size = 500
3) อ่านสายลำดับเบสวิธี Complement และกำหนด windows size = 200
4) อ่านสายลำดับเบสวิธี Complement และกำหนด windows size = 500
5) อ่านสายลำดับเบสวิธี Translate Complement และกำหนด windows size = 200
6) อ่านสายลำดับเบสวิธี Translate Complement และกำหนด windows size = 500

ขั้นที่ 3 นำผลจากโปรแกรม Glimmer version 3.0 มาเปรียบเทียบกับผลทั้ง 6 กลุ่มที่ได้ในขั้นที่ 2 ว่าผลการระบุตำแหน่งยีนจากทั้ง 2 โปรแกรมของข้อมูลแต่ละตัวมีความแตกต่างกันมากน้อยเพียงใด โดยแบ่งผลการพิจารณาเป็น 3 ส่วนคือ found, under และ over ซึ่งอธิบายในรูป 4.5 โดย found คือส่วนที่ผลการทำนายของโปรแกรมทั้งคู่ซ้อนทับกัน under คือส่วนที่โปรแกรม Fickett ทำนายได้น้อยกว่าโปรแกรม Glimmer และ over คือส่วนที่โปรแกรม Fickett ทำนายได้มากกว่าโปรแกรม Glimmer แล้วหาค่าเฉลี่ยร้อยละของ found, under และ over ในแต่ละกลุ่ม



รูป 4.5 แสดงส่วน found, under และ over

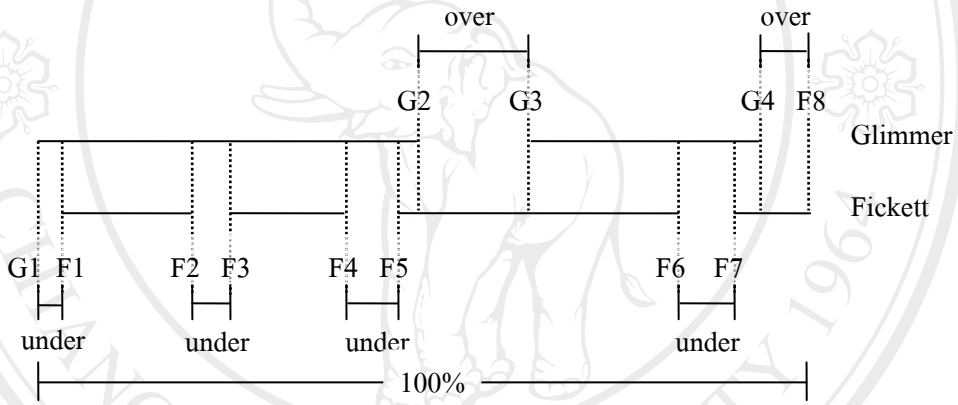
จากรูป 4.5 ลำดับเบสนี้มีความยาว 1000 เบส เมื่อลองนำผลจากโปรแกรมทั้งสองมาพิจารณาจะได้ found มีค่าเท่ากับ 700 เบส คิดเป็น 70 เปอร์เซ็นต์ over มีค่าเท่ากับ 100 เบส คิดเป็น 10 เปอร์เซ็นต์ และ under มีค่าเท่ากับ 200 เบส คิดเป็น 20 เปอร์เซ็นต์เป็นต้น

ในการทำการทดลองจริงผลการทำนายของทั้งสองโปรแกรมจะมีความซับซ้อนมากกว่าที่ยกตัวอย่างไว้ในรูป 4.5 ดั้งนั้นจึงต้องพิจารณาความสัมพันธ์ระหว่างผลการทำนายของโปรแกรมทั้ง

สองว่าควรนำผลการทำนายชุดใดบ้างมาพิจารณาหาส่วน found, under และ over ดังตัวอย่างผลการทำนายในตาราง 4.1

ตาราง 4.1 ตัวอย่างผลการทำนายของโปรแกรมทั้งสองที่ต้องนำมาพิจารณาหาความสัมพันธ์

name	Glimmer	Fickett	length
Contig156	101 - 1193 1498 - 2170	170 - 500 620 - 950 1150 - 1890 2030 - 2240 2350 - 2940	4590



รูป 4.6 แสดงการคำนวณหาส่วน found, under และ over

จากตัวอย่างผลการทดลองในตาราง 4.1 นำมาเขียนเป็นรูปคำนวณหาส่วน found, under และ over ได้ดังรูป 4.6 และมีวิธีคำนวณดังนี้

ความยาวทั้งหมดของผลการทำนายที่นำมาคำนวณ กำหนดเป็น 100 เปอร์เซ็นต์ คำนวณจาก ตำแหน่งท้ายสุดของผลการทำนายที่นำมาคำนวณลบด้วยตำแหน่งแรกสุดของผลการทำนายที่นำมาคำนวณ จากรูป 4.6 ได้ 100% = F8 – G1

over คือส่วนที่โปรแกรม Fickett ทำนายได้มากกว่าโปรแกรม Glimmer จากรูป 4.6 ได้ over = (G3 – G2) + (F8 – G4)

under คือส่วนที่โปรแกรม Fickett ทำนายได้น้อยกว่าโปรแกรม Glimmer จากรูป 4.6 ได้ under = (F1 – G1) + (F3 – F2) + (F5 – F4) + (F7 – F6)

เมื่อแทนค่าตัวแปลทั้งหมดด้วยค่าจากตาราง 4.1 ได้ผลลัพธ์ดังนี้

100% = 2,240 – 101 = 2,139 เบส

over = (1,498-1,193) + (2,240-2,170) = 305 + 70 = 375 เบส = 17.53 %

under = (170-101) + (620-500) + (1,150-950) + (2,030-1,890)

= 69 + 120 + 200 + 140 = 529 เบส = 24.73 %

found = 2,139 – (375 + 529) = 1,235 เบส = 57.74 %

เมื่อได้ผลการคำนวณส่วน found, under และ over ของแต่ละสายลำดับเบสแล้ว จึงทำการหาค่าเฉลี่ยของส่วน found, under และ over ในแต่ละกลุ่มตามวิธีการอ่านสายลำดับเบสและการกำหนดค่า windows sizeทั้ง 6 กลุ่ม ดังที่อธิบายไว้ในขั้นที่ 2 ซึ่งตัวอย่างผลการทดลองในขั้นต่าง ๆ จะแสดงไว้ในบทถัดไป บทที่ 5 ผลการทดลอง