อธิบายรูปที่ 1

ตอนแรกจะประมวลผลไฟล์ binary เป็นชุดของ non overlapping [ เพิ่มเติม รหัสพันธุกรรมไม่มีการเหลื่อมซ้อนกัน ( non overlapping ) ในกระบวนแปลรหัส พบว่ารหัสพันธุกรรมจะถูกอ่านเรียงต่อกันอย่าง เป็นลำดับๆ ละ 3 เบส นับตั้งแต่จุดเริ่มต้นโดยไม่มีการอ่านการเหลื่อมซ้อนกันของเบสที่อยู่ต่าง รหัสพันธุกรรมกัน ] จากนั้นจะทำขั้นตอนการคำนวณ 2 รอบ : Luby transform and screening. Luby transform เป็นพื้นฐานของการทำ fountain codes. โดยจะใส่ข้อมูลลงไปในจำนวนที่ต้องการของข้อความสั้นๆ ซึ่งเรียกว่า droplets. โดยจะเลือกจากการสุ่ม subset of segment จากไฟล์ที่ได้รับการแจกแจงด้วยวิธีพิเศษ จากนั้นเพิ่ม subset of segment เข้าไปแล้วทำการ bitwise ภายใน binary field. Droplet เก็บข้อมูลอยู่ 2 ส่วน : เก็บส่วนของข้อมูลที่ผลของการเพิ่ม procedure [เพิ่มเติม set of coded instructions that tell a computer how to run a program ] ตามความยาวของ seed [ เพิ่มเติม number or other value that is generated by software ]. Seed เป็นสถานะจากการสุ่มเลขที่ทำการสร้างขึ้นของการแปลงระหว่างการสร้าง Droplet และขั้นตอนการ decoder algorithm เพื่อบอกถึงความ unique ของ segments ใน Droplet. ในทางทฤษฎี เป็นไปได้ว่า Luby transform จะใช้ algorithm ที่มีประสิทธิภาพสูงในการรวม subset ของ droplet โดยขนาดของ droplet จะมีขนาดที่ใหญ่กว่าขนาดของไฟล์ต้นฉบับ

DNA Fountain เป็น algorithm ที่ใช้การแปลงซ้ำๆของแต่ละ single droplet ขั้นตอนต่อไป algorithm จะย้าย droplet ไปยัง screening stage ซึ่ง stage นี้ไม่ได้เป็นส่วนหนึ่ง original fountain code โดย stage นี้จะแปลง binary droplet เป็น ลำดับของ DNA โดยเปลี่ยน {00,01,10,11} เป็น {A,C,G,T} หลังจากนั้นต้องพิจารณาคุณสมบัติทาง biochemical คือปริมาณของ GC และ homopolymer [เพิ่มเติม พอลิเมอร์ที่ได้จากมอนอเมอร์เพียงชนิดเดียว] ถ้าคุณสมบัติผ่าน จะพิจารณาค่าและเพิ่มลงไปใน oligo

Decoding

เราใช้ขั้นตอนต่อไปนี้ในการถอดรหัสข้อมูล

1. ก่อนการประมวลผล เราจะทำการจับ paired-end reads ( โคดอนอ่านไป-อ่านย้อนเหมือนกัน เช่น AGA ) ออกให้เหลือรหัสอันเดียว โดยจะเก็บ sequences ที่มีความยาว 152nt เท่านั้น หลังจากนั้นเราจะทำการ sorted วิธีการนี้ตัว decoder จะสนใจข้อมูลที่มีคุณภาพสูงสุดก่อนและค่อย ๆ สนใจลำดับที่มีคุณภาพต่ำลง เนื่องจากความซ้ำซ้อนของวิธีนี้ ตัว decoder จะไม่ต้องการ oligos ทั้งหมดที่สามารถเปลี่ยนเป็น original file และจะหยุดก่อนที่จะถอดรหัสตัวที่สนใจ ( ? will usually stop before attempting to decode sequences that were observed a small number of times) ซึ่งจะมีแนวโน้มที่จะผิดพลาดมากกว่า

Point ของ decoder processs จะทำงานตามลำดับและ executes 2 ขั้นตอนบนแต่ละลำดับที่ยุบลงจนกว่าไฟล์จะได้รับการแก้ไขอย่างสมบูรณ์

2. การเปลี่ยน Droplet คืน : decoder จะ map ลำดับ DNA ให้เป็นเป็นรูปแบบ binary โดยจะเปลี่ยน {A, C, G, T} เป็น {0, 1, 2, 3} ถัดไปจะแยกส่วนวิเคราะห์ลำดับเพื่อแยก seed, data payload และรหัสการแก้ไขข้อผิดพลาด (ถ้ามี) หากมี code ที่ error ตัว decoder จะเป็นคนตรวจเช็คว่ามี error ที่ไหนยังไง โดยในการทดลองนี้ เราจะแยกส่วนที่ error ออกและจะไม่แก้ไขมันให้ถูกต้อง เราพบว่าถ้ายิ่ง error เยอะเนื่องจากการ เพิ่มข้อมูลสั้นและการลบ จะส่งผลถึงประสิทธิภาพในการสังเคราะห์ oligo

ReedSolomon สามารถจัดการกับ error ได้โดยการแทนทีแล้พยายามแก้ไข code ให้ถูกต้องแต่จะทำให้เกิดการกู้คืนข้อมูลผิดพลาดมากขึ้น นอกจากนี้ผู้ใช้ยังสามารถกำหนดให้ decoder จัดการ error ได้โดยการใช้ k-mers

[k-mers for GTAGAGCTGT ->

k = 1 G, T, A, G, A, G, C, T, G, T

k=2 GT, TA, AG, GA, AG, GC, CT, TG, GT จน k = len]

ภายใต้เงื่อนไขนี้ตัวถอดรหัสจะเปรียบเทียบลำดับใหม่ที่อ่านได้กับการอ่านที่ประมวลผลก่อนหน้านี้ทั้งหมดที่ใช้สำหรับการถอดรหัส