ชื่อหัวข้อที่เลือกทำ (Term project Title)

Structural variant detection

ชื่อสมาชิกในกลุ่ม (Team members)

- 6331308021 ชานน รัตนจรัสกุล
- 6331340021 รัชพล สุคนธมาน
- 6331337121 ภูรินท์ แต่งศรีวรรณ
- 6331315321 ธีรโชติ เดชสุวรรณกิจ

Progress Report #1

ที่มาและความสำคัญ (Background and Motivation / Rationale)

Structural variants ส่งผลให้เกิดโรคทางพันธุกรรม จึงได้มีการพัฒนาเทคโนโลยี และอัลกอริทึมต่างๆ เพื่อตรวจจับ DNA sequence ที่ผิดปกติ ทำให้เป็นประโยชน์ในการรักษาโรคต่างๆ ได้

งานวิจัยที่เกี่ยวข้อง (อย่างน้อย 3 เรื่อง) (Related Work)

งานวิจัยที่ 1

Paragraph: a graph-based structural variant genotyper for short-read sequence data

https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1909-7

1) เป็นงานวิจัยเกี่ยวกับอะไร ทำไมถึงเป็นปัญหา (ให้สรุปมา)
(Background and Motivation of this research)

Structural variants (SVs) ทำให้เกิดความหลากหลายของจีโนม และส่งผลให้เกิดโรคทางพันธุกรรม มีเทคโนโลยีใหม่ๆ สำหรับ long-read sequencing ที่ช่วยทำให้การตรวจจับ SVs ง่ายขึ้น รวมถึงสามารถตรวจจับ SVs ที่มีความซับซ้อนระดับต่ำ และที่อยู่ใน non-unique regions ของ genome ได้ แต่เมื่อเทียบกับ short read แล้ว แบบ long-read มีความแม่นยำและน่าเชื่อถือมากกว่า ทีมผู้วิจัยจึงได้สร้าง graph-based genotyper Paragraph ที่สามารถระบุ SVs จากตัวอย่าง short read ที่มีปริมาณมาก และเมื่อเปรียบเทียบกับ วิธี linear reference-based อื่นๆ แล้ว วิธีของ sequence graph ช่วยลด bias กับ allele ต้นแบบได้ และยังให้ผลที่ถูกต้องแม้ความหลากหลายของจีโนมจะกระจุกรวมกัน และเมื่อเปรียบเทียบกับวิธีตรวจจับ SV อื่นๆ ที่โด่งดัง Paragraph ให้ผลที่มีความถูกต้องสูงที่สุด

2) สิ่งที่งานวิจัยนี้นำเสนอ และวิธีการที่ใช้ (Proposed methodology)

Paragraph เป็นเครื่องมือในการตรวจจับและระบุ DNA ของ sequence structural variations (SVs) ที่มีความแม่นยำ จาก short-read data โดยใช้วิธีการต่างๆ ได้แก่ Graph construction, Graph alignment, Breakpoint genotyping, SV genotyping, Sequence data, Long-read ground truth and performance evaluation, Estimation of breakpoint deviation, Population genotyping and annotation

3) ข้อมูลที่ใช้ (Data)

Paragraph software เปิดให้สามารถใช้งานได้สาธารณะบน GitHub https://github.com/Illumina/paragraph ภายใต้ Apache 2.0 license

4) ผลการทดลอง (Experimental results)

Paragraph is an accurate SV genotyper for short-read sequencing data that scales to hundreds or thousands of samples. Paragraph implements a unified genotyper that works for both insertions and deletions, independent of the method by which the SVs were discovered. Thus, Paragraph is a powerful tool for studying the SV landscape in populations, human or otherwise, in addition to analyzing SVs for clinical genomic sequencing applications.

5) อภิปรายผลของงานวิจัยนี้ (Discussion)

Paragraph is an accurate graph-based SV genotyper for short-read sequencing data. It can genotype both deletions and insertions genome-wide, including those within complicated regions. Paragraph can tolerate breakpoint deviation of up to 10 bp in most genomic contexts, although performance suffers as the breakpoints deviate by more bases. Paragraph works by aligning and genotyping reads on a local sequence graph constructed for each targeted SV. This approach is different from other proposed and most existing graph methods that create a single whole-genome graph. Paragraph can accurately genotype single SVs that are not confounded by the presence of nearby SVs. The primary use case for Paragraph will be to allow investigators to genotype previously identified variants. This could be applied to known, medically relevant SVs in precision medicine initiatives or from a reference catalog.

งานวิจัยที่ 2

Newest methods for Detecting Structural Variations

https://www.cell.com/trends/biotechnology/fulltext/S0167-7799(19)30036-8

1) Background and Motivation of this research

เนื่องจากขีดจำกัดของเทคโนโลยีในสมัยก่อน เราสามารถทำได้แค่การหา Single Nucleotide Variants (SNVs) ซึ่งทำให้เราไม่สามารถหาความหลากหลายทางพันธุกรรมของมนุษย์ได้มาก แต่เมื่อเทคโนโลยีใหม่ๆ เกิดขึ้น เราสามารถพบ Structural Variations (SVs) ได้

SVs นั้นส่งผลต่อความหลากหลายทางพันธุกรรมของมนุษย์มากกว่า SNVs แม้ว่า SVs จะพบได้น้อยกว่า SNVs ก็ตาม

ตัวอย่างของเทคโนโลยีที่ใช้ในการตรวจจับ SVs ได้แก่ Strand-specific sequencing (Strand-seq), Linked-read sequencing เป็นต้น

SVs นั้นส่งผลต่อโรคทางระบบประสาทบางโรค เช่น Alzheimer's Disease, Frontotemporal Dementia เป็นต้น

2) Proposed methodology

Massive Parallel Short-Read Sequencing, Specialized Library Preparation Methods for Short Reads, Long-Read Sequencing เป็นวิธีการสำหรับการตรวจจับ SVs ได้ วิธีการทั้งหมดนี้มีพื้นฐานมาจาก Breakpoint genotyping, SV genotyping, Alignment reading to a reference genome, Tagmentation on beads

3) Data

https://github.com/mgharvey/mps-sim (For Massive Parallel Short-Read Sequencing)

4) Experimental Results

A recurrent theme is that known SVs underlying disease were only detected when the disrupted gene was already implicated with the disease or the locus was part of an unresolved linkage or association signal. In the past years, newer technologies have been used to better characterize known mutations, such as repeat expansions and the role of repeat motif interruptions. Since

screening for SVs is not commonly included in diagnostic testing, it is highly likely that genes linked to neurodegenerative diseases are harboring unknown SVs in coding regions or intronic or distant regulatory elements. Furthermore, no functional variant has yet been identified in multiple association loci. Hence, using a targeted gene panel for long-read sequencing would enable affordable characterization of all structural elements in loci linked or associated to neurodegenerative diseases.

5) Discussion

The development of enhanced methods for detecting SVs has resulted in an increased appreciation of the contribution of SVs to human genetic diversity and diseases. Technological innovations are the drivers to detect and highlight genetic variation that was absent in previous work. Detecting inversions has proved to be especially complex since the inversion breakpoints are generally located in repetitive regions, which are invisible when traditional methods are used. With the newest techniques such as long-read sequencing and especially strand-seq, many more inversions are detected. These methods will help to unravel the contribution of SVs to diseases because inversions potentially disrupt coding sequences or separate regulatory elements from the corresponding coding elements. Technological advances such as long- and linked-read sequencing allow the detection of more SVs, but they remain immature, and the field is lagging in adopting them.

งานวิจัยที่ 3

Mako: A Graph-based Pattern Growth Approach to Detect Complex Structural Variants

https://www.sciencedirect.com/science/article/pii/S1672022921001431

1) เป็นงานวิจัยเกี่ยวกับอะไร ทำไมถึงเป็นปัญหา (ให้สรุปมา) (Background and Motivation of this research)

Structural Variants (SV) เป็นที่พูดถึงอย่างมาก ในแง่ของการตรวจหาโรคทางพันธุกรรม ซึ่งปัจจุบันก็มีเทคโนโลยีที่ใช้ได้ผลดีมากมาย แต่ในหลายงานวิจัยที่เพิ่งค้นพบ แสดงให้เห็นว่าในบางกรณี การกลายพันธุ์ของยีนมีความซับซ้อนกว่านั้น เช่น มี breakpoints ที่ซับซ้อน, มีการกลายพันธุ์หลายรอบ โดยเรียกว่า Complex Structural Variants (CSV) ซึ่งไม่สามารถใช้โมเดล SV ธรรมดาตรวจสอบได้ หรือใช้ได้แต่ใช้ทรัพยากรเยอะมาก ปัจจุบันก็มีหลายงานวิจัย พยายามค้นหา CSV ในยีน แต่ก็ยังมีช่องโหว่อยู่มาก งานวิจัยนี้จึงเสนออัลกอริทึม Mako ซึ่งได้ผลดีกว่า

2) สิ่งที่งานวิจัยนี้นำเสนอ และวิธีการที่ใช้ (Proposed methodology)

Mako ระบุ CSV ด้วยสองขั้นตอนหลัก ขั้นที่หนึ่ง signal graph creation คือการนำข้อมูล sequence ที่แปลกไปจากปกติ มาสร้างเป็น graph ขั้นที่สอง ใช้ graph ที่สร้างมาใช้เทคนิคต่างๆ เช่น back tracking, maximal subgraph, minimum distance ฯลฯ มาใช้เรียงลำดับและตรวจสอบจุดผิดปกติ

3) ข้อมูลที่ใช้ (Data)

https://github.com/xjtu-omics/Mako free for non-commercial use by academic, government, and non-profit/not-for-profit institutions. A commercial version of the software is available and licensed through Xi'an Jiaotong University.

4) ผลการทดลอง (Experimental results)

Mako achieved better performance on randomized events, which included more subcomponents than the reported ones. Indeed, by comparing reported and randomized simulations, the breakpoint detection sensitivity of Mako for randomized simulation increased, while that of other algorithms dropped except for GRIDSS. Although the assembly-based method, GRIDSS, is as effective as Mako for breakpoint detection, it lacks a proper procedure to resolve the connections among breakpoints.

5) อภิปรายผลของงานวิจัยนี้ (Discussion)

Mako, utilizing the graph-based pattern growth approach, for CSV discovery with 70% accuracy and 20 bp median breakpoint shift. To the best of our knowledge, Mako is the first algorithm that utilizes the bottom-up guided model-free strategy for SV discovery, avoiding the complicated model and match procedures. Given the fact that CSVs are largely unexplored, Mako presents opportunities to broaden our knowledge of genome evolution and disease progression.

งานวิจัยที่ 4

svBreak: A New Approach for the Detection of Structural Variant Breakpoints Based on Convolutional Neural Network

https://doi.org/10.1155/2022/7196040

1) Background and Motivation of this research

Structural variation (SV) เป็นเรื่องที่สำคัญของความหลากหลายทางจีโนม โดยที่การวิเคราะห์ SV นั้นกลายมาเป็นก้าวสำคัญในการสำรวจกลไกและการรักษาโรคของมนุษย์ มีปัญหาสำคัญคือการตรวจจับจุด SV breakpoints อย่างแม่นยำ โดยใช้ next generation sequencing data แต่เนื่องการเกิดขึ้นซ้ำของหลายSV ในจีโนมของมนุษย์ ความซับซ้อนของSV ทำให้เป็นปัญหาที่ยาก ดังนั้นในงานวิจัยนี้จึงได้เสนอการใช้ Convolutional neural network (CNN) มาใช้แก้ปัญหา โดยเรียกวิธีนี้ว่า svBreak

2) Proposed methodology

การทำงานของ svBreak คือนำเอาเซตของ SV-related features จากแต่ละฐานข้อมูลจีโนมจากการอ่านแบบ sequencing มาเรียงกับจีโนมอ้างอิงแล้วสร้างเมทริกซ์ของข้อมูลโดยที่แต่ละแถวแสดงถึงฐานข้อมูล และแต่ละหลักแสดงถึง feature breakpoint แล้วเอาCNN model มาทำนาย SV breakpoint

3) Data

Data is available at https://ega-archive.org/

https://github.com/BDanalysis/svBreak

4) Experimental Results

In Simulation Studies, we set the coverage depth to 10x ,20x ,30x and 40x to generate variety of simulation data. We used F1 score as a harmonic mean of sensitivity and precision. Among the three methods, svBreak has achieved a relatively higher F1 score more than the other two methods and performs best when the coverage depth is 30x and 40x. We also explore the performance of svBreak method in detecting 7 types of the SV breakpoints. It can be observed that, the breakpoints of deletion, inversion, inverted duplication and tandem duplication are detected at larger sensitivity than the other three types.

5) Discussion

This paper combines convolutional neural networks with bioinformatics and uses convolutional neural networks to classify SV breakpoints. The central characteristic of our proposed method is that it extracts twelve SV-related features for each genome site from the sequencing reads aligned to the reference genome and adopts a CNN model for SV breakpoint prediction. In order to further improve the performance of the convolutional neural network, this paper adds a large number of labels to the training set. svBreak is able to detect and discriminate seven common SV breakpoints and is tested using simulation and real sequencing data. The experimental results demonstrate that svBreak is a valid and useful method. Thus, svBreak can be expected to be a supplementary approach in the field of SV analysis in human genomes.

Progress Report #2

นำเสนอวิธีการแก้ไขปัญหา และเปรียบเทียบกับงานวิจัยข้างต้นว่าต่างกันอย่างไรบ้าง

Your proposed method and comparison between previous related work written above.

ปัญหา: How to detect structural variations from aligned reads

Input: Reference genome และ Sequence alignment Reads1 และ Reads2

Output: ตำแหน่งและชนิดของ SV ที่เกิดขึ้น

Input

Reads1

```
>read_001
ACGCACTGGCTA
>read_002
CCACCCCGACGC
>read_003
CGCACTGGCTAC
>read_004
CCCCTCGCCAGC
>read_005
TCGCCAGCCGAT
>read_006
GATTCTGCCACC
…
```

Reads2

```
>read_001
GCCGATTCTGCC
>read_002
CGGGCGATGGCC
>read_003
CCGATTCTGCCA
>read_004
ACTGGCTACGGA
>read_005
GCCGGGAGCCGT
>read_006
CCTATCCCCTCG
…
```

Reference

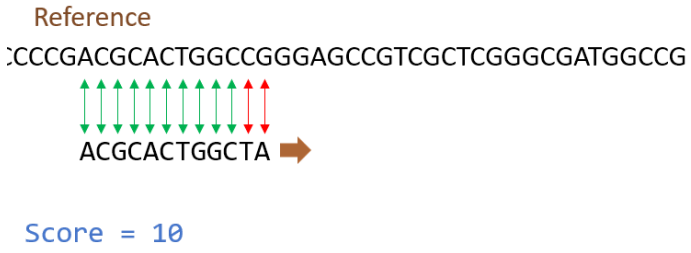
```
>reference_genome
ATAGTGCCGTGGCCGCGGGACGGTACGGAGCCTATC
CCCTCGCCAGCCGATTCTGCCACCCCGACGCACTGG
CCGGGAGCCGTCGCTCGGGCGATGGCCG
```

ขั้นตอนในการแก้ไขปัญหา:

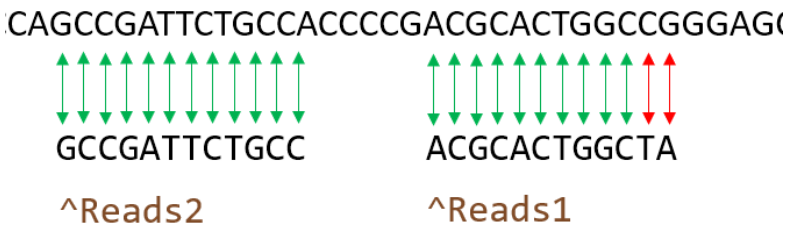
1. นำ read แต่ละอันในไฟล์ Reads1 มาเปรียบเทียบกับ Reference genome โดยเปรียบเทียบในทุกๆ ตำแหน่ง รวมถึงลอง reverse read นั้น และคิดคะแนนตามจำนวนที่มีเบสที่ติดกัน และตรงกัน Reference genome มากที่สุด

Score = 4

2. นำ read นั้นๆ มาวางบนตำแหน่งที่มีคะแนนสูงที่สุด



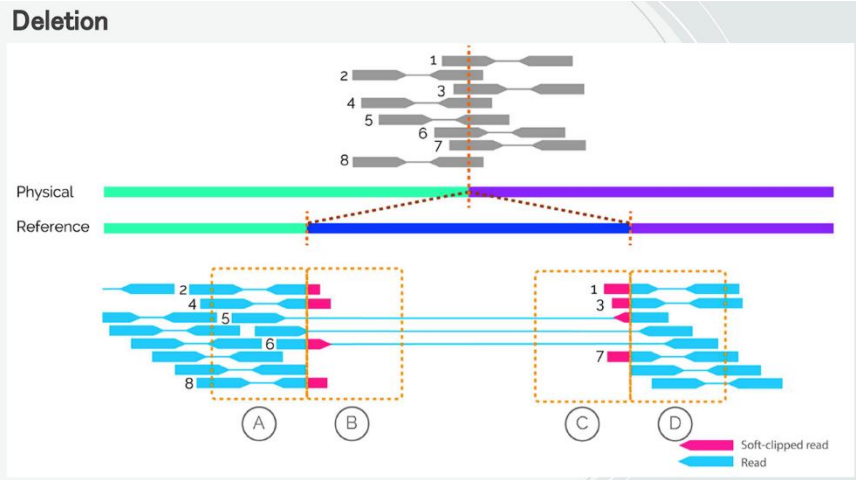
3. ทำเช่นเดียวกันกับ read ในไฟล์ Reads2 ใน Reads1 และ Reads2



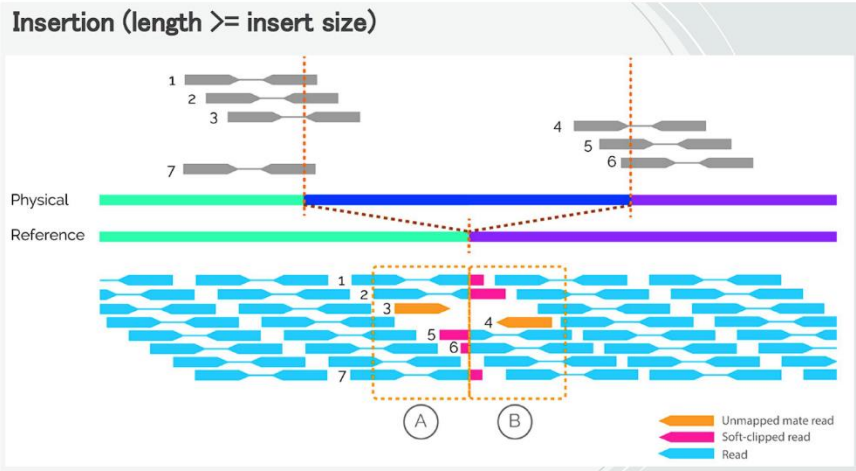
4. ทำเช่นนี้จนครบทุก read

```
GCCGTGGCCGCGGGACGGTACGGAGCCTATCCCCTCGCCAGCCGATTCTGCCACCCCGACGCACTGGCCGGGAGCCGTCGCTCGGGCGATGGCCG
                                    GCCGATTCTGCC      ACGCACTGGCTA
                                              CCACCCCGACGC                    CGGGCGATGGCC
                                     CCGATTCTGCCA      CGCACTGGCTAC
                              CCCCTCGCCAGC                  ACTGGCTACGGA
                                  TCGCCAGCCGAT                   GCCGGGAGCCGT
                          CCTATCCCCTCG      GATTCTGCCACC
                            TATCCCCTCGCC                    CGCACTGGCTAC
                       CGGAGCCTATCC                   ACCCCGACGCAC
                      ACGGAGCCTATC                    CACCCCGACGCA
                                 CCCCTCGCCAGC      TGCCACCCCGAC
                                    CTCGCCAGCCGA                  GGCCGGGAGCCG
                                              TGCCACCCCGAC                   CTCGGGCGATGG
                                               GCCACCCCGACG                   TCGGGCGATGGC
                                         ATTCTGCCACCC                   GTCGCTCGGGCG
                                    CGCCAGCCGATT      CCCCGACGCACT
                               TCCCCTCGCCAG                  CACTGGCCGGGA
                      ACGGAGCCTATC                   CACCCCGACGCA
                            CCTATCCCCTCG                    GACGCACTGGCC
    GGCCGCGGGACG                    CCAGCCGATTCT
                                     CAGCCGATTCTG                   GGAGCCGTCGCT
```

5. เปรียบเทียบว่าเป็น Structural variant ชนิดใด และตำแหน่งใดดังนี้

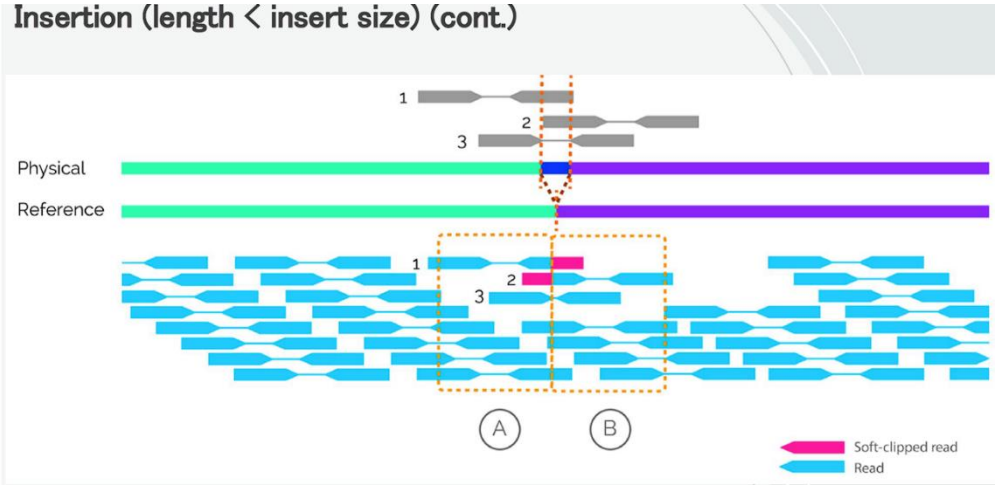


- **Deletion:** มีบริเวณ B และ C ที่ read ไม่ตรงกับ reference genome เหมือนๆ กัน หาโดยการไล่ดูทุกๆ read และจำตำแหน่งที่ read แต่ละอันเปลี่ยนจากเหมือน reference เป็น ไม่เหมือน (ฟ้า->แดง) และเปลี่ยนจากไม่เหมือน ref เป็นเหมือน (แดง->ฟ้า) จะพบว่าทุก read จะเปลี่ยนจากเหมือน reference เป็น ไม่เหมือน (ฟ้า->แดง) ในตำแหน่งเดียวกัน (ระหว่าง A, B) และเปลี่ยนจากไม่เหมือน reference เป็นเหมือน (แดง->ฟ้า) ในตำแหน่งเดียวกัน (ระหว่าง C, D) โดยที่ A, B อยู่ด้านซ้ายของ C, D

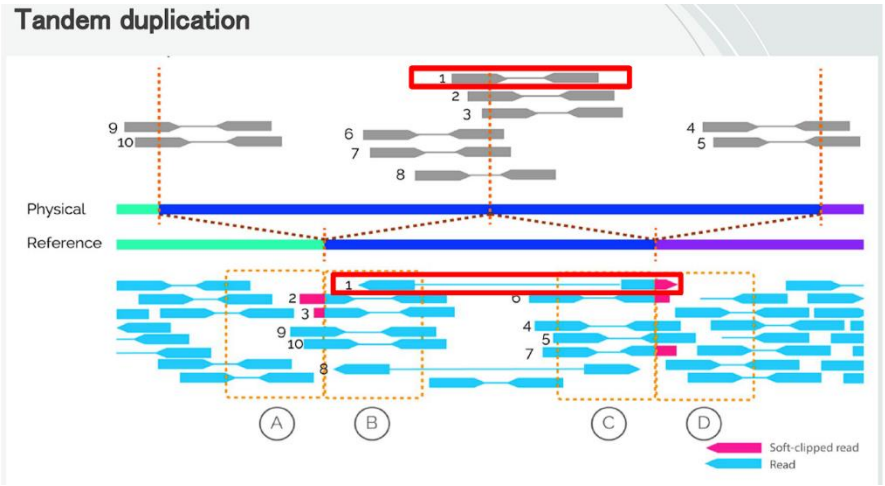


- **Insertion (length >= insert size):** มีบริเวณ A และ B ที่ read ตรงกับ reference genome บางส่วน โดย read อาจตรงกับ reference genome ในบริเวณ A และไม่ตรงเมื่ออยู่ในบริเวณ B หรืออาจมี read ที่ไม่ตรงกับ reference genome เลย (unmapped mate read) หาโดยการไล่ดูทุกๆ read และจำตำแหน่งที่ read แต่ละอันเปลี่ยนจากเหมือน reference เป็น ไม่เหมือน (ฟ้า->แดง) และเปลี่ยนจากไม่เหมือน reference เป็นเหมือน (แดง->ฟ้า) และหาว่ามี read ที่ unmapped mate read

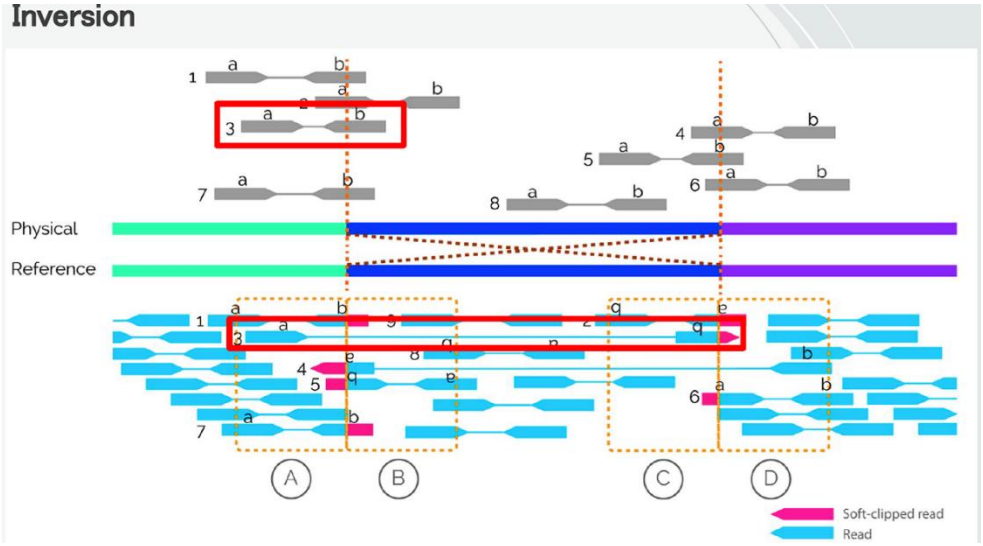
หรือไม่ (ส้ม) จะพบว่าทุก read จะเปลี่ยนจากเหมือน reference เป็น ไม่เหมือน (ฟ้า->แดง) และเปลี่ยนจากไม่เหมือน reference เป็นเหมือน (แดง->ฟ้า) ในตำแหน่งเดียวกัน (ระหว่าง A, B)



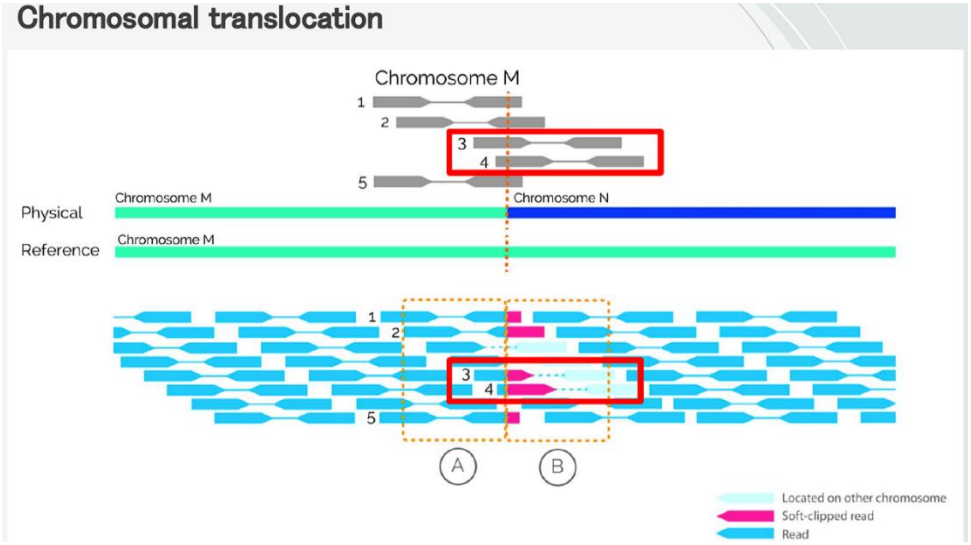
- **Insertion (length < insert size):** เหมือนกับกรณีด้านบนแต่จะไม่พบ unmapped mate read



- **Tandem duplication:** มี read ที่ตรงกับ reference genome เมื่อ inverse และมีบริเวณ A และ D ที่ read บางส่วนไม่ตรงกับ reference genome หาโดยการไล่ดูทุกๆ read และจำตำแหน่งที่ read แต่ละอันเปลี่ยนจากเหมือน reference เป็น ไม่เหมือน (ฟ้า->แดง) และเปลี่ยนจากไม่เหมือน ref เป็นเหมือน (แดง->ฟ้า) จะพบว่าทุก read จะเปลี่ยนจากเหมือน reference เป็น ไม่เหมือน (ฟ้า->แดง) ในตำแหน่งเดียวกัน (ระหว่าง C, D) และเปลี่ยนจากไม่เหมือน reference เป็นเหมือน (แดง->ฟ้า) ในตำแหน่งเดียวกัน (ระหว่าง A, B) โดย A, B อยู่ด้านซ้ายของ C, D และอาจพบ reverse read



- **Inversion:** มี read จำนวนมากที่ตรงกับ reference genome เมื่อ inverse ในช่วง B ถึง C และมีบริเวณ A และ D ที่ read บางส่วนไม่ตรงกับ reference genome หาโดยการไล่ดูทุกๆ read และจำตำแหน่ง reverse read จะพบว่าจะมีบริเวณหนึ่ง (B, C) ที่ทุก read จะ reverse



- **Chromosomal translocation:** read อ่านได้ตรงกับ reference ถึงแค่บริเวณ A และจะไม่ตรงในบริเวณ B หาโดยการไล่ดูทุกๆ read และจำตำแหน่งที่ read แต่ละอันเปลี่ยนจากเหมือน reference เป็นไม่เหมือน (ฟ้า->แดง) จะพบว่าทุก read จะเปลี่ยนในตำแหน่งเดียวกัน (ระหว่าง A, B)

ข้อจำกัด:

1. หาก read มี single nucleotide variation เกิดขึ้น อาจได้ output ผิด
2. ไม่สามารถตรวจจับกรณีมีหลาย structural variations ได้
3. ประสิทธิภาพการทำงาน = O(l*m*k)

    l = ความยาว Reference genome

    m = ความยาว read

    k = จำนวน read

เปรียบเทียบกับงานวิจัยที่ 1

Paragraph: a graph-based structural variant genotyper for short-read sequence data

- Paragraph is an accurate SV genotyper for short-read sequencing data that scales to hundreds or thousands of samples.
- Paragraph implements a unified genotyper that works for both insertions and deletions.
- Paragraph can generally tolerate breakpoint deviation of up to 10 bp in most genomic contexts.
- Paragraph also scales well to population-level studies where large sets of variants identified from different resources can be genotyped rapidly (e.g., 1000 SVs can be genotyped in 1 sample in 15 min with a single thread).
- Paragraph can accurately genotype single SVs that are not confounded by the presence of nearby SVs.
- The primary use case for Paragraph will be to allow investigators to genotype previously identified variants with high accuracy.

เปรียบเทียบกับงานวิจัยที่ 2

Newest methods for Detecting Structural Variations

• SVs underlying disease were only detected when the disrupted gene was already implicated with the disease or the locus was part of an unresolved linkage or association signal.

• Newer technologies have been used to better characterize known mutations, such as repeat expansions and the role of repeat motif interruptions.

• It is highly likely that genes linked to neurodegenerative diseases are harboring unknown SVs in coding regions or intronic or distant regulatory elements, since screening for SVs is not commonly included in diagnostic testing. Thus, using a targeted gene panel for long-read sequencing would enable affordable characterization of all structural elements in loci linked or associated to neurodegenerative diseases.

• The development of enhanced methods for detecting SVs has resulted in an increased appreciation of the contribution of SVs to human genetic diversity and diseases.

• Detecting inversions has proved to be especially complex since the inversion breakpoints are generally located in repetitive regions, which are invisible when traditional methods are used.

• Long-read sequencing and strand-seq able to detect more inversions which help to unravel the contribution of SVs to diseases because inversions potentially disrupt coding sequences or separate regulatory elements from the corresponding coding elements. However, they are still immature and medical field is lagging in adopting them.

เปรียบเทียบกับงานวิจัยที่ 3

Mako: A Graph-based Pattern Growth Approach to Detect Complex Structural Variants

- Mako's mainly focus on Complex Structural Variant (CSV) which need more complex and well-designed algorithm
- Mako, utilizing the graph-based pattern growth approach, for CSV discovery with 70% accuracy and 20 bp median breakpoint shift
- Mako is the first algorithm that utilizes the bottom-up guided model-free strategy for SV discovery, avoiding the complicated model and match procedures
- Mako detects CSVs with NGS data in two major steps, i.e., signal graph creation and subgraph detection
- Mako achieved better performance on randomized event

เปรียบเทียบกับงานวิจัยที่ 4

svBreak: A New Approach for the Detection of Structural Variant Breakpoints Based on Convolutional Neural Network

- Due to the cooccurrence of multiple types of SVs in the human genome and the intrinsic complexity of SVs, the discrimination of SV breakpoint types is a challenging task.
- Accurate detection of SVs could provide variation information for the exploration of mechanisms and precision diagnosis of cancers.
- This approach is established based on a convolutional neural network (CNN).
- svBreak is able to detect and discriminate seven common SV breakpoints and is tested using simulation and real sequencing data.
- The reason for the high accuracy of the svBreak algorithm in this paper is that its feature values are mainly obtained from the comparison information of split reads and the insertion distance of paired-end sequencing reads.

Progress Report #3

ตอบคำถามจาก comments progress #2

1. **มีแผนใช้ข้อมูลใดในการทดสอบประสิทธิภาพตัว algo ที่ใช้ในการ detect structural variant**

คำตอบ: ใช้ข้อมูลที่สร้างขึ้นมาจากโค้ด testcase_gen.ipynb โดยเป็นการสร้างตัว reads และ reference genome

2. **ใช้ข้อมูลเข้าเป็นไฟล์ประเภทไหน**

คำตอบ: ใช้ข้อมูลเข้าที่เป็นไฟล์ read.txt แสดงข้อมูล reads1 และ reads2 และ ref.txt แสดงข้อมูล reference genome

3. **ถ้าอนุญาตให้เกิด mismatches ในส่วนของการเทียบ read จะปรับปรุง algo ที่ใช้อย่างไร**

คำตอบ: ปรับ algo โดยเปลี่ยนวิธีคิด score ตามจำนวน base pair ของ read และ ref ที่ตรงกัน และเลือกวาง read ในตำแหน่งที่ score สูงสุด

4. **ถ้ามีไฟล์ข้อมูล 450 ล้าน reads ใน Reads1 และ อีก 450 ล้าน reads ใน Reads2 จะใช้ computational resources: RAM, CPU, time ประมาณเท่าไหร่**

คำตอบ: จะใช้ computational resource เป็น O(l*m*k) เมื่อ

l = ความยาว Reference genome

m = ความยาว read

k = จำนวนข้อมูล read

หากใช้ RAM 32GB, CPU 4GHz จะได้ time โดยประมาณ = 75,000 ปี

โค้ดวิธีการแก้ไขปัญหา

ไฟล์โค้ดสำหรับแก้ไขปัญหา: SVD.py

Input:

Reference genome ใน 1 ไฟล์ .txt และ Reads1 และ Reads2 ใน 1 ไฟล์ .txt

เช่น

ref.txt

```
TGAATATACAACGGGGCTGTCTATTACAATTACGAAGCCGACTGTATCGTCCAATTGAGCAGTTTAGTCTC
TTAAATCGGTAA
```

read.txt ประกอบด้วย reads1 และ reads2 ขั้นด้วย ,

```
TGAATATACAAC,ATTACAATTACG
AATATACAACGG,TACAATTACGAA
TATACAACGGGG,CAATTACGAAGC
TACAACGGGGCT,ATTACGAAGCCG
CAACGGGGCTGT,TACGAAGCCGAC
```

Output: ชนิดและตำแหน่งของ SV ที่เกิดขึ้น

เช่น

```
Deletion from index 78 to 107
```

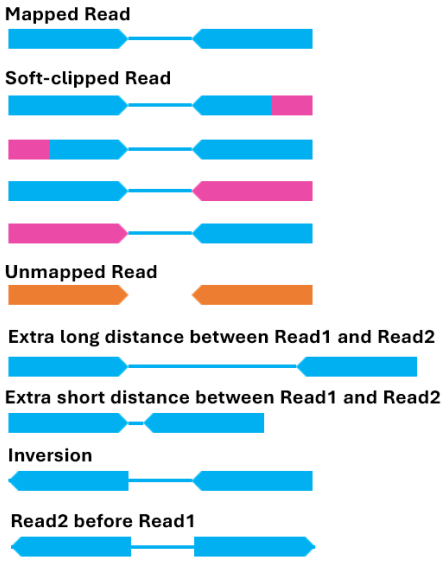
วิธีการแก้ไขปัญหาที่ได้ปรับปรุง

ปรับปรุงให้สามารถเกิด mismatches ในขั้นตอน Read Mapping ได้

1. Read Mapping ด้วยการนำ read แต่ละอันในไฟล์ read.txt มาเปรียบเทียบกับ Reference genome ในไฟล์ ref.txt เพื่อหาตำแหน่งที่เหมือนกันที่สุด โดยนำ read1, inverse read1, read2, inverse read2 มาเปรียบเทียบในทุกๆ ตำแหน่งของ Reference genome และคิดคะแนน (score) ตามจำนวน base pair ที่ read นั้นตรงกันกับ Reference genome

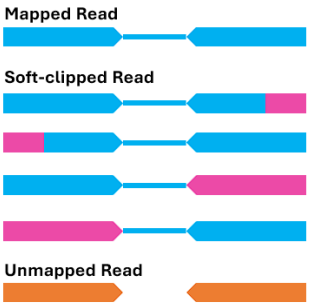
   เช่น

2. จำตำแหน่งที่มีคะแนนมากที่สุดของ read1 และจำว่า inverse แล้วได้ผลลัพธ์ที่ดีกว่าหรือไม่ โดยเลือกเปรียบเทียบระหว่างคะแนนมากที่สุดของ read1 และ inverse read1 จากนั้นทำเช่นเดียวกันใน read2
3. หาว่า read นั้นเมื่อเทียบกับ ref ในตำแหน่งที่เหมือนกันที่สุดแล้ว มีลักษณะเหมือนกันในรูปแบบใดดังนี้



   และหากเป็น soft-clipped read จะจำตำแหน่ง breakpoint ที่เปลี่ยนจากการมี base pair ตรงกับ ref เป็นไม่ตรง

4. นำข้อมูลที่ได้มาตัดสินว่าเป็น Structural variant ชนิดใด และตำแหน่งใด โดยพิจารณาจากรูปแบบที่พบได้ใน Structural variant แต่ละชนิด และตำแหน่ง breakpoint

	Deletion	Insertion (length >= insert size)	Insertion (length < insert size)	Inversion	Tandem duplication	Chromosomal translocation
	✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✓ ✓ ✓ ✓ *Soft-clipped จะอยู่ฝั่งเดียวกันเสมอ
	✓		✓			
				✓	✓	

แสดงผลการทำงานจากวิธีที่นำเสนอ

โดยใช้ไฟล์สำหรับสร้าง testcase เพื่อทดสอบประสิทธิภาพ: testcase_gen.ipynb

CPU: Intel i5-11400f

RAM: Corsair Vengeance 16GB 2400Hz

Deletion

	Reference Length	Reads Length	Reads Distance	Reads Count	Mutate Rate (%)	Execute Time (s)	Peak Memory (MB)	Result	Real Index
1	150	15	15	29	1	0.041	0.3031	Chromosomal translocation from index 70	60-86
2	500	15	15	148	1	0.766	0.1984	Deletion from index 292 to 500	292-315
3	1000	30	30	289	1	4.02	0.2647	Deletion from index 816 to 866	816-863
4	5000	50	50	802	1	75.94	2.9222	Deletion from index 469 to 526	469-527
5	5000	50	50	792	3	74.10	0.6299	Deletion from index 649 to 769	689-769

Insertion (length >= insert size)

	Reference Length	Reads Length	Reads Distance	Reads Count	Mutate Rate (%)	Execute Time (s)	Peak Memory (MB)	Result	Real
1	150	15	15	68	1	0.095	0.1735	Insertion (length >= insert size) between index 39 and 40	40
2	500	15	15	239	1	1.26	0.2291	Insertion (length >= insert size) between index 402 and 404	403
3	1000	30	30	205	1	2.89	0.3748	Unknown SV	605
4	5000	50	50	659	1	60.52	0.6079	Insertion (length >= insert size) between index 4632 and 463	4633
5	5000	50	50	832	3	76.34	0.6925	Unknown SV	4515

Insertion (length < insert size)

	Reference Length	Reads Length	Reads Distance	Reads Count	Mutate Rate (%)	Execute Time (s)	Peak Memory (MB)	Result	Real
1	150	15	15	70	1	0.099	0.315	Insertion (length >= insert size) between index 49 and 50	50
2	500	15	15	233	1	1.204	0.368	Insertion (length >= insert size) between index 412 and 413	413
3	1000	30	30	224	1	3.098	0.382	Insertion (length >= insert size) between index 856 and 857	857
4	5000	50	50	656	1	61.05	0.607	Insertion (length >= insert size) between index 2499 and 2501	2500
5	5000	50	50	654	3	60.34	0.639	Unknown SV	2239

Tandem Duplication

	Reference Length	Reads Length	Reads Distance	Reads Count	Mutate Rate (%)	Execute Time (s)	Peak Memory (MB)	Result	Real
1	150	15	15	63	1	0.086	0.314	Unknown SV	73
2	500	15	15	250	1	1.32	0.373	Unknown SV	342
3	1000	30	30	213	1	3.001	0.096	Unknown SV	493
4	5000	50	50	653	1	60.19	0.605	Unknown SV	4292
5	5000	50	50	648	3	59.89	0.603	Unknown SV	1729

Inversion

	Reference Length	Reads Length	Reads Distance	Reads Count	Mutate Rate (%)	Execute Time (s)	Peak Memory (MB)	Result	Real
1	150	15	15	46	1	0.068	0.309	Inversion from index 36 to 58	36-58
2	500	15	15	227	1	1.197	0.225	Inversion from index 285 to 288	284-304
3	1000	30	30	200	1	2.79	0.373	Inversion from index 347 to 402	347-398
4	5000	50	50	641	1	59.66	0.6	Inversion from index 3499 to 3592	3499-3592
5	5000	50	50	644	3	58.91	0.494	Inversion from index 1911 to 1969	1911-1969

Translocation

	Reference Length	Reads Length	Reads Distance	Reads Count	Mutate Rate (%)	Execute Time (s)	Peak Memory (MB)	Result	Real
1	150	15	15	57	1	0.085	0.17	Chromosomal translocation from index 96	96
2	500	15	15	226	1	1.19	0.364	Chromosomal translocation from index 432	432
3	1000	30	30	204	1	2.87	0.092	Chromosomal translocation from index 882	882
4	5000	50	50	649	1	60.24	0.462	Chromosomal translocation from index 4771	4771
5	5000	50	50	643	3	60.29	0.601	Chromosomal translocation from index 4760	4760

Final Report

แสดงผลการทำงานเพิ่มเติม ปรับปรุง จาก Progress Report #3

- ปรับปรุงขั้นตอนการ Read mapping โดยแทนที่จะนำ read มาเปรียบเทียบในทุกๆ ตำแหน่งของ reference genome เปลี่ยนเป็นเปรียบเทียบจนได้ score >= 80% ของคะแนนเต็ม และถือว่าตำแหน่งนั้นเป็นตำแหน่งที่ถูกต้อง เพื่อให้การทำงานเร็วขึ้น
- ปรับปรุงขั้นตอนการตัดสินใจรูปแบบของ structural variant และตำแหน่งที่เกิดให้มีความถูกต้องแม่นยำมากขึ้น
- ปรับปรุงขั้นตอนการสร้าง test case และการวัดความถูกต้อง คิดคะแนนด้วยการให้คะแนนเต็มที่ 3 คะแนน โดยมาจาก 1.ความถูกต้องของชนิด variance 2.ความถูกต้องของ index จุดเริ่มต้นของ variance 3.ความถูกต้องของ index จุดสิ้นสุดของ variance

Final results

ผลการทดลอง

ทดลองโดยใช้ไฟล์ main.ipynb ที่ค่าของตัวแปรดังนี้

Reference genome size: 500, 600, 700, 800, 900, 1000, 1100, 1200 ,1300, 1400, 1500, 1600, 1700, 1800, 1900 bp

Read size (เปอร์เซ็นของ reference size): 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5

ทำซ้ำ 5 ครั้ง

ตำแหน่งคลาดเคลื่อนได้สูงสุด 3 ตำแหน่ง

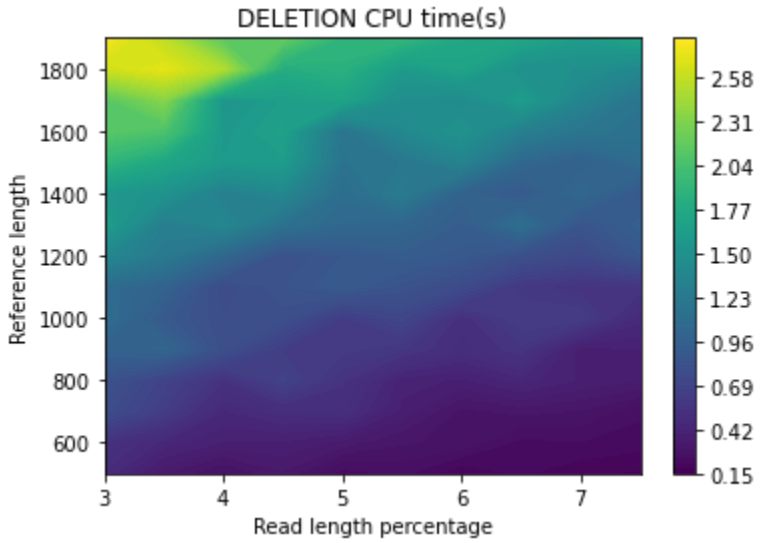
CPU: Intel i5-11400f

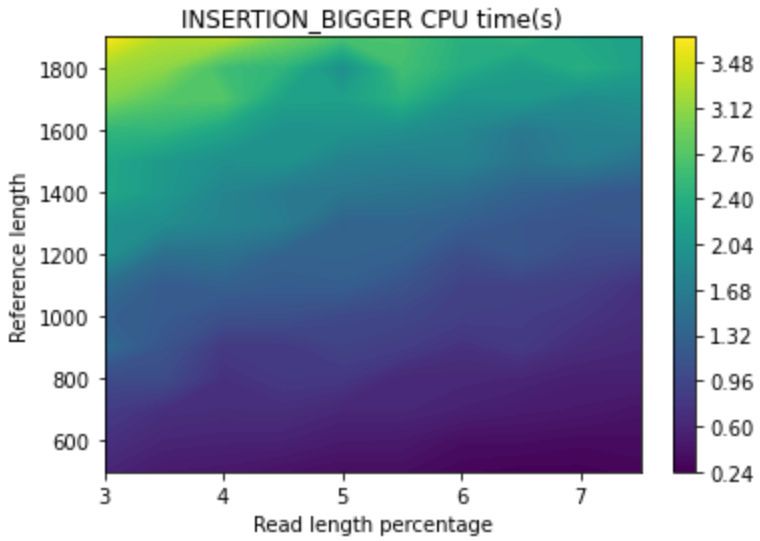
RAM: Corsair Vengeance 16GB 2400Hz

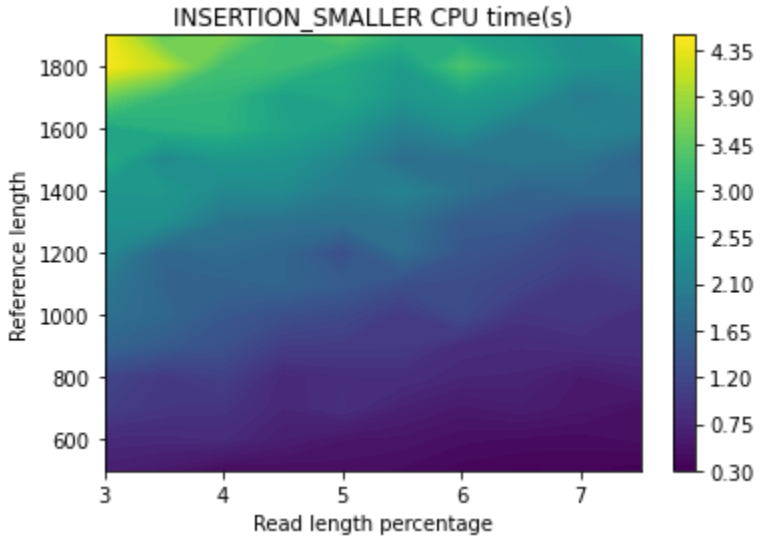
ความแม่นยำ

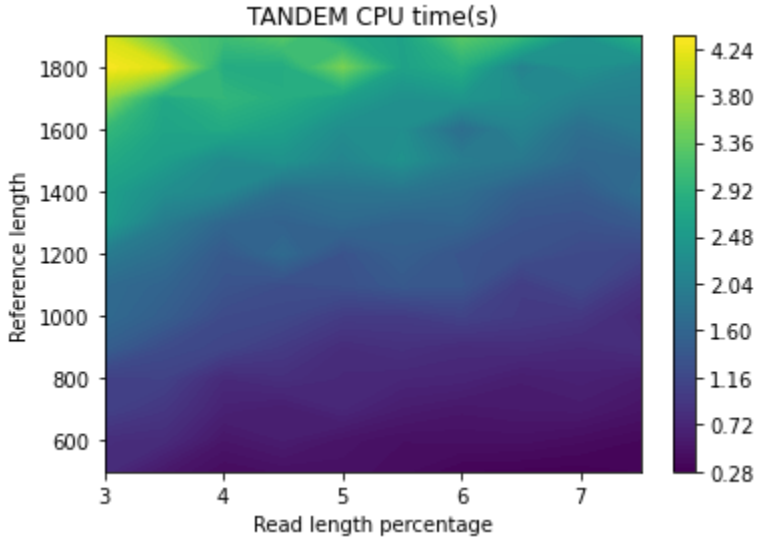
Variance	Overall accuracy
Deletion	68.62%
Insertion length < insert size	70.13%
Insertion length >= insert size	66.09%
Tandem duplication	62.58%
Inversion	54.76%
Translocation	77.07%

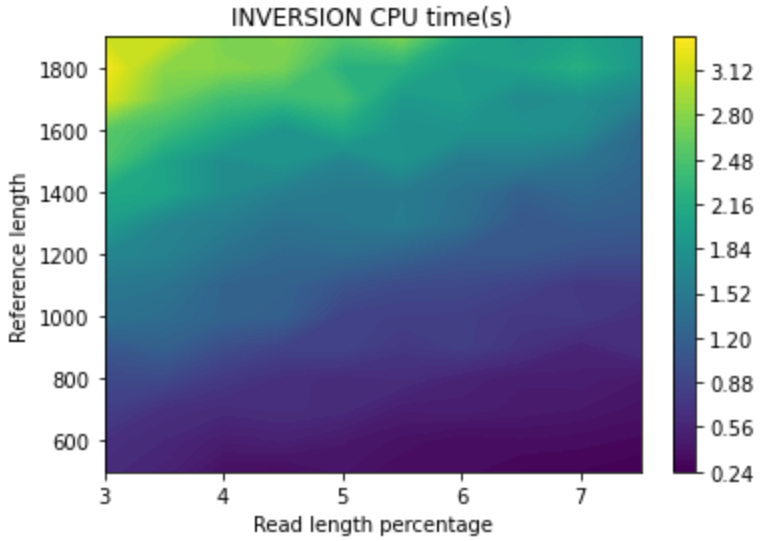
ระยะเวลาที่ใช้ในการคำนวณ

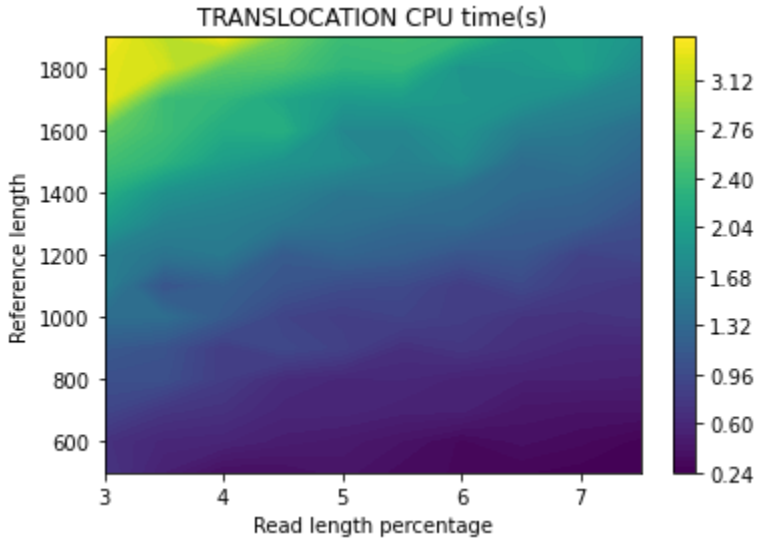



INSERTION_BIGGER CPU time(s)
Reference length
1800
1600
1400
1200
1000
800
600
3
4
5
6
7
Read length percentage
3.48
3.12
2.76
2.40
2.04
1.68
1.32
0.96
0.60
0.24


INSERTION_SMALLER CPU time(s)
Reference length
1800
1600
1400
1200
1000
800
600
3
4
5
6
7
Read length percentage
4.35
3.90
3.45
3.00
2.55
2.10
1.65
1.20
0.75
0.30


TANDEM CPU time(s)
Reference length
1800
1600
1400
1200
1000
800
600
3
4
5
6
7
Read length percentage
4.24
3.80
3.36
2.92
2.48
2.04
1.60
1.16
0.72
0.28





อภิปราย และสรุปผล (Discussion and Summary)

จากการทดลองได้มีการแก้ไขในการทำงานของ read mapping ด้วยการลดเวลาลงในการเทียบกับ reference genome ด้วยการเอาเปอร์เซนต์ที่ตรงกันสูง และได้มีการทดสอบผลของความถูกต้องกับตัวอย่าง test case ซึ่งได้ผลดีสุดกับกรณี translocation ในขณะที่ความเร็วในการรันนั้นในกรณีที่มีความยาว reference อยู่ที่ 1800 จะมีค่า CPU Time อยู่ที่ประมาณ 3 - 4 วินาที

เอกสารอ้างอิง (References)

- สไลด์วิชา Bioinformatic I Week 4: SNV & SV

ตอบคำถามเพิ่มเติมจากการนำเสนอ

หากมีหลาย SV ใน physical genome จะทำอย่างไร

- แก้ไข algorithm ใหม่โดยนำ index ที่เกิด soft-clipped, read1-read2 ระยะห่างยาวเป็นพิเศษ, สั้นเป็นพิเศษ, inversion, read2 มาก่อน read1 มาพิจารณาว่าเกิดในตำแหน่งใดบ้างเพื่อจัดเป็นกลุ่มๆ เป็นตำแหน่งการเกิด SV เพราะ index เหล่านี้บ่งบอกว่ามี SV เกิดขึ้น
- หลังจากนั้นใช้วิธีการเดิมในการหาว่าเกิด SV ชนิดใด โดยพิจารณาไปทีละกลุ่มจนครบตามจำนวน SV ที่มี