2019 JAX Long Read Sequencing Bioinformatics Workshop

Pacific Biosciences SMRT Sequencing Exercise: Working with PacBio Tools on the Command Line and Variant Analysis of HiFi (CCS) Data

In this exercise we will guide you through an example analysis of Sequel II HiFi reads. We aim to demonstrate the utility of HiFi data in the context of calling variants on human chromosome 19 from the hg38 assembly.

Input Data

- Reference sequence: human chromosome 19 from GRCh38.p13
  Summary:
  - 58,617,617 bp (including gaps)
  - 150,865 annotated features in NCBI
  - 9 gaps in primary assembly (60 kb; 2 x 50 kb; 10 kb; 6.5 kb; 4 x 100 bp)
- Experimental sequence data: GIAB HG002 Sequel II HiFi Reads, subsampled for those aligning to chromosome 19
  Summary:
  - 66,603 mapped HiFi reads
  - 738 Mb mapped bases (~13-fold coverage)
  - 11.2 kb mean mapped read length

Analysis Pipeline

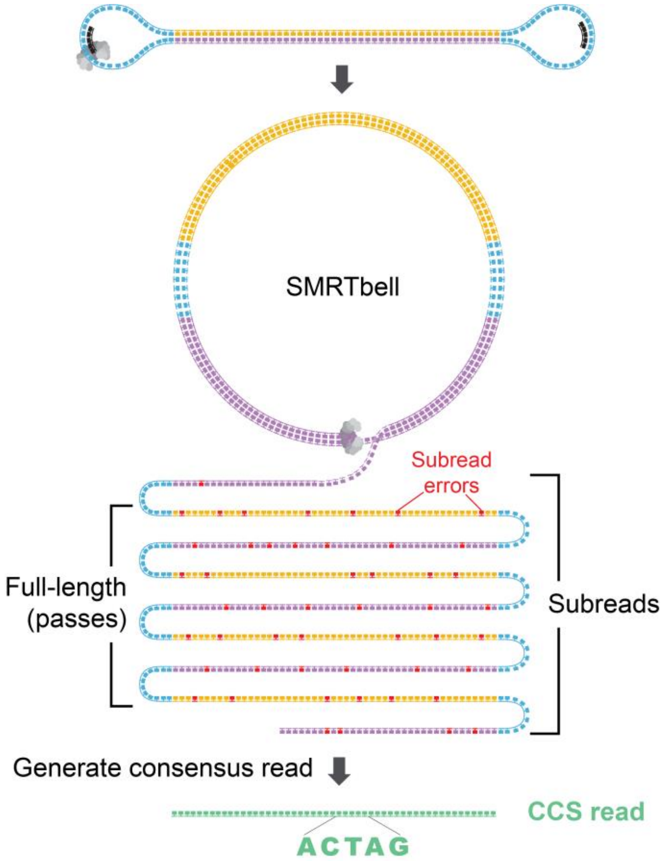
1. Index reference and align reads with pbmm2
2. Assess mapped concordance and mapping quality
3. Quality analysis and control
4. Call SNVs and indels with GATK4 and SVs with pbsv
5. Merge, assess, and filter VCF and GFF files
6. Phase variants into haplotypes with WhatsHap
7. Compile list of interesting genes to analyze
8. Analyze and visualize with IGV

Required Software

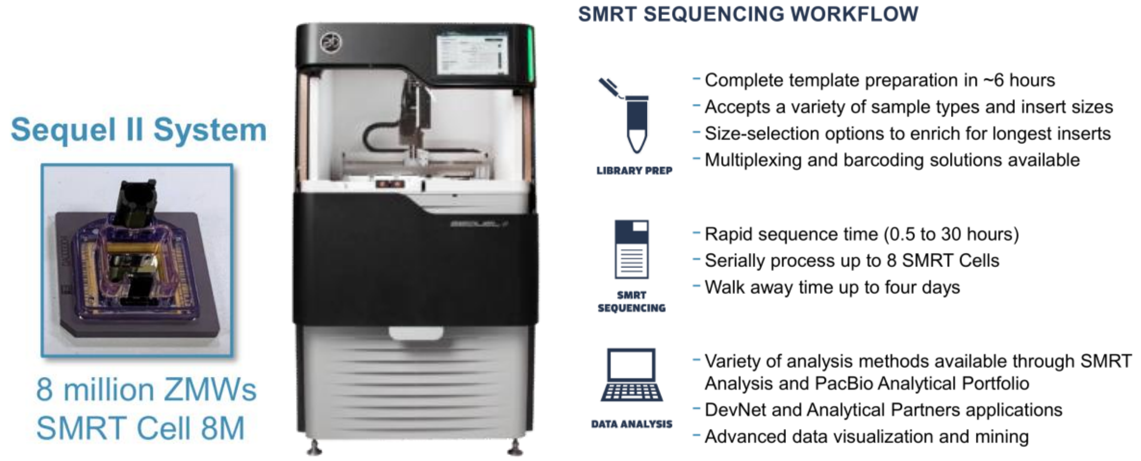
- samtools
- bedtools
- vcftools
- deepTools
- pyGenomeTracks
- pbmm2
- pbsv
- GATK4
- WhatsHap
- IGV

Introduction

Circular consensus sequencing (CCS) is a proprietary method for error-correction of DNA sequencing reads which uses all subreads generated from multiple passes over the same original DNA template to produce a consensus. This method enables highly accurate reads of single DNA molecules up to 15 kb in length. When the average Phred quality score of a CCS read is >Q20 (99% accurate), the read is called High Fidelity (HiFi). PacBio HiFi reads are ideally suited for many applications, including long amplicon analysis, variant detection and phasing, shotgun metagenomics, and more.



We have recently demonstrated that use of HiFi (CCS) reads improves variant detection in a reference sample, the son from an Ashkenazim trio whose DNA was extracted from B-lymphocytes and extensively profiled by the NIST Genome In A Bottle (GIAB) consortium. We have generated an HG002 dataset from HMW DNA and sequenced on the new Sequel II sequencing instrument, which supports SMRT cells with 8 million ZMWs, 8 times the data of the original Sequel system and 48 times the data of the RSII system.



- Complete template preparation in ~6 hours
- Accepts a variety of sample types and insert sizes
- Size-selection options to enrich for longest inserts
- Multiplexing and barcoding solutions available

- Rapid sequence time (0.5 to 30 hours)
- Serially process up to 8 SMRT Cells
- Walk away time up to four days

- Variety of analysis methods available through SMRT Analysis and PacBio Analytical Portfolio
- DevNet and Analytical Partners applications
- Advanced data visualization and mining

The Sequel II system has been optimized to efficiently produce HiFi data in addition to our suite of other sequencing applications. We will use this dataset to demonstrate the ease and utility of variant calling with HiFi reads.

Command Line Instructions for Exercise

Index reference and align reads with pbmm2

This step in pipeline was performed prior to workshop for time considerations. The “read_alignments.bam” file was generated by aligning the whole HiFi dataset against the whole hg38 assembly and using samtools to select reads mapping to chr19 with MAPQ = 60 and no secondary or supplementary alignments.

Assess mapped concordance and mapping quality

Script: 02_mapped_concordance.sh

```
samtools view alignment/read_alignments.bam \
   | grep -Eo "mc:f:[0-9]+.[0-9]+" \
   | cut -c 6- | sort -n \
   > output/sorted_conc.txt
```

```
python scripts/plot_conc.py
```

Quality analysis and control with deepTools

Script: 03_deeptools_analysis.sh

```
samtools depth -a \
    --reference reference/chr19.fa \
    alignment/read_alignments.bam \
    > output/base_coverage.txt
```

```
python scripts/plot_base.py
```

```
bamCoverage -b alignment/read_alignments.bam -p 4 \
    --binSize 58000 --effectiveGenomeSize 58440759 \
    --normalizeUsing RPGC --smoothLength 100000 \
    -o output/read_coverage.bw
```

```
make_tracks_file \
    --trackFiles output/read_coverage.bw \
    -o output/read_coverage.ini
```

```
pyGenomeTracks \
    --tracks output/read_coverage.ini \
    --outFileName output/read_coverage.png \
    --region NC_000019.10:1-58617616
```

Call SNVs, indels with GATK and SVs with pbsv

Script: 04_variant_calling.sh

```
pbsv discover -b reference/TRF_annotations.bed \
    alignment/read_alignments.bam output/disc.svsig.gz
```

```
pbsv call reference/chr19.fa \
    output/disc.svsig.gz output/pbsv_variants.vcf
```

```
gatk CreateSequenceDictionary -R reference/chr19.fa -O reference/chr19.dict
```

```
gatk HaplotypeCaller \
    --reference reference/chr19.fa \
    --input alignment/read_alignments.bam \
    --output output/gatk_variants.vcf \
    --native-pair-hmm-threads 4 \
    --pcr-indel-model AGGRESSIVE \
```

Merge, assess, and filter VCF and GFF files

Script: 05_variant_merging.sh

```
bgzip output/gatk_variants.vcf
```

```
tabix -p vcf output/gatk_variants.vcf.gz
```

```
bgzip output/pbsv_variants.vcf

tabix -p vcf output/pbsv_variants.vcf.gz

vcf-merge \
    output/gatk_variants.vcf.gz \
    output/pbsv_variants.vcf.gz \
    > output/merged_variants.vcf

bedtools intersect -wa -u \
    -a reference/chr19_annotations.gff \
    -b output/merged_variants.vcf \
    > output/mutant_alleles.gff

awk '$3=="CDS" {print $0}' output/mutant_alleles.gff \
    > output/mutant_CDS.gff

grep "#" output/merged_variants.vcf \
    > output/CDS_variants.vcf

bedtools intersect -wa -u \
    -a output/merged_variants.vcf \
    -b output/mutant_CDS.gff \
    >> output/CDS_variants.vcf
```

Phase variants into haplotypes with WhatsHap

Script: 06_whatshap_analysis.sh

```
whatshap phase -o output/phased_variants.vcf \
    -r reference/chr19.fa --mapq 20 -H 10 --indels \
    output/merged_variants.vcf alignment/read_alignments.bam

whatshap stats --tsv output/phased_stats.tsv \
    --block-list output/phased_blocks.txt \
    --gtf output/phased_blocks.gtf \
    output/phased_variants.vcf

bgzip output/phased_variants.vcf

tabix -p vcf output/phased_variants.vcf.gz

whatshap haplotag -o output/read_haplotags.bam \
    output/phased_variants.vcf.gz alignment/read_alignments.bam
```