Project: Genome Assembly and Variant Calling

This project involves genome assembly and variant calling for the sample **SRR2584868** using next-generation sequencing (NGS) data. The process utilizes a variety of bioinformatics tools, including **Trimmomatic**, **BWA**, **Samtools**, and **Bcftools**, to clean, align, and call variants from sequencing data.

---

1. Project Setup and Directory Structure

The first step is to set up a directory structure for the project to keep data organized. Here's an overview of the directory structure:

```
genome_assembly_project/
├── annotation          # Results for genome annotation (optional)
├── assembly            # Genome assembly results (if applicable)
├── raw_data            # Raw sequencing data files and quality control
reports
├── reference           # Reference genome for alignment
├── results             # Output data (e.g., aligned reads, variants,
images)
├── tools               # External tools, like Trimmomatic and BWA
└── trimmed_data        # Preprocessed (trimmed) sequencing data
```

Explanation of Directory Structure:

- **raw_data/**: This directory holds the raw sequencing files, such as the FASTQ files containing the sequencing reads, and the corresponding quality reports generated by **FastQC**.
- **reference/**: Contains the reference genome used for read alignment. It includes the genome in **FASTA format** as well as the index files created for alignment.
- **trimmed_data/**: After trimming and cleaning the raw data, this folder holds the processed FASTQ files.
- **results/**: This folder stores the results of the analysis, including aligned BAM files, called variants, and visualization files like IGV snapshots.
- **tools/**: Includes any third-party bioinformatics software used in the pipeline, such as **Trimmomatic**, **BWA**, and **Samtools**.
- **annotation/** and **assembly/**: These folders can be used for storing additional analysis files, such as gene annotations or assembly outputs, if applicable.

---

2. Quality Control of Raw Data

Before processing the raw sequencing data, we need to perform **quality control** (QC) to assess whether the data is suitable for further analysis.

FastQC Command:

Use **FastQC** to generate reports that describe the quality of the raw sequencing data.

Command:

```
fastqc SRR2584868_1.fastq.gz SRR2584868_2.fastq.gz
```

Explanation:

- `fastqc`: Command to run FastQC.
- `SRR2584868_1.fastq.gz` and `SRR2584868_2.fastq.gz`: These are the raw paired-end sequencing data files.

Important Terms:

- **FASTQ Format**: A file format containing sequence data along with quality scores for each base in the sequence.
- **Base Quality Scores**: Numeric values that indicate the confidence in the base call for each nucleotide in the sequence. They are measured using the **Phred quality score**.
- **GC Content**: The percentage of nucleotides that are **guanine (G)** or **cytosine (C)** in the sequence. Anomalies in GC content can indicate issues in the data.

---

3. Data Preprocessing: Trimming

Trimming is performed to remove low-quality bases from the sequences and to eliminate any adapter sequences that might have been introduced during the sequencing process. This is done using **Trimmomatic**.

Trimmomatic Command:

```
java -jar trimmomatic-0.39.jar PE -threads 4 SRR2584868_1.fastq.gz
SRR2584868_2.fastq.gz \
../trimmed_data/SRR2584868_1_paired.fastq.gz
../trimmed_data/SRR2584868_1_unpaired.fastq.gz \
../trimmed_data/SRR2584868_2_paired.fastq.gz
../trimmed_data/SRR2584868_2_unpaired.fastq.gz \
ILLUMINACLIP:/path/to/adapters/TruSeq3-PE.fa:2:30:10 LEADING:3 TRAILING:3
SLIDINGWINDOW:4:20 MINLEN:36
```

Explanation:

- `java -jar trimmomatic-0.39.jar PE`: Runs Trimmomatic in paired-end mode.
- `-threads 4`: Utilizes 4 CPU threads to speed up processing.

- `ILLUMINACLIP:/path/to/adapters/TruSeq3-PE.fa:2:30:10`: Removes adapter sequences using the TruSeq3 adapter file.
- `LEADING:3 TRAILING:3`: Removes bases with a quality score below 3 from the beginning (LEADING) or end (TRAILING) of the read.
- `SLIDINGWINDOW:4:20`: Trims when the average quality score in a sliding window of size 4 is below 20.
- `MINLEN:36`: Discards reads shorter than 36 bases.

Important Terms:

- **Adapter Sequences**: Short DNA sequences added during sequencing for amplification. These sequences are not part of the actual genome and must be removed.
- **Sliding Window**: A technique used to assess quality along the read by analyzing a window of a fixed size.

---

4. Read Alignment

After trimming, the next step is to align the cleaned reads to a reference genome. This helps determine where each read aligns in the genome.

BWA Command:

```
bwa mem -t 4 /path/to/reference/GCF_000005845.2_ASM584v2_genomic.fna \
/path/to/trimmed_data/SRR2584868_1_paired.fastq.gz \
/path/to/trimmed_data/SRR2584868_2_paired.fastq.gz > aligned_reads.sam
```

Explanation:

- `bwa mem`: The BWA algorithm used for aligning paired-end reads to the reference genome.
- `-t 4`: Utilizes 4 CPU threads for parallel computation.
- `/path/to/reference/`: Path to the reference genome file.
- `SRR2584868_1_paired.fastq.gz` and `SRR2584868_2_paired.fastq.gz`: Paired-end FASTQ files to be aligned.
- `aligned_reads.sam`: Output SAM file containing the alignment data.

Important Terms:

- **SAM Format**: A text-based format used for storing sequence alignments. Each line in the SAM file corresponds to a read and its alignment information.
- **Reference Genome**: A known genome sequence used as a template for aligning the sequencing reads.

---

5. Sorting and Indexing BAM Files

After alignment, we need to convert the **SAM** file to **BAM** (Binary Alignment Map) format, sort it, and create an index for faster access to specific regions of the genome.

Samtools Commands:

```
samtools view -bS aligned_reads.sam > aligned_reads.bam
samtools sort aligned_reads.bam -o aligned_reads_sorted.bam
samtools index aligned_reads_sorted.bam
```

Explanation:

- `samtools view`: Converts the SAM file to BAM format.
- `samtools sort`: Sorts the BAM file by genomic coordinates.
- `samtools index`: Creates a **BAM index** file (`*.bai`) for fast access.

Important Terms:

- **BAM Format**: A binary version of the SAM format, more efficient for storage and processing.
- **Indexing**: Creates an index file (`*.bai`) for efficient retrieval of reads from specific genomic locations.

---

6. Variant Calling

Variant calling identifies genetic variations such as **SNPs** (Single Nucleotide Polymorphisms) and **INDELs** (Insertions and Deletions) from the aligned reads.

Samtools and Bcftools Commands:

```
samtools mpileup -Ou -f
/path/to/reference/GCF_000005845.2_ASM584v2_genomic.fna \
aligned_reads_sorted.bam | bcftools call -mv -Ob -o variants.bcf
```

Explanation:

- `samtools mpileup`: Creates a pileup file that summarizes base calls at each position in the reference genome.
- `bcftools call`: Calls variants (SNPs and INDELs) from the pileup file.
- `variants.bcf`: The output BCF file containing the called variants.

Important Terms:

- **SNPs**: Variations in a single base pair.
- **INDELs**: Variations caused by insertions or deletions of bases in the genome.
- **Pileup**: A textual summary of the alignment data at each position in the genome.

---

7. Variant Filtering

Variants with low quality or those deemed unreliable need to be filtered out.

BCFtools Filter Command:

```
bcftools filter -i 'QUAL>30' variants.vcf > filtered_variants.vcf
```

Explanation:

- `-i 'QUAL>30'`: Filters variants that have a **QUALITY score** (QUAL) lower than 30.
- `variants.vcf`: Input VCF file containing the variants.
- `filtered_variants.vcf`: Output file containing high-quality filtered variants.

Important Terms:

- **VCF Format**: Variant Call Format, a standard file format for representing genetic variants.
- **Quality Score (QUAL)**: A numerical measure of the confidence in a variant call.

---

8. Visualization

Visualizing the results helps in assessing the quality of the data and the distribution of variants.

IGV (Integrative Genomics Viewer):

IGV is a visualization tool used to display sequencing alignments, coverage, and variants. It allows researchers to visually inspect how reads align to the reference genome and where variants are located.

1. Load the **aligned_reads_sorted.bam** and **variants.vcf** files into IGV.
2. Inspect the alignment and variant distribution across the genome.
