# 1.0 Introduction

## Overview

This user manual describes an automated sequence assembly pipeline designed specifically for Whole Exome Sequencing (WES). The pipeline aims to streamline the sequence assembly process, providing a comprehensive, user-friendly solution that manages various stages of data processing and analysis. By automating sequence assembly, the pipeline significantly reduces the time and effort required compared to traditional manual methods, enabling users to focus on other critical aspects of their work.

## Purpose

The primary purpose of this pipeline is to efficiently manage the sequence assembly process for WES data. Traditional manual sequence assembly methods are often time-consuming and labor-intensive, requiring extensive manual intervention and attention to detail. This automated pipeline addresses these challenges by providing a cohesive system that integrates multiple bioinformatics tools and processes, facilitating a smooth and accurate assembly of sequences. It allows users to input raw sequencing data and proceed through quality control, alignment, variant calling, and annotation with minimal manual intervention, ultimately saving valuable time and resources.

## Audience

This manual is intended for bioinformaticians, researchers, and lab technicians involved in genomic data analysis, particularly those working with WES data. The pipeline is designed to be accessible to users with a basic familiarity with genomic data and fundamental command-line usage. It provides a straightforward approach to handling the complexities of sequence assembly, making it suitable for a broad range of users within the genomic research community.

# 2.0 Installation

## System Requirements

To ensure the efficient operation of the automated sequence assembly pipeline, the following hardware and software specifications are required:

- **Operating System:** Ubuntu 22.04.4 LTS
- **RAM:** 32.0 GiB
- **Architecture:** 64-bit
- **Disk Space:** 1.0 TB
- **Swap Space:** At least 50 GiB (may need to increase if available memory is insufficient)

These specifications are necessary to handle the computational demands and data storage requirements of the pipeline, particularly when processing large-scale genomic datasets. Adequate swap space helps manage memory requirements, especially during memory-intensive operations.

## Dependencies

The pipeline relies on several key software tools and dependencies. Ensure that the following are installed and correctly configured on your system:

- **Python**:
  - Version: 3.11.7
- **Java**:
  - OpenJDK version "17.0.12"
  - OpenJDK Runtime Environment (build 17.0.12+7-Ubuntu-1ubuntu222.04)
  - OpenJDK 64-Bit Server VM (build 17.0.12+7-Ubuntu-1ubuntu222.04, mixed mode, sharing)
- **Genomic Analysis Tools**:
  - **GATK**: Version 4.6.0.0
  - **samtools**: Version 1.13
  - **BWA**: Version 0.7.17
  - **FastQC**: Version 0.12.1
  - **ANNOVAR**: Latest version

## Installation

Please refer to the official websites for detailed installation instructions and the latest version updates for each of the following tools and dependencies:

**• Python 3.11.7:**

- Source: https://www.python.org/downloads/

- Packages required:

    • os

    • json

    • threading

    • subprocess

    • tkinter (including ttk, filedialog, and scrolledtext)

    These packages are essential for ensuring the proper functioning of the pipeline.

**• OpenJDK 17.0.12:**

- Source: https://openjdk.org/install/

**• FastQC 0.12.1 (Andrews, 2010):**

- Source: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

**• GATK 4.6.0.0 (McKenna et al., 2010):**

- Source: https://github.com/broadinstitute/gatk/releases

**• Samtools 1.13 (Danecek et al., 2021):**

- Source: https://www.htslib.org/

**• Burrows-Wheeler Aligner 0.7.17 (Li & Durbin, 2009):**

- Source: https://bio-bwa.sourceforge.net/

**• AVVOVAR (Wang et al., 2010):**

- Version Date: 2020-06-07

- Source: https://annovar.openbioinformatics.org/en/latest/user-guide/download/#annovar-main-package

# 3.0 Getting Started

## Launching the Application

Before launching the application, ensure you have downloaded the necessary scripts from the project's GitHub repository. You can obtain the main.py and process_management.py files from this GitHub link. After downloading, install all the required tools and dependencies as outlined in the documentation. Next, create a directory and place your input files in it, making sure that the directory is properly organized to avoid any issues during the sequence assembly process.

To start the automated sequence assembly pipeline, you can use either a terminal or an integrated development environment (IDE). Open the main.py in the IDE and run it. Or, navigate to the directory containing the main.py file and run the following command in terminal:

```
python main.py
```

This command launches the application, initiating the graphical user interface (GUI) for the pipeline. Ensure that all necessary dependencies and environment variables are set up correctly before launching.

## User Interface Overview

The application features a user-friendly interface divided into several main sections, each accessible via the top menu bar. Here's a brief overview of each page and its primary functions:

- **MainPage**:
    - This is the central hub where users can input their sequencing data. You can provide the paths to your sequencing output files in FASTQ format and the reference sequence. The MainPage also includes a "Run" button to initiate the sequence assembly process. This page is designed for quick access and easy initiation of the pipeline.
- **ConfigurationPage**:
    - The ConfigurationPage allows users to fine-tune the settings for various tools used in the pipeline. Here, you can adjust the arguments and parameters for tools such as GATK, BWA, and ANNOVAR. This customization enables more precise control over the analysis and ensures that the pipeline can be tailored to specific research needs.
- **ProgramPage**:

    - On the ProgramPage, users can provide the paths to the executable files of the external tools required by the pipeline. This includes specifying the locations for FASTQC, GATK, BWA, and ANNOVAR. Correctly setting these paths ensures that the pipeline can correctly call and utilize these tools during the analysis process.
- **HelpPage**:
    - The HelpPage provides guidance and support resources. It contains information about the application, and troubleshooting. This page is designed to assist users in navigating the application and resolving common issues.

Each page is accessible via the navigation bar at the top of the application window. The interface is designed to be intuitive and straightforward, making it easy for users to input data, configure settings, and manage the analysis process.

# 4.0 Detailed Usage Instructions

## MainPage

The MainPage serves as the starting point for setting up your sequence assembly analysis. Follow the steps below to correctly input your data and initiate the pipeline:

1. **Setting the Directory**:
    - **Directory**: Click the "Browse" button next to the "Directory" label or directly paste the path in the provided entry field. This directory will be used to store all output files generated during the pipeline execution.
2. **Loading Input Sequences**:
    - **Forward Read Sequence**: Click the "Browse" button next to the "Forward Read Sequence" label or paste the path in the entry field. Ensure the file is in FASTQ format and located within the specified directory.
    - **Reverse Read Sequence**: Similarly, use the "Browse" button next to the "Reverse Read Sequence" label or paste the path. This file should also be in FASTQ format and stored in the directory.
    - **Reference Sequence**: Use the "Browse" button next to the "Reference Sequence" label or paste the path. The reference sequence must be in FASTA format (with a .fa extension) and also reside in the output directory.
3. **Important**: Ensure that the forward read sequence, reverse read sequence, and reference sequence files are all located in the directory you set as the output directory. This helps avoid errors later in the pipeline.
4. **Initiating the Pipeline**:
    - **Run Button**: Once all settings and paths are configured, return to the MainPage and click the "Run" button to start the sequence assembly process. The pipeline will then proceed through the various steps, including quality control, alignment, variant calling, and annotation.
5. **Monitoring Progress**:
    - A widget will display real-time updates on the progress of the pipeline. It will show the commands being executed at each step, providing transparency and allowing users to track the workflow. This feedback is useful for monitoring the analysis and identifying any issues that may arise.

By following these steps on the MainPage, you can efficiently set up and execute the sequence assembly pipeline. This page serves as the central interface for managing input data and initiating the analysis, making it a crucial component of the user experience.

## ConfigurationPage

The ConfigurationPage allows users to customize the settings and parameters for various tools used in the sequence assembly pipeline. Users can adjust both required and optional arguments for each tool to suit their specific needs. It is essential that all required arguments are filled in correctly before starting the sequence assembly process to ensure the pipeline runs smoothly and without errors.

### FastqToSam (Picard) Section

The FastqToSam tool from Picard is used to convert FASTQ files into unaligned BAM files, adding read group information and other metadata. The ConfigurationPage allows users to set various required and optional arguments for this tool.

#### Required Arguments for FastqToSam

- **FASTQ to SAM Output**
  - Default Value: "unaligned_read_pairs.bam"
  - Description: Specifies the output file name for the unaligned BAM file. This file will contain the reads from the FASTQ file(s) along with any additional metadata.
- **Sample Name**
  - Default Value: "sample001"
  - Description: The name of the sample being processed. This name will be included in the read group header and can be used to identify the sample in subsequent analyses.
- **Platform**
  - Default Value: ""
  - Description: The sequencing platform used to generate the data, such as "ILLUMINA" or "SOLID". This value is included in the read group header.

#### Optional Arguments for FastqToSam

- **Allow and Ignore Empty Lines**
  - Default Value: False
  - Description: If set to True, the tool will allow and ignore empty lines in the FASTQ files.
- **Use Sequential FASTQs**
  - Default Value: False
  - Description: When True, this option indicates that the FASTQ files should be processed sequentially.
- **Comment**
  - Default Value: ""

    - Description: A free-text field for adding comments about the data or the run.
- **Description**
    - Default Value: ""
    - Description: A description of the dataset or experiment.
- **Library Name**
    - Default Value: ""
    - Description: The name of the library from which the sequencing data was derived.
- **Max Q**
    - Default Value: 93
    - Description: The maximum quality score to be included in the output. Scores above this value will be capped.
- **Min Q**
    - Default Value: 0
    - Description: The minimum quality score to be included in the output. Scores below this value will be capped.
- **Platform Model**
    - Default Value: ""
    - Description: Specifies the model of the sequencing instrument used.
- **Platform Unit**
    - Default Value: ""
    - Description: The platform unit is typically a unique identifier for the sequencing run.
- **Predicted Insert Size**
    - Default Value: 0
    - Description: An estimate of the insert size for paired-end reads.
- **Program Group**
    - Default Value: ""
    - Description: Information about the program group, which may include details about the software version used.
- **Quality Format**
    - Default Value: ""
    - Description: Specifies the format of the quality scores (e.g., "Phred+33").
- **Read Group Name**
    - Default Value: "A"
    - Description: The name of the read group, used to distinguish between different groups of reads.
- **Run Date**
    - Default Value: ""
    - Description: The date when the sequencing run was performed.
- **Sequencing Center**

    - Default Value: ""
    - Description: The name of the center or facility where the sequencing was performed.
- **FastqToSam Sort Order**
    - Default Value: "queryname"
    - Description: The order in which the records in the output BAM file should be sorted. The default value "queryname" indicates that reads are sorted by the query name.

For further details and additional configuration options, users should refer to the FastqToSam (Picard) documentation. This documentation provides comprehensive descriptions of all parameters and their potential values, facilitating optimal configuration according to specific analytical needs.

---

**MarkIlluminaAdapters (Picard) Section**

The MarkIlluminaAdapters tool from Picard is used to identify and clip Illumina adapter sequences from reads in a BAM file. It also generates metrics detailing the clipping process. This section of the ConfigurationPage allows users to set required and optional arguments for the tool.

**Required Arguments for MarkIlluminaAdapters**

- **Mark Illumina Metrics**
    - Default Value: "metrics.txt"
    - Description: Specifies the output file for the metrics generated by the tool. This file contains a histogram showing counts of bases clipped across reads, providing insights into the extent and distribution of adapter contamination.
- **Mark Illumina Output**
    - Default Value: "MarkIlluminaAdapters_read_pairs.bam"
    - Description: Specifies the output BAM file after adapter clipping. This file will contain the reads with adapter sequences clipped based on the settings provided.

**Optional Arguments for MarkIlluminaAdapters**

- **Adapter Truncation Length**
    - Default Value: 30
    - Description: Sets the length to which adapter sequences are truncated to speed up the matching process. Setting this value to a large number effectively disables truncation.
- **Adapters**

    - Default Value: "PAIRED_END"
    - Description: Specifies the type of adapters to identify and clip. Options include "INDEXED", "DUAL_INDEXED", and "PAIRED_END".
- **Five Prime Adapter**
    - Default Value: ""
    - Description: Allows users to specify five prime adapters that are not standard Illumina adapters. This is useful for custom sequencing setups.
- **Max Error Rate PE**
    - Default Value: 0.1
    - Description: The maximum mismatch error rate tolerated when clipping paired-end reads. This value defines how strictly the tool matches adapter sequences in paired-end data.
- **Max Error Rate SE**
    - Default Value: 0.1
    - Description: The maximum mismatch error rate tolerated when clipping single-end reads. This setting is similar to Max Error Rate PE but applies to single-end data.
- **Min Match Bases PE**
    - Default Value: 6
    - Description: The minimum number of bases that must match when clipping paired-end reads. This threshold ensures that only reads with a significant match to adapter sequences are clipped.
- **Min Match Bases SE**
    - Default Value: 12
    - Description: The minimum number of bases that must match when clipping single-end reads. It ensures accuracy in identifying adapter sequences in single-end data.
- **Num Adapters to Keep**
    - Default Value: 1
    - Description: Specifies the number of adapter sequences to retain when pruning the list of possible adapters. This setting is useful for focusing on the most common adapters observed in the input data.
- **Three Prime Adapter**
    - Default Value: ""
    - Description: Allows users to specify three prime adapters that are not standard Illumina adapters. This is useful for custom or specialized sequencing setups.

For a comprehensive understanding of all the parameters and their potential values, please refer to the MarkIlluminaAdapters (Picard) documentation. This documentation provides detailed descriptions and usage guidelines, enabling users to optimize the tool settings for their specific sequencing data and research needs.

---

**SamToFastq (Picard) Section**

The SamToFastq tool from Picard is used to extract reads from a SAM or BAM file and convert them into FASTQ format. This tool can handle both single-end and paired-end reads, and offers various options for customization. The ConfigurationPage allows users to set required and optional arguments for this tool.

**Required Arguments for SamToFastq**

- **SAM to FASTQ Output**
  - Default Value: "output.fastq"
  - Description: Specifies the output file name for the FASTQ file generated from the SAM/BAM input. This output file will contain the reads extracted from the input file, formatted as FASTQ.

**Optional Arguments for SamToFastq**

- **Clipping Action**
  - Default Value: ""
  - Description: Defines the action to take with clipped reads. Options include 'X' (trim reads and qualities at the clipped position), 'N' (replace bases with Ns in the clipped region), or an integer (set base qualities to this value in the clipped region).
- **Clipping Attribute**
  - Default Value: ""
  - Description: Specifies the attribute that stores the position at which the SAM record should be clipped.
- **Clipping Min Length**
  - Default Value: 0
  - Description: Ensures that resulting reads after clipping are at least this many bases long. If the original read is shorter than this value, the original length will be maintained.
- **Include Non PF Reads**
  - Default Value: False
  - Description: When set to True, non-PF (not passing filter) reads will be included in the output FASTQ files. PF reads are those that pass quality filtering.
- **Include Non Primary Alignments**
  - Default Value: False
  - Description: If True, includes non-primary alignments in the output. Note that support for non-primary alignments in SamToFastq is limited.

- **Interleave**
    - Default Value: True
    - Description: For paired-end data, this option generates an interleaved FASTQ file, with each line indicating whether the read is from the first or second end of the pair.
- **Quality**
    - Default Value: 0
    - Description: Applies end-trimming to reads using a quality trimming algorithm, with this specified quality threshold.
- **Read1 Max Bases to Write**
    - Default Value: 0
    - Description: Specifies the maximum number of bases to write from read 1 after trimming. If not set, all bases after trimming will be written.
- **Read1 Trim**
    - Default Value: 0
    - Description: The number of bases to trim from the beginning of read 1.
- **Read2 Max Bases to Write**
    - Default Value: 0
    - Description: Specifies the maximum number of bases to write from read 2 after trimming. If not set, all bases after trimming will be written.
- **Read2 Trim**
    - Default Value: 0
    - Description: The number of bases to trim from the beginning of read 2.
- **RG Tag**
    - Default Value: "PU"
    - Description: Specifies the read group tag (PU or ID) to be used for generating a FASTQ file per read group.

---

**BWA Alignment Section**

The BWA (Burrows-Wheeler Aligner) tool is widely used for aligning sequencing reads to a reference genome. This section of the ConfigurationPage allows users to set various parameters required for the alignment process. The BWA alignment involves creating an index of the reference genome and aligning reads using different algorithms such as bwa mem, bwa aln, and others depending on the data type and study requirements.

**Required Arguments for BWA Alignment**

- **Threads**
    - Default Value: 4

    - Description: Specifies the number of threads to be used for the alignment process. Using multiple threads can speed up the alignment process, especially with large datasets.
- **Index Algorithm**
    - Default Value: "bwtsw"
    - Description: The algorithm used for constructing the Burrows-Wheeler Transform (BWT) index of the reference genome. The "bwtsw" algorithm is recommended for large genomes like the human genome. Another option is "is", which is faster for smaller genomes but requires more memory and cannot handle genomes larger than 2GB.

---

**CreateSequenceDictionary (Picard) Section**

The CreateSequenceDictionary tool from Picard is used to generate a sequence dictionary from a reference FASTA file. The dictionary provides metadata about the reference sequences, such as their names and lengths, and is used by many other bioinformatics tools for efficient data processing. This section of the ConfigurationPage allows users to set optional arguments for this tool.

**Optional Arguments for CreateSequenceDictionary**

- **Genome Assembly**
    - Default Value: ""
    - Description: Specifies the genome assembly version. This value is placed in the AS (Assembly) field of the sequence dictionary entry. It is useful for identifying the specific version of the reference genome used in the analysis.
- **Number of Sequences**
    - Default Value: 2147483647
    - Description: The maximum number of sequences to include in the output dictionary. This option is primarily used for testing purposes and should generally not be changed.
- **Species**
    - Default Value: ""
    - Description: Specifies the species associated with the reference genome. This information is included in the SP (Species) field of the sequence dictionary entry, providing additional context for the reference sequences.
- **Truncate Names at Whitespace**
    - Default Value: True
    - Description: When set to True, only the first word from the > line in the FASTA file is used as the sequence name. The default behavior includes the entire

contents of the > line, excluding leading and trailing whitespace. This option helps to standardize sequence names, particularly when they contain additional descriptive information.

- **URI**
    - Default Value: ""
    - Description: Specifies a URI (Uniform Resource Identifier) for the reference sequence. If not provided, the input reference file path is used. This field can be used to provide a reference link or additional metadata about the source of the reference sequences.

For a complete understanding of all parameters and their uses, users should refer to the CreateSequenceDictionary (Picard) documentation. This documentation provides detailed descriptions and examples, helping users configure the tool according to their specific requirements and ensuring accurate and consistent metadata in their sequence dictionary.

---

**MergeBamAlignment (Picard) Section**

The MergeBamAlignment tool from Picard is used to merge alignment data from multiple BAM files, including unmapped BAMs and aligned BAMs. It can incorporate additional alignment information, adjust metadata, and refine read mapping details. This section of the ConfigurationPage allows users to set required and optional arguments for this tool.

**Required Arguments for MergeBamAlignment**

- **Merge Align BAM Output**
    - Default Value: "merge_align.bam"
    - Description: Specifies the output file name for the merged BAM file, which includes aligned and potentially adjusted reads. This file is the primary output of the MergeBamAlignment process.

**Optional Arguments for MergeBamAlignment**

- **Add Mate Cigar**
    - Default Value: False
    - Description: If set to True, the tool adds the mate CIGAR tag (MC) to each read, providing detailed alignment information about the mate read in paired-end sequencing.
- **Aligned Reads Only**
    - Default Value: False

    - Description: When True, the output will include only reads that are aligned. Unaligned reads will be excluded from the final BAM file.
- **Aligner Proper Pair Flags**
    - Default Value: False
    - Description: Uses the aligner's definition of proper pairs, rather than computing it during the merging process. This can be useful if the aligner has specific criteria for defining proper pairs.
- **Attributes To Remove**
    - Default Value: ""
    - Description: A list of SAM attributes (tags) to remove from the alignment records during merging. This overrides the Attributes To Retain setting if they overlap.
- **Attributes To Retain**
    - Default Value: ""
    - Description: Specifies which reserved alignment attributes (tags starting with X, Y, or Z) should be retained in the output BAM file.
- **Attributes To Reverse**
    - Default Value: ""
    - Description: A list of attributes on negative strand reads that should be reversed.
- **Attributes To Reverse Complement**
    - Default Value: ""
    - Description: Attributes on negative strand reads that need to be reverse complemented.
- **Clip Adapters**
    - Default Value: True
    - Description: Determines whether to clip adapter sequences where identified in the reads.
- **Clip Overlapping Reads**
    - Default Value: True
    - Description: For paired-end reads, this option soft clips the 3' end of each read to prevent them from overlapping and extending past the 5' end of their mate.
- **Expected Orientations**
    - Default Value: ""
    - Description: Specifies the expected orientations of proper read pairs, replacing the older JUMP_SIZE parameter.
- **Include Secondary Alignments**
    - Default Value: True
    - Description: If True, secondary alignments are included in the output BAM file. Secondary alignments are additional alignments for a read that may be of lower quality or represent alternative placements.
- **MergeBamAlignment Is Bisulfite Sequence**

    - Default Value: False
    - Description: Indicates whether the data is from bisulfite sequencing, which affects the calculation of certain alignment tags like the NM tag.
- **Matching Dictionary Tags**
    - Default Value: ""
    - Description: A list of tags from the sequence dictionary that must match between the reference dictionary and the aligned files. Mismatches can cause errors or warnings.
- **Max Insertions Or Deletions**
    - Default Value: 1
    - Description: The maximum number of insertions or deletions allowed for an alignment to be included in the output. Alignments exceeding this threshold will be ignored.
- **Min Unclipped Bases**
    - Default Value: 32
    - Description: The minimum number of unclipped bases required for a read to be retained. This setting is used in conjunction with Unmap Contaminant Reads.
- **Primary Alignment Strategy**
    - Default Value: "MostDistant"
    - Description: Defines the strategy for selecting the primary alignment when multiple are present or none are marked as primary. "MostDistant" typically selects the alignment that is the farthest from the reference.
- **Program Group Command Line**
    - Default Value: ""
    - Description: The command line used for the program group, if not supplied by the aligned file.
- **Program Group Name**
    - Default Value: ""
    - Description: The name of the program group, providing information about the tool used for alignment.
- **Program Group Version**
    - Default Value: ""
    - Description: The version of the program group, useful for tracking the specific version of the tool used.
- **Program Record Id**
    - Default Value: ""
    - Description: The ID of the program record, linking alignments to the program that generated them.
- **Read1 Aligned Bam**
    - Default Value: ""

    - Description: Specifies the BAM file(s) containing alignment data from the first read of a pair.
- **Read1 Trim**
    - Default Value: 0
    - Description: The number of bases trimmed from the beginning of read 1 prior to alignment.
- **Read2 Aligned Bam**
    - Default Value: ""
    - Description: Specifies the BAM file(s) containing alignment data from the second read of a pair.
- **Read2 Trim**
    - Default Value: 0
    - Description: The number of bases trimmed from the beginning of read 2 prior to alignment.
- **MergeBamAlignment Sort Order**
    - Default Value: "unsorted"
    - Description: Specifies the order in which the reads are output in the merged BAM file. Common options include "coordinate" and "queryname".
- **Unmap Contaminant Reads**
    - Default Value: False
    - Description: If True, reads suspected to originate from contaminant sources (e.g., foreign organisms) are unmapped and labeled accordingly.
- **Unmapped Read Strategy**
    - Default Value: "COPY_TO_TAG"
    - Description: Defines how to handle alignment information in reads that are being unmapped, typically due to cross-species contamination.

For additional details and complete usage guidelines, users should refer to the MergeBamAlignment (Picard) documentation. This resource provides in-depth explanations of each parameter, examples, and best practices for optimizing the merging process.

---

**SortSam (Picard) For Merged BAM Section**

The SortSam tool from Picard is used to sort BAM or SAM files according to a specified order. This tool is essential for organizing reads in a meaningful way, such as by coordinate or queryname, and can also create associated indices and MD5 files for verification. This section of the ConfigurationPage allows users to set required and optional arguments for sorting merged BAM files.

**Required Arguments for SortSam**

- **SortSam For Merged BAM Output**
    - Default Value: "sorted_merge.bam"
    - Description: Specifies the output file name for the sorted BAM file. This file contains the reads from the input BAM file arranged in the specified sort order.
- **SortSam For Merged BAM Sort Order**
    - Default Value: "coordinate"
    - Description: Determines the order in which the reads are sorted in the output file. Common sort orders include "coordinate" (by genomic position) and "queryname" (by read name).

**Optional Arguments for SortSam**

- **SortSam For Merged BAM Create Index**
    - Default Value: False
    - Description: When set to True, an index file is created for the sorted BAM file. This index is useful for quickly accessing specific regions within the BAM file, particularly for downstream analysis.
- **SortSam For Merged BAM Create MD5 File**
    - Default Value: False
    - Description: If True, an MD5 checksum file is generated for the output BAM file. This file can be used to verify the integrity of the BAM file, ensuring that it has not been altered or corrupted.

For comprehensive details and best practices, users should consult the SortSam (Picard) documentation. This documentation provides full descriptions of all parameters, including examples of different sorting strategies and use cases.

---

**SetNmMdAndUqTags (Picard) Section**

The SetNmMdAndUqTags tool from Picard is used to set or correct the NM, MD, and UQ tags in a BAM or SAM file. These tags are crucial for downstream analysis, as they provide information about the number of mismatches, exact match positions, and base qualities. This section of the ConfigurationPage allows users to set required and optional arguments for this tool.

**Required Arguments for SetNmMdAndUqTags**

- **Setnmmduqtags Output**
    - Default Value: "fixed.bam"

    - Description: Specifies the output file name for the BAM file with updated or corrected NM, MD, and UQ tags. This file is the primary output, reflecting any changes made by the tool.

**Optional Arguments for SetNmMdAndUqTags**

- **SetnmmdUqtags Is Bisulfite Sequence**
    - Default Value: False
    - Description: Indicates whether the input file contains bisulfite sequence data. This is important for calculating the NM tag correctly, as bisulfite sequencing can affect base pairing and mismatches.
- **Set Only Uq**
    - Default Value: False
    - Description: If True, the tool will only set the UQ tag and ignore the MD and NM tags. This option is useful when only the UQ tag needs to be updated or corrected.
- **SetnmmdUqtags Create Index**
    - Default Value: False
    - Description: Specifies whether to create an index for the output BAM file. An index allows for efficient querying of the BAM file, which is useful for large datasets.
- **SetnmmdUqtags Create MD5 File**
    - Default Value: False
    - Description: If set to True, the tool generates an MD5 checksum for the output BAM file. This checksum can be used to verify the integrity of the file.

For complete details and additional options, users should refer to the SetNmMdAndUqTags (Picard) documentation. The documentation provides full descriptions of the parameters, including examples and best practices for ensuring accurate and reliable data processing.

---

**MarkDuplicates (Picard) Section**

The MarkDuplicates tool from Picard is used to identify and flag duplicate reads in a BAM or SAM file. Duplicate reads can arise during the PCR amplification process and can bias downstream analyses. The tool marks these duplicates, allowing them to be excluded from subsequent analysis, or in some cases, removed altogether. This section of the ConfigurationPage allows users to set required and optional arguments for identifying and marking duplicates.

**Required Arguments for MarkDuplicates**

- **Metrics File**

    - Default Value: "marked_dup_metrics.txt"
    - Description: Specifies the output file for the duplication metrics. This file contains detailed information about the number and types of duplicates detected, providing essential insights for quality control.
- **Mark Duplicates Output**
    - Default Value: "marked_duplicates.bam"
    - Description: Specifies the output BAM file where the duplicates have been marked. This file is essential for downstream processes that need to identify and potentially exclude duplicate reads.

**Optional Arguments for MarkDuplicates**

- **Assume Sort Order**
    - Default Value: "queryname"
    - Description: Indicates the assumed sort order of the input file. If the header indicates a different sort order, this parameter overrides it.
- **Barcode Tag**
    - Default Value: ""
    - Description: Specifies the SAM tag containing the barcode information, which can be used for identifying duplicates. This is especially useful in single-cell sequencing and other specialized sequencing methods.
- **Clear Dt**
    - Default Value: True
    - Description: Clears the DT tag from input SAM records. This is set to True by default, as the tag should not exist in the output.
- **MarkDuplicates Comment**
    - Default Value: ""
    - Description: Adds comments to the output file's header, useful for annotating the processing steps and settings used.
- **Duplicate Scoring Strategy**
    - Default Value: "SUM_OF_BASE_QUALITIES"
    - Description: Determines the strategy used for scoring and selecting the best representative among duplicate reads. Options include "SUM_OF_BASE_QUALITIES" and others.
- **Max File Handles For Read Ends Map**
    - Default Value: 8000
    - Description: Limits the number of file handles that can be open when spilling read ends to disk. This setting helps manage memory usage.
- **Max Optical Duplicate Set Size**
    - Default Value: 300000

    - Description: The maximum size of a set of duplicate reads considered for determining optical duplicates. This value can be adjusted based on the dataset size and characteristics.
- **Max Sequences For Disk Read Ends Map**
    - Default Value: 50000
    - Description: This option is obsolete. It was previously used to manage the number of sequences spilled to disk.
- **Optical Duplicate Pixel Distance**
    - Default Value: 2500
    - Description: Defines the maximum pixel distance between two clusters of duplicates for them to be considered optical duplicates. Adjusting this value is necessary for different sequencing platforms.
- **Program Group Command Line**
    - Default Value: ""
    - Description: The command line used for the program group in the output file header.
- **Program Group Name**
    - Default Value: ""
    - Description: The name of the program group for the process, useful for tracking the tool and version used.
- **Program Group Version**
    - Default Value: ""
    - Description: The version of the program group, if not automatically detected.
- **Program Record Id**
    - Default Value: ""
    - Description: The ID of the program record, used for tracking processing steps in the output file.
- **Read Name Regex**
    - Default Value: ""
    - Description: A regular expression used to parse read names for extracting tile, x, and y coordinates. This is important for identifying optical duplicates. If set to null, optical duplicate detection is disabled.
- **Read One Barcode Tag**
    - Default Value: ""
    - Description: Specifies the barcode tag for read one, used in identifying duplicates.
- **Read Two Barcode Tag**
    - Default Value: ""
    - Description: Specifies the barcode tag for read two, used in identifying duplicates.
- **Sorting Collection Size Ratio**
    - Default Value: 0.25

    - Description: Determines the memory footprint used for sorting collections. Adjusting this can help manage memory usage, especially in low-memory environments.
- **Tag Duplicate Set Members**
    - Default Value: False
    - Description: When True, adds tags to indicate the size of the duplicate set and provide a unique identifier for the duplicate set.
- **Tagging Policy**
    - Default Value: "DontTag"
    - Description: Specifies the policy for tagging duplicate types in the DT attribute. Options include "DontTag", "TagAll", and "TagRepresentative".
- **MarkDuplicates Create Index**
    - Default Value: False
    - Description: Indicates whether to create an index for the output BAM file. This is useful for quickly accessing specific regions in the file.
- **MarkDuplicates Create MD5 File**
    - Default Value: False
    - Description: Specifies whether to create an MD5 checksum file for the output BAM file. This file can verify the file's integrity.

For a comprehensive understanding of all parameters and their appropriate settings, users should refer to the MarkDuplicates (Picard) documentation. This resource provides detailed explanations and best practices for configuring the tool to detect and handle duplicates effectively.

---

**SortSam (Picard) For Marked Duplicate BAM Section**

The SortSam tool from Picard, specifically for marked duplicate BAM files, sorts the BAM or SAM file based on a specified order. After marking duplicates with MarkDuplicates, it's often necessary to sort the BAM file to prepare it for downstream analyses, such as variant calling or visualization. This section of the ConfigurationPage allows users to set required and optional arguments for sorting the BAM file that contains marked duplicates.

**Required Arguments for SortSam**

- **SortSam For Marked Duplicate Output**
    - Default Value: "sorted_marked.bam"
    - Description: Specifies the output file name for the sorted BAM file containing marked duplicates. This file is the result of sorting and is typically coordinate-sorted for further analysis.

- **SortSam For Marked Duplicate Sort Order**
    - Default Value: "coordinate"
    - Description: Determines the order in which reads are sorted in the output file. The "coordinate" sort order arranges reads by their genomic coordinates, which is required for many downstream applications, such as variant calling.

**Optional Arguments for SortSam**

- **SortSam For Marked Duplicate Create Index**
    - Default Value: False
    - Description: Indicates whether to create an index for the output BAM file. An index file allows efficient querying of specific regions in the BAM file, which is particularly useful for large datasets.
- **SortSam For Marked Duplicate Create MD5 File**
    - Default Value: False
    - Description: Specifies whether to generate an MD5 checksum file for the output BAM file. This checksum can be used to verify the integrity of the file, ensuring that it has not been altered or corrupted.

**Optional Common Arguments**

- **COMPRESSION_LEVEL**
    - Default Value: 5
    - Description: Specifies the compression level for the output BAM file. A higher compression level reduces file size but increases processing time.

For detailed usage guidelines and additional options, users should refer to the SortSam (Picard) documentation. The documentation provides comprehensive descriptions of all parameters, examples of different sort orders, and best practices for configuring the tool for various types of genomic data.

---

**BaseRecalibrator (Picard) Section**

The BaseRecalibrator tool from Picard is used to perform base quality score recalibration (BQSR). This process adjusts the quality scores of sequencing reads to correct systematic errors made by the sequencing machine when it estimates the quality score of each base call. The recalibration is based on known sites of variation and a model of machine cycle and sequence context effects. This section of the ConfigurationPage allows users to set required and optional arguments for base recalibration.

**Required Arguments for BaseRecalibrator**

- **Base Recal Output**
    - Default Value: "recal_data.table"
    - Description: Specifies the output file name for the recalibration table. This table contains the recalibration data generated by the tool, which can be used to adjust the base quality scores of the input reads.
- **Base Recal Known Sites**
    - Default Value: ""
    - Description: One or more databases of known polymorphic sites, which are used to identify known variants in the reads. This information helps exclude these regions from the recalibration process, ensuring that true variants are not treated as errors.

**Optional Arguments for BaseRecalibrator**

- **BQSR BAQ Gap Open Penalty**
    - Default Value: 40.0
    - Description: Specifies the gap open penalty used in the BAQ calculation, which is Phred-scaled. The default value is 40, but for whole genome call sets, a value of 30 might be more appropriate.
- **Cloud Index Prefetch Buffer**
    - Default Value: -1
    - Description: Sets the size of the cloud-only prefetch buffer in megabytes. A value of 0 disables the buffer. This option is particularly useful when working with data stored in cloud environments.
- **Cloud Prefetch Buffer**
    - Default Value: 40
    - Description: Specifies the size of the prefetch buffer for cloud environments, in megabytes. This buffer helps manage data transfer between cloud storage and local processing.
- **Default Base Qualities**
    - Default Value: -1
    - Description: Assigns a default base quality score to all bases. If set to a value other than -1, this overrides the quality scores in the original data.
- **Deletions Default Quality**
    - Default Value: 45
    - Description: The default quality score for base deletions, used as part of the recalibration process.
- **Disable BAM Index Caching**
    - Default Value: False

    - Description: When set to True, this option disables caching of BAM indexes, reducing memory usage but potentially affecting performance when processing many intervals.
- **GCS Max Retries**
    - Default Value: 20
    - Description: Specifies the maximum number of retry attempts for accessing data in Google Cloud Storage (GCS) if there are connection issues.
- **Indels Context Size**
    - Default Value: 3
    - Description: Defines the size of the k-mer context used for base insertions and deletions during recalibration.
- **Insertions Default Quality**
    - Default Value: 45
    - Description: The default quality score for base insertions, applied during the recalibration process.
- **Interval Merging Rule**
    - Default Value: "ALL"
    - Description: Determines how to merge adjacent intervals. The default "ALL" merges all intervals, but other settings can specify different merging behaviors.
- **Intervals**
    - Default Value: ""
    - Description: Specifies one or more genomic intervals to operate on. This option allows for focused recalibration on specific regions of interest.
- **Low Quality Tail**
    - Default Value: 2
    - Description: Sets the minimum quality score for bases at the tail ends of reads to be considered for recalibration.
- **Maximum Cycle Value**
    - Default Value: 500
    - Description: The maximum cycle value for the Cycle covariate. This setting helps limit the recalibration model to a reasonable range of cycle values.
- **Mismatches Context Size**
    - Default Value: 2
    - Description: The size of the k-mer context used for base mismatches during recalibration.
- **Mismatches Default Quality**
    - Default Value: -1
    - Description: Specifies the default quality score for base mismatches. A value of -1 means no override; the original scores are used.
- **Preserve QScores Less Than**

    - Default Value: 6
    - Description: Specifies a threshold below which base quality scores are not recalibrated. This helps preserve very low-quality scores, which may indicate problematic bases.
- **Quantizing Levels**
    - Default Value: 16
    - Description: Sets the number of distinct quality scores in the quantized output. This option reduces the number of quality score levels to simplify downstream processing.
- **Use Original Qualities**
    - Default Value: False
    - Description: If True, the tool uses the original quality scores from the OQ tag, if present, instead of the recalibrated scores.

**Optional Common Arguments**

- **Add Output SAM Program Record**
    - Default Value: True
    - Description: Adds a PG tag to created SAM/BAM/CRAM files, recording the tool and command line used.
- **Add Output VCF Command Line**
    - Default Value: True
    - Description: Adds a command line header line to created VCF files, documenting the tool and settings used.
- **Create Output BAM Index**
    - Default Value: True
    - Description: Indicates whether to create an index for the output BAM/CRAM file when it is coordinate-sorted.
- **Create Output BAM MD5**
    - Default Value: False
    - Description: Specifies whether to create an MD5 checksum for the output BAM/SAM/CRAM file, ensuring the integrity of the file.

For more detailed information and usage guidelines, users should refer to the BaseRecalibrator (Picard) documentation. The documentation provides comprehensive descriptions of all parameters, best practices for setting them, and examples to guide users in performing base quality score recalibration effectively.

---

**ApplyBQSR (Picard) Section**

The ApplyBQSR tool from Picard applies the base quality score recalibration (BQSR) data generated by BaseRecalibrator to a BAM or SAM file. This process adjusts the quality scores of the sequencing reads to correct systematic errors made by the sequencing machine. This section of the ConfigurationPage allows users to set required and optional arguments for applying the recalibration data to the input reads.

**Required Arguments for ApplyBQSR**

- **ApplyBQSR Output**
    - Default Value: "bqsr_output.bam"
    - Description: Specifies the output BAM file that will contain the recalibrated reads. The recalibrated quality scores are written to this file, replacing the original scores.

**Optional Arguments for ApplyBQSR**

- **ApplyBQSR Cloud Index Prefetch Buffer**
    - Default Value: -1
    - Description: Sets the size of the cloud-only prefetch buffer in megabytes. A value of 0 disables the buffer. This option is useful for optimizing data transfer when working with data stored in cloud environments.
- **ApplyBQSR Cloud Prefetch Buffer**
    - Default Value: 40
    - Description: Specifies the size of the prefetch buffer for cloud environments, in megabytes. This helps manage data transfer between cloud storage and local processing.
- **ApplyBQSR Disable BAM Index Caching**
    - Default Value: False
    - Description: When set to True, this option disables caching of BAM indexes, reducing memory usage but potentially affecting performance if many intervals are specified.
- **ApplyBQSR Emit Original Quals**
    - Default Value: False
    - Description: If set to True, the tool emits the original base qualities in the OQ tag, allowing comparison between the original and recalibrated quality scores.
- **ApplyBQSR GCS Max Retries**
    - Default Value: 20
    - Description: Specifies the maximum number of retry attempts for accessing data in Google Cloud Storage (GCS) if there are connection issues.
- **ApplyBQSR Global Qscore Prior**

    - Default Value: -1.0
    - Description: Sets a global Q-score Bayesian prior for BQSR. This is used to adjust the recalibration process based on prior knowledge of the sequencing quality.
- **ApplyBQSR Interval Merging Rule**
    - Default Value: "ALL"
    - Description: Determines how to merge adjacent intervals. The default "ALL" merges all intervals, but other settings can specify different merging behaviors.
- **ApplyBQSR Intervals**
    - Default Value: ""
    - Description: Specifies one or more genomic intervals to operate on. This option allows focused recalibration on specific regions of interest.
- **ApplyBQSR Preserve Qscores Less Than**
    - Default Value: 6
    - Description: Specifies a threshold below which base quality scores are not recalibrated, preserving low-quality scores that may indicate problematic bases.
- **ApplyBQSR Quantize Quals**
    - Default Value: 0
    - Description: Quantizes the quality scores to a specified number of levels. This reduces the complexity of the data and can help with downstream analysis.
- **ApplyBQSR Use Original Qualities**
    - Default Value: False
    - Description: If True, the tool uses the base quality scores from the OQ tag, if present, instead of the recalibrated scores.

**Optional Common Arguments**

- **Add Output SAM Program Record**
    - Default Value: True
    - Description: Adds a PG tag to created SAM/BAM/CRAM files, recording the tool and command line used.
- **Add Output VCF Command Line**
    - Default Value: True
    - Description: Adds a command line header line to created VCF files, documenting the tool and settings used.
- **Create Output BAM Index**
    - Default Value: True
    - Description: Indicates whether to create an index for the output BAM/CRAM file when it is coordinate-sorted.
- **Create Output BAM MD5**
    - Default Value: False

    - Description: Specifies whether to create an MD5 checksum for the output BAM/SAM/CRAM file, ensuring the integrity of the file.

For more detailed information and usage guidelines, users should refer to the ApplyBQSR (Picard) documentation. The documentation provides comprehensive descriptions of all parameters, best practices for setting them, and examples to guide users in applying base quality score recalibration effectively.

---

**HaplotypeCaller (Picard) Section**

The HaplotypeCaller tool from Picard is used to call variants (SNPs and indels) using local de novo assembly of haplotypes in a region showing evidence of variation. This tool is particularly powerful for its ability to detect complex variants and for its sophisticated handling of sequencing errors. This section allows users to configure the tool with the necessary inputs and fine-tune the analysis using various optional parameters.

**Required Arguments for HaplotypeCaller**

- **HaplotypeCaller Output**
    - Default Value: "haplotypecaller.vcf"
    - Description: Specifies the output file where the variants detected by HaplotypeCaller will be written. This file is typically in VCF format.

**Optional Arguments for HaplotypeCaller**

- **HaplotypeCaller Activity Profile Out**
    - Default Value: ""
    - Description: Specifies a file to output the raw activity profile results, which can be useful for debugging and visualizing the regions where HaplotypeCaller is performing variant calling.
- **HaplotypeCaller Alleles**
    - Default Value: ""
    - Description: Defines a set of alleles to force-call, regardless of the evidence. This is useful in targeted resequencing studies or when verifying known variants.
- **HaplotypeCaller Annotate With Num Discovered Alleles**
    - Default Value: False
    - Description: If enabled, this option annotates records with the number of alternate alleles discovered at a given site, which may not necessarily be genotyped.
- **HaplotypeCaller Annotation**
    - Default Value: ""

    - Description: Specifies one or more specific annotations to add to the variant calls, providing additional information about each variant.
- **HaplotypeCaller Annotation Group**
    - Default Value: ""
    - Description: Allows users to apply groups of annotations to the variant calls, facilitating streamlined and consistent annotation.
- **HaplotypeCaller Annotations To Exclude**
    - Default Value: ""
    - Description: Specifies annotations to exclude from the variant calls, useful for omitting unnecessary or irrelevant annotations.
- **HaplotypeCaller Assembly Region Out**
    - Default Value: ""
    - Description: Outputs the assembly region to a file in IGV format, providing a visual representation of the regions assembled during variant calling.
- **HaplotypeCaller Base Quality Score Threshold**
    - Default Value: 18
    - Description: Sets the minimum base quality score below which bases will be treated as having the minimum score (typically 6). This threshold helps to filter out low-quality bases from the analysis.
- **HaplotypeCaller Cloud Index Prefetch Buffer**
    - Default Value: -1
    - Description: Sets the size of the cloud-only prefetch buffer in megabytes. A value of 0 disables the buffer, optimizing data transfer for cloud-stored inputs.
- **HaplotypeCaller Cloud Prefetch Buffer**
    - Default Value: 40
    - Description: Specifies the size of the cloud prefetch buffer in megabytes, improving data access efficiency when working with cloud-stored data.
- **HaplotypeCaller Contamination Fraction To Filter**
    - Default Value: 0.0
    - Description: Sets the fraction of contamination in the sequencing data to aggressively filter out, crucial for accurate variant calling.
- **HaplotypeCaller Correct Overlapping Quality**
    - Default Value: False
    - Description: An undocumented option that may affect the handling of overlapping reads during variant calling.
- **HaplotypeCaller Dbsnp**
    - Default Value: ""
    - Description: Specifies a dbSNP file to annotate known variants, enhancing the output VCF with known reference SNP information.
- **HaplotypeCaller Disable Bam Index Caching**

    - Default Value: False
    - Description: Disables BAM index caching, reducing memory usage at the potential cost of slower performance, particularly when working with numerous intervals.
- **HaplotypeCaller Disable Sequence Dictionary Validation**
    - Default Value: False
    - Description: Skips validation of sequence dictionaries from inputs, which can be risky but may be necessary in specific workflows.
- **HaplotypeCaller Founder Id**
    - Default Value: ""
    - Description: Specifies samples representing the population "founders," relevant in pedigree and population studies.
- **HaplotypeCaller Gcs Max Retries**
    - Default Value: 20
    - Description: Sets the maximum number of retries for accessing data in Google Cloud Storage, ensuring robustness against transient errors.
- **HaplotypeCaller Gcs Project For Requester Pays**
    - Default Value: ""
    - Description: Specifies the billing project for accessing "requester pays" buckets in GCS, necessary for handling costs associated with data access.
- **HaplotypeCaller Graph Output**
    - Default Value: ""
    - Description: Outputs debug assembly graph information, useful for troubleshooting and understanding the variant calling process.
- **HaplotypeCaller Heterozygosity**
    - Default Value: 0.001
    - Description: Sets the heterozygosity value used to compute prior likelihoods for loci, influencing variant calling accuracy.
- **HaplotypeCaller Heterozygosity Stdev**
    - Default Value: 0.01
    - Description: Specifies the standard deviation of heterozygosity for SNP and indel calling, helping to model population genetics.
- **HaplotypeCaller Indel Heterozygosity**
    - Default Value: 1.25E-4
    - Description: Sets the heterozygosity for indel calling, crucial for accurately identifying insertions and deletions.
- **HaplotypeCaller Interval Merging Rule**
    - Default Value: "ALL"
    - Description: Determines how adjacent genomic intervals are merged, affecting the scope and granularity of the analysis.

- **HaplotypeCaller Intervals**
    - Default Value: ""
    - Description: Specifies genomic intervals to focus the analysis on specific regions of interest.
- **HaplotypeCaller Max Reads Per Alignment Start**
    - Default Value: 50
    - Description: Limits the number of reads per alignment start position, preventing excessive memory usage and improving processing speed.
- **HaplotypeCaller Min Base Quality Score**
    - Default Value: 10
    - Description: Sets the minimum base quality score required for a base to be considered in variant calling, ensuring data quality.
- **HaplotypeCaller Native Pair Hmm Threads**
    - Default Value: 4
    - Description: Specifies the number of threads for the native pair-HMM implementation, balancing speed and resource usage.
- **HaplotypeCaller Native Pair Hmm Use Double Precision**
    - Default Value: False
    - Description: Uses double precision in the native pair-HMM, providing greater accuracy at the cost of performance.
- **HaplotypeCaller Num Reference Samples If No Call**
    - Default Value: 0
    - Description: Specifies the number of hom-ref genotypes to infer at sites not present in a panel, relevant in population studies.
- **HaplotypeCaller Output Mode**
    - Default Value: "EMIT_VARIANTS_ONLY"
    - Description: Determines the types of calls output, typically "EMIT_VARIANTS_ONLY" for variant detection.
- **HaplotypeCaller Pedigree**
    - Default Value: ""
    - Description: A pedigree file defining relationships between samples, useful in family-based studies.
- **HaplotypeCaller Population Callset**
    - Default Value: ""
    - Description: Specifies a callset for calculating genotype priors, enhancing population-based analyses.
- **HaplotypeCaller Sample Name**
    - Default Value: ""
    - Description: Specifies a single sample to analyze from a multi-sample BAM file, useful for targeted analysis.

- **HaplotypeCaller Sample Ploidy**
    - Default Value: 2
    - Description: Sets the ploidy of the sample, which is the number of sets of chromosomes present.
- **HaplotypeCaller Sites Only Vcf Output**
    - Default Value: False
    - Description: If true, the output VCF will contain only site-level information, excluding genotype data.
- **HaplotypeCaller Stand Call Conf**
    - Default Value: 30.0
    - Description: The minimum confidence threshold for calling variants, influencing the stringency of variant detection.

---

**SelectVariants SNVs Extraction Section**

The SelectVariants tool from GATK is used to subset and filter VCF files based on various criteria. In the context of Single Nucleotide Variants (SNVs) extraction, it allows the user to select and output only the SNVs from a given VCF file, using a range of optional parameters to refine the selection.

**Required Arguments for SNV Extraction**

- **SNV Output**
    - Default Value: "snps.vcf"
    - Description: Specifies the output file where the selected SNVs will be written. The output is typically a VCF file containing only the variants that match the specified criteria.

**Optional Arguments for SNV Extraction**

- **SNV Cloud Index Prefetch Buffer**
    - Default Value: -1
    - Description: Sets the size of the cloud-only prefetch buffer in megabytes. A value of 0 disables the buffer, optimizing data transfer for cloud-stored inputs.
- **SNV Cloud Prefetch Buffer**
    - Default Value: 40
    - Description: Specifies the size of the cloud prefetch buffer in megabytes, enhancing data access efficiency when working with cloud-stored data.
- **SNV Concordance**
    - Default Value: ""

    - Description: Specifies a comparison track to output variants that are also called in this track, allowing for concordance checks.
- **SNV Disable Bam Index Caching**
    - Default Value: False
    - Description: Disables BAM index caching, reducing memory usage at the potential cost of slower performance, particularly when working with numerous intervals.
- **SNV Discordance**
    - Default Value: ""
    - Description: Outputs variants that are not called in the specified comparison track, useful for identifying discordant variants.
- **SNV Exclude Filtered**
    - Default Value: False
    - Description: When set to true, excludes filtered variants from the output.
- **SNV Exclude Ids**
    - Default Value: ""
    - Description: List of variant rsIDs to exclude from the output, allowing the removal of specific variants.
- **SNV Exclude Non Variants**
    - Default Value: False
    - Description: When set to true, excludes non-variant sites from the output, focusing only on variants.
- **SNV Exclude Sample Expressions**
    - Default Value: ""
    - Description: List of sample expressions to exclude, allowing the exclusion of specific samples based on expressions.
- **SNV Exclude Sample Name**
    - Default Value: ""
    - Description: Excludes genotypes from the specified sample, useful for removing unwanted samples.
- **SNV Gcs Max Retries**
    - Default Value: 20
    - Description: Sets the maximum number of retries for accessing data in Google Cloud Storage, ensuring robustness against transient errors.
- **SNV Interval Merging Rule**
    - Default Value: "ALL"
    - Description: Determines how adjacent genomic intervals are merged, affecting the scope and granularity of the analysis.
- **SNV Intervals**
    - Default Value: ""

    - Description: Specifies genomic intervals to focus the analysis on specific regions of interest.
- **SNV Invert Mendelian Violation**
    - Default Value: False
    - Description: When set to true, outputs only non-mendelian violation sites.
- **SNV Invert Select**
    - Default Value: False
    - Description: Inverts the selection criteria, selecting variants that do not meet the specified criteria.
- **SNV Keep Ids**
    - Default Value: ""
    - Description: List of variant rsIDs to include, ensuring specific variants are retained in the output.
- **SNV Keep Original Ac**
    - Default Value: False
    - Description: Retains the original allele counts (AC) after subsetting, useful for tracking original data.
- **SNV Keep Original Dp**
    - Default Value: False
    - Description: Retains the original depth of coverage (DP) after subsetting, useful for maintaining original data context.
- **SNV Max Filtered Genotypes**
    - Default Value: 2147483647
    - Description: Sets the maximum number of filtered genotypes allowed in the output, used for quality control.
- **SNV Max Fraction Filtered Genotypes**
    - Default Value: 1.0
    - Description: Sets the maximum fraction of filtered genotypes allowed, ensuring a minimum quality threshold.
- **SNV Max Indel Size**
    - Default Value: 2147483647
    - Description: Specifies the maximum size of indels to include, filtering out large indels if desired.
- **SNV Max Nocall Fraction**
    - Default Value: 1.0
    - Description: Sets the maximum fraction of samples with no-call genotypes, controlling the quality of the output data.
- **SNV Max Nocall Number**
    - Default Value: 2147483647

    - Description: Specifies the maximum number of samples with no-call genotypes, ensuring the output data meets quality criteria.
- **SNV Mendelian Violation**
    - Default Value: False
    - Description: Outputs only mendelian violation sites, useful for pedigree analysis and quality control.
- **SNV Mendelian Violation Qual Threshold**
    - Default Value: 0.0
    - Description: Sets the minimum quality score for accepting a site as a violation, filtering low-quality data.
- **SNV Min Filtered Genotypes**
    - Default Value: 0
    - Description: Specifies the minimum number of filtered genotypes required, used for quality control.
- **SNV Min Fraction Filtered Genotypes**
    - Default Value: 0.0
    - Description: Sets the minimum fraction of filtered genotypes required, ensuring a baseline quality threshold.
- **SNV Min Indel Size**
    - Default Value: 0
    - Description: Specifies the minimum size of indels to include, useful for focusing on specific indel sizes.
- **SNV Pedigree**
    - Default Value: ""
    - Description: Specifies a pedigree file for analyzing family-based data, useful for detecting inherited variants.
- **SNV Preserve Alleles**
    - Default Value: False
    - Description: When set to true, preserves original alleles, ensuring no data is lost during subsetting.
- **SNV Remove Fraction Genotypes**
    - Default Value: 0.0
    - Description: Randomly sets a fraction of genotypes to no-call, useful for simulations or quality control.
- **SNV Remove Unused Alternates**
    - Default Value: False
    - Description: Removes alternate alleles not present in any genotypes, simplifying the output data.
- **SNV Restrict Alleles To**
    - Default Value: "ALL"

    - Description: Selects only variants of a particular allelicity, allowing for focused analysis.
- **SNV Sample Expressions**
    - Default Value: ""
    - Description: Specifies regular expressions to select multiple samples, useful for complex sample selection criteria.
- **SNV Sample Name**
    - Default Value: ""
    - Description: Includes genotypes from the specified sample, useful for targeted analysis.
- **SNV Select Random Fraction**
    - Default Value: 0.0
    - Description: Selects a random fraction of variants from the input, useful for subsampling data.
- **SNV Select Type To Exclude**
    - Default Value: ""
    - Description: Excludes specific types of variants from the input, such as indels or structural variants.
- **SNV Select Type To Include**
    - Default Value: "SNP"
    - Description: Includes only specific types of variants from the input, typically SNPs for SNV extraction.
- **SNV Select Expressions**
    - Default Value: ""
    - Description: Specifies one or more criteria to use when selecting the data, allowing for custom filtering.
- **SNV Set Filtered Gt To Nocall**
    - Default Value: False
    - Description: Sets filtered genotypes to no-call, ensuring the output data does not contain low-quality genotypes.

For detailed information and usage instructions regarding the SelectVariants tool, please refer to the official GATK documentation. This resource provides comprehensive guidance on configuration and the various options available.

---

**SelectVariants Indel Extraction Section**

Similar to the SNVs extraction section, the Indel Extraction section allows users to subset and filter VCF files to output only the indels. The tool offers a range of options for fine-tuning the selection criteria.

**Required Arguments for Indel Extraction**

- **Indel Output**
    - Default Value: "indel.vcf"
    - Description: Specifies the output file where the selected indels will be written, typically in VCF format.

**Optional Arguments for Indel Extraction**

- **Indel Cloud Index Prefetch Buffer**
    - Default Value: -1
    - Description: Sets the size of the cloud-only prefetch buffer in megabytes. A value of 0 disables the buffer, optimizing data transfer for cloud-stored inputs.
- **Indel Cloud Prefetch Buffer**
    - Default Value: 40
    - Description: Specifies the size of the cloud prefetch buffer in megabytes, enhancing data access efficiency when working with cloud-stored data.
- **Indel Concordance**
    - Default Value: ""
    - Description: Specifies a comparison track to output variants that are also called in this track, allowing for concordance checks.
- **Indel Disable Bam Index Caching**
    - Default Value: False
    - Description: Disables BAM index caching, reducing memory usage at the potential cost of slower performance, particularly when working with numerous intervals.
- **Indel Discordance**
    - Default Value: ""
    - Description: Outputs variants that are not called in the specified comparison track, useful for identifying discordant variants.
- **Indel Exclude Filtered**
    - Default Value: False
    - Description: When set to true, excludes filtered variants from the output.
- **Indel Exclude Ids**
    - Default Value: ""

    - Description: List of variant rsIDs to exclude from the output, allowing the removal of specific variants.
- **Indel Exclude Non Variants**
    - Default Value: False
    - Description: When set to true, excludes non-variant sites from the output, focusing only on variants.
- **Indel Exclude Sample Expressions**
    - Default Value: ""
    - Description: List of sample expressions to exclude, allowing the exclusion of specific samples based on expressions.
- **Indel Exclude Sample Name**
    - Default Value: ""
    - Description: Excludes genotypes from the specified sample, useful for removing unwanted samples.
- **Indel Gcs Max Retries**
    - Default Value: 20
    - Description: Sets the maximum number of retries for accessing data in Google Cloud Storage, ensuring robustness against transient errors.
- **Indel Interval Merging Rule**
    - Default Value: "ALL"
    - Description: Determines how adjacent genomic intervals are merged, affecting the scope and granularity of the analysis.
- **Indel Intervals**
    - Default Value: ""
    - Description: Specifies genomic intervals to focus the analysis on specific regions of interest.
- **Indel Invert Mendelian Violation**
    - Default Value: False
    - Description: When set to true, outputs only non-mendelian violation sites.
- **Indel Invert Select**
    - Default Value: False
    - Description: Inverts the selection criteria, selecting variants that do not meet the specified criteria.
- **Indel Keep Ids**
    - Default Value: ""
    - Description: List of variant rsIDs to include, ensuring specific variants are retained in the output.
- **Indel Keep Original Ac**
    - Default Value: False

    - Description: Retains the original allele counts (AC) after subsetting, useful for tracking original data.
- **Indel Keep Original Dp**
    - Default Value: False
    - Description: Retains the original depth of coverage (DP) after subsetting, useful for maintaining original data context.
- **Indel Max Filtered Genotypes**
    - Default Value: 2147483647
    - Description: Sets the maximum number of filtered genotypes allowed in the output, used for quality control.
- **Indel Max Fraction Filtered Genotypes**
    - Default Value: 1.0
    - Description: Sets the maximum fraction of filtered genotypes allowed, ensuring a minimum quality threshold.
- **Indel Max Indel Size**
    - Default Value: 2147483647
    - Description: Specifies the maximum size of indels to include, filtering out large indels if desired.
- **Indel Max Nocall Fraction**
    - Default Value: 1.0
    - Description: Sets the maximum fraction of samples with no-call genotypes, controlling the quality of the output data.
- **Indel Max Nocall Number**
    - Default Value: 2147483647
    - Description: Specifies the maximum number of samples with no-call genotypes, ensuring the output data meets quality criteria.
- **Indel Mendelian Violation**
    - Default Value: False
    - Description: Outputs only mendelian violation sites, useful for pedigree analysis and quality control.
- **Indel Mendelian Violation Qual Threshold**
    - Default Value: 0.0
    - Description: Sets the minimum quality score for accepting a site as a violation, filtering low-quality data.
- **Indel Min Filtered Genotypes**
    - Default Value: 0
    - Description: Specifies the minimum number of filtered genotypes required, used for quality control.
- **Indel Min Fraction Filtered Genotypes**
    - Default Value: 0.0

    - Description: Sets the minimum fraction of filtered genotypes required, ensuring a baseline quality threshold.
- **Indel Min Indel Size**
    - Default Value: 0
    - Description: Specifies the minimum size of indels to include, useful for focusing on specific indel sizes.
- **Indel Pedigree**
    - Default Value: ""
    - Description: Specifies a pedigree file for analyzing family-based data, useful for detecting inherited variants.
- **Indel Preserve Alleles**
    - Default Value: False
    - Description: When set to true, preserves original alleles, ensuring no data is lost during subsetting.
- **Indel Remove Fraction Genotypes**
    - Default Value: 0.0
    - Description: Randomly sets a fraction of genotypes to no-call, useful for simulations or quality control.
- **Indel Remove Unused Alternates**
    - Default Value: False
    - Description: Removes alternate alleles not present in any genotypes, simplifying the output data.
- **Indel Restrict Alleles To**
    - Default Value: "ALL"
    - Description: Selects only variants of a particular allelicity, allowing for focused analysis.
- **Indel Sample Expressions**
    - Default Value: ""
    - Description: Specifies regular expressions to select multiple samples, useful for complex sample selection criteria.
- **Indel Sample Name**
    - Default Value: ""
    - Description: Includes genotypes from the specified sample, useful for targeted analysis.
- **Indel Select Random Fraction**
    - Default Value: 0.0
    - Description: Selects a random fraction of variants from the input, useful for subsampling data.
- **Indel Select Type To Exclude**
    - Default Value: ""

    - Description: Excludes specific types of variants from the input, such as SNPs or structural variants.
- **Indel Select Type To Include**
    - Default Value: "INDEL"
    - Description: Includes only specific types of variants from the input, typically indels for indel extraction.
- **Indel Select Expressions**
    - Default Value: ""
    - Description: Specifies one or more criteria to use when selecting the data, allowing for custom filtering.
- **Indel Set Filtered Gt To Nocall**
    - Default Value: False
    - Description: Sets filtered genotypes to no-call, ensuring the output data does not contain low-quality genotypes.

For detailed information and usage instructions regarding the SelectVariants tool, please refer to the official GATK documentation. This resource provides comprehensive guidance on configuration and the various options available.

---

**CollectVariantCallingMetrics (Picard) Section**

The CollectVariantCallingMetrics tool from Picard calculates metrics for variant calls, helping to assess the quality of the variant calling process. It provides metrics related to variant calls in VCF files, such as the number of SNPs, indels, and their distribution.

**Required Arguments for CollectVariantCallingMetrics**

- **Collectvariantcallingmetrics Output**
    - Default Value: "variant_metrics"
    - Description: Specifies the path (excluding the file extension) for the output metrics files. The tool will generate multiple metrics files with different extensions, detailing various aspects of the variant calls.
- **Collectvariantcallingmetrics Dbsnp**
    - Default Value: ""
    - Description: Path to the reference dbSNP file in dbSNP or VCF format. This file is used to annotate the variants in the input file and categorize them as known or novel.

**Optional Arguments for CollectVariantCallingMetrics**

- **Collectvariantcallingmetrics Gvcf Input**
    - Default Value: False
    - Description: Indicates whether the input is a single-sample GVCF file. When set to true, the tool will process the file accordingly.
- **Collectvariantcallingmetrics Sequence Dictionary**
    - Default Value: ""
    - Description: Specifies the sequence dictionary file. If provided, this can speed up the loading of the dbSNP file. If not present, the tool will look for a dictionary in the VCF file.
- **Collectvariantcallingmetrics Target Intervals**
    - Default Value: ""
    - Description: Specifies target intervals to restrict the analysis to specific regions of interest, such as exons or specific chromosomes.
- **Collectvariantcallingmetrics Thread Count**
    - Default Value: 10
    - Description: Sets the number of threads to use for processing. Increasing the thread count can speed up the analysis by parallelizing the work across multiple cores.

**Common Optional Arguments**

- **COMPRESSION_LEVEL**
    - Default Value: 5
    - Description: Sets the compression level for any compressed files created, such as BAM or VCF files. The level can be adjusted to balance file size and compression time.
- **CREATE_INDEX**
    - Default Value: False
    - Description: Specifies whether to create a BAM index when writing a coordinate-sorted BAM file. Indexing allows for faster random access to specific regions of the BAM file.
- **CREATE_MD5_FILE**
    - Default Value: False
    - Description: Indicates whether to create an MD5 digest for any BAM or FASTQ files created. An MD5 digest can be used to verify file integrity.

For detailed information and usage instructions regarding the CollectVariantCallingMetrics (Picard) tool, please refer to the official GATK documentation. This resource provides comprehensive guidance on configuration and the various options available.

---

## ANNOVAR Section

The ANNOVAR tool is used for functional annotation of genetic variants. It annotates variants based on their potential impact, such as predicting whether a mutation is benign or pathogenic, identifying known variants, and locating mutations in regulatory regions.

### Required Arguments for ANNOVAR

- **Database Directory**
  - Default Value: ""
  - Description: Specifies the directory containing the ANNOVAR databases. These databases are required for annotation and must be downloaded separately.
- **Genome Build Version**
  - Default Value: "hg38"
  - Description: Specifies the genome build version to be used for annotation. Common build versions include "hg19" and "hg38" for human genomes. The build version must match the version used in the reference genome and variant calling steps.
- **Output Prefix**
  - Default Value: "myanno"
  - Description: Specifies the prefix for output files generated by ANNOVAR. The output files will contain the annotation results and will use this prefix in their filenames.

### Optional Arguments for ANNOVAR

- **Protocol**
  - Default Value: "refGene,cytoBand,exac03,avsnp147,dbnsfp30a"
  - Description: Specifies the list of databases or protocols to use for annotation. Each protocol corresponds to a specific type of annotation, such as gene-based, region-based, or filter-based annotations.
- **Operation**
  - Default Value: "g,r,f,f,f"
  - Description: Defines the type of operation for each protocol specified. The operations can include gene-based ("g"), region-based ("r"), and filter-based ("f") annotations. The order of operations must match the order of protocols.
- **Nastring**
  - Default Value: "."
  - Description: Specifies the string to use for missing values in the output. This string will replace missing or unavailable data in the annotation results.
- **No Polish**
  - Default Value: False

    - Description: If set to True, ANNOVAR will not "polish" the output files, meaning it will not remove unnecessary spaces or standardize formatting. This option is useful for debugging or when exact output formatting is required.
- **Remove Intermediate Files**
    - Default Value: False
    - Description: If set to True, intermediate files generated during the annotation process will be removed after the final output is produced. This option helps conserve disk space.

For detailed information and usage instructions regarding ANNOVAR, please refer to the official ANNOVAR documentation. This resource provides comprehensive guidance on installation, configuration, and the various annotation protocols available.

## ProgramPage

The ProgramPage serves as the central hub for configuring the locations of essential external tools used in the pipeline, such as FASTQC, GATK, BWA, and ANNOVAR. It provides a user-friendly interface for setting and managing these paths, ensuring that the pipeline can efficiently access the necessary software for various genomic analyses.

**Set Program Locations for External Tools**

1. **FASTQC Location**:
    - Use the "Browse" button next to "FASTQC Location" to select the path to the FASTQC executable file. The selected path will be displayed in the entry field.
2. **GATK Location**:
    - Click the "Browse" button next to "GATK Location" to navigate and select the GATK executable. The path will be shown in the corresponding entry field.
3. **BWA Location**:
    - Similar to other tools, use the "Browse" button next to "BWA Location" to find and select the BWA executable. The path will appear in the entry field.
4. **ANNOVAR Location**:
    - The "Browse" button next to "ANNOVAR Location" allows you to select the directory where ANNOVAR is installed. The directory path will be displayed in the entry field.

**Description of the "Save as Default" and "Load Default" Features**

- **Save as Default**:
    - This feature allows you to save the currently set paths for the external tools as the default configuration.
    - When you click the "Save as Default" button, the program saves the paths to a configuration file (external_program_location.json). A message will confirm that the configuration has been successfully saved.
- **Load Default**:
    - Use the "Load Default" button to load the previously saved default paths from the configuration file.
    - This is useful if you have a standard setup you frequently use and want to quickly apply it without manually entering each path.

These functionalities ensure that your tool paths are easily manageable and can be quickly reset to a known configuration, enhancing the efficiency and ease of use of the pipeline setup process.

# 5.0 Pipeline Workflow

**Initial Setup and Configuration:**

- Install necessary toolkits and programs including Python, OpenJDK, FastQC, GATK, Samtools, BWA, and ANNOVAR.
- Set up a directory for raw sequence files (FASTQ format) and reference genome (FASTA format).
- Configure the pipeline with parameters and paths for the tools and input data.

**Quality Control:**

- **Tool:** FastQC
- **Process:** Assess the quality of raw sequencing data.
- **Outcome:** Generates metrics such as quality scores, GC content distribution, and sequence length distribution to identify any irregularities that could impact downstream analyses.

**Preprocessing:**

- **Tool:** Picard (FastqToSam, MarkIlluminaAdapters, SamToFastq)
- **Process:**
    - Convert raw FASTQ sequences to BAM format, adding read group information.
    - Mark adapter sequences in BAM files to avoid misinterpretation.
    - Convert BAM files back to FASTQ format for alignment.
- **Outcome:** Generates a BAM file with marked adapters, ready for alignment.

**Sequence Alignment:**

- **Tool:** BWA (BWA-MEM), Picard (MergeBamAlignment, CreateSequenceDictionary, SortSam), Samtools
- **Process:**
    - Index the reference genome.
    - Align sequencing reads to the reference genome using BWA-MEM.
    - Merge alignment data with unmapped BAM files.
    - Sort and prepare BAM files for downstream analysis.
- **Outcome:** A merged and sorted BAM file containing aligned reads.

**Post-Alignment Processing:**

- **Tool:** Picard (SetNmMdAndUqTags, MarkDuplicates, SortSam), GATK (BaseRecalibrator, ApplyBQSR)

- **Process:**
    - Calculate and add crucial tags (NM, MD, UQ) to BAM files.
    - Mark duplicate reads to reduce redundancy.
    - Recalibrate base quality scores using BaseRecalibrator and ApplyBQSR.
- **Outcome:** A high-quality, duplicate-marked, and recalibrated BAM file.

**Variant Calling:**

- **Tool:** GATK (HaplotypeCaller, SelectVariants)
- **Process:**
    - Call variants (SNVs and Indels) from the recalibrated BAM file.
    - Extract specific variant types (SNVs and Indels) for further analysis.
- **Outcome:** VCF files containing detailed information about detected variants.

**Quality Evaluation:**

- **Tool:** Picard (CollectVariantCallingMetrics)
- **Process:** Collect metrics to evaluate the quality of variant calls, including comparison with known variant databases.
- **Outcome:** Detailed metrics and statistics that assess the accuracy and quality of the variant calls.

**Variant Annotation:**

- **Tool:** ANNOVAR
- **Process:** Annotate variants with additional information such as gene names, predicted effects, allele frequencies, and clinical significance.
- **Outcome:** Annotated VCF files that provide comprehensive insights into the biological and clinical relevance of the identified variants.

Each output file serves as a valuable resource for interpreting the sequencing data, assessing the quality of the experiments, and understanding the biological implications of the identified variants. Proper interpretation of these files requires familiarity with genomic data analysis and the specific tools used.

# 6.0 Troubleshooting

## Common Issues

1. **Input File Format Errors**:
    - *Issue*: The pipeline does not recognize the input files.
    - *Solution*: Ensure that the input files are in the correct format (FASTQ for sequencing reads and FASTA for the reference genome). Verify that the files are properly named and located in the specified input directory.
2. **Insufficient Resources**:
    - *Issue*: The pipeline crashes or runs very slowly.
    - *Solution*: Check that your system meets the minimum hardware requirements, including RAM and disk space. Consider increasing swap space if memory is insufficient. Ensure that your system has enough free disk space for temporary files and output data.
3. **Tool Path Configuration**:
    - *Issue*: The pipeline cannot find external tools (e.g., GATK, BWA).
    - *Solution*: Verify that the paths to external tools are correctly set in the ProgramPage. Ensure that the specified paths are accurate and point to the correct executable files.
4. **Missing Required Arguments**:
    - *Issue*: The pipeline does not start or produces errors due to missing required arguments.
    - *Solution*: Ensure all required fields are filled in the ConfigurationPage, such as input files, output directories, and reference files.
5. **Known Site and Reference Sequence Mismatch**:
    - *Issue*: The known sites file does not match the reference sequence being used.
    - *Solution*: Ensure that the known sites file (e.g., dbSNP, 1000 Genomes) corresponds to the same genome build as the reference sequence. Using known sites from a different build can result in errors or incorrect variant calling.
6. **Invalid Directory Names**:
    - *Issue*: The directory name contains spaces or special characters, causing issues with file retrieval and output.
    - *Solution*: Ensure that directory names used in the pipeline do not contain spaces or special characters. Use underscores (_) or hyphens (-) instead of spaces, and avoid characters like &, %, $, etc. Incorrect directory names can prevent the pipeline from correctly accessing or saving files.

## Logs and Debugging

**Accessing Logs**: All logs and error messages are recorded in the "log" folder, located within the same directory as the pipeline. This folder contains detailed logs of each step, including command executions, outputs, and error messages.

**Interpreting Logs**: Logs provide information on the progress and status of each step in the pipeline. They can be used to identify where the pipeline encountered issues and provide details about what went wrong. Look for lines marked as "ERROR" or "WARNING" to quickly find potential problems.

**Important Note**: The log files in the "log" folder will be overwritten each time a new sequence assembly is run. If you need to retain logs from a specific analysis, please save them to another directory before starting a new run. This ensures that you have a permanent record of the logs for future reference or troubleshooting. You can copy the log files to a designated backup location or rename them to include relevant details about the analysis they correspond to.