1. Issue with Installing Java and Trimmomatic

When I first tried to use Trimmomatic, I faced an issue where it was unable to locate a Java Runtime, even though I had installed Java using Homebrew. This meant I couldn't run Trimmomatic for trimming my sequencing reads.

Solution: To resolve this, I had to set up the Java environment variables properly. I added the Java path to my `.bash_profile` and reloaded the shell using `source ~/.bash_profile`. Once the environment variables were set correctly, I was able to run Trimmomatic without any issues.

---

2. Trouble with Running BWA

While running the BWA alignment, I initially faced errors related to the reference genome and input files. The alignment didn't run because I had missed ensuring that my reference genome and data files were properly linked.

Solution: I double-checked the paths for both the reference genome and the input fastq files. I corrected the path to the reference genome and re-ran the BWA alignment using the `bwa mem` command. Once I ensured that everything was in the correct format and location, the alignment completed successfully.

---

3. Problems with BAM File Conversion and Sorting

After alignment, I tried converting the `.sam` file to `.bam` using `samtools view`, but the process was taking a long time and sometimes would crash due to the size of the files.

Solution: I tackled this by using a subset of the data. Initially, I used the `samtools view` command on a specific genomic region instead of the entire genome. This reduced the file size significantly, making it more manageable. Once I processed smaller chunks of the data, I was able to continue with the rest of the steps.

---

4. Trouble with Variant Calling

During the variant calling step, I used `samtools mpileup` and `bcftools call` but ran into an issue where some of the variants were missing, or the output was incorrect.

Solution: After reviewing the command parameters, I found that the reference genome had not been indexed properly. I re-indexed the reference genome using `samtools faidx` and re-ran the variant calling step. This solved the issue, and I was able to successfully identify variants in the genomic region.

---

5. Issues with Large Data Files

Throughout the project, I was working with large sequencing files (e.g., `.fastq.gz`, `.bam`). These files were too large to upload to GitHub directly.

Solution: I decided to use Git LFS (Large File Storage) for handling large files. I installed Git LFS, tracked the large files with `git lfs track`, and added them to the repository. This allowed me to store large files in a separate storage while keeping the main repository clean.

---

6. Data Visualization Problems

When trying to visualize my results in IGV (Integrative Genomics Viewer), I faced issues where the files weren't displaying correctly, or there were discrepancies between the variants shown in the VCF file and the alignment data.

Solution: The problem was due to the mismatch between the reference genome version used in the alignment and the one used in the visualization tool. I downloaded the correct reference genome and ensured it was the same version as the one used for the alignment. After reloading the data into IGV, the visualization worked as expected.

---

7. Understanding the Bioinformatics Terminology

I often found it challenging to understand the bioinformatics terms and commands used during the analysis. Terms like "SNP," "INDEL," "SAM," and "BAM" were initially unclear, and I wasn’t sure how they fit into the workflow.

Solution: To address this, I took the time to research each term and its relevance to the analysis. I read tutorials and documentation for each tool (e.g., BWA, Samtools, Bcftools) and watched videos explaining sequencing, alignment, and variant calling. I also reached out to peers and instructors for clarification on certain steps.

Conclusion

Despite the challenges I faced, I was able to successfully complete the genome assembly and variant calling project. I learned a lot about bioinformatics tools and how to handle large datasets. The project not only enhanced my understanding of sequence analysis but also taught me how to troubleshoot common issues that arise when working with complex biological data.

By breaking down each step and tackling problems systematically, I was able to navigate through the difficulties and build a solid understanding of genome assembly, variant calling, and bioinformatics workflows.