Task 1: Data Handling and Statistical Analysis

1. Coverage Analysis:

a. Calculate the median and coefficient of variation (CV) for single CpG coverage in each tissue.

Step 1: Prepare the Data: Ensure your CSV file has two essential columns:

Tissue: The tissue type for each CpG site (cancer, normal).

Coverage: The coverage (number of reads) at each CpG site.

Step 2: Load the CSV Data: We will use pandas to load the CSV file and manipulate the data.

Step 3: Calculate the Median Coverage for Each Tissue:

We can use the groupby method in pandas to calculate the median coverage for each tissue.

Step 4: Calculate the Coefficient of Variation (CV) for Each Tissue:

The CV is calculated as the ratio of the standard deviation to the mean coverage, multiplied by 100 to express it as a percentage.

Step 5: Combine Results: You can combine the median and CV calculations into a single DataFrame for easier comparison.

Step 6: Visualize the Results

You can plot the Median Coverage and CV for each tissue using matplotlib or seaborn.

Plotting Boxplots: If you want to visualize the distribution of coverage values for each tissue type, you can use a boxplot.

b. Generate plots summarizing the coverage statistics.

1. Histogram: Displays the distribution of coverage values.

2. Boxplot: Compares coverage distribution across tissues.

3. Violin plot: Visualizes both distribution and density for each tissue.

4. Scatter plot: Visualizes the relationship between coverage and other variables.

5. Density plot (KDE): Provides a smoothed distribution of coverage values.

2. Biomarker Identification:

a. Identify PMPs with high specificity for tissue differentiation, minimizing false positives for Tissue #1 while allowing some false negatives. Use statistical or machine learning approaches to assign confidence (e.g., p-values) to each PMP.

1. Data Preparation: You need a dataset that contains expression levels for different PMPs across Tissue #1 and other tissues.

2. Preprocessing and Normalization: Normalization: Normalize the expression levels of PMPs across tissues to ensure comparability. This can be done using Z-score normalization or min-max scaling.

Log Transformation: If necessary, apply a log transformation to stabilize the variance of expression data.

3. Statistical Analysis: Identifying Differentially Expressed PMPs

t-test: Use a two-sample t-test to identify PMPs that are differentially expressed between Tissue #1 and other tissues. The t-test will help assess whether the mean expression of a PMP differs significantly between tissues.

Null hypothesis (H0): There is no significant difference in PMP expression between Tissue #1 and the other tissues.

Alternative hypothesis (Ha): There is a significant difference in PMP expression between Tissue #1 and other tissues.

You will get p-values for each PMP. A lower p-value indicates a higher likelihood that the PMP is differentially expressed between the tissues.

Multiple Testing Correction: Use Benjamini-Hochberg correction (FDR) to control for multiple comparisons and adjust the p-values.

4. Machine Learning: Classifier for Tissue Differentiation

To identify PMPs that best differentiate Tissue #1 from other tissues, use a machine learning classifier like Random Forest, Support Vector Machine (SVM), or Logistic Regression. This classifier will be trained on the expression data to distinguish Tissue #1 from other tissues.

Random Forest Classifier: A Random Forest classifier can be used to predict whether a given sample belongs to Tissue #1 or not based on the PMP expression data.

Feature Importance: After training the model, you can extract the feature importance to identify which PMPs are most important for differentiating Tissue #1 from other tissues.

Probability Scores: In addition to predictions, the Random Forest classifier also provides probability scores. These scores can be used as confidence scores for each PMP, indicating the likelihood of it being specific to Tissue #1.

5. Optimize Specificity and Minimize False Positives

To minimize false positives (incorrectly identifying a PMP as specific to Tissue #1), adjust the decision threshold. A higher threshold will make the classifier more conservative (i.e., only identify a PMP as specific to Tissue #1 if the probability score is very high).

6. Assign Confidence to Each PMP

Use the p-values from the statistical tests and the probability scores from the machine learning classifier to assign a confidence score to each PMP.

For PMPs with high p-values (i.e., no significant difference), discard them or assign a low confidence score.

For PMPs with high probability scores (from machine learning) and low p-values, assign a high confidence score.

7. Final List of PMPs

After applying the statistical and machine learning approaches, compile a list of PMPs with high specificity for Tissue #1. Include both the p-values and confidence scores for each PMP.

8. Evaluate Performance

Use performance metrics like ROC curves, AUC (Area Under the Curve), and confusion matrices to evaluate how well your model is distinguishing between Tissue #1 and other tissues.

b. Calculate the mean variant read fraction (VRF) for each PMP in both tissues.

Step 1: Prepare the Data: Assume you have a CSV file that contains the read counts for variants and total reads across different PMPs and Tissues.

Step 2: Load the Data and Calculate VRF: You can use pandas to load the data and perform the calculations.

Step 3: Calculate Mean VRF for Each PMP in Both Tissues: Now, we can group the data by PMP and calculate the mean VRF for each PMP across both tissues.

Step 4: Optional - Visualize the Mean VRF: You can plot the mean VRF for each PMP to visually compare the values.

3. Address the following questions:

a. How does sequencing depth affect specificity confidence?.

1. **Greater Sequencing Depth Boosts Variant Call Confidence:** More Independent Observations Support the Variant Calls Deeper sequencing (more reads per sample) increases the evidence for variants. Because a variety that is real will be seen more than once, this lowers the possibility of false positives. Decreased Random Errors: Since the existence of a

variation will be verified by several reads, whereas errors are likely to happen in a small number of reads, sequencing faults (such base-calling errors) are less likely to impact the precision of variant calls.

2. **Lower Sequencing Depth Decreases Confidence in Specificity:** Increased False Positives: Random mistakes are more likely to result in false positives when sequencing depth is low since there are fewer reads supporting a variant. This is particularly true for low-frequency variations, which are mutations that occur in a tiny percentage of cells. Unable to Find Seldom Occurring Variants: Variants that are present at low frequencies may not be detected by low depth, which could lead to false negatives where genuine variants are overlooked.

3. **Variant Allele Fraction (VAF) Impact:** Low VAF Detection: Low sequencing depth makes it more difficult to find variants with low allele frequency, or a small percentage of all reads. Higher sequencing depth is required, for instance, to reliably detect a mutation in a cancer sample where it is present in only 5% of the reads and differentiate it from sequencing errors. Trust in Seldom Occurring Variants: Because there are more reads to back the call, the confidence in identifying low-frequency mutations rises with sequencing depth. With more sequencing depth, a mutation found in a tiny percentage of reads will be more likely to be confirmed.

4. **Effect on Statistical Confidence and p-value:** Statistical Confidence: Because the variation is detected in more reads, deeper sequencing strengthens the evidence for its existence and increases the statistical confidence (p-value) for detecting a true variant. False Positive Rate: The false discovery rate (FDR) is reduced and specificity is raised when the number of random sequencing errors falls with increased sequencing depth.

5. **Cutoffs for Thresholding and Confidence:** Depth-Dependent Thresholds: Usually, thresholds are used to specify the bare minimum of supporting readings needed to invoke a variant (a variant is only invoked if it has at least five reads supporting it). The specificity can be increased by setting these thresholds higher without missing true variants thanks to deeper sequencing. Heterozygous Call Confidence: Higher sequencing depth guarantees that the allelic balance—the ratio of reference to variant allele—is more reliably detected for heterozygous variations (one copy of the mutant allele), boosting confidence in the variant call.

b. For the top 10 PMPs, estimate the threshold of reads required to confidently call Tissue #2 at a sequencing depth of 1 million reads.

1. **The Variant Allele Fraction (VAF) is defined as follows:** The percentage of readings that contain a specific variant is known as the Variant Allele Fraction, or VAF. For instance, the VAF is 0.001 (or 0.1%) if 1000 out of 1,000,000 reads support a variant. In a normal tissue sample, a heterozygous variant's VAF may be approximately 50%, whereas a somatic mutation in cancer may have a substantially lower VAF (5%).

2. **Calculate the Fewest Reads Needed to Identify a Variant with Confidence:** The coverage depth and the VAF set the threshold for calling a variant with confidence. We usually need at least 10–20 readings supporting the variant in order to reduce false positives and call the variant with confidence. However, the estimated VAF, background noise (sequencing mistakes), and desired confidence level (often at least 95% confidence) can all affect how many reads are needed.

3. **Determine the Threshold Using VAF:** The number of reads needed to reliably identify the variant with a given VAF at a certain sequencing depth establishes the threshold.

• Suppose the following: One million reads is the sequencing depth.

• We wish to estimate the number of variation-supporting readings required for a given VAF in order to call the variant with confidence. For instance, the estimated number of reads supporting a variant with a VAF of 5% at a sequencing depth of one million reads would be as follows:

$$\text{Variant Read Count} = \text{VAF} \times \text{SequencingDepth}$$

$$\text{Variant Read Count} = 0.05 \times 1{,}000{,}000 = 50{,}000 \text{ reads supporting the variant}$$

$$\text{Variant Read Count} = 0.10 \times 1{,}000{,}000 = 100{,}000 \text{ reads supporting the variant}$$

4. **Calculate Confidence and Variability:** To achieve 95% confidence in detecting a variant, we want to ensure that the variant read count is significantly higher than any potential background noise (sequencing errors, etc.). A typical method to calculate the confidence in variant detection is using binomial statistics. The number of variant-supporting reads must

be significantly higher than the expected number of reference allele reads (background noise). The threshold read count required for confident variant calling typically varies depending on the false positive rate you are willing to tolerate.

5. **Think about the False Positive Rate:** Setting a minimal threshold for the quantity of supporting reads can help reduce the false positive rate in the context of variant calling. The probability of a false positive falls as the number of reads supporting a variant rises. For instance, 5–10 supportive reads per million reads at a 5–10% VAF is a typical threshold used to identify low-frequency variants (such as uncommon mutations or somatic mutations in cancer).

6. **Calculate the Read Threshold:** Using the information provided, you can calculate the threshold using the reasoning below data.

c. Validate the hypothesis by comparing the specificity of the top 10 PMPs against individual CpG sites.(10 points).

1. **Gather Information for PMPs and CpG Sites:** For every PMP and CpG site, you should have information.

2. **Determine Specificity for Every CpG and PMP Site:** Determine specificity by counting the number of true positive variant calls and false positives found using the information given. To do this, the sequencing data must be aligned with a reference genome, known mutations must be annotated, and the variant calls must be compared.

3. **Statistical Comparison:** Use the following statistical test to compare the specificity of PMPs and CpG sites: To compare the specificity distributions of PMPs and CpG sites, use a t-test or a Mann-Whitney U test. Given their focused nature and stronger data backing, PMPs should have a better specificity if the hypothesis is correct.

4. **Visualise the Results:** It is easier to comprehend the differences between PMPs and CpG sites when the specificity for each is visualised. A violin plot or boxplot can be used to show the specificity distributions.

5. **Analyse the findings:** A low p-value (e.g., < 0.05) from the t-test indicates that PMPs have a significantly higher specificity than CpG sites. The notion is further supported if PMPs with higher specificity values (less variance) are displayed in the boxplot.

Task 2: NGS Data Analysis

1. Perform quality checks using tools fastqc and sumarized quality matrics (sequence count, per base quality, read duplication level)

1. Install FastQC: If your system does not already have FastQC installed, do so now. It is available for download on the official FastQC website.

```
sudo apt-get install fastqc
```

2. Get Your Data Ready: Make sure the sequencing data you have is in FASTQ format. The most used format for raw sequencing data files is this one. Each sample (single-end or paired-end) should have one or more.fastq or.fq files.

3. Apply FastQC to the Data: Apply FastQC to the data to produce quality reports. If you have many samples, you can run it on multiple files simultaneously.

```
fastqc sample1.fastq sample2.fastq -o fastqc_reports/
```

4. Examine the output documents
• Following completion, FastQC creates a.zip file with extra files (fastqc_data.txt) and an HTML report.
• To locate the fastqc_reports/ directory, navigate to the

5. Provide an overview of the quality metrics.
• FastQC offers a number of important quality measurements, such as:
• Sequence Count: The FASTQ file's total number of sequences.The distribution of base quality scores for each base in the sequence is known as per-base quality.
• Read Duplication Level: The proportion of readings that are duplicated.
While the fastqc_data.txt file has more in-depth statistics, the HTML report offers a graphical summary.

6. Look at the number of sequences
• To make sure you have adequate data for further analysis, look for the total number of sequences in the HTML report's "Basic Statistics" section.

7. Per-Base Quality Analysis

• The HTML report's "Per Base Sequence Quality" section displays a graph of quality scores (Phred scores) for each point in the reads.

• For the majority of the bases, especially at the conclusion of the reads, look for a constant quality score (usually ≥30, or a Phred value of ≥Q30).

8. Check Read Duplication Level

• The proportion of duplicate reads in the dataset will be displayed in the "Duplicate Reads" column.
• Excessive duplication rates (>20%) could indicate problems like biassed PCR amplification or troublesome sequencing.

9. If applicable, do FastQC on paired-end data.

• FastQC can handle R1 and R2 files independently if your data is paired-end. To guarantee the quality of both ends of the reads, run FastQC on both files. -o fastqc_reports/ fastqc sample_R1.fastq sample_R2.fastq

```
fastqc sample_R1.fastq sample_R2.fastq -o fastqc_reports/
```

10. Examine the Summary Reports from FastQC
• Following FastQC, you can evaluate the general quality of the data:
◦ Sequence Count: Make sure there are enough sequences for a useful analysis.
◦ Per-Base Quality: Verify that the reads' quality scores are consistently high (Q30 or higher).
◦ Read Duplication Level: While a certain amount of duplication (10–15%) is typical, lower levels are desired. Think about removing or filtering your data or taking care of the underlying sequencing problems if any of these indicators are troublesome.
Extra Information:
• FastQC Flags: For further control, you can use FastQC with particular flags, like -t for multiple threads (parallel processing) or -f for file format specification. **MultiQC**: If you have multiple FastQC reports for different samples, you can use **MultiQC** to summarize the results in one consolidated report.

```
multiqc fastqc_reports/
```

2. Align the samples to the human genome using tools like bwa.

1. Set up the required software on your computer, including Samtools and BWA. Sequence alignment and processing are common uses for these techniques.
sudo apt-get install bwa samtools

2. Acquire the Reference Genome
• A repository such as Ensembl or UCSC may offer the human reference genome (GRCh38) for download.
Wget ftp://ftp.ensembl.org/pub/release-104/fasta/homo_sapiens/dna/Homo_sapiens.GRCh38.dna.primary_assembly.fa.gz
gunzip Homo_sapiens.GRCh38.dna.primary_assembly.fa.gz

3. To get the reference genome ready for alignment, index it using BWA.
• The BWA index GRC38.dna.primary_assembly.fa Homo sapiens

4. Get Your FastQ Files Ready
• You should have two files for paired-end reads: one for reverse readings (R2) and one for forward reads (R1).

5. Use BWA to align the sequence. Both paired-end readings should be aligned with the reference genome.

bwa mem Homo_sapiens.GRCh38.dna.primary_assembly.fa sample_R1.fastq sample_R2.fastq > aligned_reads.sam

6. SAM to BAM Format Conversion
• Convert the SAM file to the more processing and storage-efficient BAM format after alignment.

samtools view -Sb aligned_reads.sam > aligned_reads.bam

7. Arrange the BAM Document
To facilitate further analysis, arrange the BAM file according to genomic coordinates.

samtools sort aligned_reads.bam -o aligned_reads_sorted.bam

8. Eliminate Repeated Readings
• To eliminate duplicate reads that might have been added during library preparation, use Samtools or Picard's MarkDuplicates.
picard MarkDuplicates I=aligned_reads_sorted.bam O=aligned_reads_no_duplicates.bam M=metrics.txt REMOVE_DUPLICATES=true

9. Index the BAM File After Sorting
• For quick access during variant calling and other downstream analysis, create an index for the sorted BAM file.

samtools index aligned_reads_no_duplicates.bam
10. Check for Alignment
• Verify the process's proper operation by looking at the alignment quality and statistics. You can use Samtools flagstat or Qualimap for this:
samtools flagstat aligned_reads_no_duplicates.bam

b. Identify somatic mutations present in the cancer sample but absent in the normal tissue.

• Identify Mutations Present in Cancer but Absent in Normal Tissue
The final step is to identify somatic mutations that are present in the cancer sample but absent in the normal tissue. This can be done by comparing the variant calls between the cancer and normal sample using tools like bcftools or bedtools.

bcftools isec -n-1 -c all cancer_vs_normal.vcf normal_marked.vcf -p isec_output

• The mutations present only in the cancer file and not in the normal tissue file will be found in the isec_output directory.

Interpret and Visualize the Results
Visualize the somatic mutations using tools like IGV (Integrative Genomics Viewer) to ensure that the identified mutations are correctly aligned in the context of the genome.
Validate the results by comparing with known somatic mutation databases (COSMIC, dbSNP) and using additional filtering techniques to improve the accuracy of mutation calling.

i. Benchmark Software: Use established tools such as Mutect2 for somatic mutation identification and background mutation estimation. (10 points)

- Prepare Data: Ensure you have aligned BAM files for both cancer and normal tissue samples, ideally preprocessed with duplicate marking and base quality score recalibration (BQSR).
- Run Mutect2: Use Mutect2 (from GATK) to identify somatic mutations by comparing the cancer and normal BAM files:

  gatk Mutect2 -R reference.fasta -I cancer.bam -I normal.bam --tumor-sample cancer --normal-sample normal --output somatic_mutations.vcf

- Filter Results: Apply filters to remove false positives based on quality scores, depth of coverage, and allele frequency:

  gatk FilterMutectCalls -V somatic_mutations.vcf -O filtered_mutations.vcf

- Estimate Background Mutation Level: Use Mutect2's background model to estimate sequencing errors and biases. This is often included in the output as artifact detection and is important for distinguishing true somatic mutations.

- Interpret the Results: Review the VCF file for somatic mutations, ensuring proper annotation and validation against known mutation databases.

ii. Custom Code Development: Write your own scripts, leveraging tools like Samtools, bcftools, or Python/R libraries, to perform mutation detection and calculate the required metrics. (15 points)

Note: Scripts add in github repository.

c. Use the normal tissue to calculate the median background mutation level. The background mutation level accounts for sequencing errors or biases that can mimic true mutations. Determine how many reads per million are required to confidently call a given mutation.

1. Obtain Normal Tissue Data
• Use the normal tissue sequencing data as a reference for determining the background mutation level. Ensure the data is aligned and processed into a variant call format (VCF) file after variant calling (using GATK).

2. Preprocess the Data
• Align the normal tissue sequencing data to the human reference genome (GRCh38).
• Call the variants using a variant calling tool (GATK HaplotypeCaller).
• Filter out known germline variants and sequencing errors by using quality filters on the VCF file (QUAL score, DP (depth), and GQ (genotype quality)).
3. Identify Background Mutations
• The background mutation level refers to sequencing errors or biases that appear as false positive mutations in the normal tissue data. These should be identified by counting low-frequency variants that are not consistent across biological replicates or known somatic mutations.
• Focus on single nucleotide variants (SNVs) or small insertions/deletions (indels) with low allele frequency (<5%) and not supported by sufficient read depth (<20 reads).
4. Calculate the Median Background Mutation Level
• Once you have filtered and identified the potential sequencing errors or background mutations, calculate the mutation density. This involves counting the number of variants in the normal tissue VCF file and calculating the frequency of these variants across the genome.
grep -v "^#" normal_tissue.vcf | wc -l # Count the number of variants in the VCF file

Then, compute the mutation density as:

Mutation Density= Number of Variants / Total Number of Bases Covered

• After obtaining the density, the median background mutation level can be estimated by computing the median variant frequency across the genome.

5. Calculate Reads Per Million (RPM)
• To determine how many reads per million are required to confidently call a given mutation, you need to calculate the required depth of coverage for confident mutation calling.

• RPM calculation involves determining the number of mutations identified per million reads. The confidence threshold for calling a mutation can be assessed based on the following formula:

RPM for Mutation Confidence = Number of Mutations Detected / Total Reads × 10^6

• Typically, for high-confidence mutation calling, a minimum coverage of 30x is recommended, with at least 20-30 reads supporting a mutation at a particular locus.

Example Calculation:
1. Count the number of variants detected in the normal tissue:
Let's say there are 500 variants in the normal tissue VCF file.

2. Calculate the total number of reads used for variant calling (50 million reads).

3. Calculate the mutation density and background mutation level:

Mutation density = 500 variants / (50 million reads) = 0.00001 mutations per read.

4. Calculate the Reads per Million (RPM):

RPM for normal tissue = (500 mutations / 50,000,000 reads) * 1,000,000 = 10 RPM.

5. Confident Mutation Call Threshold:

To confidently call mutations, typically >= 10 RPM is required, depending on the sequencing technology and error rate.