Breast cancer prediction

Step 1: Data Acquisition

- **Objective:** Obtain raw NGS data for both healthy individuals and breast cancer patients.
- **Action Items:**
  - Use the **NCBI SRA database** to download datasets.
  - Install and use the **SRA Toolkit** (e.g., using fastq-dump) to convert SRA files to FASTQ format.

Step 2: Quality Control & Preprocessing

- **Objective:** Ensure high-quality sequencing data.
- **Tools:**
  - **FastQC**: For quality assessment of FASTQ files.
  - **Trimmomatic** (or **Cutadapt**): For adapter removal and trimming low-quality bases.
- **Action Items:**
  - Run FastQC on raw FASTQ files.
  - Trim adapters and low-quality ends using Trimmomatic.
- **Outcome:** Clean, high-quality reads ready for alignment.

Step 3: Sequence Alignment

- **Objective:** Align the cleaned reads to the human reference genome.
- **Tools:**
  - **BWA** or **Bowtie2**: Both are free and widely used.
  - **SAMtools**: To convert SAM files to BAM and perform sorting/indexing.
- **Action Items:**
  - Align reads against the human reference genome (e.g., GRCh38) to generate SAM files.
  - Convert SAM to BAM, sort, and index using SAMtools.

Step 4: Variant Calling

- **Objective:** Identify genetic variants (SNPs and Indels) from the aligned data.
- **Tools:**
  - **SAMtools** and **BCFtools**
- **Action Items:**
  - Use SAMtools/BCFtools to call variants from the BAM files.
  - Generate VCF files for each sample containing potential mutations.

Step 5: Variant Annotation

- **Objective:** Determine the functional impact of identified variants.

- **Tools:**
    - **SnpEff** or **ANNOVAR** (free versions available)
- **Action Items:**
    - Annotate the VCF files to identify variants causing protein alterations (non-synonymous changes, frameshifts, etc.).
    - Focus on mutations in genes known to be associated with breast cancer (e.g., BRCA1, BRCA2).

Step 6: Feature Extraction for Machine Learning

- **Objective:** Create a structured dataset for machine learning.
- **Action Items:**
    - For each sample, generate features such as:
        - **Presence/absence** of specific mutations.
        - **Mutation counts** in important genes.
        - **Functional impact scores** from annotation.
    - Label the samples as "Healthy (0)" or "Breast Cancer (1)".
    - Organize the data in a tabular format (e.g., CSV file) with rows as samples and columns as features.

Step 7: Machine Learning Model Development

- **Objective:** Build a predictive model to classify samples.
- **Tools:**
    - **Python** with libraries such as **pandas**, **scikit-learn**, and **matplotlib/seaborn** for visualization.
- **Action Items:**
    - **Data Loading & Preprocessing:** Import the dataset, handle missing values, and scale features if needed.
    - **Train/Test Split:** Divide data into training and testing sets.
    - **Model Training:** Use a classifier (e.g., Logistic Regression, Random Forest) to train on the data.
    - **Model Evaluation:** Assess the model using metrics like accuracy, precision, recall, and F1-score.
    - **Feature Importance:** Identify which variants/features contribute most to the prediction.

Step 8: Results Analysis and Visualization

- **Action Items:**
    - Visualize the performance metrics using plots (e.g., ROC curves, confusion matrices).
    - Use feature importance (for tree-based models) or model coefficients (for logistic regression) to interpret which genetic variants are most indicative of breast cancer.

Step 9: Conclusion & Future Directions

- **Action Items:**

- Summarize your findings, discussing how certain genetic alterations correlate with breast cancer.
- Address limitations (e.g., dataset size, noise) and propose future work (e.g., testing on independent datasets, integrating additional omics data).

Key Points to Remember

- **Toolchain:** All tools (SRA Toolkit, FastQC, Trimmomatic, BWA/Bowtie2, SAMtools, BCFtools, SnpEff/ANNOVAR, scikit-learn) are free and widely used in bioinformatics.
- **Complexity:** This pipeline is designed to be straightforward yet comprehensive for a final year project.
- **Reproducibility:** Document every step, and consider using environments (e.g., Conda) and version control (e.g., Git) to maintain reproducibility.