Step-by-step workflow

1. Database Setup

* Connects to PostgreSQL using the DB\_CONFIG
* Creates a table gene\_variants if it doesn’t exist. This table holds:
* Variant identifiers
* Gene info
* HGVS cDNA/protein notations
* Clinical significance
* Allele info
* ACMG criteria results

1. Data Download & Filtering

* Downloads variant\_summary.txt.gz from NCBI
* Keeps only TP53 variants from GRCh38, plus header row.

1. Data Cleaning & Parsing

* Loads the filtered file into a Pandas DataFrame
* Normalizes column names
* Extracts HGVS\_c (DNA change) and HGVS\_p (protein change) using regex
* Calculates protein\_pos (numeric protein position) from HGVS\_p
* Filters out variants missing protein\_pos

1. Clinical Significance Simplification

* Converts verbose clinicalsignificance values into a simplified form:

“pathogenic”, “likely pathogenic”, “vus”, “likely benign”, “benign”, “conflicting”, etc.

* Detects variants with conflicting interpretations (conflictinginterpretations column)

1. Variant Consequences

* Classifies DNA and protein changes into categories like:
* “missense\_variant”
* “nonsense”
* “frameshift\_variant”
* “synonymous\_variant”
* “deletion”, “insertion”, “duplication”, etc.
* Combines them into a general variant\_type.

1. ACMG Support Criteria

* Builds support groups for comparing variants with:
* Same cDNA change (PP5/BP6)
* Same protein change but different DNA change (PS1)
* Same protein position but different amino acid (PM5)
* Applies ACMG rules:
* PS1 – same protein effect, different DNA change, known pathogenic
* PM5 - different missense change at same position as known pathogenic
* PP5/BP6- multiple submitters agree variant is pathogenic or benign
* Combines results from:
* Group-based method (mark\_acmg\_criteria)
* Direct dataset comparisons (apply\_ps1\_pm5\_pp5\_bp6)

1. Filtering for Reliable Variants

* Keeps only “reliable” variants:
* Reviewed by expert panel or
* At least 3 submitters in categories 2 or 3 of SubmitterCategories with no conflicts

1. Database Insertion

* Inserts (or updates) each variant into gene\_variants using ON CONFLICT on variation\_id

1. Cleanup

* Deletes temporary files
* Closes database connection

Main Purpose:

This script automatically retrieves TP53 ClinVar data, processes and classifies variants with ACMG support criteria and loads the results into a structured PostgreSQL database for downstream analysis