**Powerpoints notes full** *ελπίζω..*

Introduction

*Slide 1*

This project focused on the application of ACMG criteria to genetic variant classification, with a specific emphasis on the TP53 gene.

*Slide 2*

The ACMG 2015 criteria, published by the American College of Medical Genetics and Genomics together with the Association for Molecular Pathology, provide standards and guidelines for interpreting sequence variants.

These guidelines define five tiers of classification: pathogenic, likely pathogenic, uncertain significance, likely benign and benign.

In total, 28 criteria are applied – 16 supporting pathogenicity and 12 supporting benign classification.

The project centered around applying a subset of these rules to real-world data from the ClinVar database.

*Slide 3*

In this project, we applied four specific ACMG support rules.

* PS1, which identifies variants that result in the same amino acid change as an already established pathogenic variant
* PM5, which applies when a novel missense change occurs at a residue where another missense pathogenic variant has already been observed
* PP5, which considers a reputable external source reporting a variant as pathogenic
* and BP6, which is the benign counterpart, when a reputable reports a variant as benign.

These criteria were selected because they could be implemented computationally and applied systematically to variant data.

In summary, the ACMG 2015 criteria provide a comprehensive, evidence-based framework to guide clinical laboratories and genetics professionals in interpreting the clinical significance of genetic variants, facilitating standardized decision-making to support patient care and research.

Due to the huge amount of data that exists in in ClinVar, we focused mainly on the variants of only one gene, TP53. TP53 is an extremely important gene because it encodes the p53 protein, a critical tumor suppressor often called the “guardian of the genome”. p53 regulates key cellular processes such as cell cycle arrest, DNA repair, apoptosis (programmed cell death), senescence and metabolism to maintain genetic stability and prevent cancer development. Mutations in TP53 are found in about 50% of human cancers and lead to loss of this protective function, allowing uncontrolled cell proliferation and tumor progression. Additionally, many TP53 mutations give the protein oncogenic gain-of-function properties that further promote malignancy, metastasis and resistance to chemotherapy and radiation. Because of its central role in guarding against cancer, TP53 mutations are one of the most common consequential genetic alterations in cancer biology, making it a major focus for research and potential targeted therapies.

*Slide 4*

The main goal was to design and implement a script that could automatically download, process and classify TP53 variants from ClinVar according to the ACMG support rules.

The script retrieves the ClinVar dataset, filters for TP53 variants under the GRCh38 reference genome, processes and extracts relevant genomic and protein change information and finally applies ACMG classification rules based on:

1. known and trusted data
2. grouping of similar variants

The results are stored in a structured PostgreSQL database, named clinvar\_db, enabling downstream analysis and querying.

The advantage that comes with such evaluation is that it gives the possibility for a new variant, which does not have a previous ClinVar entry, to be pathogenic.

*Slide 5*

In order to reach the point of evaluation and classification of a variant based on data that already exists in ClinVar, it is needed a suitable background.

The workflow followed a step-by-step approach. We began with database setup and the download of ClinVar data. Next, we applied filtering and data cleaning, followed by clinical significance simplification and annotation of variant consequences. The ACMG support criteria were then systematically applied, filtering for reliable variants. Finally, the processed data was inserted into the PostgreSQL database, followed by a clean-up step to maintain efficiency.

Python was used as programming language and PostgreSQL was implemented for the creation of the database.

*Slide 6*

This slide illustrates the logical workflow of the main function in the script. The function coordinates downloading, parsing, filtering and classification, ensuring each step produces structured, usable data.

By modularizing the process, the system is easier to maintain, extend and integrate into further analyses.

Code structure:

1. Settings

* Definition of ClinVar file’s URL
* PostgreSQL connection details

1. Helper functions:

* Create tables in the database (create\_tables)
* Extract HGVS from variant description (extract\_HGVS)
* Calculate amino acid position (extract\_protein\_pos)
* Classification of molecular consequence in DNA/ protein (consequence, consequence\_dna, combine\_consequence)

1. ClinVar data processing:

* Process\_clinvar\_data:
* Use of zcat + grep for filtering of entries (e.g. TP53 and GRCh38)
* Cleaning and normalization of column names
* HGVS, amino acid position *(protein\_pos)*, number of submitters extraction
* Filtering only on entries with valid *protein\_pos*

1. ACMG criteria evaluation:

2 strategies are used:

* Group-based: Grouping of variants based on same:
* cDNA change (PP5/BP6)
* amino acid change, but different cDNA change (PS1)
* amino acid position, but different amino acid change (PM5)
* Direct matching with raw data: Compares the current entry with other in the DataFrame

1. Clinical significance simplification
2. Filtering for trusted entries:

* get\_reliable\_variation\_ids\_from\_variant\_summary:

1. Entries with review status by Expert Panel
2. Entries with more than 3 submitters belonging in categories 2 or 3 of Submitter Categories, with no conflicts
3. Insert to database

Output data

In gene\_variants database are saved:

1. Main variant details (HGVS, gene, position, consequence)
2. Clinical significance simplified
3. ACMG criteria:

* acmg\_criteria: list with grouping and direct algorithms
* acmg\_from\_grouping: only from grouping
* acmg\_combined\_criteria: final combined result

1. Indications of conflicting interpretations
2. RCV references
3. Metadata (latest update etc.)

*Slide 7*

To build this workflow, we relied on a range of Python libraries for data processing, parsing and database integration.

These tools ensured that the data pipeline was both reproducible and scalable, which is critical for bioinformatics applications where large datasets are frequently updated.

*Slide 8*

Beyond classification and database storage, the project also included the development of a FastAPI interface.

This API allows users to query and analyze the stored ClinVar TP53 data. Through the API, it is possible to: check system health, retrieve specific variant details, perform ACMG rule classification, search by protein position or molecular consequence type and aggregate classification summaries. This provides both flexibility and accessibility for researchers.

The script implements a REST API with FastAPI that:

1. Connects with the PostgreSQL database clinvar\_db and draws variant data from the gene\_variants table
2. Provides endpoints for searching, filtering and counting variants based on gene, position, molecular consequences and pathogenicity
3. Implements the ACMG 2015 logic to evaluate variants (PS1, PM5, PP5, BP6)
4. Provides statistics summary (e.g. counts per consequence/ significance)
5. Supports advanced queries (range queries, multifilter search)

Technologies used:

1. FastAPI: REST API for Python (high speed, auto docs)
2. PostgreSQL: Database for storing variants
3. psycopg2: PostgreSQL connector for Python
4. pandas: processing of data (mainly for grouping/ criteria evaluation)
5. re (Regex): Extraction of amino acid positions from HGVS\_p strings
6. custom module (*apply\_ps1\_pm5\_pp5\_bp6*): implementation of ACMG criteria logic

Code structure:

1. Settings:

* DB\_CONFIG: PostgreSQL connection details
* Libraries insertion (fastapi, psycopg2, pandas, re, collections)
* Custom module insertion: *apply\_ps1\_pm5\_pp5\_bp6*

1. Health endpoint:

* GET /: health check to check whether the API is active

Returns status JSON if the API is running

1. Search variants endpoints:

* GET /user\_classify\_variant: Returns either direct summary of the variation or calculates ACMG criteria (PS1, PM5, PP5, BP6)
* GET /variants\_by\_genomic\_range: Searches variants based on genomic coordinates (assembly version should be specified)
* GET /variant\_by\_protein\_pos: Searches variants in a specific amino acid position or a range of amino acid positions
* GET /search\_variants: Complex search with multiple filters

1. Helper functions endpoints:

* GET /variant\_counts: Returns the number of variants based on the consequence, significance or position
* GET /summary: grouping by molecular consequence
* GET /significance\_summary: grouping by clinical significance
* GET /available\_genes: returns a list of all the genes in the database
* GET /available\_consequences: List of available types of consequences

1. ACMG Support:

* calculate\_pp5\_bp6\_from\_summary: Calculate PP5/BP6 based on the consistency of submitters
* GET /acmg\_criteria\_bp6\_pp5: Combination of gene variants with ACMG criteria (contains PS1/PM5/PP5/BP6 through pandas DataFrame grouping)

Output data:

The API returns JSON with:

1. Variant information: variant\_id, gene\_symbol, hgvs\_c, hgvs\_p, protein\_pos, molecular\_consequence, clinicalsignificance, review\_status
2. Summary: counts per consequence/ significance
3. ACMG criteria: number of conflicting/ active criteria
4. Available filters: list with genes and consequnces

*Slide 9*

This slide highlights example queries that can be made through the API. By enabling structured queries, users can efficiently retrieve and analyze genetics variant data without needing to directly interact with the raw database. This ensures that the pipeline is accessible not only to bioinformaticians, but also to clinicians and researchers who may have less technical expertise.

*Slide 11*

Important characteristics of API:

1. Durability: all endpoints have try/except/finally blocks for safe closing of the connection
2. Extension potential: The ACMG criteria are implemented modularly (*apply\_ps1\_pm5\_pp5\_bp6*), so new criteria can be added easily
3. Presentation ease: results are in JSON format, ready to be used in dashboards or other applications

*Slide 12*

The results confirm that the script and database structure successfully classified TP53 variants using ACMG rules. By automating the process, we created a reproducible system that can serve as a foundation for future variant analysis projects.

*Slide 17*

In summary, the code creates a full API for searching of variants that exist in ClinVar with:

1. Support of deep searches (gene, cDNA, protein, genomic ranges)
2. Helper tools (counts, summaries)
3. Implementation of ACMG criteria in order to assist the interpretation of variants
4. Extension potential for extra criteria or filtering

In conclusion, this project demonstrated the feasibility of automating ACMG classification for TP53 variants and storing the results in a structured database. For future extensions, the same framework can be applied to additional genes of clinical interest. It can also be linked with other databases for further confirmation, expanded with additional query tables and extended by implementing more ACMG criteria such as PS1. This work provides a solid foundation for scaling bioinformatics variant interpretation guidelines.