# Project Spotlight: Variant Reanalysis Case Study

Interview Coding Challenge

## Context

You are a scientist working in a clinical genomics laboratory specialising in the diagnosis of rare genetic diseases. Patients referred to your lab often remain unsolved after their initial genome or exome analysis. As new knowledge emerges—new gene–disease associations, improved variant annotations, and better prioritisation methods—**variant reanalysis** has become an essential part of clinical diagnostics.

Your lab is investigating how different computational approaches perform when reanalysing existing sequencing data from previously solved rare disease cases. The goal is to understand the strengths and limitations of different reanalysis strategies and to inform future diagnostic workflows.

This case study is inspired by real-world benchmarking work carried out in clinical and research genomics settings.

—

## Aim of the Case Study

The aim of this exercise is to evaluate and compare two variant prioritisation approaches using synthetic but realistic rare disease data.

Specifically, you will:

- Work with standard genomics-style tabular data
- Implement filtering and ranking logic inspired by real diagnostic pipelines
- Benchmark variant prioritisation performance using ranking-based metrics
- Communicate your analytical approach and findings clearly

This case study focuses on **data analysis, bioinformatics reasoning, and coding practices**, not on producing a single "correct" answer.

—

## Available Tools and Pipelines

You will benchmark two variant prioritisation approaches:

### Tool 1 (provided output)

Tool 1 is a machine-learning–based variant prioritisation tool that assigns a score and rank to individual variants.

In this case study:

- You are given example output as a ranked list of variants per case
- Each variant includes genomic coordinates, gene name, rank, and score

### Tool 2 (to be implemented by you)

Tool 2 is a rule-based reanalysis framework used in rare disease genomics. It prioritises variants using:

- Population allele frequency thresholds
- Predicted functional impact (e.g. VEP annotations)
- Pathogenicity annotations (e.g. ClinVar)
- External scores (e.g. AlphaMissense)
- Gene panels and phenotype context

In this exercise, you will implement a **simplified filtering and ranking approach** using provided configuration thresholds.

—

## Provided Data

All data are fully synthetic but designed to resemble real clinical genomics datasets.

- `variants.csv` Annotated variants per case, including:
  - Genomic position and gene
  - VEP-like consequence and impact
  - gnomAD allele frequency
  - AlphaMissense score
  - ClinVar annotation
- `truth_set.csv` One known causal variant per case (ground truth).
- `tool1_results.csv` Ranked variant output from aiDIVA.
- `cases_metadata.csv` Case-level metadata including inheritance mode and gene panel.
- `gene_panels.csv` Mapping of genes to diagnostic panels.
- `tool2_config.toml` Filtering thresholds and allowed annotations for the rule-based logic.

—

## Objectives

By completing this case study, you should demonstrate your ability to:

1. Load, clean, and integrate multiple genomics-style datasets in Python
2. Apply biologically motivated filtering criteria to variant data
3. Implement ranking-based evaluation metrics (e.g. Top-1, Top-5, Top-10 accuracy)
4. Compare two different variant prioritisation strategies in a fair and reproducible way
5. Interpret results and discuss limitations of the analysis

—

## Expected Output

You are expected to submit:

- A Python script or notebook containing your analysis
- Output tables and at least one figure comparing pipeline performance
- A short written summary (approximately $\frac{1}{2}$–1 page) describing:
  - Your approach
  - Key findings
  - One or two limitations or possible improvements

You will be asked to present and discuss your solution during the interview.
—

## Time Expectations

This exercise is designed to take approximately **3–5 hours**. You are not expected to optimise performance or implement every possible edge case.
Clarity, reasoning, and correctness are more important than completeness.
—

## Final Notes

There is no single "correct" solution. We are primarily interested in how you:

- Think about biological data
- Structure your analysis
- Make and justify design decisions

Good luck—we look forward to discussing your work with you.