OXFORD UNIVERSITY PRESS | Bioinformatics

Application of qualifying variants for genomic analysis

Journal:	*Bioinformatics*
Manuscript ID	BIOINF-2025-1188.R1
Manuscript type:	Applications Note
Date Submitted by the Author:	n/a
Complete List of Authors:	Lawless, Dylan; University Children's Hospital Zürich, University of Zürich, Department of Intensive Care and Neonatology Saadat, Ali; École polytechnique fédérale de Lausanne Oumelloul, Mariam Ait; École Polytechnique Fédérale de Lausanne Boutry, Simon; École polytechnique fédérale de Lausanne Stadler, Veronika; University Children's Hospital Zürich Oesterle, Sabine; SIB Swiss Institute of Bioinformatics Armida, Jan; SIB Swiss Institute of Bioinformatics Haerry, David; Positive Council Froese, D. Sean; University Children's Hospital Zürich, University of Zürich Schlapbach, Luregn J.; University Children's Hospital Zürich, University of Zürich, Department of Intensive Care and Neonatology Fellay, Jacques; Ecole Polytechnique Federale de Lausanne, School of Life Sciences
Portal Keywords:	Annotation, Genome analysis, Classification
Keywords:	qualifying, variants, genomic

October 21, 2025

Dear Reviewers,

We thank you sincerely for your thoughtful and constructive comments on our manuscript.

We agree fully with your suggestions and believe that the revisions have substantially improved the manuscript.

Reviewer: 1

Major comments:

1. *The reviewer requested a clear example showing how a clinician or analyst would use the QV set ID to retrieve the exact criteria applied, for audit, re-analysis, or regulatory review.*

Response: We added a concrete example in the Implications section. It now explains how a clinician using EPIC or another EHR system can access a patient's analysis results linked to their specific QV set ID, retrieving the corresponding YAML or JSON file from an institutional or DOI-linked repository. The example illustrates how a report referencing a QV ID confirms that ACMG v3.3 criteria, including BRCA1/2 breast cancer screening and associated thresholds, were applied, demonstrating practical traceability for audit.

2. *The reviewer requested broader validation across different datasets and use cases.*

Response: We have expanded the validation to three distinct use cases covering major analysis types and tools. In addition to (1) the existing in-house rare disease WES cohort (940 individuals), we now include (2) a GWAS using HapMap Phase 3 on 1,397 individuals and (3) a WGS trio analysis using the Genome in a Bottle reference from NIST with Exomiser. These are presented in the Methods/Results section and illustrated in Figures S1-S3.

3. *The reviewer requested benchmarking of computational overhead and scalability when using YAML-based QV files versus the traditional approach.*

Response: We added a dedicated computational benchmark section with a new figure. The supplemental result and Figure S4 shows that preprocessing times were effectively identical between traditional and YAML-based workflows, with a small median improvement in favour of the QV YAML approach. This confirms that YAML-based QV files introduce no computational overhead and scale equivalently to conventional implementations.

4. *The reviewer requested a clear example of how patient preferences and PPIE input are recorded in QV files and influence reporting.*

Response: We added explicit examples in "Example QV structure", showing patient context and PPIE review notes. In "FAIR mapping and patient involvement", we explain how such inputs can be collected through consent-linked forms or patient/public review groups within the same FAIR-compliant file. Broader implications are discussed in "Traceability and confirmation of applied

clinical standards".

Minor comments:

5. The reviewer noted minor typographical and grammatical errors. We have now carefully reviewed and corrected the full manuscript for grammar, spelling, and typographical consistency.

6. The reviewer noted that, in addition to Box 1, readers unfamiliar with YAML may benefit from a brief explanation of its syntax and structure. We have revised the Implementation section to include a concise description of the logic immediately before the example box. The updated text explains that QV files use simple key=value statements to define filters, criteria, and metadata, making the content clear even for readers without prior experience.

Reviewer: 2

1. *"The authors propose a framework to apply and combine different filtering sets to identify qualifying variants for genomic analysis. While this approach might be convenient, I think it lacks novelty as various analysis tools offer user-friendly filtering approaches (e.g. VarFish). Please find my major comments below. I would appreciate a more extensive description of the gap that this framework is filling."*

Response: We appreciate this comment and recognise that our earlier presentation may have led to a misunderstanding of where the novelty of our framework lies. We have clarified this point in the revised manuscript.

Tools such as VarFish are essential for variant interpretation, and our framework is designed to work alongside them rather than replace them. The QV framework serves a distinct role: it externalises key filtering rules and thresholds from pipelines into a simple, portable format that any analysis tool can use.

For example, one team may use VarFish, another GATK with Exomiser, and a third DeepVariant with VEP and SnpEff in Snakemake. Each can retain its preferred software stack, yet by sharing a single public QV file, all apply identical criteria. This ensures reproducibility and transparency without requiring shared implementations.

While sharing a Docker container or complete workflow allows full technical replication, it is not always practical when institutional or commercial environments must retain internal infrastructure. The QV framework achieves reproducibility at the level of analytic logic, making analyses transferable, auditable, and interoperable between systems.

2. *"The chapter regarding implementation lacks technical clarity, i.e. how is the framework implemented, which input does it require, which is the output, etc."*

Response: We have thoroughly revised the implementation section to describe these points explicitly, outlining how QV files are read, which parameters they define, and how they integrate into existing analysis pipelines.

3. *"The authors mention that their criteria are aligned with FAIR principles. Could they elaborate on this and justify this statement?"*

Response: We have added a dedicated section titled "FAIR mapping ..." describing the technical alignment with FAIR principles, including identifiers,

accessibility, interoperability, and reusability. The practical implications of FAIR compliance are further discussed in "Implications - Traceability and confirmation of applied clinical standards".

4. *"The validation study showcases the agreement with hard-coded criteria for one example data set. Could the authors add a more comprehensive comparison by extending the validation study to different data sets and use cases?"*

Response: We have expanded the validation to three distinct use cases covering major analysis types and tools. In addition to (1) the existing in-house rare disease WES cohort (940 individuals), we now include (2) a GWAS using HapMap Phase 3 on 1,397 subjects and (3) a WGS trio analysis using the Genome in a Bottle reference from NIST with Exomiser. These are presented in the Methods/Results section and illustrated in Figures S1-S3.

5. *"Can users specify their own filtering criteria for QVs and save them as a QV set? Can the authors showcase how to do that?"*

Response: Yes, users can fully define and save their own filtering criteria as QV sets. We have clarified this in the revised "Implementation" section, which now explains the process step by step, includes an example QV file structure (Box 1) with both technical and plain-language descriptions. We have also included an HTML-based QV builder that enables researchers and commercial users to create and export their own QV sets. It can be used as a standalone tool or easily embedded into existing websites.

We thank you again for your valuable feedback and hope you will agree that the manuscript has been strengthened considerably through your recommendations.

Yours sincerely,
On behalf of all authors,

Dylan Lawless, PhD

Application of qualifying variants for genomic analysis

Dylan Lawless*[1], Ali Saadat[2], Mariam Ait Oumelloul[2], Simon Boutry[2], Veronika Stadler[1], Sabine Österle[3], Jan Armida[3], David Haerry[4], D. Sean Froese[5], Luregn J. Schlapbach[1], and Jacques Fellay[2]

[1]Department of Intensive Care and Neonatology, University Children's Hospital Zürich, University of Zürich, Switzerland.
[2]Global Health Institute, School of Life Sciences, École Polytechnique Fédérale de Lausanne, Switzerland.
[3]SPHN Data Coordination Center, SIB Swiss Institute of Bioinformatics, Basel, Switzerland.
[4]Positive Council, Zürich, Switzerland.
[5]Division of Metabolism and Children's Research Center, University Children's Hospital Zürich, University of Zurich, Zurich, Switzerland.

October 22, 2025

*Addresses for correspondence: Dylan.Lawless@kispi.uzh.ch

Abstract

Motivation:

Qualifying variants (QVs) are genomic alterations selected by defined criteria within analysis pipelines. Although crucial for both research and clinical diagnostics, QVs are often seen as simple filters rather than dynamic elements that influence the entire workflow. In practice these rules are embedded within pipelines, which hinders transparency, audit, and reuse across tools. A unified, portable specification for QV criteria is needed.

Results:

Our aim is to embed the concept of a "QV" into the genomic analysis vernacular, moving beyond its treatment as a single filtering step. By decoupling QV criteria from pipeline variables and code, the framework enables clearer discussion, application, and reuse. It provides a flexible reference model for integrating QVs into analysis pipelines, improving reproducibility, interpretability, and interdisciplinary communication. Validation across diverse applications confirmed that QV based workflows match conventional methods while offering greater clarity and scalability.

Availability:

The source code and data are accessible at the Zenodo repository https://doi.org/10.5281/zenodo.17414191. Manuscript files are available at https://github.com/DylanLawless/qvApp2025lawless. The QV framework is available under the MIT licence, and the dataset will be maintained for at least two years following publication.

Acronyms

ACMG American College of Medical Genetics and Genomics 9

EHR Electronic Health Record 13

FAIR Findable, Accessible, Interoperable, and Reusable 7

GIAB Genome in a Bottle 12

GWAS Genome-Wide Association Study 4

HPO Human Phenotype Ontology 11

MAF Minor Allele Frequency 9

MD5 Message-Digest Algorithm 5 11

PCA Principal Component Analysis 11

PPIE Public and Patient Involvement and Engagement 8

PRS Polygenic Risk Score 4

QC Quality Control 4

QV Qualifying Variant 4

RDF Resource Description Framework 8

SNOMED CT Systematized Nomenclature of Medicine-Clinical Terms . . . 8

VCF Variant Call Format 4

VEP Variant Effect Predictor 9

WES Whole Exome Sequencing 9

WGS Whole Genome Sequencing 4

1 Introduction

Qualifying Variant (QV)s are genomic alterations selected by specific criteria within genome processing pipelines, serving as dynamic elements essential for both research and clinical diagnostics. QVs are not merely static filters applied at a single step in an analysis pipeline; rather, they are dynamic, multifaceted elements that permeate the entire workflow, from initial data quality control to final result interpretation. This nuanced perspective underscores that QVs play an integral role in shaping the fidelity and reproducibility of genomic analyses, enabling the iterative refinement of data and facilitating the integration of diverse analytical strategies throughout the pipeline.

Often, QV selection adheres to established variant classification and reporting standards (1–5) and standardised workflows (6–8). However, a unified framework for QVs is lacking, despite the recognised benefits of similar initiatives, such as Polygenic Risk Score (PRS) reporting standards (9; 10). Tools such as vcfexpress (11) enable flexible filtering and formatting of Variant Call Format (VCF) files using user-defined expressions. Treating QV criteria as an external parameter layer complements these tools by externalising their thresholds and logic. This approach improves reproducibility across distributed computing environments (12) and integrates seamlessly with workflow managers like Snakemake (13) or Nextflow (14).

QV selection criteria vary by application. In Genome-Wide Association Study (GWAS), thresholds favour common variants, yielding datasets with over 500,000 variants per subject, whereas rare disease analyses use stringent filters producing fewer than 1,000 variants, often limited to known genes or pathogenic loci. Although targeted filtering is valuable (15; 16), no unified approach exists. In practice, QV sets range from broad quality control filters to specific disease panels, and their definition is critical for reproducibility and accurate reporting, influencing results as much as the pipeline itself (17).

As Whole Genome Sequencing (WGS) becomes standard for large cohorts (18; 19), the integration of diverse QV protocols is critical for data cleaning and analysis. During sequencing analysis several layers can be responsible for triggering QV protocols, including pre-existing metadata, technical Quality Control (QC) results, and post-calling annotations, highlighting the need for a clear, unified approach.

We introduce the QV as a standalone entity, independent from other pipeline variables. Structured human- and machine-readable criteria, aligned with FAIR principles (20), facilitate integration across databases (21; 22). We advocate for the use

of standard vocabularies, unique identifiers, and flexible file formats to support this integration.

Building on this framework, we propose an openly documented registry model for QV files that assigns a unique `qv_set_id` and records a SHA-256 checksum for each release, enabling direct retrieval and verification for audit and re-analysis. Our accompanying HTML-based QV builder converts simple `key=value` statements into structured YAML and can be embedded in public, private, or commercial websites to simplify the authoring of consistent criteria (Zenodo repository). The framework is designed to support the emergence of a shared, widely adopted registry over time.

2 Methods

2.1 Implementation

The QV file provides a structured, human- and machine-readable definition of variant qualifying criteria. It is composed of five logical components that define its structure and metadata. It is portable across tools, transparent in content, and verifiable through unique identifiers and checksums. Each file is a lightweight YAML or JSON document specifying the variables and thresholds used in analysis. It can be read programmatically at runtime, for example using `yq` in shell-based workflows or `yaml::read_yaml()` in R, providing the same parameters that would otherwise be embedded within pipeline configurations, as illustrated in **Figure 1**. The output is identical to that of the native workflow, with the added benefit of an explicit, versioned, and shareable configuration file.

- **1. Meta**: Descriptive metadata including `qv_set_id`, title, version, author list, creation date, and tags. These fields ensure traceability and version control across analyses.
- **2. Filters**: Simple rule-based statements that apply inclusion or exclusion logic based on variable thresholds (for example, minimum allele frequency or coverage depth). Filters can also restrict the analysis to defined genomic regions, such as a target gene panel or BED file.
- **3. Criteria**: Compound logic blocks that combine one or more conditions into interpretable rules, corresponding to concepts such as ACMG criteria or study-specific thresholds.

- **4. Notes**: Optional free-text annotations providing context, assumptions, or technical caveats.
- **5. Descriptions (optional)**: Plain-language fields, such as `description_patient` and `description_ppie`, that can record patient preferences or public involvement input. These complement the technical definitions without affecting computational logic.

Example QV structure

We include an HTML-based QV builder that can be embedded in research or commercial platforms to simplify the creation of consistent, versioned criteria files (available via Zenodo repository). A minimal QV YAML file is shown in **Box 1** , equivalent to the configuration generated by this builder. QV files are composed of `key=value` statements, ensuring that all filtering and interpretation rules are explicit, versioned, and reproducible. In simple terms, **Box 1** specifies that only variants overlapping a curated disease gene panel are retained and that variants classified as pathogenic or likely pathogenic are prioritised. It also records patient context and patient-public involvement notes, thereby linking the technical filtering logic with its clinical and ethical rationale.

Box 1: qv_disease_panel_example.yaml

```
meta:
  qv_set_id: qv_disease_panel_v1_20250828
  version: 1.0.0
  title: Disease panel filter
filters:
  region_include:
    description: >
      Restrict to curated disease gene panel
    logic: keep_if
    field: OVERLAP(targets.disease_panel.bed)
    operator: '>='
    value: 1
criteria:
  pathogenic:
    description: >
      Variant classified as pathogenic or likely pathogenic
    logic: and
    conditions:
      - group: any_of:start
      - { field: CLASS, operator: '==', value: P }
      - { field: CLASS, operator: '==', value: LP }
      - group: any_of:end
meta:
  description_patient: >
    We have a strong family history of early heart attacks.
  description_ppie: >
    The PPIE group reviewed the criteria and approved them
    on 2025-08-15.
notes:
  - Gene panel file defines the target regions.
  - Additional quality filters may be added as needed.
```

FAIR mapping and patient involvement

Each QV file includes a persistent identifier (`qv_set_id`) that links criteria across analyses and databases. The framework aligns with the Findable, Accessible, Inter-

operable, and Reusable (FAIR) principles of findability, accessibility, interoperability, and reusability (20). Findability is achieved through unique identifiers; accessibility through open, human- and machine-readable YAML or JSON files; interoperability through standardised syntax (i.e. `key=value`) and semantic mappings such as Resource Description Framework (RDF) or Systematized Nomenclature of Medicine-Clinical Terms (SNOMED CT) (21; 22); and reusability through embedded metadata, checksum verification, and versioned registry records.

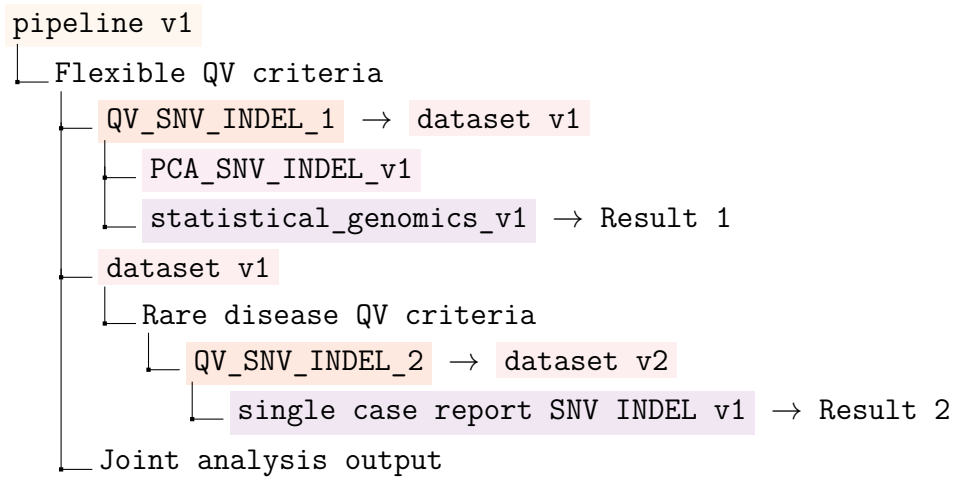
Optional metadata fields such as `description_patient` and `description_ppie` allow patient input and Public and Patient Involvement and Engagement (PPIE) feedback to be recorded in a manner appropriate to the study or application, with patient notes provided through consent-linked forms and PPIE groups offering structured review or approval of criteria within the same FAIR-compliant file.

Example QVs in WGS analysis

A typical WGS pipeline applies several QV sets sequentially, as the genetic cause of disease may stem from different variant types such as SNVs, CNVs, or structural variants. Each pass filters data for its purpose, producing both cohort-level and single-patient results within one reproducible framework (23; 24). As illustrated in **Figure 1**, the description can be written as:

"A cohort of patient WGS data was analysed to identify genetic determinants for phenotype X. A flexible QV set was applied using the `pipeline v1`, which implements the `QV_SNV_INDEL_1` criteria to produce the prepared dataset (`dataset v1`). This dataset was analysed alongside other modules (e.g. `PCA_SNV_INDEL_v1` and `statistical_genomics_v1`) to derive a cohort-level association signal (Result 1). It was then re-filtered with stricter `QV_SNV_INDEL_2` criteria to identify known causal variants, yielding (`dataset v2`) and single-patient reports (Result 2)."

Box 2: Example diagrammatic representation



Joint analysis output from:
Result 1 = Cohort-level association signal (e.g. variant P-value).
Result 2 = Single variant report per patient.

2.2 Usage in a rare disease cohort validation study

We validated the QV framework on an in-house rare disease cohort of 940 individuals using Whole Exome Sequencing (WES) comparing a conventional manual implementation with a QV-based YAML configuration. The analysis targeted rare variants (Minor Allele Frequency (MAF) < 0.01) in known disease genes from the Genomics England "Primary immunodeficiency or monogenic inflammatory bowel disease" panel, retrieved via PanelAppRex (25). This yielded 6,026 candidate variants annotated with 376 information sources, prepared in R using the GuRu variant interpretation tool and imported from gVCF files processed by Variant Effect Predictor (VEP).

We applied the first eight American College of Medical Genetics and Genomics (ACMG) criteria for pathogenicity scoring (1), six of which were relevant to this cohort. The manual pipeline encoded each criterion directly, while the QV workflow read the same definitions from a YAML file. The YAML criteria included `ACMG_PS1` (known pathogenic amino acid change), `ACMG_PS3` (supporting functional evidence), `ACMG_PS5` (compound heterozygosity), and frequency- and segregation-based criteria (`PM2`, `PM3`). Criteria `PS2` and `PS4` were not applicable in this cohort.

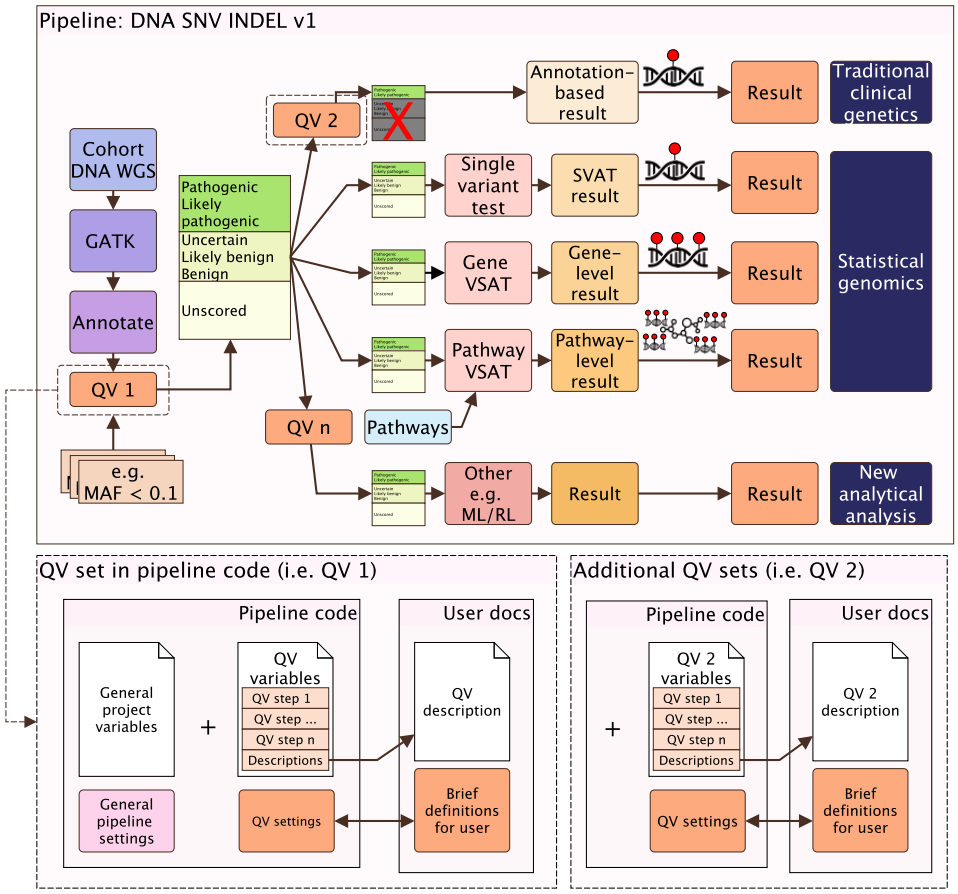


Figure 1: Summary of the QV application for a WGS pipeline. QV1 and QV2 are applied as sequential protocol steps. In this example, QV2 differs from QV1 by retaining only likely/pathogenic variants (indicated by a red X). The QV file loaded by the analysis pipeline comprises a description field (optional) and a variables field (mandatory). The QV criteria may be distributed across multiple pipeline steps.

2.3 Usage in a GWAS validation study

We next applied the QV criteria framework to a GWAS using HapMap3 Phase 3 (R3) consensus genotypes on 1397 individuals (26). Again, two pipelines were executed with identical inputs and parameters: one hard-coded and one driven by the QV file. This QV set defined common GWAS thresholds: restriction to autosomal, biallelic SNPs; minimum sample call rate of 95%; variant call rate of 95%; minor allele frequency $\geq 1\%$; and Hardy–Weinberg equilibrium $p \geq 1 \times 10^{-6}$. After quality control, variants were LD-pruned and principal components (PC1–PC10) were computed, with sex included as an additional covariate. Logistic regression under an additive model was then performed with a binary simulated phenotype using PLINK. The outputs of the two pipelines were captured and compared across each main PLINK stage.

Manhattan plots, Principal Component Analysis (PCA) plots, and `md5` checksums were used to confirm exact reproducibility between the hard-coded and QV-driven analyses.

For benchmarking, Message-Digest Algorithm 5 (MD5) checksums were uniquely reported for the GWAS study because PLINK output files are exactly reproducible between runs. In contrast, VCF files used in the other validation studies include variable header fields such as BCFtools view command with a timestamp, which changes with each run and alters the MD5 value. For those cases, we instead report variant count and content.

2.4 Usage in a WGS validation study with GIAB and Exomiser

We next applied the QV framework to a WGS trio analysis using the Genome In A Bottle Chinese Trio (HG005-HG007, PRJNA200694, GRCh38 v4.2.1) of the National Institute of Standards and Technology (27). Two pipeline phases were executed with identical inputs and parameters: one hard-coded and one driven by the QV file. Both phases applied identical QC and study filters and included a gene-panel style analysis using the paediatric disorders panel (panel 486; 3,853 genes (25)). The upstream processing used BCFtools for region restriction using BED overlap, site-level thresholds on QUAL and INFO/DP (using computed site depth from per-sample FORMAT/DP when absent), and per-sample thresholds on FORMAT/DP and FORMAT/GQ with exclusion of missing genotypes. Composite criteria were applied to require either all samples to pass or at least one sample to pass. The downstream filtered trio VCF was analysed with Exomiser using the same trio `.ped` input and without using Human Phenotype Ontology (HPO) terms.

3 Results

3.1 Validation rare disease cohort case study

We validated the QV framework using WES analysis with ACMG-based criteria on a rare disease cohort of 940 individuals, comparing a conventional pipeline with parameters defined internally (QV manual) to the new external YAML-based implementation (QV yaml). As shown in **Figure S1**, the outputs from both methods were identical, demonstrating a 100% match. This confirmed that our framework of a standalone,

shareable, QV criteria file can be imported and applied programmatically with equivalent accuracy, providing a reproducible resource that is adaptable across different pipelines and programming environments.

3.2 Validation in a common variant GWAS

To demonstrate the integration of the QV framework with established best practices in GWAS (28), we validated it in a standard HapMap3 Phase 3 GWAS by again running two equivalent analyses: a conventional pipeline with parameters defined internally and a YAML-based implementation that externalised all settings. As shown in **Figure S2**, the Manhattan and PCA plots were identical between the two methods, and the MD5 checksums of all PLINK outputs matched exactly. These results confirm that QV parameterisation reproduces the original workflow precisely while improving clarity, transparency, and reusability.

3.3 Validation in a WGS study with GIAB and Exomiser

To demonstrate the ease and benefit of using QV parameterisation in established WGS analysis pipelines, we conducted a trio validation study using the Genome in a Bottle (GIAB) Chinese Trio (HG005-HG007, GRCh38 v4.2.1) and the Exomiser tool for variant annotation and interpretation (29). Two equivalent analyses were run: one with hard-coded thresholds and one using an external QV YAML file specifying the same parameters. Both applied identical QC and study filters and restricted analysis to the PanelAppRex paediatric disorders panel (3,853 genes). Results were identical: variant counts matched at each step, and Exomiser outputs produced the same candidate genes and variants. **Figure S3** shows this agreement. This validation confirms that a shareable QV file reproduces the full variant interpretation workflow exactly, while aligning with established variant effect predictors and interpretation tools (29–31). Benchmarking showed that QV files introduce no computational overhead and scale equivalently to conventional implementations (**Supplemental 10.2**, **Figure S4**).

3.4 Implications

General applicability and reproducibility

Across validation studies, the QV framework reproduced conventional workflows in which parameters are embedded within scripts, while externalising those same variables into a portable, shareable format. The framework itself performs no filtering, calling, annotation, or interpretation, but provides a machine-readable layer for defining and reusing the qualifying variables that underpin these analyses. It complements tools such as GATK and BCFtools for processing, Ensembl VEP, SnpEff, FAVOR, and WGSA for variant effect prediction (30), and Exomiser and VarFish for interpretation (29; 31), by making their analytic criteria explicit.

Scalability and interoperability with genomic tools

The validation studies, covering clinical interpretation, genome-wide association analysis, and WGS trio interpretation, demonstrate that the QV framework generalises across distinct genomic contexts without altering analytical outcomes or adding computational overhead. The format further allows users to define, combine, and extend their own QV sets using simple declarative syntax, providing a scalable approach for reproducible genomics.

Traceability and confirmation of applied clinical standards

Each QV file includes a persistent identifier and checksum that can be stored in Electronic Health Record (EHR) or laboratory systems such as EPIC, Cerner, Clinisys, or REDCap. This links each patient's analysis (including any associated PPIE input) to the exact QV set used, enabling transparent, auditable, and FAIR-compliant reporting. A clinician or molecular pathologist viewing a result in EPIC or Cerner can access the linked `qv_set_id` to verify the applied standards and filtering criteria. Automated genomic reports should include these details by default, ensuring full traceability without requiring access to the pipeline. For example, if a patient asks whether their genome was screened for breast cancer due to variants in *BRCA1* or *BRCA2*, the EHR-linked report referencing "qv acmg sf v3.3 20250828.json" confirms that the ACMG secondary findings guideline (v3.3) (32) was applied, including its defined gene set, thresholds, version, and standard.

4 Summary

This paper introduces a framework for integrating qualifying variants into genomic analysis pipelines, enhancing reproducibility, interpretability and the seamless translation of research findings into clinical practice.

5 Funding

This project was supported through the grant Swiss National Science Foundation 320030_201060, and NDS-2021-911 (SwissPedHealth) from the Swiss Personalized Health Network and the Strategic Focal Area 'Personalized Health and Related Technologies' of the ETH Domain (Swiss Federal Institutes of Technology).

6 Acknowledgements

Acknowledgements We would like to thank all the patients and families who have been providing advice on SwissPedHealth and its projects, as well as the clinical and research teams at the participating institutions.

7 Contributions

DL designed the work and contributed to the manuscript. AS, SB, VS, DH, SÖ, JA, SF contributed to the manuscript. LJS and JF supervised the work, manuscript, and applied for funding.

8 Competing interests

The authors declare no competing interests.

9 Ethics statement

The projects were approved by the respective ethics committees of all participating centers (Cantonal Ethics Committee Bern, approval number KEK-029/11) and the

study was conducted in accordance with the Declaration of Helsinki.

References

[1] Sue Richards, Nazneen Aziz, Sherri Bale, David Bick, Soma Das, Julie Gastier-Foster, Wayne W Grody, Madhuri Hegde, Elaine Lyon, Elaine Spector, et al. Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the american college of medical genetics and genomics and the association for molecular pathology. *Genetics in medicine*, 17 (5):405–423, 2015.

[2] Marilyn M Li, Michael Datto, Eric J Duncavage, Shashikant Kulkarni, Neal I Lindeman, Somak Roy, Apostolia M Tsimberidou, Cindy L Vnencak-Jones, Daynna J Wolff, Anas Younes, et al. Standards and guidelines for the interpretation and reporting of sequence variants in cancer: a joint consensus recommendation of the association for molecular pathology, american society of clinical oncology, and college of american pathologists. *The Journal of molecular diagnostics*, 19(1):4–23, 2017.

[3] Quan Li and Kai Wang. Intervar: clinical interpretation of genetic variants by the 2015 acmg-amp guidelines. *The American Journal of Human Genetics*, 100 (2):267–280, 2017.

[4] Erin Rooney Riggs, Erica F Andersen, Athena M Cherry, Sibel Kantarci, Hutton Kearney, Ankita Patel, Gordana Raca, Deborah I Ritter, Sarah T South, Erik C Thorland, et al. Technical standards for the interpretation and reporting of constitutional copy-number variants: a joint consensus recommendation of the american college of medical genetics and genomics (acmge and the clinical genome resource (clingen). *Genetics in Medicine*, 22(2):245–257, 2020.

[5] Sean V Tavtigian, Steven M Harrison, Kenneth M Boucher, and Leslie G Biesecker. Fitting a naturally scaled point system to the acmg/amp variant classification guidelines. *Human mutation*, 41(10):1734–1737, 2020.

[6] Brent S Pedersen, Joe M Brown, Harriet Dashnow, Amelia D Wallace, Matt Velinder, Martin Tristani-Firouzi, Joshua D Schiffman, Tatiana Tvrdik, Rong Mao, D Hunter Best, et al. Effective variant filtering and expected candidate variant yield in studies of rare human disease. *NPJ Genomic Medicine*, 6(1):1–8, 2021.

[7] Carl A Anderson, Fredrik H Pettersson, Geraldine M Clarke, Lon R Cardon, Andrew P Morris, and Krina T Zondervan. Data quality control in genetic case-control association studies. *Nature protocols*, 5(9):1564–1573, 2010. URL https://doi.org/10.1038/nprot.2010.116.

[8] Emil Uffelmann, Qin Qin Huang, Nchangwi Syntia Munung, Jantina De Vries, Yukinori Okada, Alicia R Martin, Hilary C Martin, Tuuli Lappalainen, and Danielle Posthuma. Genome-wide association studies. *Nature Reviews Methods Primers*, 1(1):59, 2021. URL https://doi.org/10.1038/s43586-021-00056-9.

[9] Hannah Wand, Samuel A Lambert, Cecelia Tamburro, Michael A Iacocca, Jack W O'Sullivan, Catherine Sillari, Iftikhar J Kullo, Robb Rowley, Jacqueline S Dron, Deanna Brockman, et al. Improving reporting standards for polygenic scores in risk prediction studies. *Nature*, 591(7849):211–219, 2021.

[10] Samuel A Lambert, Laurent Gil, Simon Jupp, Scott C Ritchie, Yu Xu, Annalisa Buniello, Aoife McMahon, Gad Abraham, Michael Chapman, Helen Parkinson, et al. The polygenic score catalog as an open database for reproducibility and systematic evaluation. *Nature Genetics*, 53(4):420–425, 2021.

[11] Brent S Pedersen and Aaron R Quinlan. Vcfexpress: flexible, rapid user-expressions to filter and format VCFs. *Bioinformatics*, 41(3): btaf097, March 2025. ISSN 1367-4811. doi: 10.1093/bioinformatics/btaf097. URL https://academic.oup.com/bioinformatics/article/doi/10.1093/bioinformatics/btaf097/8051444.

[12] Henri E. Bal, Jennifer G. Steiner, and Andrew S. Tanenbaum. Programming languages for distributed computing systems. *ACM Computing Surveys*, 21(3): 261–322, September 1989. ISSN 0360-0300, 1557-7341. doi: 10.1145/72551.72552. URL https://dl.acm.org/doi/10.1145/72551.72552.

[13] Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B. Hall, Christopher H. Tomkins-Tinch, Vanessa Sochat, Jan Forster, Soohyun Lee, Sven O. Twardziok, Alexander Kanitz, Andreas Wilm, Manuel Holtgrewe, Sven Rahmann, Sven Nahnsen, and Johannes Köster. Sustainable data analysis with Snakemake. *F1000Research*, 10:33, January 2021. ISSN 2046-1402. doi: 10.12688/f1000research.29032.1. URL https://f1000research.com/articles/10-33/v1.

[14] Paolo Di Tommaso, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. Nextflow enables reproducible computational workflows. *Nature Biotechnology*, 35(4):316–319, April 2017. ISSN 1087-0156, 1546-1696. doi: 10.1038/nbt.3820. URL https://www.nature.com/articles/nbt.3820.

[15] Gundula Povysil, Slavé Petrovski, Joseph Hostyk, Vimla Aggarwal, Andrew S. Allen, and David B. Goldstein. Rare-variant collapsing analyses for complex traits: guidelines and applications. *Nature Reviews Genetics*, 20(12):747–759, 2019. doi: 10.1038/s41576-019-0177-4. URL https://doi.org/10.1038/s41576-019-0177-4.

[16] Elizabeth T Cirulli, Brittany N Lasseigne, Slavé Petrovski, Peter C Sapp, Patrick A Dion, Claire S Leblond, Julien Couthouis, Yi-Fan Lu, Quanli Wang, Brian J Krueger, et al. Exome sequencing in amyotrophic lateral sclerosis identifies risk genes and pathways. *Science*, 347(6229):1436–1441, 2015.

[17] Nathan D Olson, Justin Wagner, Nathan Dwarshuis, Karen H Miga, Fritz J Sedlazeck, Marc Salit, and Justin M Zook. Variant calling and benchmarking in an era of complete human genome sequences. *Nature Reviews Genetics*, 24(7):464–483, 2023.

[18] James J Lee, Robbee Wedow, Aysu Okbay, Edward Kong, Omeed Maghzian, Meghan Zacher, Tuan Anh Nguyen-Viet, Peter Bowers, Julia Sidorenko, Richard Karlsson Linnér, et al. Gene discovery and polygenic prediction from a 1.1-million-person gwas of educational attainment. *Nature genetics*, 50(8):1112, 2018.

[19] Philip R Jansen, Kyoko Watanabe, Sven Stringer, Nathan Skene, Julien Bryois, Anke R Hammerschlag, Christiaan A de Leeuw, Jeroen S Benjamins, Ana B Muñoz-Manchado, Mats Nagel, et al. Genome-wide analysis of insomnia in 1,331,010 individuals identifies new risk loci and functional pathways. *Nature genetics*, 51(3):394–403, 2019.

[20] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9, 2016.

[21] Eelke van der Horst, Deepak Unni, Femke Kopmels, Jan Armida, Vasundra Touré, Wouter Franke, Katrin Crameri, Elisa Cirillo, and Sabine Österle. Bridging clinical and genomic knowledge: An extension of the sphn rdf schema for seamless integration and fairification of omics data. 2023.

[22] Vasundra Touré, Philip Krauss, Kristin Gnodtke, Jascha Buchhorn, Deepak Unni, Petar Horki, Jean Louis Raisaro, Katie Kalt, Daniel Teixeira, Katrin Crameri, et al. Fairification of health-related data using semantic web technologies in the swiss personalized health network. *Scientific Data*, 10(1):127, 2023.

[23] Geraldine Van der Auwera and Brian D. O'Connor. *Genomics in the cloud: using Docker, GATK, and WDL in Terra*. O'Reilly, Beijing Boston Farnham Sebastopol Tokyo, first edition edition, 2020. ISBN 978-1-4919-7519-0 978-1-4919-7516-9 978-1-4919-7512-1.

[24] Xihao Li, Han Chen, Margaret Sunitha Selvaraj, Eric Van Buren, Hufeng Zhou, Yuxuan Wang, Ryan Sun, Zachary R McCaw, Zhi Yu, Min-Zhi Jiang, et al. A statistical framework for multi-trait rare variant analysis in large-scale whole-genome sequencing studies. *Nature Computational Science*, pages 1–19, 2025.

[25] Dylan Lawless. PanelAppRex aggregates disease gene panels and facilitates sophisticated search. March 2025. doi: 10.1101/2025.03.20.25324319. URL http://medrxiv.org/lookup/doi/10.1101/2025.03.20.25324319.

[26] Susan Fairley, Ernesto Lowy-Gallego, Emily Perry, and Paul Flicek. The International Genome Sample Resource (IGSR) collection of open human genomic variation resources. *Nucleic Acids Research*, 48(D1):D941–D947, January 2020. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkz836.

[27] Justin Wagner, Nathan D. Olson, Lindsay Harris, Ziad Khan, Jesse Farek, Medhat Mahmoud, Ana Stankovic, Vladimir Kovacevic, Byunggil Yoo, Neil Miller, Jeffrey A. Rosenfeld, Bohan Ni, Samantha Zarate, Melanie Kirsche, Sergey Aganezov, Michael C. Schatz, Giuseppe Narzisi, Marta Byrska-Bishop, Wayne Clarke, Uday S. Evani, Charles Markello, Kishwar Shafin, Xin Zhou, Arend Sidow, Vikas Bansal, Peter Ebert, Tobias Marschall, Peter Lansdorp, Vincent Hanlon, Carl-Adam Mattsson, Alvaro Martinez Barrio, Ian T. Fiddes, Chunlin Xiao, Arkarachai Fungtammasan, Chen-Shan Chin, Aaron M. Wenger, William J. Rowell, Fritz J. Sedlazeck, Andrew Carroll, Marc Salit, and Justin M. Zook. Benchmarking challenging small variants with linked and

long reads. *Cell Genomics*, 2(5):100128, May 2022. ISSN 2666979X. doi: 10.1016/j.xgen.2022.100128.

[28] Emil Uffelmann, Qin Qin Huang, Nchangwi Syntia Munung, Jantina De Vries, Yukinori Okada, Alicia R. Martin, Hilary C. Martin, Tuuli Lappalainen, and Danielle Posthuma. Genome-wide association studies. *Nature Reviews Methods Primers*, 1(1):59, August 2021. ISSN 2662-8449. doi: 10.1038/s43586-021-00056-9.

[29] Valentina Cipriani, Nikolas Pontikos, Gavin Arno, Panagiotis I. Sergouniotis, Eva Lenassi, Penpitcha Thawong, Daniel Danis, Michel Michaelides, Andrew R. Webster, Anthony T. Moore, Peter N. Robinson, Julius O.B. Jacobsen, and Damian Smedley. An Improved Phenotype-Driven Tool for Rare Mendelian Variant Prioritization: Benchmarking Exomiser on Real Patient Whole-Exome Data. *Genes*, 11(4):460, April 2020. ISSN 2073-4425. doi: 10.3390/genes11040460.

[30] Cristian Riccio, Max L. Jansen, Linlin Guo, and Andreas Ziegler. Variant effect predictors: A systematic review and practical guide. *Human Genetics*, 143(5):625–634, May 2024. ISSN 1432-1203. doi: 10.1007/s00439-024-02670-5.

[31] Manuel Holtgrewe, Oliver Stolpe, Mikko Nieminen, Stefan Mundlos, Alexej Knaus, Uwe Kornak, Dominik Seelow, Lara Segebrecht, Malte Spielmann, Björn Fischer-Zirnsak, Felix Boschann, Ute Scholl, Nadja Ehmke, and Dieter Beule. VarFish: Comprehensive DNA variant analysis for diagnostics and research. *Nucleic Acids Research*, 48(W1):W162–W169, July 2020. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkaa241.

[32] David T Miller, Kristy Lee, Noura S Abul-Husn, Laura M Amendola, Kyle Brothers, Wendy K Chung, Michael H Gollob, Adam S Gordon, Steven M Harrison, Ray E Hershberger, et al. Acmg sf v3. 2 list for reporting of secondary findings in clinical exome and genome sequencing: a policy statement of the american college of medical genetics and genomics (acmg). *Genetics in Medicine*, 25(8):100866, 2023.

10 Supplemental

Application of qualifying variants for genomic analysis.

10.1 Validation study figures

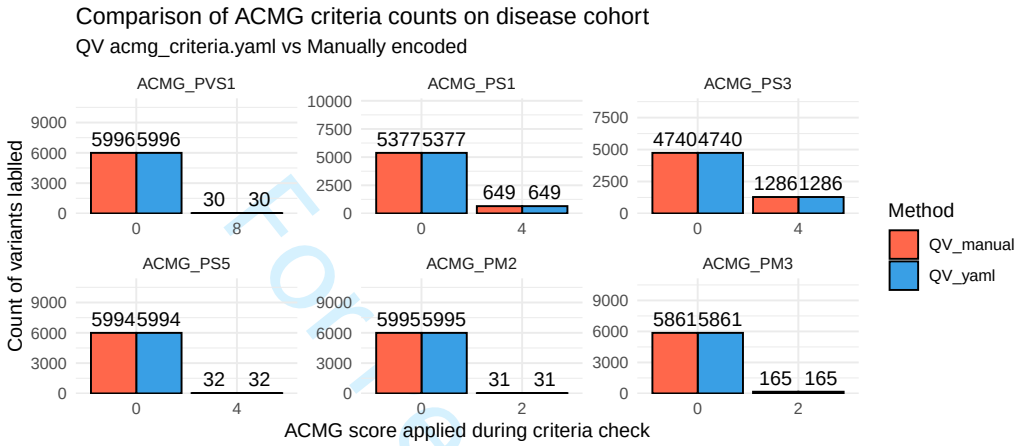


Figure S1: Validation case study of a rare disease cohort of 940 WES individuals using an ACMG criteria subset, demonstrating a 100% match between manually encoded and standalone YAML-based QV for assigning pathogenicity scores.

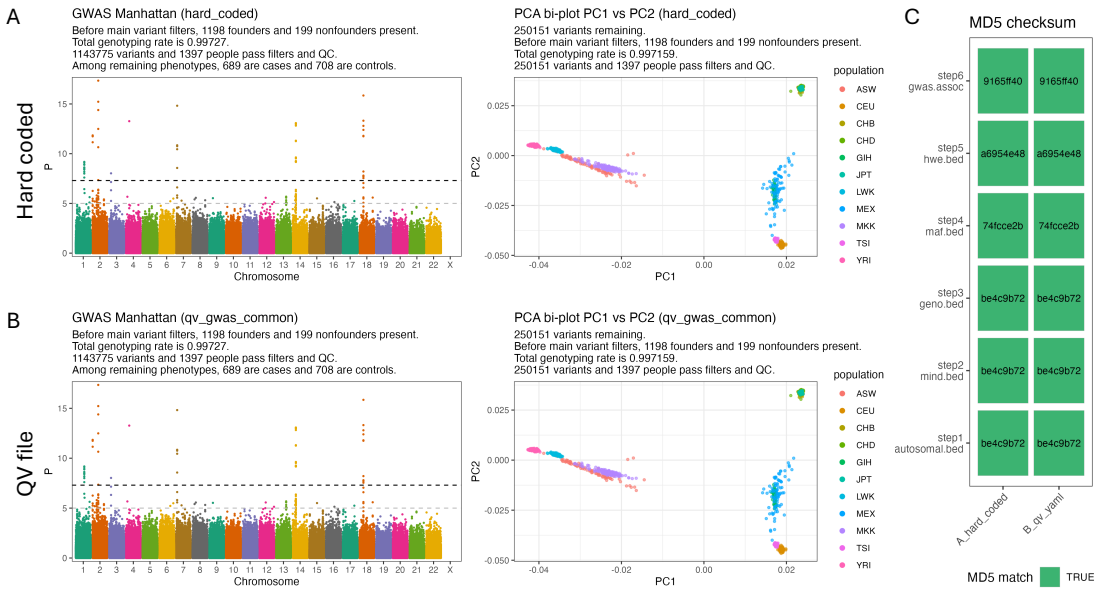
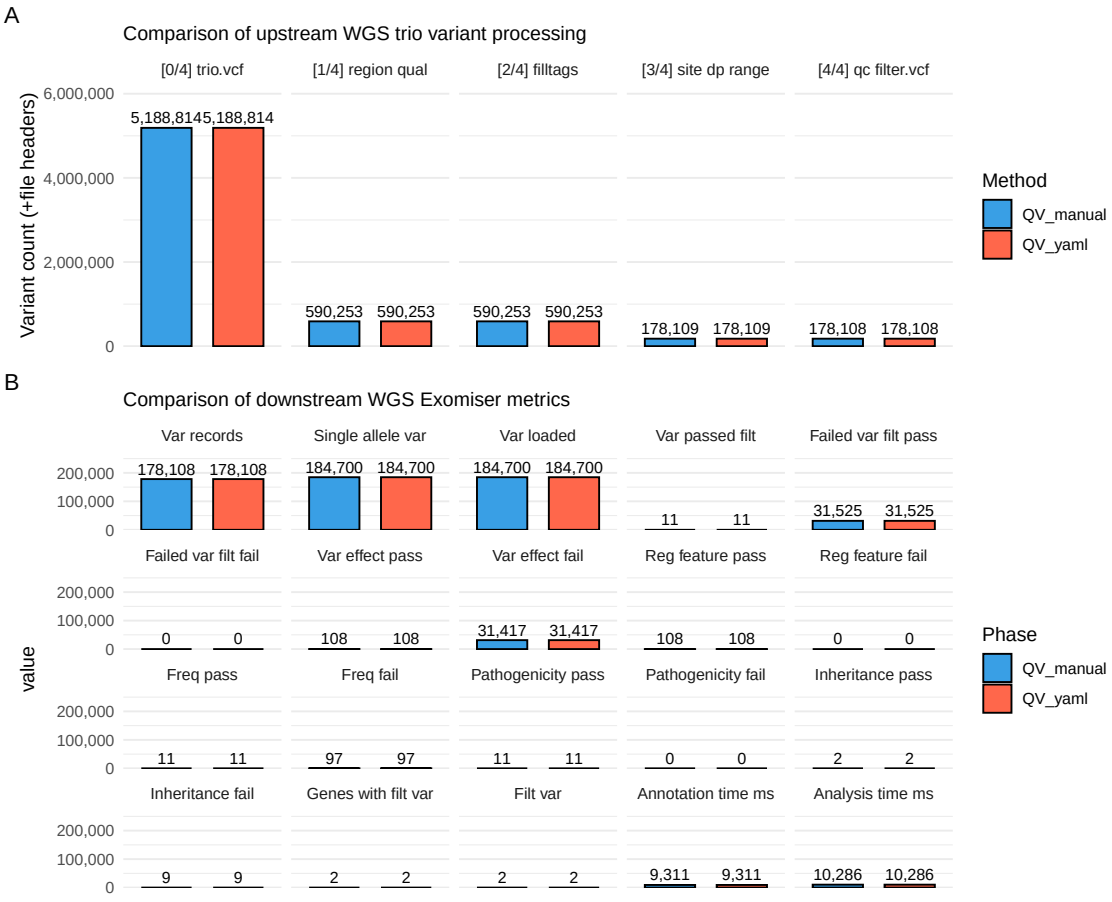


Figure S2: Validation in GWAS using QV parameterisation. (A) GWAS of simulated binary phenotypes in HapMap3 Phase 3 (R3) using a traditional variable embedded pipeline. Shown are the Manhattan plot of logistic regression results (left) and correction for population structure with principal component analysis (PC1 vs PC2, right). (B) Identical GWAS using a QV YAML configuration file. The Manhattan and PCA results are indistinguishable from panel A. (C) Verification of reproducibility. MD5 checksums of the main PLINK outputs are identical between panels A and B. The steps included processing of autosomal biallelic SNPs, sample call rate, variant call rate, minor allele frequency, Hardy–Weinberg equilibrium, and association results. The QV file encoded these thresholds (sample call rate $\geq$95%, variant call rate $\geq$95%, MAF $\geq$1%, HWE p $\geq$1e-6, autosomal biallelic SNPs only) together with covariates (sex and PC1-PC10) and logistic regression settings. This confirms that a shareable QV file reproduces hard-coded pipelines exactly while improving transparency and reusability.



phase	MOI	Effect	HGVS	Genotype
QV_manual	AD	frameshift	RYR2:ENST00000366574.7:c.6493dup:p.(Leu2165Profs*44)	0/1:./.:./.
QV_yaml	AD	frameshift	RYR2:ENST00000366574.7:c.6493dup:p.(Leu2165Profs*44)	0/1:./.:./.
QV_manual	AR	missense	TTN:ENST00000589042.5:c.36982A>G:p.(Thr12328Ala)	1|1:0/1:0/1
QV_yaml	AR	missense	TTN:ENST00000589042.5:c.36982A>G:p.(Thr12328Ala)	1|1:0/1:0/1

Figure S3: Validation of the trio Exomiser pipeline using QV parameterisation. **(A)** summarises upstream processing counts by file, **(B)** compares downstream Exomiser metrics, and **(C)** shows the key variant fields for the two variants identified. Variant counts in all panels confirm that intermediate files and final outputs are identical between configurations. The five preprocessing stages shown in (A) are: (0) input trio VCF, (1) gene panel region and quality filtering, (2) tag annotation, (3) site-level depth range filtering, and (4) final QC-filtered VCF. MOI, mode of inheritance; HGVS, Human Genome Variation Society nomenclature.

10.2 Computational benchmark

Runtime performance was equivalent between traditional and QV-based pipelines, as both read identical parameters from different sources. In the WGS trio validation study, pre-processing steps including filtering, QC, and gene panel selection completed in 16–17 seconds, with a median difference of ~0.5 seconds favouring the QV YAML pipeline (**Figure S4**). An incidental one-off 5 second delay arose from Singularity initialisation for the `yq` utility (step 0), a system-specific effect on our HPC and unrelated to the framework itself.

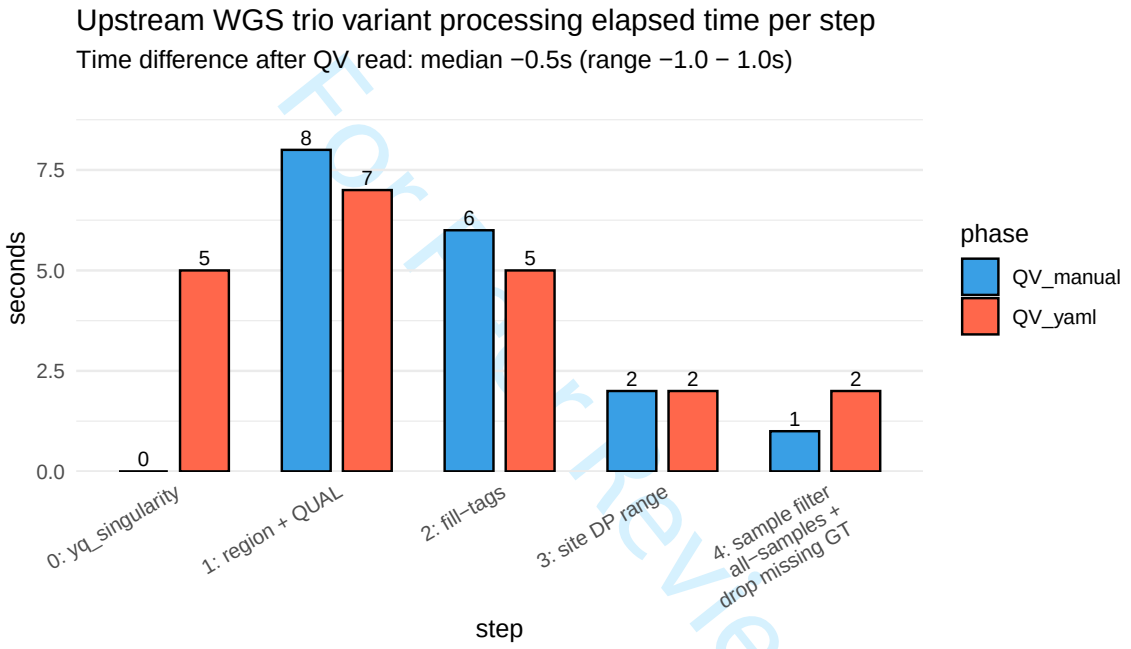


Figure S4: Benchmark of upstream preprocessing times in the WGS trio pipeline comparing QV-based and traditional (manually parameterised) configurations. Stepwise elapsed times were nearly identical across both methods (median difference ~0.5 s), with a fixed 5 s overhead from optional Singularity initialisation of `yq` in the QV pipeline. The four preprocessing steps correspond to: (1) gene panel region and quality filtering of the trio VCF, (2) annotation of variant tags, (3) site-level depth range filtering, and (4) per-sample genotype filtering and exclusion of missing genotypes. All steps used `BCFtools` on VCF preprocessing, as illustrated in **Figure S3 (A)**.

10.3 How to build a QV file

We recommend YAML or JSON for portability and adoption. You can build a QV in three ways:

Option 1: use the HTML QV builder (Zenodo)

1. Open the HTML builder from the Zenodo repository.
2. Enter simple `key=value` statements in the left pane.
3. Copy or download the generated YAML.

Example input lines:

```
meta qv_set_id="qv_gwas_common_v1_20250827"
meta version="1.0.0"
meta title="GWAS common QC"
meta authors=Alice,Bob
meta tags=GWAS,QC,PCA
filter maf_minimum field=MAF operator=">=" value=0.01 desc="Minimum MAF"
filter hwe field=HWE_P operator=">=" value=1e-6 logic=keep_if
filter region_include desc="include panel" field=OVERLAP(targets.exome.bed)
    >>>> operator=">=" value=1 logic=keep_if
criteria disease_panel logic=and desc="HIGH impact within panel"
criteria disease_panel field=IMPACT operator="==" value=HIGH
criteria disease_panel field=OVERLAP(targets.exome.bed) operator=">=" value=1
meta description_patient=
    >>>> "There is a strong family history of early heart attacks."
meta description_ppie=
    >>>> "The PPIE group reviewed and approved the criteria on 2025-08-15."
```

Option 2: write YAML by hand

Minimal pattern:

```
meta:
  qv_set_id: qv_disease_panel_v1_20250828
  version: 1.0.0
  title: Disease panel filter
filters:
  region_include:
    description: Restrict to curated disease gene panel
    logic: keep_if
```

```
    field: OVERLAP(targets.disease_panel.bed)
    operator: ">="
    value: 1
criteria:
  pathogenic:
    description: Variant classified as pathogenic or likely pathogenic
    logic: and
    conditions:
      - group: any_of:start
      - { field: CLASS, operator: "==", value: P }
      - { field: CLASS, operator: "==", value: LP }
      - group: any_of:end
notes:
  - Gene panel file defines the target regions
```

Option 3: write JSON

JSON equivalent of the minimal example:

```
{
  "meta": {
    "qv_set_id": "qv_disease_panel_v1_20250828",
    "version": "1.0.0",
    "title": "Disease panel filter"
  },
  "filters": {
    "region_include": {
      "description": "Restrict to curated disease gene panel",
      "logic": "keep_if",
      "field": "OVERLAP(targets.disease_panel.bed)",
      "operator": ">=",
      "value": 1
    }
  },
  "criteria": {
    "pathogenic": {
      "description": "Variant classified as pathogenic or likely pathogenic",
      "logic": "and",
```

```
      "conditions": [
        { "group": "any_of:start" },
        { "field": "CLASS", "operator": "==", "value": "P" },
        { "field": "CLASS", "operator": "==", "value": "LP" },
        { "group": "any_of:end" }
      ]
    }
  },
  "notes": [
    "Gene panel file defines the target regions"
  ]
}
```

Checksum and register

Record the checksum and register the release:

```
sha256sum qv/examples/qv_disease_panel_v1_20250828.yaml

# qv/registry/releases.csv
qv_set_id, version, checksum, file, date
qv_disease_panel_v1_20250828,1.0.0, ef6cf810b994...,
    > qv_disease_panel_v1_20250828.yaml, 2025-08-28
```

Versioning and IDs

Use a stable `qv_set_id` plus semantic version. Update the version on any change that affects selection or interpretation. Keep one file per release and never mutate published files.

Use in a workflow

Point your pipeline to the QV file:

```
# workflows/.../config.yaml
qv_file: ".../qv/registry/qv_disease_panel_v1_20250828.yaml"
```

It can be read programmatically at runtime, for example using `yq` in shell-based workflows or `yaml::read_yaml()` in R, providing the same parameters that would otherwise be embedded within pipeline configurations.

Application of qualifying variants for genomic analysis

Dylan Lawless*[1], Ali Saadat[2], Mariam Ait Oumelloul[2], Simon Boutry[2], Veronika Stadler[1], Sabine Österle[3], Jan Armida[3], David Haerry[4], D. Sean Froese[5], Luregn J. Schlapbach[1], and Jacques Fellay[2]

[1]Department of Intensive Care and Neonatology, University Children's Hospital Zürich, University of Zürich, Switzerland.
[2]Global Health Institute, School of Life Sciences, École Polytechnique Fédérale de Lausanne, Switzerland.
[3]SPHN Data Coordination Center, SIB Swiss Institute of Bioinformatics, Basel, Switzerland.
[4]Positive Council, Zürich, Switzerland.
[5]Division of Metabolism and Children's Research Center, University Children's Hospital Zürich, University of Zurich, Zurich, Switzerland.

October 22, 2025

*Addresses for correspondence: Dylan.Lawless@kispi.uzh.ch

Abstract

Motivation:

Qualifying variants (QVs) are genomic alterations selected by defined criteria within analysis pipelines. Although crucial for both research and clinical diagnostics, QVs are often seen as simple filters rather than dynamic elements that influence the entire workflow. ~~While best practices follow variant classification standards and standardised workflows, a unified framework to integrate and optimise QVs for advanced applications is missing~~In practice these rules are embedded within pipelines, which hinders transparency, audit, and reuse across tools. A unified, portable specification for QV criteria is needed.

Results:

Our aim is to embed the concept of a "QV" into the genomic analysis vernacular, moving beyond its treatment as a single filtering step. By decoupling QV criteria from ~~other~~ pipeline variables and code, ~~our approach facilitates easier discussionand application. Our framework, with its new terminology and reference model, offers a flexible approach~~ the framework enables clearer discussion, application, and reuse. It provides a flexible reference model for integrating QVs into analysis pipelines, ~~thereby enhancing~~ improving reproducibility, interpretability, and interdisciplinary communication. ~~A validation case study implementing ACMG criteria in a disease cohort shows that our approach matches~~ Validation across diverse applications confirmed that QV based workflows match conventional methods while offering ~~improved~~ greater clarity and scalability.

Availability:

The source code and data are accessible at ~~. The QV file used in this work is available from (`qv_acmg_svnindel_criteria_20250225.yaml`)~~the Zenodo repository `https://doi.org/10.5281/zenodo.17414191`. Manuscript files are available at `https://github.com/DylanLawless/qvApp2025lawless`. The QV framework is available under the MIT licence, and the dataset will be maintained for at least two years following publication.

Acronyms

ACMG American College of Medical Genetics and Genomics 12

EHR Electronic Health Record 16

FAIR Findable, Accessible, Interoperable, and Reusable 9

GIAB Genome in a Bottle 15

GWAS Genome-Wide Association Study 4

HPO Human Phenotype Ontology 14

MAF Minor Allele Frequency 11

MD5 Message-Digest Algorithm 5 13

PCA Principal Component Analysis 12

PPIE Public and Patient Involvement and Engagement 9

PRS Polygenic Risk Score 4

QC Quality Control 5

QV Qualifying Variant 4

RDF Resource Description Framework 9

SNOMED CT Systematized Nomenclature of Medicine-Clinical Terms . . . 9

VCF Variant Call Format 4

VEP Variant Effect Predictor 12

WES Whole Exome Sequencing 11

WGS Whole Genome Sequencing 5

1 Introduction

Qualifying Variant (QV)s are genomic alterations selected by specific criteria within genome processing pipelines, serving as dynamic elements essential for both research and clinical diagnostics. QVs are not merely static filters applied at a single step in an analysis pipeline; rather, they are dynamic, multifaceted elements that permeate the entire workflow, from initial data quality control to final result interpretation. This nuanced perspective underscores that QVs play an integral role in shaping the fidelity and reproducibility of genomic analyses, enabling the iterative refinement of data and facilitating the integration of diverse analytical strategies throughout the pipeline.

Often, QV selection adheres to established variant classification and reporting standards (1–5) and standardised workflows (6–8). However, a unified framework for QVs is lacking, despite the recognised benefits of similar initiatives, such as Polygenic Risk Score (PRS) reporting standards (9; 10). ~~For instance, tools like~~ Tools such as vcfexpress (11) enable flexible ~~, rapid~~ filtering and formatting of ~~VCF~~ Variant Call Format (VCF) files using user-defined expressions. ~~The application of independently defined~~ Treating QV criteria ~~would complement such tools . This role is particularly important for~~ as an external parameter layer complements these tools by externalising their thresholds and logic. This approach improves reproducibility across distributed computing environments (12) and ~~would also integrate~~ integrates seamlessly with workflow managers ~~such as~~ like Snakemake (13) or Nextflow (14)~~, streamlining genomic processing tasks~~.

~~The criteria for~~ QV selection criteria vary by application. ~~For example,~~ In Genome-Wide Association Study (GWAS)~~may focus on common variants, while clinical analyses usually target rare or known pathogenic variants. Previous studies have demonstrated the utility of s (15; 16), yet no common approach exists. Here, we detail four typical applications of sets: **passing only**: Generates large datasets (e.g. > 500, 000 variantsper subject) for or initial pre-processing. **Flexible** : Balances between and false positives, yielding intermediate datasets (e.g. fewer than 100,~~thresholds favour common variants, yielding datasets with over 500,000 variants per subject~~) for uses such as rare variant association testing. **for rare disease**: Applies stringent filtering to produce smaller datasets (e.g. < , whereas rare disease analyses use stringent filters producing fewer than 1~~,~~,000 variants~~per subject), targeting~~ , often limited to known genes or ~~single causal variants. **Known disease panel set**: Focuses on well-established gene panels with pathogenic variants (e.g. the set) for clinical reporting (17).~~

~~These examples illustrate a few common applications without providing an exhaustive classification of all possible~~ pathogenic loci. Although targeted filtering is valuable (15; 16), no unified approach exists. In practice, QV ~~uses. The careful selection and categorisation of s are thus critical for accurate reporting and reproducibility, sometimes even more so than the choice of the analysis~~ sets range from broad quality control filters to specific disease panels, and their definition is critical for reproducibility and accurate reporting, influencing results as much as the pipeline itself (18).

As Whole Genome Sequencing (WGS) becomes standard for large cohorts (19; 20), the integration of diverse QV protocols is critical for data cleaning and analysis. During sequencing analysis several layers can be responsible for triggering QV protocols, including pre-existing metadata, technical Quality Control (QC) results, and post-calling annotations, highlighting the need for a clear, unified approach.

~~We propose treating~~ We introduce the QV as a standalone entity, independent from other pipeline variables. ~~We suggest structured~~ Structured human- and machine-readable criteria, aligned with FAIR principles (21)~~to~~, facilitate integration across databases (22; 23). We advocate for the use of standard vocabularies, unique identifiers, and flexible file formats to support this integration.

Building on this framework, we propose an openly documented registry model for QV files that assigns a unique `qv_set_id` and records a SHA-256 checksum for each release, enabling direct retrieval and verification for audit and re-analysis. Our accompanying HTML-based QV builder converts simple `key=value` statements into structured YAML and can be embedded in public, private, or commercial websites to simplify the authoring of consistent criteria (Zenodo repository). The framework is designed to support the emergence of a shared, widely adopted registry over time.

2 Methods

2.1 Implementation

~~Implementation configurations and roles within analysis pipelines include, for example: theoretical pipelining of~~ The QV ~~sets, establishing publicor standardised sets for specific analytical scenarios, and recognition that s are integral throughout the analysis pipeline rather than confined to a single end-stage. We introduce a simple framework for the effective use of protocols, comprising four components~~ file provides a structured, human- and machine-readable definition of variant qualifying criteria. It is composed

of five logical components that define its structure and metadata. It is portable across tools, transparent in content, and verifiable through unique identifiers and checksums. Each file is a lightweight YAML or JSON document specifying the variables and thresholds used in analysis. It can be read programmatically at runtime, for example using `yq` in shell-based workflows or `yaml::read_yaml()` in R, providing the same parameters that would otherwise be embedded within pipeline configurations, as illustrated in **Figure~~?? (A)~~ 1**~~:~~. The output is identical to that of the native workflow, with the added benefit of an explicit, versioned, and shareable configuration file.

- ~~**1. Variables**: The criteria variables sourced as part of the pipeline (see **Box ??**).~~
- ~~**2a. Technical description**: An optional narrative detailing each step within the overall set (see **Box ??**).~~
- ~~**2b. description**: An optional narrative providing a patient-focused interpretation of the protocol, incorporating preferences and priorities.~~
- ~~**3. set ID**~~**1. Meta**: ~~A unique identifier that links analysis records.~~ Descriptive metadata including `qv_set_id`, title, version, author list, creation date, and tags. These fields ensure traceability and version control across analyses.
- ~~**4. Source code**~~**2. Filters**: ~~The implementation of the variables file within the pipeline code, for examplethrough custom scripts or workflow managers.~~ Simple rule-based statements that apply inclusion or exclusion logic based on variable thresholds (for example, minimum allele frequency or coverage depth). Filters can also restrict the analysis to defined genomic regions, such as a target gene panel or BED file.
- **3. Criteria**: Compound logic blocks that combine one or more conditions into interpretable rules, corresponding to concepts such as ACMG criteria or study-specific thresholds.
- **4. Notes**: Optional free-text annotations providing context, assumptions, or technical caveats.
- **5. Descriptions (optional)**: Plain-language fields, such as `description_patient` and `description_ppie`, that can record patient preferences or public involvement input. These complement the technical definitions without affecting computational logic.

We propose the set ID as a unique identifier linking variant sets used in analyses. This facilitates integration into databases, by representing data in formats such as schemas (23), and allows for features including hash functions, s, semantic combinations, incorporation, registry-based allocation, and standard mapping such as . The results can be used alongside other genomic-specific concepts spanning from sample processing to the sequencing run (22).

Example QV structure

This framework efficiently manages We include an HTML-based QV -specific variables (e.g. allele frequency thresholds)separately from general pipeline settings, ensuring clarity and specificity. Its versatile format supports applications across genomic analyses and by linking the builder that can be embedded in research or commercial platforms to simplify the creation of consistent, versioned criteria files (available via Zenodo repository). A minimal QV set ID to both results and raw data sources in a database for downstream interpretation and reportingYAML file is shown in **Box 1**, equivalent to the configuration generated by this builder. QV files are composed of `key=value` statements, ensuring that all filtering and interpretation rules are explicit, versioned, and reproducible. In simple terms, **Box 1** specifies that only variants overlapping a curated disease gene panel are retained and that variants classified as pathogenic or likely pathogenic are prioritised. It also records patient context and patient-public involvement notes, thereby linking the technical filtering logic with its clinical and ethical rationale.

2.2 ~~Example application of qualifying variants in WGS analysis~~

Box 1: qv_disease_panel_example.yaml

```yaml
meta:
  qv_set_id: qv_disease_panel_v1_20250828
  version: 1.0.0
  title: Disease panel filter
filters:
  region_include:
    description: >
      Restrict to curated disease gene panel
    logic: keep_if
    field: OVERLAP(targets.disease_panel.bed)
    operator: '>='
    value: 1
criteria:
  pathogenic:
    description: >
      Variant classified as pathogenic or likely pathogenic
    logic: and
    conditions:
      - group: any_of:start
      - { field: CLASS, operator: '==', value: P }
      - { field: CLASS, operator: '==', value: LP }
      - group: any_of:end
meta:
  description_patient: >
    We have a strong family history of early heart attacks.
  description_ppie: >
    The PPIE group reviewed the criteria and approved them
    on 2025-08-15.
notes:
  - Gene panel file defines the target regions.
  - Additional quality filters may be added as needed.
```

~~Multiple~~

FAIR mapping and patient involvement

Each QV ~~protocols can be combined to generate progressively filtered datasets tailored to specific analytical needs. Often, different~~ file includes a persistent identifier (`qv_set_id`) that links criteria across analyses and databases. The framework aligns with the Findable, Accessible, Interoperable, and Reusable (FAIR) principles of findability, accessibility, interoperability, and reusability (21). Findability is achieved through unique identifiers; accessibility through open, human- and machine-readable YAML or JSON files; interoperability through standardised syntax (i.e. `key=value`) and semantic mappings such as Resource Description Framework (RDF) or Systematized Nomenclature of Medicine-Clinical Terms (SNOMED CT) (22; 23); and reusability through embedded metadata, checksum verification, and versioned registry records.

Optional metadata fields such as `description_patient` and `description_ppie` allow patient input and Public and Patient Involvement and Engagement (PPIE) feedback to be recorded in a manner appropriate to the study or application, with patient notes provided through consent-linked forms and PPIE groups offering structured review or approval of criteria within the same FAIR-compliant file.

Example QVs in WGS analysis

A typical WGS pipeline applies several QV ~~sets are applied sequentially, with the final outcomes merged to address distinct objectives. For instance, a comprehensive analysis pipeline might integrate:~~

- ~~,~~
- ~~,~~
- ~~,~~
- ~~, and~~
- ~~.~~

~~The final analysis yields (1) a joint cohort disease association (e.g. variant P-values) and (2) individual single-case results (e.g. clinical genetics diagnosis for a patient) (24; 25). As an example, in **Figure ?? (A)** we focus on a SNV/INDEL pipeline employing two~~ sets ~~: for flexible cohort-level filtering, and for stricter filtering in subsequent single-case analysis. The pipeline is illustrated in~~ sequentially, as the

genetic cause of disease may stem from different variant types such as SNVs, CNVs, or structural variants. Each pass filters data for its purpose, producing both cohort-level and single-patient results within one reproducible framework (24; 25). As illustrated in ~~Box 2~~**Figure 1**, ~~and can be summarised asfollows~~the description can be written as:

"A cohort of patient WGS data was analysed to identify genetic determinants for phenotype X. ~~Initially, a~~ A flexible QV set was applied using the `pipeline v1`, which implements the `QV_SNV_INDEL_1` criteria to produce the prepared dataset (`dataset v1`). This dataset was ~~then~~ analysed alongside other modules (e.g. ~~and~~ `PCA_SNV_INDEL_v1` and `statistical_genomics_v1`) to derive a cohort-level association signal (Result 1). ~~Next, the same prepared dataset was~~ It was then re-filtered with ~~the stricter~~ stricter `QV_SNV_INDEL_2` criteria to identify known causal variants~~for each patient, yielding the final dataset () and resulting in individual case~~, yielding (`dataset v2`) and single-patient reports (Result 2)."

Box 2: Example diagrammatic representation

```
pipeline v1
└─ Flexible QV criteria
   ├─ QV_SNV_INDEL_1 → dataset v1
   │  ├─ PCA_SNV_INDEL_v1
   │  └─ statistical_genomics_v1 → Result 1
   ├─ dataset v1
   │  └─ Rare disease QV criteria
   │     └─ QV_SNV_INDEL_2 → dataset v2
   │        └─ single case report SNV INDEL v1 → Result 2
   └─ Joint analysis output
```

Joint analysis output from:
Result 1 = Cohort-level association signal (e.g. variant P-value).
Result 2 = Single variant report per patient.

2.2 ~~Usage in a Validation Study~~

~~In a validation study, we demonstrate the use of our~~

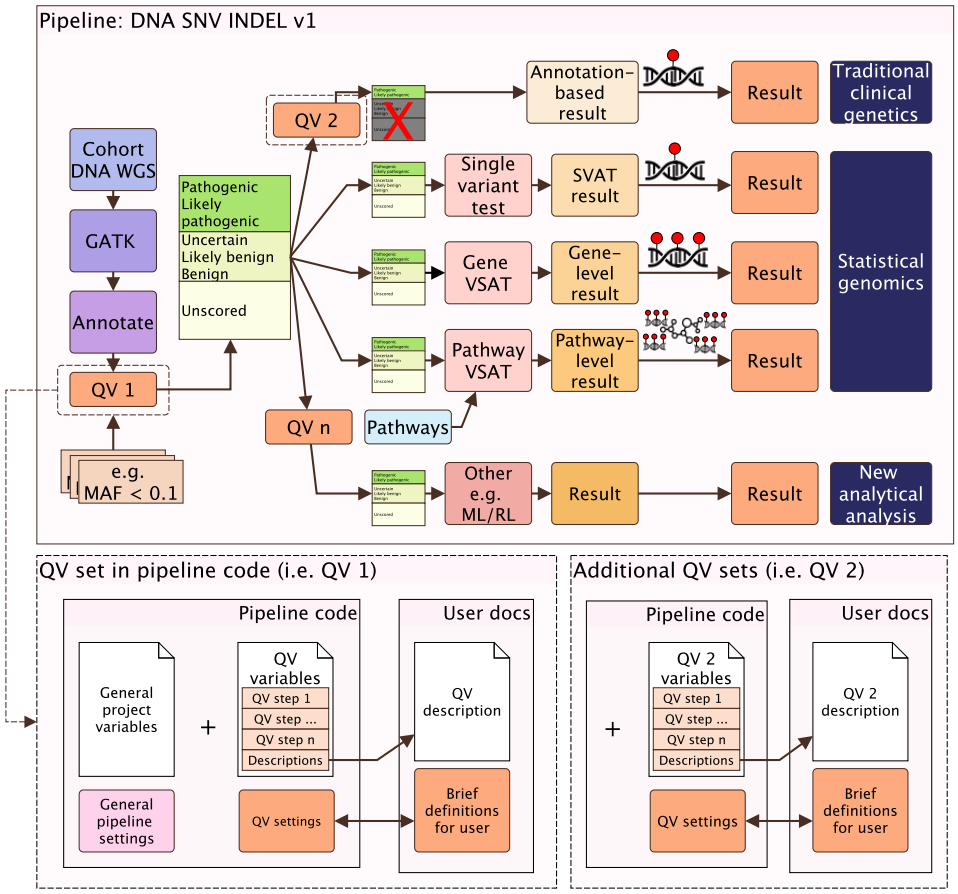


Figure 1: Summary of the QV application for a WGS pipeline. QV1 and QV2 are applied as sequential protocol steps. In this example, QV2 differs from QV1 by retaining only likely/pathogenic variants (indicated by a red X). The QV file loaded by the analysis pipeline comprises a description field (optional) and a variables field (mandatory). The QV criteria may be distributed across multiple pipeline steps.

2.2 Usage in a rare disease cohort validation study

We validated the QV ~~criteria framework compared to the conventional manual approach. This analysis was performed~~ framework on an in-house rare disease cohort of 940 individuals ~~, which had been pre-processed for . We used genome-wide set of variants which was filtering to target rare varaints~~ using Whole Exome Sequencing (WES) comparing a conventional manual implementation with a QV-based YAML configuration. The analysis targeted rare variants (Minor Allele Frequency (MAF) ~~< 0.01) restricted to~~ < 0.01) in known disease genes ~~based on~~ from the Genomics England ~~panel~~ "Primary immunodeficiency or monogenic inflammatory bowel disease~~," retrieved using our PanelAppRex R repository () "~~ panel, retrieved via PanelAppRex (26). This ~~provided us with a prepared dataset of 6026~~ yielded 6,026 candidate variants anno-

tated with 376 information sources. The dataset was , prepared in R using GuRu, our the GuRu variant interpretation tool that consolidates all annotation sources and scores variants as candidate causal, and was and imported from gVCF format as output files processed by Variant Effect Predictor (VEP).

We selected We applied the first eight American College of Medical Genetics and Genomics (ACMG) criteria for assigning pathogenicity scores to variants (1); six of these were relevant for pathogenicity scoring (1), six of which were relevant to this cohort. First, the analysis was performed manually by hard-coding each criterion in the pipeline script, reflecting a typical workflow . Second, the same criteria were imported from the YAML filefor the new framework approach, using the file "qv acmg svnindel criteria 20250225.yaml" (see **Box ??** or). The outputs from both methods were captured and compared.

Additional details of the YAML criteria in this set included definitions for The manual pipeline encoded each criterion directly, while the QV workflow read the same definitions from a YAML file. The YAML criteria included `ACMG_PS1` (identifying previously established known pathogenic amino acid changeschange), `ACMG_PS3` (supporting functional studies with matching inheritance patterns), and evidence), `ACMG_PS5` (covering compound heterozygositywith high-impact variants). The criteria for `ACMG_PM2` and `ACMG_PM3` assess variant frequency and in trans occurrences, respectively, while compound heterozygosity), and frequency- and segregation-based criteria (`PM2`, `PM3`). Criteria `PS2` and `PS4` were not applicable to in this cohort.

2.3 Usage in a GWAS validation study

We next applied the QV criteria framework to a GWAS using HapMap3 Phase 3 (R3) consensus genotypes on 1397 individuals (27). Again, two pipelines were executed with identical inputs and parameters: one hard-coded and one driven by the QV file. This QV set defined common GWAS thresholds: restriction to autosomal, biallelic SNPs; minimum sample call rate of 95%; variant call rate of 95%; minor allele frequency $\geq 1\%$; and Hardy–Weinberg equilibrium $p \geq 1 \times 10^{-6}$. After quality control, variants were LD-pruned and principal components (PC1–PC10) were computed, with sex included as an additional covariate. Logistic regression under an additive model was then performed with a binary simulated phenotype using PLINK. The outputs of the two pipelines were captured and compared across each main PLINK stage. Manhattan plots, Principal Component Analysis (PCA) plots, and `md5` checksums were used to confirm exact reproducibility between the hard-coded and QV-driven

analyses.

```
qv_set_id: acmg_sf_v3.2
-
acmg_pvs1:
  description_technical: >
    Null variants (IMPACT = HIGH) in genes where
    loss-of-function causes disease.
    Includes homozygous variants, dominant inheritance,
    and compound heterozygous cases.
    Compound heterozygosity is considered when both
    variants are HIGH impact. WARNING: Not phase checked.
  logic: "or"
  conditions:
    - condition:
        field: IMPACT
        value: "HIGH"
        operator: "=="
...
shasum -a 256 acmg_criteria.yaml | fold -w 32
d91fde41a5fff48631adecba38773d61
9ae8cd5cff9b9b42ef7f5efbd6bbfcdf
acmg_criteria.yaml
```

For benchmarking, Message-Digest Algorithm 5 (MD5) checksums were uniquely reported for the GWAS study because PLINK output files are exactly reproducible between runs. In contrast, VCF files used in the other validation studies include variable header fields such as BCFtools view command with a timestamp, which changes with each run and alters the MD5 value. For those cases, we instead report variant count and content.

3 Results

2.1 Validation Case StudyUsage in a WGS validation study with GIAB and Exomiser

We next applied the QV framework to a WGS trio analysis using the Genome In A

Bottle Chinese Trio (HG005-HG007, PRJNA200694, GRCh38 v4.2.1) of the National Institute of Standards and Technology (28). Two pipeline phases were executed with identical inputs and parameters: one hard-coded and one driven by the QV file. Both phases applied identical QC and study filters and included a gene-panel style analysis using the paediatric disorders panel (panel 486; 3,853 genes (26)). The upstream processing used BCFtools for region restriction using BED overlap, site-level thresholds on QUAL and INFO/DP (using computed site depth from per-sample FORMAT/DP when absent), and per-sample thresholds on FORMAT/DP and FORMAT/GQ with exclusion of missing genotypes. Composite criteria were applied to require either all samples to pass or at least one sample to pass. The downstream filtered trio VCF was analysed with Exomiser using the same trio `.ped` input and without using Human Phenotype Ontology (HPO) terms.

~~We validated our~~

3 Results

3.1 Validation rare disease cohort case study

We validated the QV ~~protocol using~~ framework using WES analysis with ACMG-based criteria ~~for~~ on a rare disease cohort of 940 individuals~~. We then conducted the variant classification using two approaches: a conventional manual method with hard-coded criteria, and our new~~ , comparing a conventional pipeline with parameters defined internally (QV manual) to the new external YAML-based implementation (QV yaml). As shown in **Figure ?? ~~(B)~~S1**, the outputs from both methods were identical, demonstrating a 100% match. This confirmed that our framework of a standalone, shareable, QV criteria file can be imported and applied programmatically with equivalent accuracy, providing a reproducible resource that is adaptable across different pipelines and programming environments.

~~ABSummary of the QV application for a WGS pipeline. In panel (A), 1 and 2 are presented as sequentially piped protocol steps. In this example, 2 differs from 1 by retaining only likely/pathogenic variants (indicated by a red X). The QV file loaded by the analysis pipeline comprise a description field (optional) and a variables field (mandatory). The criteria may be spread throughout the pipeline. (B) Validation case study using an criteria subset, demonstrating a 100% match between manually encoded and standalone YAML-based (`qv_files/acmg_criteria.yaml`) for assigning pathogenicity scores.~~

3.2 Validation in a common variant GWAS

To demonstrate the integration of the QV framework with established best practices in GWAS (29), we validated it in a standard HapMap3 Phase 3 GWAS by again running two equivalent analyses: a conventional pipeline with parameters defined internally and a YAML-based implementation that externalised all settings. As shown in **Figure S2**, the Manhattan and PCA plots were identical between the two methods, and the MD5 checksums of all PLINK outputs matched exactly. These results confirm that QV parameterisation reproduces the original workflow precisely while improving clarity, transparency, and reusability.

3.3 ~~Implications~~In the validation Validation in a WGS study ~~, we applied criteria for variant interpretation~~with GIAB and Exomiser

To demonstrate the ease and benefit of using QV parameterisation in established WGS analysis pipelines, we conducted a trio validation study using the Genome in a Bottle (GIAB) Chinese Trio (HG005-HG007, GRCh38 v4.2.1) and the Exomiser tool for variant annotation and interpretation (30). Two equivalent analyses were run: one with hard-coded thresholds and one using an external QV YAML file specifying the same parameters. Both applied identical QC and study filters and restricted analysis to the PanelAppRex paediatric disorders panel (3,853 genes). Results were identical: variant counts matched at each step, and Exomiser outputs produced the same candidate genes and variants. ~~In clinical genetics,for instance, the resulting output can be used to retrieve candidate pathogenic variantsusing scoring methods (1; 5). Application of additional~~ **Figure S3** shows this agreement. This validation confirms that a shareable QV ~~sets, such as the widely used set for clinical reporting (17), can be used to confirm any secondary findings~~file reproduces the full variant interpretation workflow exactly, while aligning with established variant effect predictors and interpretation tools (30–32). Benchmarking showed that QV files introduce no computational overhead and scale equivalently to conventional implementations (**Supplemental 10.2**, **Figure S4**).

~~In a clinical setting it is necessary to bridge the gap between technical detail and lay understanding. By explicitly documenting variant qualifying criteria and making~~

3.4 Implications

General applicability and reproducibility

Across validation studies, the QV ~~data accessible, our framework builds trust and supports meaningful (33). The~~ framework reproduced conventional workflows in which parameters are embedded within scripts, while externalising those same variables into a portable, shareable format. The framework itself performs no filtering, calling, annotation, or interpretation, but provides a machine-readable layer for defining and reusing the qualifying variables that underpin these analyses. It complements tools such as GATK and BCFtools for processing, Ensembl VEP, SnpEff, FAVOR, and WGSA for variant effect prediction (31), and Exomiser and VarFish for interpretation (30; 32), by making their analytic criteria explicit.

Scalability and interoperability with genomic tools

The validation studies, covering clinical interpretation, genome-wide association analysis, and WGS trio interpretation, demonstrate that the QV ~~file adapts by integrating the main criteria variables with optionally dedicated fields for both technical description and description. This approach captures the analysis intent defined by the~~ framework generalises across distinct genomic contexts without altering analytical outcomes or adding computational overhead. The format further allows users to define, combine, and extend their own QV ~~set creator and embeds patient preferences from the start.~~ sets using simple declarative syntax, providing a scalable approach for reproducible genomics.

Traceability and confirmation of applied clinical standards

Each QV file includes a persistent identifier and checksum that can be stored in Electronic Health Record (EHR) or laboratory systems such as EPIC, Cerner, Clinisys, or REDCap. This links each patient's analysis (including any associated PPIE input) to the exact QV set used, enabling transparent, auditable, and FAIR-compliant reporting. A clinician or molecular pathologist viewing a result in EPIC or Cerner can access the linked `qv_set_id` to verify the applied standards and filtering criteria. Automated genomic reports should include these details by default, ensuring full traceability without requiring access to the pipeline. For example, ~~patient preferences recorded in~~ if a patient asks whether their genome was screened for breast cancer due to variants in *BRCA1* or *BRCA2*, the ~~description can be automatically incorporated~~

into a genetic report without additional interpretation, ensuring clarity and consistency throughout the analysis. This transparency guarantees that both experts and laypersons receive information in a format suited to their needs, thereby improving diagnostic traceability and accelerating the translation of genetic research into clinical practiceEHR-linked report referencing "qv acmg sf v3.3 20250828.json" confirms that the ACMG secondary findings guideline (v3.3) (17) was applied, including its defined gene set, thresholds, version, and standard.

4 Summary

This paper introduces a framework for integrating qualifying variants into genomic analysis pipelines, enhancing reproducibility, interpretability and the seamless translation of research findings into clinical practice.

5 Funding

This project was supported through the grant Swiss National Science Foundation 320030_201060, and NDS-2021-911 (SwissPedHealth) from the Swiss Personalized Health Network and the Strategic Focal Area 'Personalized Health and Related Technologies' of the ETH Domain (Swiss Federal Institutes of Technology).

6 Acknowledgements

Acknowledgements We would like to thank all the patients and families who have been providing advice on SwissPedHealth and its projects, as well as the clinical and research teams at the participating institutions.

7 Contributions

DL designed the work and contributed to the manuscript. AS, SB, VS, DH, SÖ, JA, SF contributed to the manuscript. JF, SF, LJS LJS and JF supervised the work, manuscript, and applied for funding.

8 Competing interests

The authors declare no competing interests.

9 Ethics statement

~~Summary statistics were used from studies which have been previously reported and~~ The projects were approved by the respective ethics committees of all participating centers (Cantonal Ethics Committee Bern, approval number KEK-029/11) and the study was conducted in accordance with the Declaration of Helsinki.

References

[1] Sue Richards, Nazneen Aziz, Sherri Bale, David Bick, Soma Das, Julie Gastier-Foster, Wayne W Grody, Madhuri Hegde, Elaine Lyon, Elaine Spector, et al. Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the american college of medical genetics and genomics and the association for molecular pathology. *Genetics in medicine*, 17 (5):405–423, 2015.

[2] Marilyn M Li, Michael Datto, Eric J Duncavage, Shashikant Kulkarni, Neal I Lindeman, Somak Roy, Apostolia M Tsimberidou, Cindy L Vnencak-Jones, Daynna J Wolff, Anas Younes, et al. Standards and guidelines for the interpretation and reporting of sequence variants in cancer: a joint consensus recommendation of the association for molecular pathology, american society of clinical oncology, and college of american pathologists. *The Journal of molecular diagnostics*, 19(1):4–23, 2017.

[3] Quan Li and Kai Wang. Intervar: clinical interpretation of genetic variants by the 2015 acmg-amp guidelines. *The American Journal of Human Genetics*, 100 (2):267–280, 2017.

[4] Erin Rooney Riggs, Erica F Andersen, Athena M Cherry, Sibel Kantarci, Hutton Kearney, Ankita Patel, Gordana Raca, Deborah I Ritter, Sarah T South, Erik C Thorland, et al. Technical standards for the interpretation and reporting of constitutional copy-number variants: a joint consensus recommendation of the american college of medical genetics and genomics (acmge and the clinical genome resource (clingen). *Genetics in Medicine*, 22(2):245–257, 2020.

[5] Sean V Tavtigian, Steven M Harrison, Kenneth M Boucher, and Leslie G Biesecker. Fitting a naturally scaled point system to the acmg/amp variant classification guidelines. *Human mutation*, 41(10):1734–1737, 2020.

[6] Brent S Pedersen, Joe M Brown, Harriet Dashnow, Amelia D Wallace, Matt Velinder, Martin Tristani-Firouzi, Joshua D Schiffman, Tatiana Tvrdik, Rong Mao, D Hunter Best, et al. Effective variant filtering and expected candidate variant yield in studies of rare human disease. *NPJ Genomic Medicine*, 6(1):1–8, 2021.

[7] Carl A Anderson, Fredrik H Pettersson, Geraldine M Clarke, Lon R Cardon, Andrew P Morris, and Krina T Zondervan. Data quality control in genetic case-control association studies. *Nature protocols*, 5(9):1564–1573, 2010. URL https://doi.org/10.1038/nprot.2010.116.

[8] Emil Uffelmann, Qin Qin Huang, Nchangwi Syntia Munung, Jantina De Vries, Yukinori Okada, Alicia R Martin, Hilary C Martin, Tuuli Lappalainen, and Danielle Posthuma. Genome-wide association studies. *Nature Reviews Methods Primers*, 1(1):59, 2021. URL https://doi.org/10.1038/s43586-021-00056-9.

[9] Hannah Wand, Samuel A Lambert, Cecelia Tamburro, Michael A Iacocca, Jack W O'Sullivan, Catherine Sillari, Iftikhar J Kullo, Robb Rowley, Jacqueline S Dron, Deanna Brockman, et al. Improving reporting standards for polygenic scores in risk prediction studies. *Nature*, 591(7849):211–219, 2021.

[10] Samuel A Lambert, Laurent Gil, Simon Jupp, Scott C Ritchie, Yu Xu, Annalisa Buniello, Aoife McMahon, Gad Abraham, Michael Chapman, Helen Parkinson, et al. The polygenic score catalog as an open database for reproducibility and systematic evaluation. *Nature Genetics*, 53(4):420–425, 2021.

[11] Brent S Pedersen and Aaron R Quinlan. Vcfexpress: flexible, rapid user-expressions to filter and format VCFs. *Bioinformatics*, 41(3): btaf097, March 2025. ISSN 1367-4811. doi: 10.1093/bioinformatics/btaf097. URL https://academic.oup.com/bioinformatics/article/doi/10.1093/bioinformatics/btaf097/8051444.

[12] Henri E. Bal, Jennifer G. Steiner, and Andrew S. Tanenbaum. Programming languages for distributed computing systems. *ACM Computing Surveys*, 21(3): 261–322, September 1989. ISSN 0360-0300, 1557-7341. doi: 10.1145/72551.72552. URL https://dl.acm.org/doi/10.1145/72551.72552.

[13] Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B. Hall, Christopher H. Tomkins-Tinch, Vanessa Sochat, Jan Forster, Soohyun Lee, Sven O. Twardziok, Alexander Kanitz, Andreas Wilm, Manuel Holtgrewe, Sven Rahmann, Sven Nahnsen, and Johannes Köster. Sustainable data analysis with Snakemake. *F1000Research*, 10:33, January 2021. ISSN 2046-1402. doi: 10.12688/f1000research.29032.1. URL https://f1000research.com/articles/10-33/v1.

[14] Paolo Di Tommaso, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. Nextflow enables reproducible computational workflows. *Nature Biotechnology*, 35(4):316–319, April 2017. ISSN 1087-0156, 1546-1696. doi: 10.1038/nbt.3820. URL https://www.nature.com/articles/nbt.3820.

[15] Gundula Povysil, Slavé Petrovski, Joseph Hostyk, Vimla Aggarwal, Andrew S. Allen, and David B. Goldstein. Rare-variant collapsing analyses for complex traits: guidelines and applications. *Nature Reviews Genetics*, 20(12):747–759, 2019. doi: 10.1038/s41576-019-0177-4. URL https://doi.org/10.1038/s41576-019-0177-4.

[16] Elizabeth T Cirulli, Brittany N Lasseigne, Slavé Petrovski, Peter C Sapp, Patrick A Dion, Claire S Leblond, Julien Couthouis, Yi-Fan Lu, Quanli Wang, Brian J Krueger, et al. Exome sequencing in amyotrophic lateral sclerosis identifies risk genes and pathways. *Science*, 347(6229):1436–1441, 2015.

[17] David T Miller, Kristy Lee, Noura S Abul-Husn, Laura M Amendola, Kyle Brothers, Wendy K Chung, Michael H Gollob, Adam S Gordon, Steven M Harrison, Ray E Hershberger, et al. Acmg sf v3. 2 list for reporting of secondary findings in clinical exome and genome sequencing: a policy statement of the american college of medical genetics and genomics (acmg). *Genetics in Medicine*, 25(8): 100866, 2023.

[18] Nathan D Olson, Justin Wagner, Nathan Dwarshuis, Karen H Miga, Fritz J Sedlazeck, Marc Salit, and Justin M Zook. Variant calling and benchmarking in an era of complete human genome sequences. *Nature Reviews Genetics*, 24(7): 464–483, 2023.

[19] James J Lee, Robbee Wedow, Aysu Okbay, Edward Kong, Omeed Maghzian, Meghan Zacher, Tuan Anh Nguyen-Viet, Peter Bowers, Julia Sidorenko, Richard Karlsson Linnér, et al. Gene discovery and polygenic prediction from a

1.1-million-person gwas of educational attainment. *Nature genetics*, 50(8):1112, 2018.

[20] Philip R Jansen, Kyoko Watanabe, Sven Stringer, Nathan Skene, Julien Bryois, Anke R Hammerschlag, Christiaan A de Leeuw, Jeroen S Benjamins, Ana B Muñoz-Manchado, Mats Nagel, et al. Genome-wide analysis of insomnia in 1,331,010 individuals identifies new risk loci and functional pathways. *Nature genetics*, 51(3):394–403, 2019.

[21] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9, 2016.

[22] Eelke van der Horst, Deepak Unni, Femke Kopmels, Jan Armida, Vasundra Touré, Wouter Franke, Katrin Crameri, Elisa Cirillo, and Sabine Österle. Bridging clinical and genomic knowledge: An extension of the sphn rdf schema for seamless integration and fairification of omics data. 2023.

[23] Vasundra Touré, Philip Krauss, Kristin Gnodtke, Jascha Buchhorn, Deepak Unni, Petar Horki, Jean Louis Raisaro, Katie Kalt, Daniel Teixeira, Katrin Crameri, et al. Fairification of health-related data using semantic web technologies in the swiss personalized health network. *Scientific Data*, 10(1):127, 2023.

[24] Geraldine Van der Auwera and Brian D. O'Connor. *Genomics in the cloud: using Docker, GATK, and WDL in Terra*. O'Reilly, Beijing Boston Farnham Sebastopol Tokyo, first edition edition, 2020. ISBN 978-1-4919-7519-0 978-1-4919-7516-9 978-1-4919-7512-1.

[25] Xihao Li, Han Chen, Margaret Sunitha Selvaraj, Eric Van Buren, Hufeng Zhou, Yuxuan Wang, Ryan Sun, Zachary R McCaw, Zhi Yu, Min-Zhi Jiang, et al. A statistical framework for multi-trait rare variant analysis in large-scale whole-genome sequencing studies. *Nature Computational Science*, pages 1–19, 2025.

[26] Dylan Lawless. PanelAppRex aggregates disease gene panels and facilitates sophisticated search. March 2025. doi: 10.1101/2025.03.20.25324319. URL http://medrxiv.org/lookup/doi/10.1101/2025.03.20.25324319.

[27] Susan Fairley, Ernesto Lowy-Gallego, Emily Perry, and Paul Flicek. The International Genome Sample Resource (IGSR) collection of open human genomic variation resources. *Nucleic Acids Research*, 48(D1):D941–D947, January 2020. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkz836.

[28] Justin Wagner, Nathan D. Olson, Lindsay Harris, Ziad Khan, Jesse Farek, Medhat Mahmoud, Ana Stankovic, Vladimir Kovacevic, Byunggil Yoo, Neil Miller, Jeffrey A. Rosenfeld, Bohan Ni, Samantha Zarate, Melanie Kirsche, Sergey Aganezov, Michael C. Schatz, Giuseppe Narzisi, Marta Byrska-Bishop, Wayne Clarke, Uday S. Evani, Charles Markello, Kishwar Shafin, Xin Zhou, Arend Sidow, Vikas Bansal, Peter Ebert, Tobias Marschall, Peter Lansdorp, Vincent Hanlon, Carl-Adam Mattsson, Alvaro Martinez Barrio, Ian T. Fiddes, Chunlin Xiao, Arkarachai Fungtammasan, Chen-Shan Chin, Aaron M. Wenger, William J. Rowell, Fritz J. Sedlazeck, Andrew Carroll, Marc Salit, and Justin M. Zook. Benchmarking challenging small variants with linked and long reads. *Cell Genomics*, 2(5):100128, May 2022. ISSN 2666979X. doi: 10.1016/j.xgen.2022.100128.

[29] Emil Uffelmann, Qin Qin Huang, Nchangwi Syntia Munung, Jantina De Vries, Yukinori Okada, Alicia R. Martin, Hilary C. Martin, Tuuli Lappalainen, and Danielle Posthuma. Genome-wide association studies. *Nature Reviews Methods Primers*, 1(1):59, August 2021. ISSN 2662-8449. doi: 10.1038/s43586-021-00056-9.

[30] Valentina Cipriani, Nikolas Pontikos, Gavin Arno, Panagiotis I. Sergouniotis, Eva Lenassi, Penpitcha Thawong, Daniel Danis, Michel Michaelides, Andrew R. Webster, Anthony T. Moore, Peter N. Robinson, Julius O.B. Jacobsen, and Damian Smedley. An Improved Phenotype-Driven Tool for Rare Mendelian Variant Prioritization: Benchmarking Exomiser on Real Patient Whole-Exome Data. *Genes*, 11(4):460, April 2020. ISSN 2073-4425. doi: 10.3390/genes11040460.

[31] Cristian Riccio, Max L. Jansen, Linlin Guo, and Andreas Ziegler. Variant effect predictors: A systematic review and practical guide. *Human Genetics*, 143(5):625–634, May 2024. ISSN 1432-1203. doi: 10.1007/s00439-024-02670-5.

[32] Manuel Holtgrewe, Oliver Stolpe, Mikko Nieminen, Stefan Mundlos, Alexej Knaus, Uwe Kornak, Dominik Seelow, Lara Segebrecht, Malte Spielmann, Björn Fischer-Zirnsak, Felix Boschann, Ute Scholl, Nadja Ehmke, and Dieter Beule. VarFish: Comprehensive DNA variant analysis for diagnostics and research. *Nu-*

cleic Acids Research, 48(W1):W162–W169, July 2020. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkaa241.

[33] Zoë Slote Morris, Steven Wooding, and Jonathan Grant. The answer is 17 years, what is the question: understanding time lags in translational research. *Journal of the Royal Society of Medicine*, 104(12):510–520, December 2011. ISSN 0141-0768, 1758-1095. doi: 10.1258/jrsm.2011.110180. URL https://journals.sagepub.com/doi/10.1258/jrsm.2011.110180.

10 Supplemental

Application of qualifying variants for genomic analysis.

10.1 Validation study figures

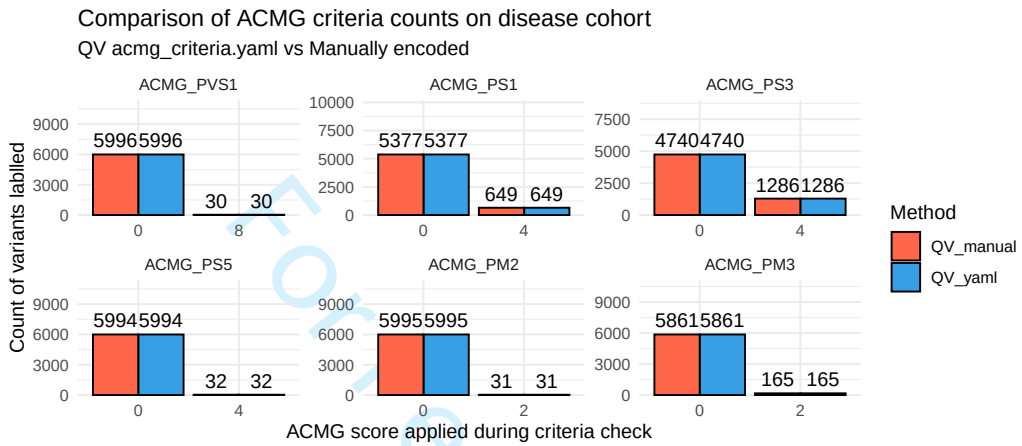


Figure S1: Validation case study of a rare disease cohort of 940 WES individuals using an ACMG criteria subset, demonstrating a 100% match between manually encoded and standalone YAML-based QV for assigning pathogenicity scores.

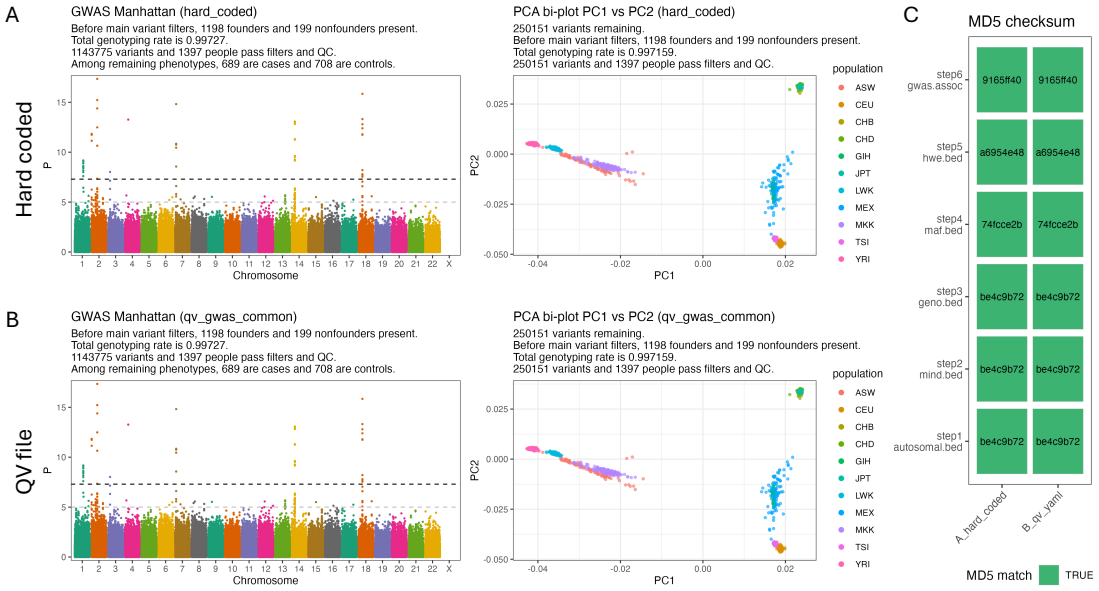
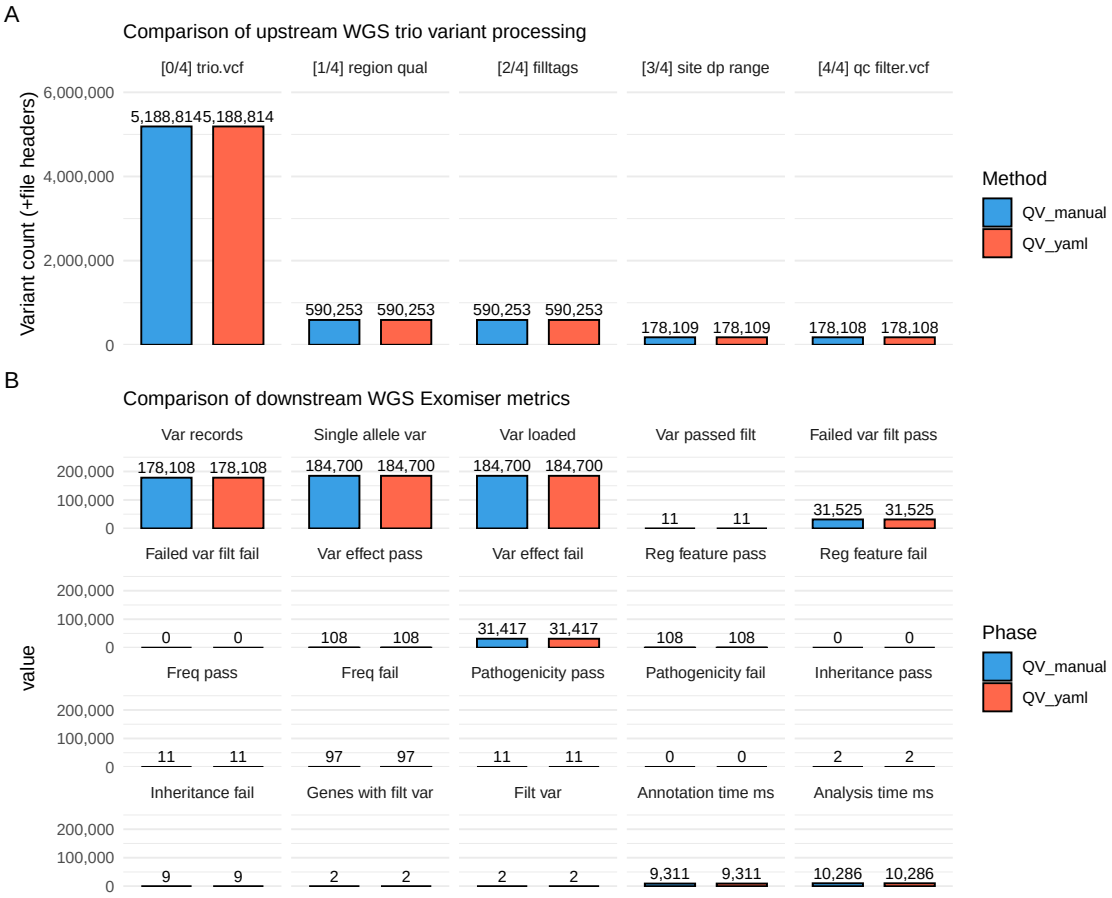


Figure S2: Validation in GWAS using QV parameterisation. (A) GWAS of simulated binary phenotypes in HapMap3 Phase 3 (R3) using a traditional variable embedded pipeline. Shown are the Manhattan plot of logistic regression results (left) and correction for population structure with principal component analysis (PC1 vs PC2, right). (B) Identical GWAS using a QV YAML configuration file. The Manhattan and PCA results are indistinguishable from panel A. (C) Verification of reproducibility. MD5 checksums of the main PLINK outputs are identical between panels A and B. The steps included processing of autosomal biallelic SNPs, sample call rate, variant call rate, minor allele frequency, Hardy–Weinberg equilibrium, and association results. The QV file encoded these thresholds (sample call rate ≥95%, variant call rate ≥95%, MAF ≥1%, HWE p ≥1e-6, autosomal biallelic SNPs only) together with covariates (sex and PC1-PC10) and logistic regression settings. This confirms that a shareable QV file reproduces hard-coded pipelines exactly while improving transparency and reusability.



phase	MOI	Effect	HGVS	Genotype
QV_manual	AD	frameshift	RYR2:ENST00000366574.7:c.6493dup:p.(Leu2165Profs*44)	0/1:./.:./.
QV_yaml	AD	frameshift	RYR2:ENST00000366574.7:c.6493dup:p.(Leu2165Profs*44)	0/1:./.:./.
QV_manual	AR	missense	TTN:ENST00000589042.5:c.36982A>G:p.(Thr12328Ala)	1|1:0/1:0/1
QV_yaml	AR	missense	TTN:ENST00000589042.5:c.36982A>G:p.(Thr12328Ala)	1|1:0/1:0/1

Figure S3: Validation of the trio Exomiser pipeline using QV parameterisation. **(A)** summarises upstream processing counts by file, **(B)** compares downstream Exomiser metrics, and **(C)** shows the key variant fields for the two variants identified. Variant counts in all panels confirm that intermediate files and final outputs are identical between configurations. The five preprocessing stages shown in (A) are: (0) input trio VCF, (1) gene panel region and quality filtering, (2) tag annotation, (3) site-level depth range filtering, and (4) final QC-filtered VCF. MOI, mode of inheritance; HGVS, Human Genome Variation Society nomenclature.

10.2 Computational benchmark

Runtime performance was equivalent between traditional and QV-based pipelines, as both read identical parameters from different sources. In the WGS trio validation study, pre-processing steps including filtering, QC, and gene panel selection completed in 16–17 seconds, with a median difference of ~0.5 seconds favouring the QV YAML pipeline (**Figure S4**). An incidental one-off 5 second delay arose from Singularity initialisation for the `yq` utility (step 0), a system-specific effect on our HPC and unrelated to the framework itself.

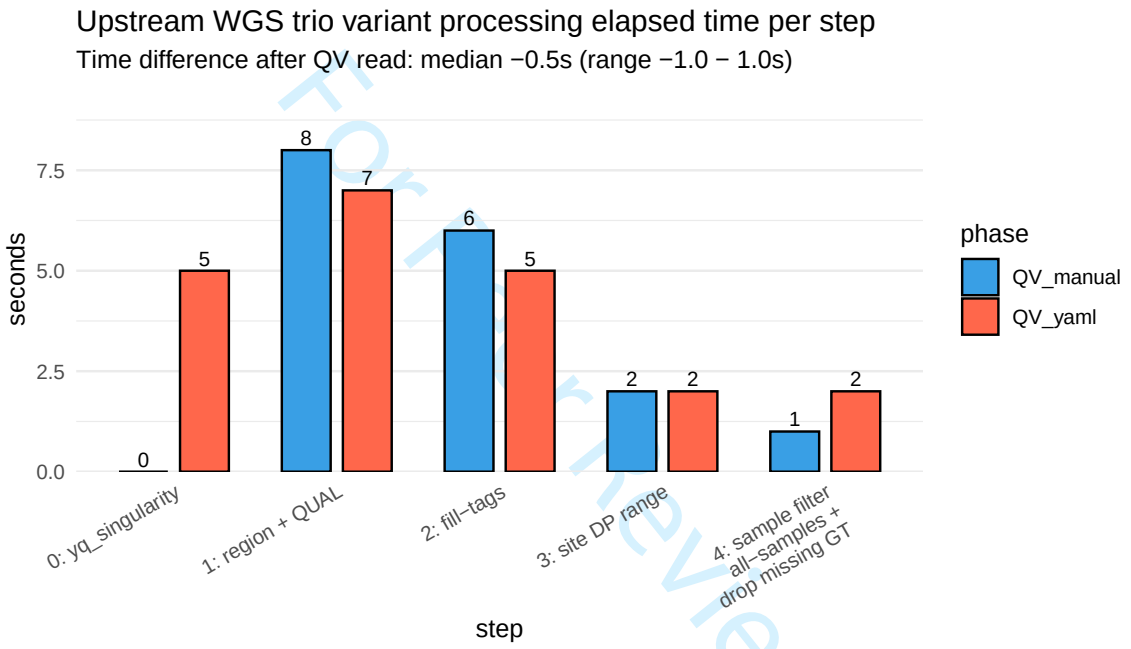


Figure S4: Benchmark of upstream preprocessing times in the WGS trio pipeline comparing QV-based and traditional (manually parameterised) configurations. Stepwise elapsed times were nearly identical across both methods (median difference ~0.5 s), with a fixed 5 s overhead from optional Singularity initialisation of `yq` in the QV pipeline. The four preprocessing steps correspond to: (1) gene panel region and quality filtering of the trio VCF, (2) annotation of variant tags, (3) site-level depth range filtering, and (4) per-sample genotype filtering and exclusion of missing genotypes. All steps used `BCFtools` on VCF preprocessing, as illustrated in **Figure S3 (A)**.

10.3 How to build a QV file

We recommend YAML or JSON for portability and adoption. You can build a QV in three ways:

Option 1: use the HTML QV builder (Zenodo)

1. Open the HTML builder from the Zenodo repository.
2. Enter simple `key=value` statements in the left pane.
3. Copy or download the generated YAML.

Example input lines:

```
meta qv_set_id="qv_gwas_common_v1_20250827"
meta version="1.0.0"
meta title="GWAS common QC"
meta authors=Alice,Bob
meta tags=GWAS,QC,PCA
filter maf_minimum field=MAF operator=">=" value=0.01 desc="Minimum MAF"
filter hwe field=HWE_P operator=">=" value=1e-6 logic=keep_if
filter region_include desc="include panel" field=OVERLAP(targets.exome.bed)
    >>>> operator=">=" value=1 logic=keep_if
criteria disease_panel logic=and desc="HIGH impact within panel"
criteria disease_panel field=IMPACT operator="==" value=HIGH
criteria disease_panel field=OVERLAP(targets.exome.bed) operator=">=" value=1
meta description_patient=
    >>>> "There is a strong family history of early heart attacks."
meta description_ppie=
    >>>> "The PPIE group reviewed and approved the criteria on 2025-08-15."
```

Option 2: write YAML by hand

Minimal pattern:

```
meta:
  qv_set_id: qv_disease_panel_v1_20250828
  version: 1.0.0
  title: Disease panel filter
filters:
  region_include:
    description: Restrict to curated disease gene panel
    logic: keep_if
```

```
    field: OVERLAP(targets.disease_panel.bed)
    operator: ">="
    value: 1
criteria:
  pathogenic:
    description: Variant classified as pathogenic or likely pathogenic
    logic: and
    conditions:
      - group: any_of:start
      - { field: CLASS, operator: "==", value: P }
      - { field: CLASS, operator: "==", value: LP }
      - group: any_of:end
notes:
  - Gene panel file defines the target regions
```

Option 3: write JSON

JSON equivalent of the minimal example:

```
{
  "meta": {
    "qv_set_id": "qv_disease_panel_v1_20250828",
    "version": "1.0.0",
    "title": "Disease panel filter"
  },
  "filters": {
    "region_include": {
      "description": "Restrict to curated disease gene panel",
      "logic": "keep_if",
      "field": "OVERLAP(targets.disease_panel.bed)",
      "operator": ">=",
      "value": 1
    }
  },
  "criteria": {
    "pathogenic": {
      "description": "Variant classified as pathogenic or likely pathogenic",
      "logic": "and",
```

```
      "conditions": [
        { "group": "any_of:start" },
        { "field": "CLASS", "operator": "==", "value": "P" },
        { "field": "CLASS", "operator": "==", "value": "LP" },
        { "group": "any_of:end" }
      ]
    }
  },
  "notes": [
    "Gene panel file defines the target regions"
  ]
}
```

Checksum and register

Record the checksum and register the release:

```
sha256sum qv/examples/qv_disease_panel_v1_20250828.yaml

# qv/registry/releases.csv
qv_set_id, version, checksum, file, date
qv_disease_panel_v1_20250828,1.0.0, ef6cf810b994...,
    > qv_disease_panel_v1_20250828.yaml, 2025-08-28
```

Versioning and IDs

Use a stable `qv_set_id` plus semantic version. Update the version on any change that affects selection or interpretation. Keep one file per release and never mutate published files.

Use in a workflow

Point your pipeline to the QV file:

```
# workflows/.../config.yaml
qv_file: ".../qv/registry/qv_disease_panel_v1_20250828.yaml"
```

It can be read programmatically at runtime, for example using `yq` in shell-based workflows or `yaml::read_yaml()` in R, providing the same parameters that would otherwise be embedded within pipeline configurations.