MOLGENIS VIP: an open-source and modular pipeline for high-throughput and integrated DNA variant analysis

Authors

W.T.K. Maassen[1,2], L.F. Johansson[1,2], B. Charbon[1,2], D. Hendriksen[1,2], S. van den Hoek[1,2], M.K. Slofstra[1,2], R. Mulder[2], M.T. Meems-Veldhuis[2], R. Sietsma[1,2], H.H. Lemmink[2], C.C. van Diemen[2], M.E. van Gijn[3], M.A. Swertz†[1,2], K.J. van der Velde†[1,2]

† These authors contributed equally to this work and share last authorship

Affiliations

1. Genomics Coordination Center, University of Groningen and University Medical Center Groningen, Antonius Deusinglaan 1, 9713 AV, Groningen, the Netherlands.
2. Department of Genetics, University of Groningen and University Medical Center Groningen, Antonius Deusinglaan 1, 9713 AV, Groningen, the Netherlands.
3. Amsterdam UMC Genome Diagnostics (AGDx), Dept of Human Genetics, Amsterdam UMC, AGDx, Meibergdreef 9, 1105 AZ Amsterdam, Department of Genetics, University Medical Center Groningen, University Groningen, Groningen, the Netherlands.
4.

Abstract

In silico variant interpretation pipelines have become an integral part of genetics research and genome diagnostics. However, challenges remain for automated variant interpretation and candidate shortlisting. Their reliability is affected by variability in input data caused due the use of differing sequencing platforms, erroneous nomenclature and changing experimental conditions. Similarly, differences in predictive algorithms can result in discordant results. Finally, scalability is essential to accommodate large amounts of input data, such as in whole genome sequencing (WGS). To accelerate causal variant detection and innovation in genome diagnostics and research, we developed the MOLGENIS Variant Interpretation Pipeline (VIP). VIP is a flexible open-source computational pipeline that generates interactive reports of variants in whole exome sequencing (WES) and WGS data for expert interpretation. VIP can process short- and long-read data from different platforms and offers tools for increased sensitivity: a configurable decision-tree, filters based on human phenotype ontology (HPO) and gene inheritance that can be used to pinpoint disease-causing variants or finetune a query for specific variants. Here, alongside presenting VIP, we provide a step-by-step protocol for how to use VIP to annotate, classify and filter genetic variants of patients with a rare disease that has a suspected genetic cause. Finally, we demonstrate how VIP performs using 25,664 previously classified variants from the data sharing initiative of the Vereniging van Klinisch Genetische Laboratoriumdiagnostiek (VKGL), a cohort of 18 diagnosed patients from routine diagnostics and a cohort of 41 patients with a rare disease (RD) who were not diagnosed in routine diagnostics but were diagnosed using novel omics approaches within the EU-wide project to solve rare diseases (EU-Solve-RD). VIP requires bioinformatic knowledge to configure, but once configured, any diagnostic professional can perform an analysis within 5 hours.

Introduction

The field of clinical genetics focuses on the identification of genetic variants that cause disease. By understanding the pathogenesis of genetic diseases, we can provide patients with a prognosis and appropriate treatment[1-3]. Worldwide, around 350 million people are affected by one of 4440 RDs with a known genetic cause[4]. Pinpointing the causal RD variants among all the variants detected in targeted-enrichment sequencing, such as adaptive sampling and whole exome sequencing (WES), or even in whole genome sequencing (WGS) data, is not a trivial task and cannot be performed using a single strategy[2,5-7]. Several *in silico* pipelines have been created to combine published tools into a single solution[2,5]. Alissa Interpret (Agilent) is one example of a commercially developed tool in genome diagnostics, but there are also open-source alternatives, such as Kipoi and Scout[8-10].

Although the rapid growth of next generation sequencing (NGS) technologies has also accelerated the development of *in silico* pipelines, many challenges remain. To ensure reliability and reproducibility, pipelines must deal with the variability in quality and types of input data caused by different sequencing platforms and experimental conditions. Pipelines also need to accommodate different sources of information with different nomenclatures and ontologies, such as reference genomes, DNA annotation files and knowledge bases[2,11-15]. Further, variability in prediction algorithms, which originates from inherent biases in the datasets used for the development and training of these tools, often causes discordant results[2,16].

Additionally, the increasing volumes of sequencing data resulting from advances in NGS such as long-read and WGS require large-scale translation of results into the clinic[17]. To accomplish this, scalability must be ensured. Pipelines are also required to adapt to the continuous emergence of new bioinformatic methods and knowledge bases. Ideally, they should also include tools to interpret variants according to the American College of Medical Genetics (ACMG) guidelines[5,6,18], which helps to accommodate more advanced variant interpretation and prioritization methods[17,19].

Finally, because variant interpretation pipelines involve different algorithms and resources, their operation can require bioinformatic expertise. Most pipelines are restricted to individual institutions or bound to specific hardware configurations[17]. There are solutions that solve these problems using containerization and parallelization of the processes within the pipelines, but these are far from perfect[17]. Another example that often requires some level of bioinformatic expertise is the presentation and visualization of the results. This is typically of poor quality and difficult to understand, which negatively affects interpretation and translation toward the clinic[20].

To accelerate causal variant detection and innovation in genome diagnostics and research, we present the MOLGENIS Variant Interpretation Pipeline (VIP). VIP is a flexible open-source pipeline to generate self-contained, interactive reports of variants in WES, WGS, targeted NGS (tNGS) and adaptive sampling data for expert interpretation. To our knowledge, VIP is the only pipeline to incorporate the aforementioned into a single solution. It features integrated best-in-practice algorithms and protocols from routine diagnostics to facilitate classification of coding and non-coding variants according to the ACMG guidelines. VIP applies experience from the VKGL, the EU-wide project to solve rare diseases (EU-Solve-RD), the European Joint Program on Rare Diseases (EJP-RD,) and European infrastructural collaboration to accelerate access and sharing of research data (CINECA) (table 1)[12,21]. We containerized and parallelized VIP for straightforward deployment and large-scale data analysis. VIP is easily modified by updating its tools, annotations, and classification trees. Results are presented in coherent interactive reports for expert interpretation. VIP has already been used in several studies, such as the curation and expansion of human phenotype ontology (HPO) for systemic

autoinflammatory diseases and to study the use of a targeted gene panel in Dutch NGS-based newborn screening[22,23].

Here, we explain the different capabilities of VIP and how to use them. These include integrated best-in-practice annotation sources, such as our variant pathogenicity predictor CAPICE, and a step-by-step protocol for the analysis of NGS sequencing data, supporting researchers and diagnostic professionals in variant interpretation and classification. We demonstrate VIPs ability to classify and report any clinically relevant germline variants. Additionally, we demonstrate the classification of causal variants in representative cohorts of RD patients from the University Medical Center Groningen (UMCG) and the EU-Solve-RD project[21].

Resource	**Link**
CINECA	https://www.cineca-project.eu/
EJP RD	https://www.ejprarediseases.org/
File formats	https://gatk.broadinstitute.org/hc/en-us/articles/360035531812-GVCF-Genomic-Variant-Call-Format/
Git download	https://git-scm.com/downloads/
Illumina	https://emea.illumina.com/
Nextflow	https://www.nextflow.io/docs/
Oxford Nanopore Technology	https://nanoporetech.com/
PacBio HiFi	https://www.pacb.com/technology/hifi-sequencing/
VIP Github repository	https://github.com/molgenis/vip/
VIP online documentation	https://molgenis.github.io/vip/
VIP releases	https://github.com/molgenis/vip/releases/
VKGL	https://vkgl.molgeniscloud.org/

Table 1 **Online resources**

MOLGENIS VIP

VIP supports short- and long-read WGS, WES and otherwise targeted-enrichment WGS data from different sequencing platforms: PacBio HiFi long-read sequencing, Oxford Nanopore Tech (ONT) long-read sequencing and Illumina short-read sequencing (table 1). To annotate both coding and non-coding variants, VIP integrates different algorithms and knowledge bases and is divided into four modules: pre-processing, annotation, filtering and interactive reporting (figure 1). Depending on the input format, VIP starts one of four workflows: the FASTQ (raw human-readable sequencing output) workflow, the CRAM/BAM (Compressed Reference-oriented Alignment Map/Binary Alignment Map) workflow, the gVCF (Genomic Variant Call Format) or the VCF (Variant Call Format) workflow (figure 1). The sequence of the pre-processing steps is different for each workflow, but after arriving at a VCF file, the steps performed by the remaining modules are the same. From here, all variants are annotated using the different algorithms, knowledge bases, known genotype-phenotype relationships and inheritance patterns (figure 1, supplementary table 1). Finally, all variants are filtered using a user-specified decision tree and inheritance modes. This results in an interactive report containing a prioritized shortlist of classified variants of interest. Possible classifications are likely benign (LB), benign (B), variant of unknown significance (VUS), likely pathogenic (LP) and pathogenic (P). The raw output with the different annotations is also available as an unfiltered VCF file.

VIP has been developed in a combination of Java, JavaScript, Python, Perl and Shell scripting and runs on the command line. All processes performed by VIP are implemented using the workflow manager Nextflow, allowing parallel processing per chromosome to maximize performance. Apptainer is used for containerization, ensuring straightforward installation and reproducible deployment in different research or diagnostic environments[24,25]. Nextflow also creates intermediate files, allowing users to stop and resume the pipeline at any time during a run. This feature also allows users to analyze large amounts of data step-by-step and helps to accurately follow and reproduce all processes.

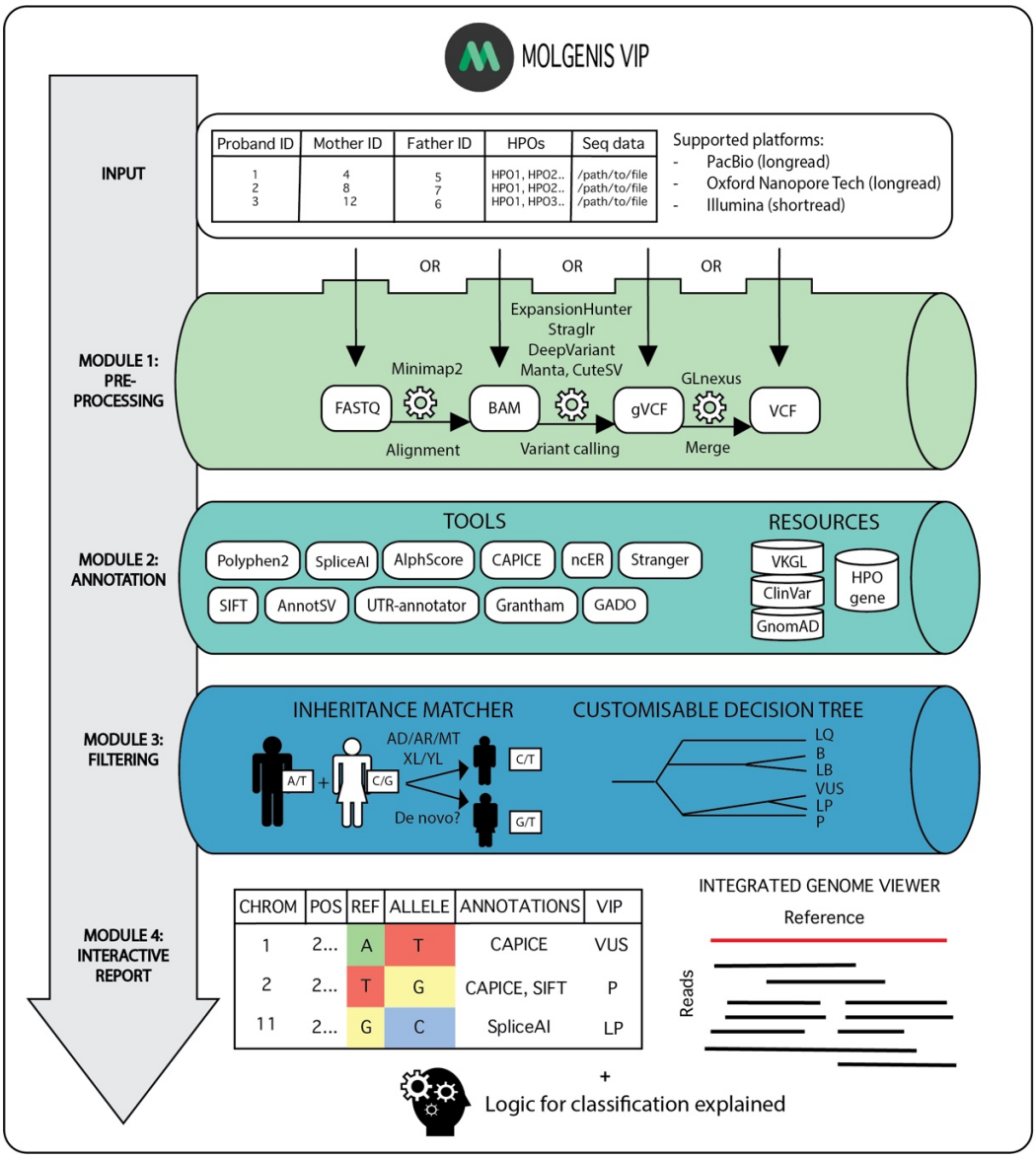


Figure 1 **Summary of modules in VIP.** As input, VIP requires a sample sheet in which the patient information is specified. In module 1, the input is validated and pre-processed, resulting in a VCF file. The workflow can be started at all points in the pre-processing step. Module 2 provides the variants in the VCF file with annotations from bioinformatic tools and resources (supplementary table 1). Module 3 filters the variants using a customizable decision tree and inheritance information from the previous modules. Finally, VIP generates an interactive report in which the logic for the classifications is explained. Diagnosticians and researchers can use this report for further interpretation of the variants and sharing of the results.

Application of the method

With VIP we aim to provide a complete computational procedure to align reads, followed by calling, annotating, classifying and finally filtering genetic variants for research and diagnostics of patients with a RD with a suspected genetic cause. VIP offers tools to increase sensitivity to find previously unsolved or unknown variants or to finetune a query to find specific variants. The annotation module and decision tree can be altered or expanded to incorporate new annotation tools or resources. Therefore, we encourage users to develop and validate their own decision trees and to expand VIP for their own needs.

Description of the workflow

Input

Diagnosticians and researchers can provide a sample sheet in the tab-separated values (TSV) format in which each row represents a sample taken from an individual. It contains the location of the sequencing data and metadata associated to the sample, such as the location of maternal and paternal sequencing data, phenotypic information, and the reference genome specified and sequencing method used. VIP supports several standard input formats for sequencing data produced by NGS: FASTQ, CRAM, BAM, VCF and gVCF files (table 1)[26-28]. In addition, phenotypic information can be provided using terms from the HPO system[29].

Module 1: pre-processing

When sequencing data is provided in FASTQ format, VIP initiates the FASTQ workflow and maps the files to the specified reference genome using Minimap2[30]. Next, BAM files are aligned and indexed by Samtools. Structural tandem repeats are detected using ExpansionHunter for short-read data and Straglr for long-read data[31,32]. Single nucleotide variants (SNVs), and insertions and deletions (indels) are detected by DeepVariant. GLnexus is used to merge proband, maternal and paternal VCF files into a single VCF for further processing[33]. Finally, structural variants are detected using Manta and cuteSV[34,35]. Sequencing data generated with the following platforms are supported: Illumina short-read (single and paired-end), ONT long-read sequencing and PacBio HiFi long-read sequencing (table 1).

Module 2: annotation

Next, the detected variants are annotated with different levels of information. Examples are allele frequencies from GnomAD, known gene-phenotype relationships from the HPO database and classifications validated by the VKGL. VIP also integrates our powerful CAPICE machine learning (ML)–based variant pathogenicity predictor with other best-in-class annotation sources, such as Ensembl Variant Effect Predictor (VEP) and SpliceAI. Supplementary table 1 provides a complete list of the default tools and knowledge bases, including references, that VIP uses to annotate variants. These annotations are used to classify and display variants in the interactive report. We use the plugin framework of Ensembl VEP to extend the existing functionality with new tools and annotations[36].

The two main annotation tools in VIP are CAPICE and SpliceAI. CAPICE is a ML- based method to predict the pathogenicity of SNVs and indels using diverse genomic features, such as genetic context, gene model annotations and evolutionary constraints. We chose CAPICE as it outperforms other similar pathogenicity predictors, such as CADD and REVEL[37]. Based on the study performed by Li et al. we

updated and re-trained CAPICE. Our benchmark study showed that a cut-off value of 0.5 results in a recall rate of 0.95, which we used in our decision tree. SpliceAI is a ML-based method that exploits the sequence surrounding a variant to determine the likelihood that the position in the pre-mRNA transcript is a splice donor or acceptor site[38]. SpliceAI outperforms most other splice variant predictors. Based on a study in which we performed a sensitivity and specificity analysis on published splice variants, we chose a cut-off value of 0.42 as this resulted in the highest sensitivity (supplementary data 1).

Module 3: filtering

Inheritance matcher

To study families for which genotypic and phenotypic information is available, diagnosticians can evaluate if variants segregate with disease or if they occurred *de novo*. This provides an additional level of evidence to determine if a variant is disease-causing. Therefore, VIP uses pedigree information consisting of trios (proband, mother and father) to check if the variant segregation matches disease disease-affected status, while taking into consideration the known inheritance modes of the gene of interest in the Clinical Genomic Database (CGD)[39]. Users can also provide VIP with known inheritance modes from the Online Mendelian Inheritance in Man (OMIM) database or other sources[39,40]. Both maternal and paternal genetic information can be specified in the sample sheet. VIP supports X-linked recessive, X-linked dominant, Y-linked recessive, Y-linked dominant, mitochondrial (MT), autosomal recessive, autosomal dominant and *de novo* inheritance modes. Additionally, VIP supports compound recessive inheritance patterns and inheritance with incomplete penetrance.

Customizable decision tree based on best-practices

VIP uses a decision tree to filter variants based on the results of the annotation module, which are collected in an interactive report. Users can compose a decision tree to fit their diagnostic workflow or research question. The decision tree is specified in a human-readable JavaScript object notation (JSON) formatted file. It can be customized by editing the different objects within this file. Each object represents criteria that ultimately lead to the predicted variant classification by VIP. VIP also produces intermediate files so that classification and filtering can be performed repeatedly using different decision trees. Figure 2 shows the default decision tree which is based on the VKGL and the Association for Clinical and Genetic Science (ACGS) guidelines and on collaboration with experts from Genome Diagnostics at the UMCG Genetics department[12,41,42].

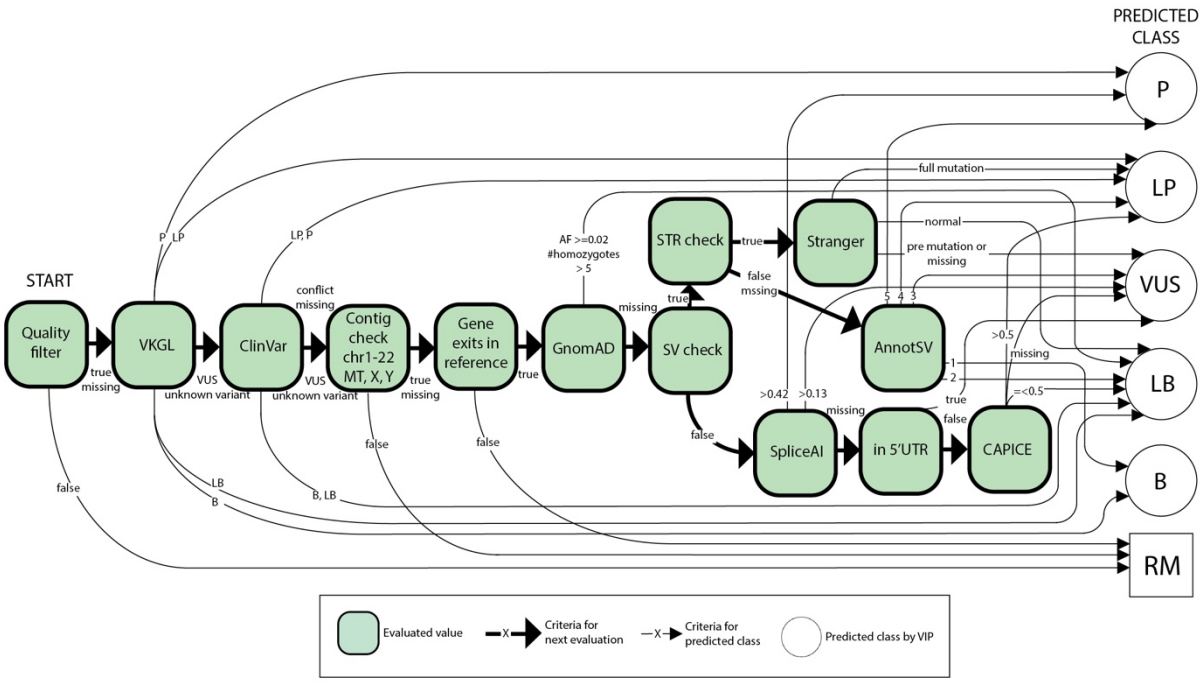


Figure 2 **Default decision tree.** The figure shows a schematic version of the default decision tree. The green blocks and bold arrows represent the general sequence of filtering steps and the values that are evaluated for each variant (see legend). Each value is calculated in the annotation module (supplementary table 1). Small arrows represent the decisions for the consequence classifications by VIP. VIP classifies the different consequences as B, LB, VUS, LP or P. After a variant is classified by VIP, it exits the filter tree. Variants with incorrect contigs or genes and low quality are removed (RM). Using the JSON format, each component in the decision tree can be customized to fit the workflow of the user.

Module 4: interactive report

The final set of variants that passed through the decision tree is provided as a self-contained, interactive hypertext markup language (HTML) report that can be opened in any recent internet browser (figure 3). This report contains the filtered list of annotated variants. Diagnosticians can use these annotations to navigate the report and study the criteria that were used to classify the variants. Additionally, basic information, such as the genomic position, reference and alternative alleles, the consequence, and the gene in which the variant is located, are provided. A more detailed view of all available annotations for each transcript is also available. Experts can perform diagnostic-relevant filtering, such as hiding variants with a genotype quality ≤ 20, variants with allelic imbalance and all variants predicted to be LB and B. Similarly, using the HPO-match filter button, variants not located in a gene associated to the HPO terms provided are hidden. Variants that do not match known monogenic inheritance patterns or variants that are not *de novo* can be hidden using the inheritance-match buttons. The report is also provided as a raw VCF file as input for other tools. For the FASTQ and BAM workflows, a genome viewer is available to view the reads that overlap with a variant.

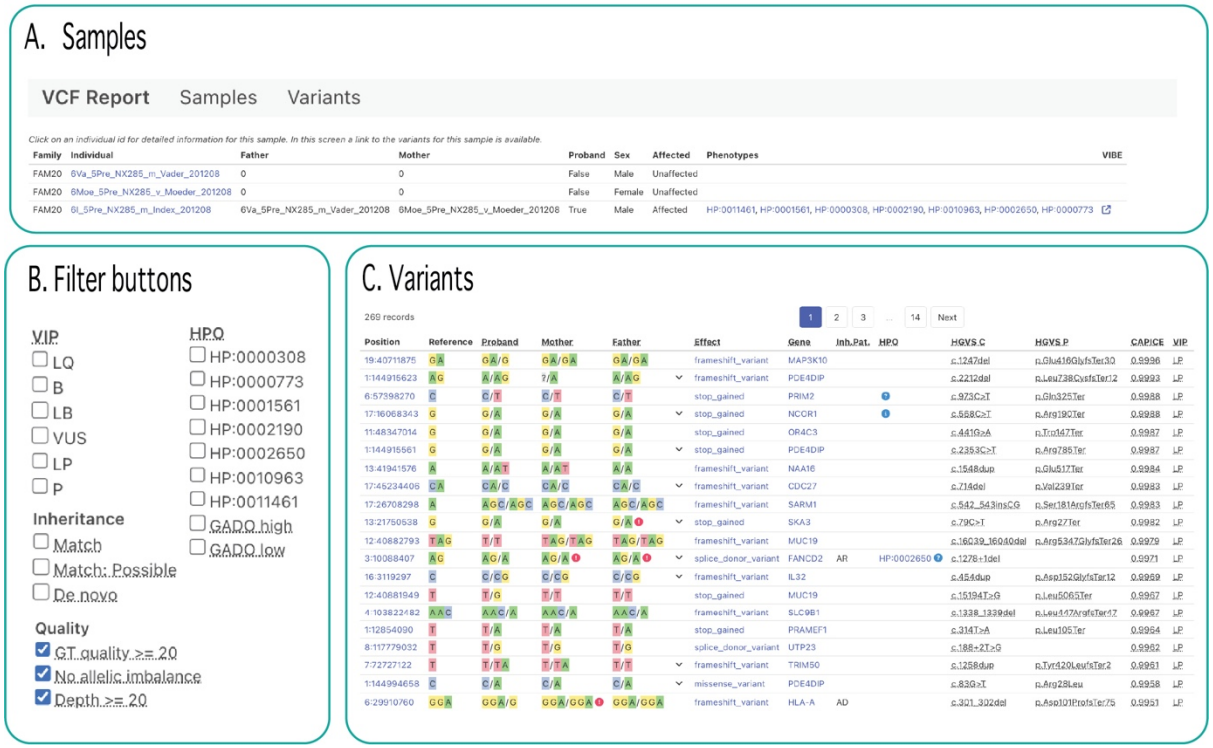


Figure 3 **Interactive report**. This figure shows an example of the base view of the interactive report. The interactive report opens in the sample screen (A). Here, all the individuals within a family are shown. To navigate to the list of variants with their predicted class, users can navigate to the variant view by clicking one of the individuals (C). The default variant view shows the variants and the consequence of the transcript with the highest CAPICE score. The variants in the list can be filtered using different filters, such as HPO filters, inheritance filters and predicted class filters (B). The report also contains more detailed views to show all annotations for each transcript and which criteria were used to classify a specific variant. When the BAM workflow is used, the built-in genome viewer can be used to study the context of variants within reads that were mapped to the reference genome.

Expertise needed

We advise that the installation, configuration and customization of the decision tree be done by a bioinformatician or system administrator. Subsequently, the protocol can be followed by any diagnostician or researcher.

Materials

Software

VIP has been developed for GNU-based Linux (e.g. Ubuntu, Windows Subsystem for Linux with x86_64 architecture with installed versions of Bash (version 3.2 or higher), Java (version 11 or higher), Apptainer (version 3.18 or higher) and Nextflow (version 23.10.0). Installation of Git is required to download and install VIP (table 1).

Hardware

To install the version of VIP that is described in this paper (7.4.0), 220 GB of free diskspace is required. A multicore computer with at least 4 CPUs and 8 GB of RAM is advised, as this allows for parallel processing per chromosome to optimize performance. However, for the FASTQ and BAM workflows, the computing resources required are heavily dependent on the number of input files and the read depth. Similarly, the resources required for the VCF workflow depend heavily on the number of VCF files and the number of variants per sample. To install the most recent version of VIP, we refer to table 1.

Procedure

Below we describe the procedure to download, install and run the VCF workflow and analyze the results. The same procedure applies to the FASTQ and BAM workflows. For more information on running the FASTQ and BAM workflows, see the online VIP documentation (table 1).

Installation (1-2 hours)

1. Open a command-line Unix-like terminal, define a directory in which VIP will be installed and navigate to that directory. Download VIP using the following command: `git clone https://github.com/molgenis/vip.git`. Execute `bash vip/install` to install VIP. The install process creates a directory `vip` with all files required to run the application.

Optional configuration (1 hour)

2. The default configuration files are stored in the directory `vip/config`. These files specify values for the different parameters in JSON format. For each workflow a custom configuration file can be created to overwrite values in the default configuration. All parameters that are not hardcoded in the config files can be overwritten. Our experience is that the most frequently adjusted parameters are the process parameters for allocating RAM, CPU and processing time limits to specific sub workflows, the reference genome parameters to specify the reference genome used, and the filter parameters to specify which variants are shown in the interactive report. Additionally, users can specify the decision tree that they want to use. For a detailed description of all parameters, with default and example values, see the online VIP documentation (table 1). The custom configuration file used to run the VCF workflow for the demonstration data set is available in supplementary data 2.

3. The file containing the default decision tree in a JSON formatted file is stored in the directory `vip/resources`. The decision tree can be customized by creating a copy of the default decision tree file and editing or changing the order of the different parameters. For a description of the parameters that can be edited, see the online VIP documentation (table 1).

Create sample sheet (1 hour)

4. Users need to create a sample sheet containing the information of the individuals within each family. The required and default values are shown in table 2 and the online VIP documentation (table 1). When performing a trio analysis, we recommend providing the family identifier, individual identifier of the proband, the paternal and maternal identifiers, the affected status of the individuals, HPO terms of the proband, the sequencing method used (WES or WGS), the reference assembly used (GRCh37 or GRCh38) and the location of the stored VCF files. Examples of sample sheets used to run the VCF workflow for the demonstration data set are available in supplementary tables 4, 5 and 6.

Category	Column	Description
General	project_id	Unique identifier for each project. Each projects can contain multipe families and individuals. Default is 'vip'.
	family_id	Unique identifier for each family. A family can exit out of multiple layers of a father, mother and children. Default is 'fam_<integer>'.
	individual_id	Unique identifier for each family member. This field is mandatory.
	paternal_id	Unique identifier for each father within a family.
	maternal_id	Unique identifier for each mother within a family.
	sex	Biological sex of family member. Possible values are 'male' and 'female'. Default is 'unknown'.
	affected	Specifies whether the family member is affected by the disease phenotype. Allowed values are 'yes' or 'no'. Default is 'unknown'.
	proband	Specifies whether the family member is a proband. Allowed values are 'yes' or 'no'. If no probands are defined in the sample sheet, then all samples are considered to be probands.
	hpo_ids	Comma-sperated unique identifiers from the HPO database describing the phenotypes of the family members within the sample sheet.
	sequencing_method	Specifies the sequencing method used. This determines which workflow-specific processes are used for the processing of the input data. Allowed values are 'WES' or 'WGS'. Default is 'WGS'.
Workflow specific		
Category	**Column**	**Description**
FASTQ	fastq	Specifies the absolute path to the file containing raw sequencing data in FASTQ format. Only for single read files. Allowed file extensions are: fastq, fastq.gz, fq and fq.gz.
	fastq_r1	Specifies the absolute path to the file containg raw sequencing data in FASTQ format for one of the pairs in paired-end sequencing. Allowed file extensions are: fastq, fastq.gz, fq and fq.gz.
	fastq_r2	Specifies the absolute path to the file containg raw sequencing data in FASTQ format for one of the pairs in paired-end sequencing. Allowed file extensions are: fastq, fastq.gz, fq and fq.gz.
	sequencing_platform	Specifies the sequencing platform used. This determines which workflow-specific processes are used for the processing of the input data. Allowed values are: 'illumina', 'nanopore' or 'pacbio_hifi'. Value must be the same for all project samples.
BAM	bam	Specifies the absolute path to the file containing sequencing data in BAM format. Allowed file extensions are: bam, cram and sam.
	sequencing_platform	Specifies the sequencing platform used. This determines which workflow-specific processes are used for the processing of the input data. Allowed values are: 'illumina', 'nanopore' or 'pacbio_hifi'. Value must be the same for all project samples.
gVCF	assembly	Identifier specifying the used build of the reference genome. Allowed values are: 'GRCh37', 'GRCh38' and 'T2T'.
	gvcf	Specifies the absolute path to the file containing the called variants in gVCF format. In case a multisample gVCF is used, the individual_id has to match one of the samples in the VCF file.
VCF	assembly	Identifier specifying the used build of the reference genome. Allowed values are: 'GRCh37', 'GRCh38' and 'T2T'.
	vcf	Specifies the absolute path to the file containing the called variants in VCF format. In case a multisample VCF is used, the individual_id has to match one of the samples in the VCF file.

Table 2 **Sample sheet format** The first column describes for which workflow the fields need to be filled in. The second column represents the different columns within the samplesheet. The third column contains the description and the requirements of the infromation that can be provided.

Running the pipeline (4 hours)

5. To run the VCF workflow, navigate to the installation folder vip and execute vip -workflow vcf -input <path_to_directory>/<sample_sheet_name>.tsv -output <path_to_directory>/<output_folder_name>. If custom configuration values are provided, add -config <path_to_directory>/<config_file_name>.cfg. The pipeline's progress is displayed in the standard output of the terminal. After VIP has performed a quality check on the sample sheet, it will run the pre-processing, annotation, filtering and interactive reporting modules. The filtered VCF file, intermediate files, supporting files and interactive reports are stored in the respective output folders (table 3). An interactive report is created for each project specified in the sample sheet. See table 4 for potential problems and suggestions for troubleshooting potential errors.

File or folder[1]	Description
.nextflow[2]	Contains files with run statistics for each Nextflow run.
.nxf.home[2], .nxf.tmp[2]	Contains files and executables used by Nextflow.
.nxf.log[2]	Logfile containing history of all performed processes. Can be used to follow each run and debug failed runs.
.nxf.work[2]	Folder containing temporary files created by Nextflow for each process. Needed by Nextlfow to start subsequent processes. This folder is also used by Nextflow to resume a failed run.
nxf_report.html[2]	Self-contained interactive summary of the statistics for each process that Nextflow performed, like runtime, memory and CPU usage. Can be opened in any modern web browser.
nxf_timeline.html[2]	Self-contained schematic overview of the runtime and walltime of each Nextflow process. Can be opend in any modern web browser.
vip.html	Self-contained interactive report based on variant calls and annotations in vip.vcf.gz.
vip.vcf.gz	Compressed VCF file containing all variant calls, annotations and predicted classifications. This file can be used as input for other tools or subsequent analyses that require VCF files as input.
vip.vcf.gz.csi	Index file for vip.vcf.gz
intermediates	Compressed VCF and index files. Allows user to monitor or rerun VIP starting from pre-defined modules. For different possible rerun options, refer to the VIP documentation (table 1). Contents are dependend on the workflow started. A run of the VCF workflow containing WES data for one trio results in the following most important files: - <project_id>_<fam_id>_<individual_id>_liftover_accepted.vcf.gz - <project_id>_<fam_id>_<individual_id>_liftover_accepted.vcf.gz.csi - <project_id>_<fam_id>_<individual_id>_liftover_rejected.vcf.gz - <project_id>_<fam_id>_<individual_id>_liftover_rejected.vcf.gz.csi - <project_id>_<fam_id>_<individual_id>_annotations.vcf.gz - <project_id>_<fam_id>_<individual_id>_annotations.vcf.gz.csi - <project_id>_<fam_id>_<individual_id>_classifications.vcf.gz - <project_id>_<fam_id>_<individual_id>_classifications.vcf.gz.csi - <project_id>.sample_classifications.vcf.gz - <project_id>.sample_classifications.vcf.gz.csi
1) Path to files is specified by the user with the '--output' option when running VIP on the command line 2) For a detailed explanation, refer to the documentation of Nextflow (table 1)	

Table 3 **Output files**. The first column shows the name of the different output files that are created during a VIP run. The second column shows a description and function of the output files.

Step	Problem	Possible reason	Solution
5	VIP returns unkown contig in VCF header.	Used reference genome does not match input VCF.	Edit the config file for the corresponding workflow to adjust the reference genome.
	Process runs out of memory.	Not enough memory allocated in the corresponding config file.	Edit the config file for the corresponding process to increase RAM and CPU usage, and time limits for the failed process.
	VIP returns parsing error in the sample sheet validation process.	VIP has found a parsing error in the sample sheet.	Check the nxf.log output file for details about the error. Avoid special characters. Check if the columns line up with tab-separated values. Compare input with required and allowed values by using online VIP documentation and check for inconsitencies (table 1).
	VIP fails during the annotation process (no out-of-memory error).	One of the processes within the annotation module has failed.	Check the nextflow log file for details about the error. Find the process working directory and locate it in the nextflow working directory. Check the command-files to pinpoint which process caused the error. Adjust your input accordingly and rerun your command by adding the resume option.

Table 4 **Troubleshooting table**

Navigating the interactive report

6. To interpret the results, download and open an interactive report <project_id>.html from your output directory in a locally installed web browser. Navigate to the 'samples tab' and the proband of interest (figure 3a). Clicking 'variants' opens the base view, which displays the possible filters and all the effects of a variant on each possible transcript (figure 3b,c). Each line represents the effect of a variant with the highest CAPICE score. To show all effects for each variant, press the arrow next to an effect. By default, the report only displays the gene symbol, expected inheritance pattern, matching HPO terms, HGVS notations, CAPICE scores, VKGL and ClinVar classifications, GnomAD allele frequencies, predicted classifications by VIP and links to available literature. To display all available annotations, click 'Variants' in the header to open a detailed view of all annotations. To use a different sorting method, select an annotation by which to sort in the top right corner.
7. In the base view, use the filters on the left side of the report to filter the list (figure 3b). By default, the read quality, read depth and allelic imbalance filters are applied. To also view low quality variant calls, deselect these filters.
8. When VIP is run using the FASTQ or BAM workflow, click a chromosomal position in the base view to show the genomic context and the reads overlapping the variant for the different transcripts.
9. In the base view, click on an effect to navigate to a list of all annotations for the specific effect. All the annotations added by VIP are displayed in the left panel in the 'consequence section'. The path through the decision tree is displayed in the right panel. This information can be used to verify which information VIP used to predict a class for a variant. Record- and sample-specific information, such as the chromosomal position and known inheritance mode, is shown in the bottom panel.
10. In the base view, click on a gene symbol to open the gene in a webpage of the HUGO Gene Nomenclature Committee (HGNC) with links to related genomic, clinical and proteomic information[43].
11. If available, click on the citation in the base view to navigate to the related research article in PubMed.
12. In the base view, click on an HPO term in the variant screen to navigate to a webpage of the HPO database. This contains a description of the phenotype, its ontology and its associations to diseases and genes.
13. In the base view, click on the ClinVar classification to navigate to the associated webpage of ClinVar. This webpage contains detailed information about the variant and the submitted interpretations and evidence for its classifications[44].

Reanalysis

14. To re-analyze data from a previous run, VIP can be restarted from the filter module. This is, for example, useful for applying a different decision tree to classify variants without having to annotate all variants again. To rerun VIP, the basic procedure remains the same. In addition, the correct intermediate files should be provided in the sample sheet and the module from which to start should be added in the custom configuration file using the parameter vcf.start. For more instructions to restart VIP from a different module, see the online VIP documentation (table 1).

Adding additional annotation sources

We also provide bioinformaticians and system administrators the ability to develop their own plugins to add new annotation sources. We use the VEP plugin framework to develop new plugins for VIP. The /vip/resources/vep/plugins directory contains examples with existing plugins. After a plugin is developed, the /vip/modules/vcf/templates/annotate.sh needs to be expanded with the newly developed plugin.

Timing

Using VIP version 7.4.0, 16 GB of memory and 4 CPUs, the average runtime for WES data in VCF format for 20 samples (average size ~216 MB) was 4 hours. However, the performance is highly dependent on the size of the dataset, the workflow used and the number of available CPUs and RAM.

Preparation (once):

- Step 1. Installing MOLGENIS VIP: 1–2 hours.
- Step 2-3. Creating custom configuration file and decision tree: 1 hour. When the default configuration and decision tree is used, this step can be skipped.

For each analysis:

- Step 4. Creating the sample sheet: 1 hour.
- Step 5. Running the pipeline: 4 hours (waiting for output).

Anticipated results

To demonstrate how VIP performs and how it can be used to interpret clinically relevant germline data, we used VIP version 7.4.0, 4 CPUs and 16 GB of RAM to analyze variants from the VKGL database and two anonymized patient cohorts. The total runtime was 4 hours.

Input data

Previously classified variants

We collected 25,664 variants previously classified by experts as LP or P based on the consensus between the different Dutch genome diagnostic laboratories within the VKGL. For this demonstration, we used the LP and P variants in VKGL release 2023-11 and all newly added LP and P variants (added between release 2023-11 and 2024-2) to showcase how VIP classifies previously classified variants (table 1).

Routine diagnostics cohort

To demonstrate VIP in diagnostics, we created two anonymized patient cohorts. The first cohort was created to demonstrate how VIP is applied in routine diagnostics at the UMCG. This routine diagnostics cohort contained 18 patients with a molecular genetic diagnosis (supplementary table 2). These patients were selected out of 70 monthly interpretation requests by a clinical geneticist. Virtual gene panels were used to home in on potential disease-causing genes and prevent incidental findings. The panels consisted of genes specific for developmental delay, dilated cardiomyopathy and the clinical exome (supplementary data 3). For all patients HPO terms, WES data and maternal and paternal WES data were available.

Solve-RD research cohort

The second cohort contained 41 patients with RDs who had not been diagnosed following routine diagnostics but whose cases were solved within the EU-Solve-RD project using novel omics approaches [21]. These patients were selected based on the availability of a molecular diagnosis, HPO terms and maternal and paternal WES data. The resulting patient selection originated from three different European Reference Networks (ERNs): the ERN for neuromuscular disorders (ERN-EURO NMD), the ERN for rare neurological diseases (ERN-RND) and the ERN for rare malformation syndromes, intellectual and other neurodevelopmental disorders (ERN-ITHACA)[21]. The verified causal variants were extracted from the Genome Phenome analysis platform (GPAP) on the 12th of February 2024[45] (supplementary table 3).

Comparing the number of candidate variants, recall rate and ranking

Create sample sheets

First, we created a sample sheet containing one file with all LP and P variants in VKGL release 2024-2, including variants that are part of the version of VIP used. Another sample sheet was created containing only newly added variants that are not part of the version of VIP used (supplementary table 4).

Following this step, we created one sample sheet with the patients from the routine diagnostics cohort and one containing the patients from the Solve-RD research cohort. Both sample sheets specified the HPO terms used, affected status, reference genomes, sequencing types and the proband, maternal and paternal WES data (supplementary table 5, 6).

Run VIP and analysis of results

VIP was used with the three sample sheets and the default decision tree to create an interactive report for each case. By default, variants with a genotype quality ≤ 20 and allelic imbalance and all LB and B variants were filtered out. We then collected the number of candidate variants when applying: (1) the HPO-match filter, (2) the inheritance-match filter and (3) both the HPO- and inheritance-match filters.

We studied the number of candidate variants per patient and the total number of confirmed molecular diagnoses found in order to compare the recall rates for the previously classified variants, the routine diagnostics cohort and the Solve-RD research cohort. To see how well VIP prioritized the molecular diagnoses, and demonstrate its added value for clinical geneticists, we also compared the assigned rank with the number of candidate variants.

Expectation

Because the variants of VKGL release 2023-11 were used by VIP in the VKGL and CAPICE annotation steps, we expect a 100% recall rate for these known variants. To showcase how VIP classifies known VKGL variants that were not part of the VKGL release used by VIP, we used the variants added between release 2023-11 and 2024-2. Because these are also known variants, we expect a recall rate of almost 100%.

The VCF files for the routine diagnostics cases are pre-filtered based on a virtual gene panel. Therefore, we expect that the average number of candidate variants in the interactive report for the routine diagnostics cohort will be smaller than for the Solve-RD research cohort. We also expect that the filtering based on VUS, LP and P variants, HPO term matches and inheritance matches results in the highest ranking of the causal variants. However, this could lead to a decreased recall rate. Not all genes in the HPO database are equally well annotated with HPO terms, and phenotypic features could have been missed when the patient's phenotype was described in the clinic[22]. Similarly, not all genomic positions for the proband are also covered for both parents, resulting in a missing inheritance match.

The patients from the Solve-RD research cohort that we collected were initially unsolved by the individual expertise centers. For this reason, they were submitted to the Solve-RD project to be analyzed using novel omics methods, such as gene expression analysis and RNA-sequencing, instead of relying only on NGS. We expect that the recall rate for this cohort is lower compared to the routine diagnostics cohort.

Results

Previously classified variants

Figure 4 shows the number of known variants recalled. As expected, VIP detected 100% (25,664/25,664) of these variants. In addition, VIP recalled 558 out of 597 (93.47%) of the variants that were newly added. Twenty-one variants missed because they were classified as LB based on the CAPICE scores. Two variants were missed because one was classified as LB in the ClinVar database and

another variant had a gnomAD minor allele frequency (MAF) higher than 0.02. Finally, 16 variants were missed because they had a SpliceAI score between 0.13 and 0.42.

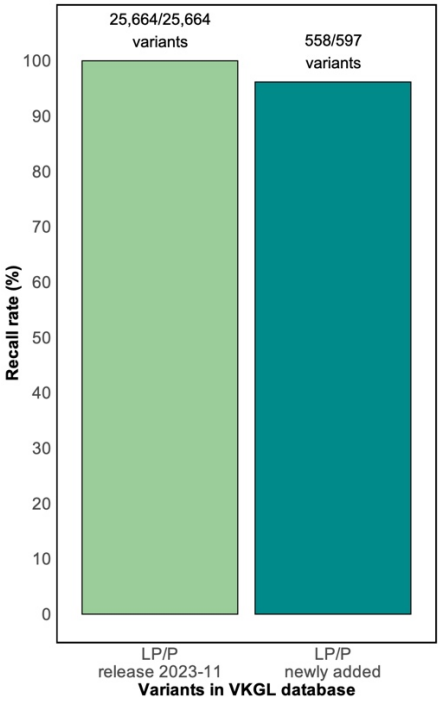


Figure 4 **Number of recalled variants that are previously classified.** Y-axis shows the percentage of variants that VIP was able to recall as LP or P. The first bar at left represents the variants of VKGL release 2024-2 that were used by VIP in the VKGL and CAPICE annotation steps. The second bar represents the newly added variants between VKGL release 2023-11 and 2024-2 that were not used by VIP. The absolute number of variants that were recalled is shown at the top of the bars.

Routine diagnostics cohort

Figure 5a shows the average number of candidate variants per patient and the total number of recalled molecular diagnoses for the routine diagnostics cohort. When LB and B variants are filtered out, VIP returned an average of 338 candidate variants per patient and recalled 18/18 molecular diagnoses. Applying the HPO-match and inheritance-match filters resulted in an average of 8 and 18 candidate variants per patient and 14/18 and 15/18 recalled molecular diagnoses, respectively. The most significant decrease in the number of candidates came after applying the HPO-match and inheritance-match filters simultaneously, resulting in four candidate variants per patient. However, the recall rate decreased to 11/18.

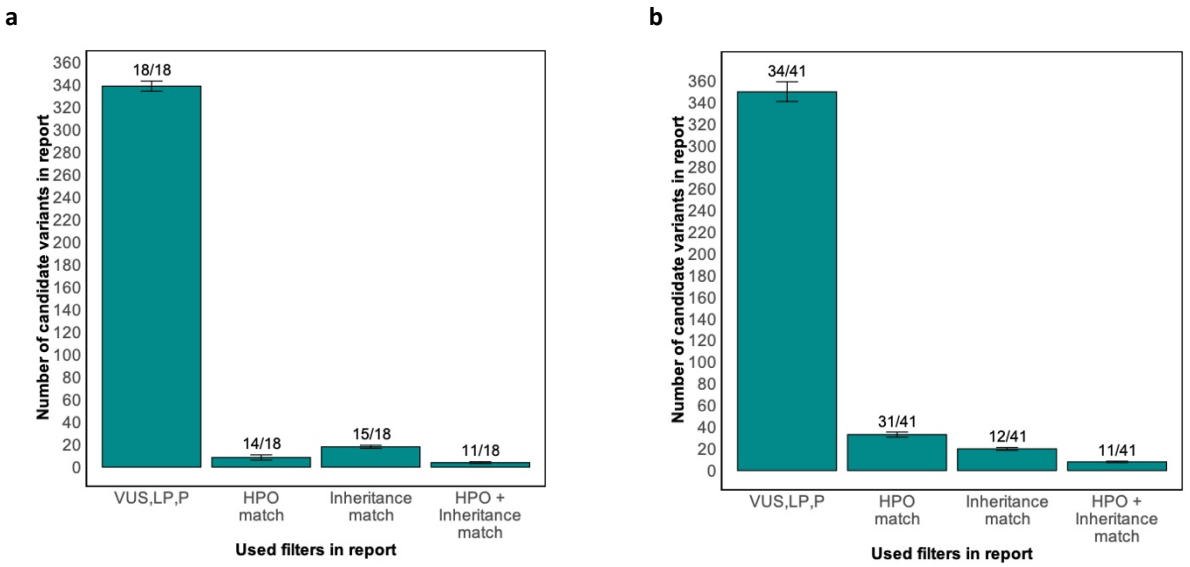


Figure 5 **Number of candidate variants and recall rate. a**, The average numbers of candidate variants per patient in the routine diagnostics cohort. **b**, The average number of candidate variants per patient in the Solve-RD research cohort. On the x-axis, the different filters used in the interactive report are specified. The recall rate per filter is displayed above the individual bars.

Solve-RD research cohort

Figure 5b shows the same results for the Solve-RD research cohort. After LB and B variants were filtered out, VIP returned an average of 349 candidate variants per patient and recalled 34/41 molecular diagnoses. Applying the HPO-match and inheritance-match filters resulted in an average of 33 and 20 candidate variants per patient and 31/41 and 12/41 recalled molecular diagnoses, respectively. The most significant decrease in the number of candidate variants came from applying both the HPO-match and inheritance-match filters simultaneously, resulting in eight candidate variants per patient. However, the recall rate decreased to 11/41.

Missed variants

In the Solve-RD research cohort, VIP missed 7 molecular diagnoses due to a CAPICE score that did not meet the cut-off value specified in the default decision tree (figure 2). The decrease in recall rate after applying the HPO-match filter can be explained by the fact that 4 patients in the routine diagnostics and 10 patients in the Solve-RD research were annotated with HPO terms that do not match the gene in which the molecular diagnosis is located. This was not unexpected. As described previously by W. Maassen et al., it is possible that not every patient's phenotype is equally well described in HPO terms or that the gene is not annotated with HPO terms in the HPO database[22]. The decrease in recall rate after applying the inheritance-match filter for patients in the routine diagnostics and the Solve-RD research cohort can be explained by the fact that, for 3 and 29 patients, respectively, NGS did not cover both alleles for at least one parent.

Assigned rank versus number of candidate variants

In figure 6, the average rank of the molecular diagnoses was compared with the number of candidate variants per patient in the routine diagnostics and Solve-RD research cohort. This figure shows that the average rank increased and that the average number of candidate variants per patient decreased when the different filters were applied. This shows that VIP was better able to discriminate causal variants from other variants when the number of candidate variants decreases. The lowest number of candidate variants was returned when both the HPO- and the inheritance-match filter were applied.

For the routine diagnostics and Solve-RD research cohort, this resulted in an average rank of 2 and 4, respectively.

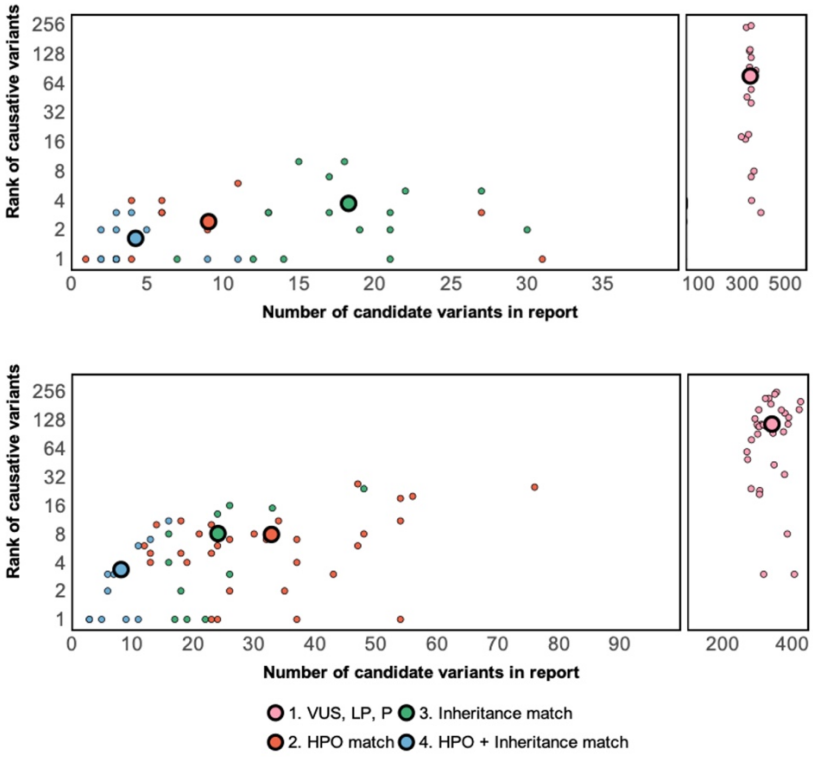


Figure 6 **Number of candidate variants per patient and rank of molecular diagnoses.** Panels show the average numbers of candidate variants per patient plotted versus the average rank of the molecular diagnosis for the patients in the (**a**) routine diagnostics and (**b**) Solve-RD research cohorts. Patients for whom the molecular diagnoses were not found after applying the different filters are not included.

Data availability

The previously classified variants from the VKGL can be downloaded from https://vkgl.molgeniscloud.org/. The routine diagnostics cohort contains patient data from patients within the UMCG and can therefore only be shared upon request. The Solve-RD research cohort is available as a dataset (EGAD50000000390) in the European Genome-Phenome Archive and can be accessed by sending a data access request to the data access committee of the EU-Solve-RD project.

Ethics statement

The studies involving humans were approved by the ethical committee of the University Medical Center Groningen. The studies were conducted in accordance with the local legislation and institutional requirements. Written informed consent was obtained from the 18 UMCG patients for sharing their WES data. To access patient data from the EU-Solve-RD project, we received confirmation from the Solve-RD Data Access Committee.

Code availability

MOLGENIS VIP is publicly available at https://github.com/molgenis/vip under the GNU Lesser General Public License v3.0. See https://github.com/molgenis/vip/blob/main/LICENSE for details. VIP is an aggregate work of many individual tools, each covered by their own license(s). Therefore, the individual license(s) of the relevant tools should also be considered.

Author contributions statements

W.T.K.M., L.F.J. wrote manuscript. W.T.K.M., M.E.G., M.A.S., K.J.V. conceived and designed the experiments. W.T.K.M. performed analysis. W.T.K.M., L.F.J., C.C.D. gathered patient data. L.F.J., B.C., D.H., S.H., M.A.S., R.M., M.M-V., R.S., H.H.L., C.C.D., M.E.G., M.A.S., K.J.V. discussed results and reviewed paper. M.E.G., M.A.S., K.J.V. supervised project. D.H., B.C., S.H., M.A.S., R.S., W.T.K.M., L.F.J., K.J.V. developed and tested the software. All authors contributed to the results and approved the submitted version.

Acknowledgements

This study has been performed with the support of the members of the UMCG Genomics Coordination Center, the MOLGENIS VIP development team and the Development and Innovation team at the Genetics Department of the UMCG. We want to thank all members and patients participating in the Solve-RD project for access to the Solve-RD database and the use of the GPAP analysis software. Finally, we would like to thank Kate Mc Intyre for editorial assistance.

This study received funding from the EU projects Solve-RD, EJP-RD and CINECA (H2020 779257, H2020 825575, H2020 825775, respectively) and NWO grant numbers 917.164.455 and 184.034.019.

Competing interests

All authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1 Claussnitzer, M. *et al.* A brief history of human disease genetics. *Nature* **577**, 179-189, doi:10.1038/s41586-019-1879-7 (2020).

2 Liu, Y., Yeung, W. S. B., Chiu, P. C. N. & Cao, D. Computational approaches for predicting variant impact: An overview from resources, principles to applications. *Front Genet* **13**, 981005, doi:10.3389/fgene.2022.981005 (2022).

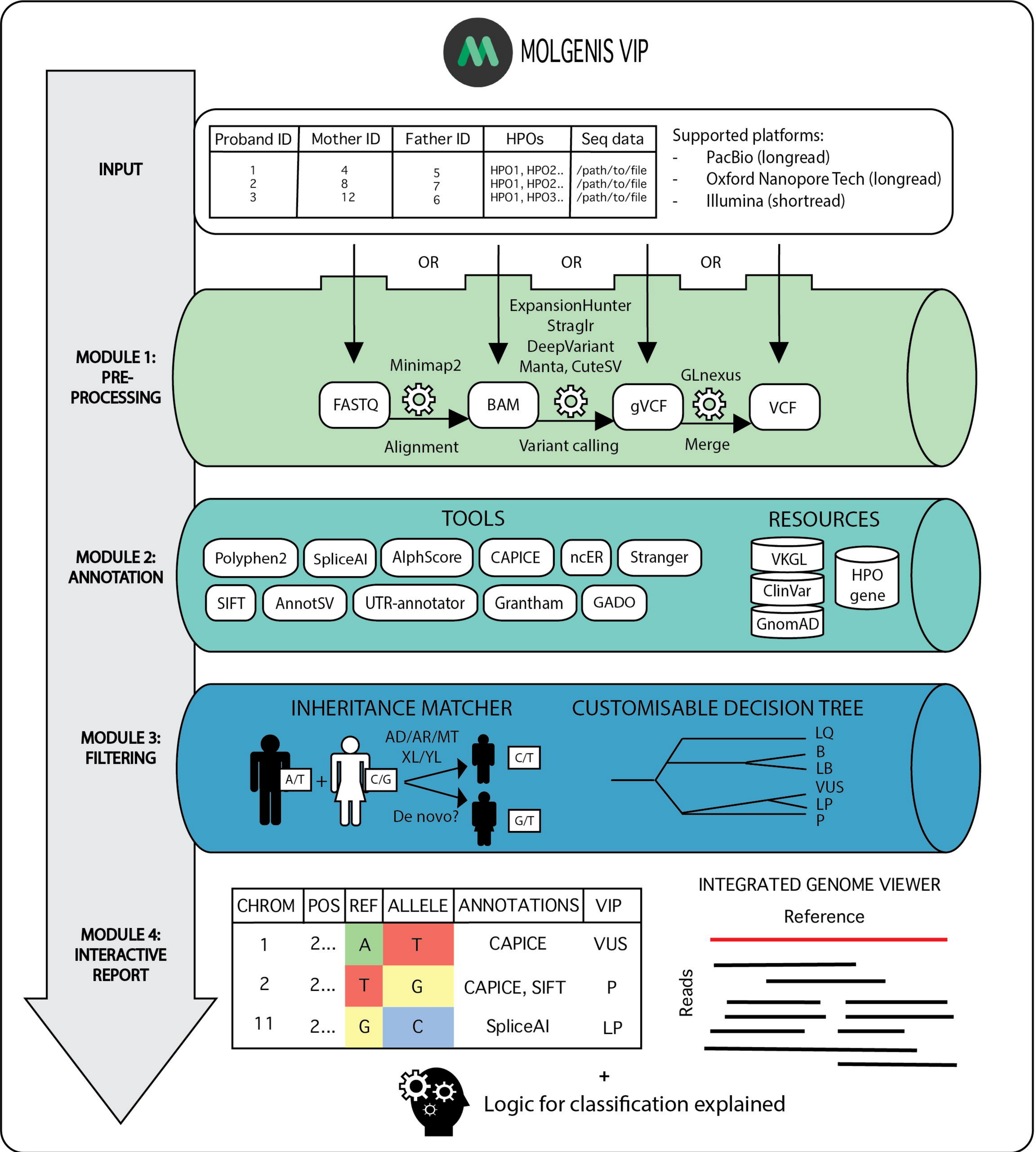
3 The Lancet Diabetes, E. Spotlight on rare diseases. *Lancet Diabetes Endocrinol* **7**, 75, doi:10.1016/S2213-8587(19)30006-3 (2019).

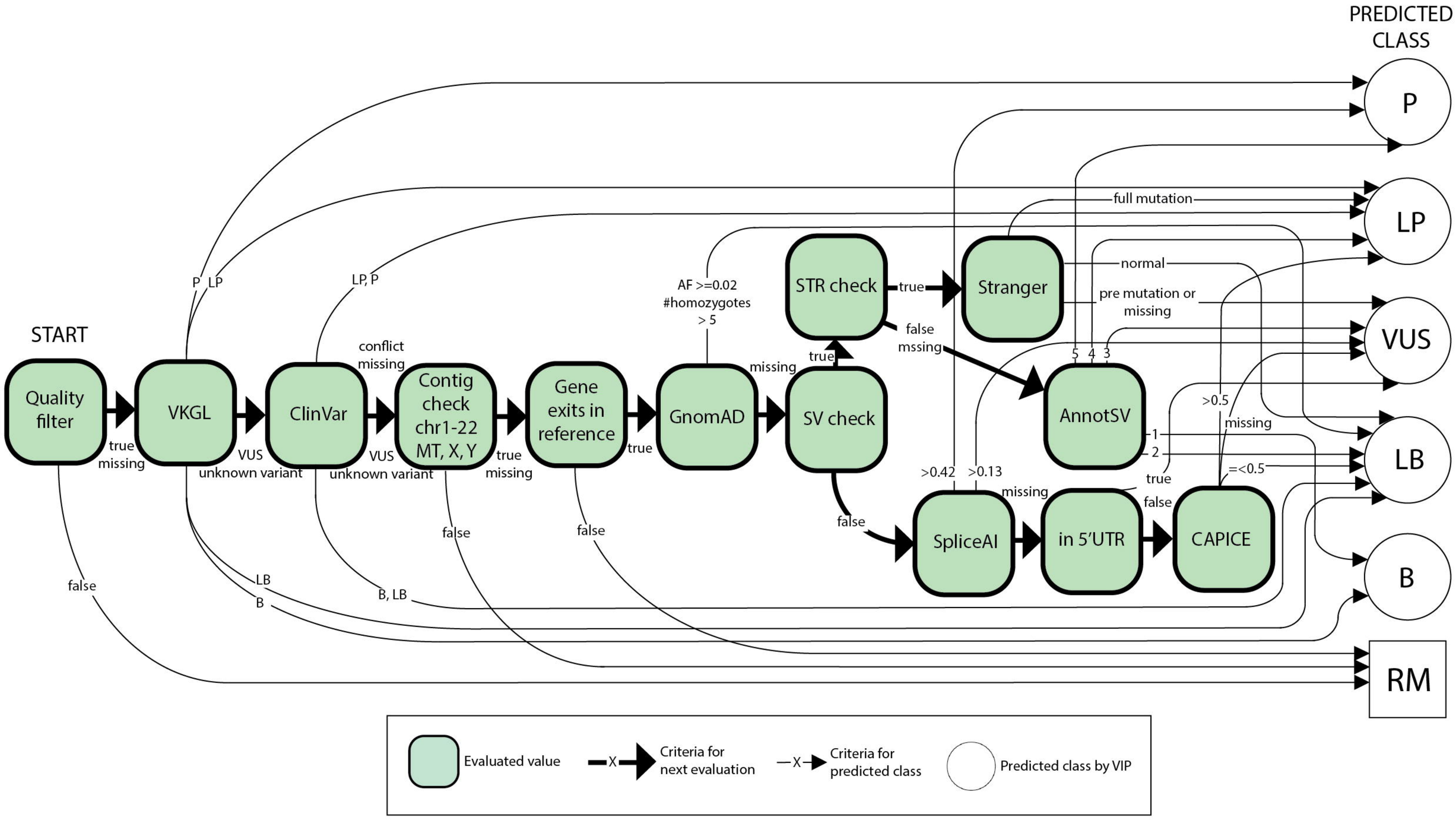
4 Nguengang Wakap, S. *et al.* Estimating cumulative point prevalence of rare diseases: analysis of the Orphanet database. *Eur J Hum Genet* **28**, 165-173, doi:10.1038/s41431-019-0508-0 (2020).

5 He, X. *et al.* DIVIS: Integrated and Customizable Pipeline for Cancer Genome Sequencing Analysis and Interpretation. *Front Oncol* **11**, 672597, doi:10.3389/fonc.2021.672597 (2021).

6 Liu, Z. *et al.* Annotation Query (AnnoQ): an integrated and interactive platform for large-scale genetic variant annotation. *Nucleic Acids Res* **5**, 50(W51):W57-W65, doi:10.1093/nar/gkac418 (2022).
7 Weilguny, L. *et al.* Dynamic, adaptive sampling during nanopore sequencing using Bayesian experimental design. *Nat Biotechnol* **41**, 1018-1025, doi:10.1038/s41587-022-01580-z (2023).
8 Exome Enrichment using Agilent SureSelect Human All Exon V8 Panel and Agilent Alissa Clinical Informatics Platform (ed Agilent Technologies) (2023).
9 Avsec, Z. *et al.* The Kipoi repository accelerates community exchange and reuse of predictive models for genomics. *Nat Biotechnol* **37**, 592-600, doi:10.1038/s41587-019-0140-0 (2019).
10 Stranneheim, H. *et al.* Integration of whole genome sequencing into a healthcare setting: high diagnostic rates across multiple clinical entities in 3219 rare disease patients. *Genome Med* **13**, 40, doi:10.1186/s13073-021-00855-5 (2021).
11 Léon, A. & Pastor, O. Enhancing Precision Medicine: A Big Data-Driven Approach for the Management of Genomic Data. *Big Data Research* **26**, 100253, doi:https://doi.org/10.1016/j.bdr.2021.100253 (2021).
12 Fokkema, I. *et al.* Dutch genome diagnostic laboratories accelerated and improved variant interpretation and increased accuracy by sharing data. *Hum Mutat* **40**, 2230-2238, doi:10.1002/humu.23896 (2019).
13 Koboldt, D. C. Best practices for variant calling in clinical sequencing. *Genome Med* **12**, 91, doi:10.1186/s13073-020-00791-w (2020).
14 Cheng, C., Fei, Z. & Xiao, P. Methods to improve the accuracy of next-generation sequencing. *Front Bioeng Biotechnol* **11**, 982111, doi:10.3389/fbioe.2023.982111 (2023).
15 Rehm, H. L. & Fowler, D. M. Keeping up with the genomes: scaling genomic variant interpretation. *Genome Med* **12**, 5, doi:10.1186/s13073-019-0700-4 (2019).
16 Kim, Y. E., Ki, C. S. & Jang, M. A. Challenges and Considerations in Sequence Variant Interpretation for Mendelian Disorders. *Ann Lab Med* **39**, 421-429, doi:10.3343/alm.2019.39.5.421 (2019).
17 Field, M. A. Bioinformatic Challenges Detecting Genetic Variation in Precision Medicine Programs. *Front Med (Lausanne)* **9**, 806696, doi:10.3389/fmed.2022.806696 (2022).
18 Ghosh, R., Oak, N. & Plon, S. E. Evaluation of in silico algorithms for use with ACMG/AMP clinical variant interpretation guidelines. *Genome Biol* **18**, 225, doi:10.1186/s13059-017-1353-5 (2017).
19 Hamzeh, A. R., Andrews, T. D. & Field, M. A. Detecting Causal Variants in Mendelian Disorders Using Whole-Genome Sequencing. *Methods Mol Biol* **2243**, 1-25, doi:10.1007/978-1-0716-1103-6_1 (2021).
20 Schrijver, I. *et al.* Opportunities and challenges associated with clinical diagnostic genome sequencing: a report of the Association for Molecular Pathology. *J Mol Diagn* **14**, 525-540, doi:10.1016/j.jmoldx.2012.04.006 (2012).
21 Zurek, B. *et al.* Solve-RD: systematic pan-European data sharing and collaborative analysis to solve rare diseases. *Eur J Hum Genet* **29**, 1325-1331, doi:10.1038/s41431-021-00859-0 (2021).
22 Maassen, W. *et al.* Curation and expansion of the Human Phenotype Ontology for systemic autoinflammatory diseases improves phenotype-driven disease-matching. *Front Immunol* **14**, 1215869, doi:10.3389/fimmu.2023.1215869 (2023).
23 Kiewiet, G. *et al.* Future of Dutch NGS-Based Newborn Screening: Exploring the Technical Possibilities and Assessment of a Variant Classification Strategy. *Int J Neonatal Screen* **10**, doi:10.3390/ijns10010020 (2024).
24 Di Tommaso, P. *et al.* Nextflow enables reproducible computational workflows. *Nat Biotechnol* **35**, 316-319, doi:10.1038/nbt.3820 (2017).
25 Kurtzer, G. M., Sochat, V. & Bauer, M. W. Singularity: Scientific containers for mobility of compute. *PLoS One* **12**, e0177459, doi:10.1371/journal.pone.0177459 (2017).
26 Cock, P. J., Fields, C. J., Goto, N., Heuer, M. L. & Rice, P. M. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Res* **38**, 1767-1771, doi:10.1093/nar/gkp1137 (2010).
27 Danecek, P. *et al.* The variant call format and VCFtools. *Bioinformatics* **27**, 2156-2158, doi:10.1093/bioinformatics/btr330 (2011).
28 Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078-2079, doi:10.1093/bioinformatics/btp352 (2009).
29 Robinson, P. N. *et al.* The Human Phenotype Ontology: a tool for annotating and analyzing human hereditary disease. *Am J Hum Genet* **83**, 610-615, doi:10.1016/j.ajhg.2008.09.017 (2008).
30 Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094-3100, doi:10.1093/bioinformatics/bty191 (2018).

31 Dolzhenko, E. *et al.* ExpansionHunter: a sequence-graph-based tool to analyze variation in short tandem repeat regions. *Bioinformatics* **35**, 4754-4756, doi:10.1093/bioinformatics/btz431 (2019).
32 Chiu, R., Rajan-Babu, I. S., Friedman, J. M. & Birol, I. Straglr: discovering and genotyping tandem repeat expansions using whole genome long-read sequences. *Genome Biol* **22**, 224, doi:10.1186/s13059-021-02447-3 (2021).
33 Yun, T. *et al.* Accurate, scalable cohort variant calls using DeepVariant and GLnexus. *Bioinformatics* **36**, 5582-5589, doi:10.1093/bioinformatics/btaa1081 (2021).
34 Chen, X. *et al.* Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics* **32**, 1220-1222, doi:10.1093/bioinformatics/btv710 (2016).
35 Jiang, T., Liu, S., Cao, S. & Wang, Y. Structural Variant Detection from Long-Read Sequencing Data with cuteSV. *Methods Mol Biol* **2493**, 137-151, doi:10.1007/978-1-0716-2293-3_9 (2022).
36 Hunt, S. E. *et al.* Annotating and prioritizing genomic variants using the Ensembl Variant Effect Predictor-A tutorial. *Hum Mutat* **43**, 986-997, doi:10.1002/humu.24298 (2022).
37 Li, S. *et al.* CAPICE: a computational method for Consequence-Agnostic Pathogenicity Interpretation of Clinical Exome variations. *Genome Med* **12**, 75, doi:10.1186/s13073-020-00775-w (2020).
38 Jaganathan, K. *et al.* Predicting Splicing from Primary Sequence with Deep Learning. *Cell* **176**, 535-548 e524, doi:10.1016/j.cell.2018.12.015 (2019).
39 Solomon, B. D., Nguyen, A. D., Bear, K. A. & Wolfsberg, T. G. Clinical genomic database. *Proc Natl Acad Sci U S A* **110**, 9851-9855, doi:10.1073/pnas.1302575110 (2013).
40 Amberger, J., Bocchini, C. A., Scott, A. F. & Hamosh, A. McKusick's Online Mendelian Inheritance in Man (OMIM). *Nucleic Acids Res* **37**, D793-796, doi:10.1093/nar/gkn665 (2009).
41 Wallis, Y. *et al.* Practice Guidelines for the Evaluation of Pathogenicity and the Reporting of Sequence Variants in Clinical Molecular Genetics (ed Verenging Klinisch Genetische Laboratoriumdiagnostiek Association for Clinical Genetic Science) (2013).
42 Durkie, M. *et al.* ACGS Best Practice Guidelines for Variant Classification in Rare Disease 2023 (ed Association for Clinical Genetic Science) (2023).
43 Seal, R. L. *et al.* Genenames.org: the HGNC resources in 2023. *Nucleic Acids Res* **51**, D1003-D1009, doi:10.1093/nar/gkac888 (2023).
44 Landrum, M. J. *et al.* ClinVar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Res* **42**, D980-985, doi:10.1093/nar/gkt1113 (2014).
45 Matalonga, L. *et al.* Solving patients with rare diseases through programmatic reanalysis of genome-phenome data. *Eur J Hum Genet* **29**, 1337-1347, doi:10.1038/s41431-021-00852-7 (2021).


MOLGENIS VIP
INPUT
Proband ID
Mother ID
Father ID
HPOs
Seq data
1 4 5 HPO1, HPO2.. /path/to/file
2 8 7 HPO1, HPO2.. /path/to/file
3 12 6 HPO1, HPO3.. /path/to/file
Supported platforms:
- PacBio (longread)
- Oxford Nanopore Tech (longread)
- Illumina (shortread)
OR
OR
OR
MODULE 1: PRE-PROCESSING
FASTQ
Minimap2
Alignment
BAM
ExpansionHunter
Straglr
DeepVariant
Manta, CuteSV
Variant calling
gVCF
GLnexus
Merge
VCF
MODULE 2: ANNOTATION
TOOLS
Polyphen2
SpliceAI
AlphScore
CAPICE
ncER
Stranger
SIFT
AnnotSV
UTR-annotator
Grantham
GADO
RESOURCES
VKGL
ClinVar
GnomAD
HPO gene
MODULE 3: FILTERING
INHERITANCE MATCHER
A/T
C/G
AD/AR/MT
XL/YL
C/T
De novo?
G/T
CUSTOMISABLE DECISION TREE
LQ
B
LB
VUS
LP
P
MODULE 4: INTERACTIVE REPORT
CHROM POS REF ALLELE ANNOTATIONS VIP
1 2... A T CAPICE VUS
2 2... T G CAPICE, SIFT P
11 2... G C SpliceAI LP
INTEGRATED GENOME VIEWER
Reference
Reads
+
Logic for classification explained


PREDICTED CLASS
START
Quality filter
VKGL
ClinVar
Contig check chr1-22 MT, X, Y
Gene exits in reference
GnomAD
SV check
STR check
Stranger
AnnotSV
SpliceAI
in 5'UTR
CAPICE
P
LP
VUS
LB
B
RM
P, LP
LP, P
true
missing
VUS
unknown variant
conflict
missing
false
AF >=0.02
#homozygotes
> 5
missing
mssing
full mutation
normal
pre mutation or
missing
5 4 3
1
2
>0.42
>0.13
>0.5
=<0.5
LB
B
B, LB

A. Samples

VCF Report Samples Variants

Click on an individual id for detailed information for this sample. In this screen a link to the variants for this sample is available.

Family	Individual	Father	Mother	Proband	Sex	Affected	Phenotypes	VIBE
FAM20	6Va_5Pre_NX285_m_Vader_201208	0	0	False	Male	Unaffected		
FAM20	6Moe_5Pre_NX285_v_Moeder_201208	0	0	False	Female	Unaffected		
FAM20	6I_5Pre_NX285_m_Index_201208	6Va_5Pre_NX285_m_Vader_201208	6Moe_5Pre_NX285_v_Moeder_201208	True	Male	Affected	HP:0011461, HP:0001561, HP:0000308, HP:0002190, HP:0010963, HP:0002650, HP:0000773	

B. Filter buttons

VIP

- [ ] LQ
- [ ] B
- [ ] LB
- [ ] VUS
- [ ] LP
- [ ] P

Inheritance

- [ ] Match
- [ ] Match: Possible
- [ ] De novo

Quality

- [x] GT quality >= 20
- [x] No allelic imbalance
- [x] Depth >= 20

HPO

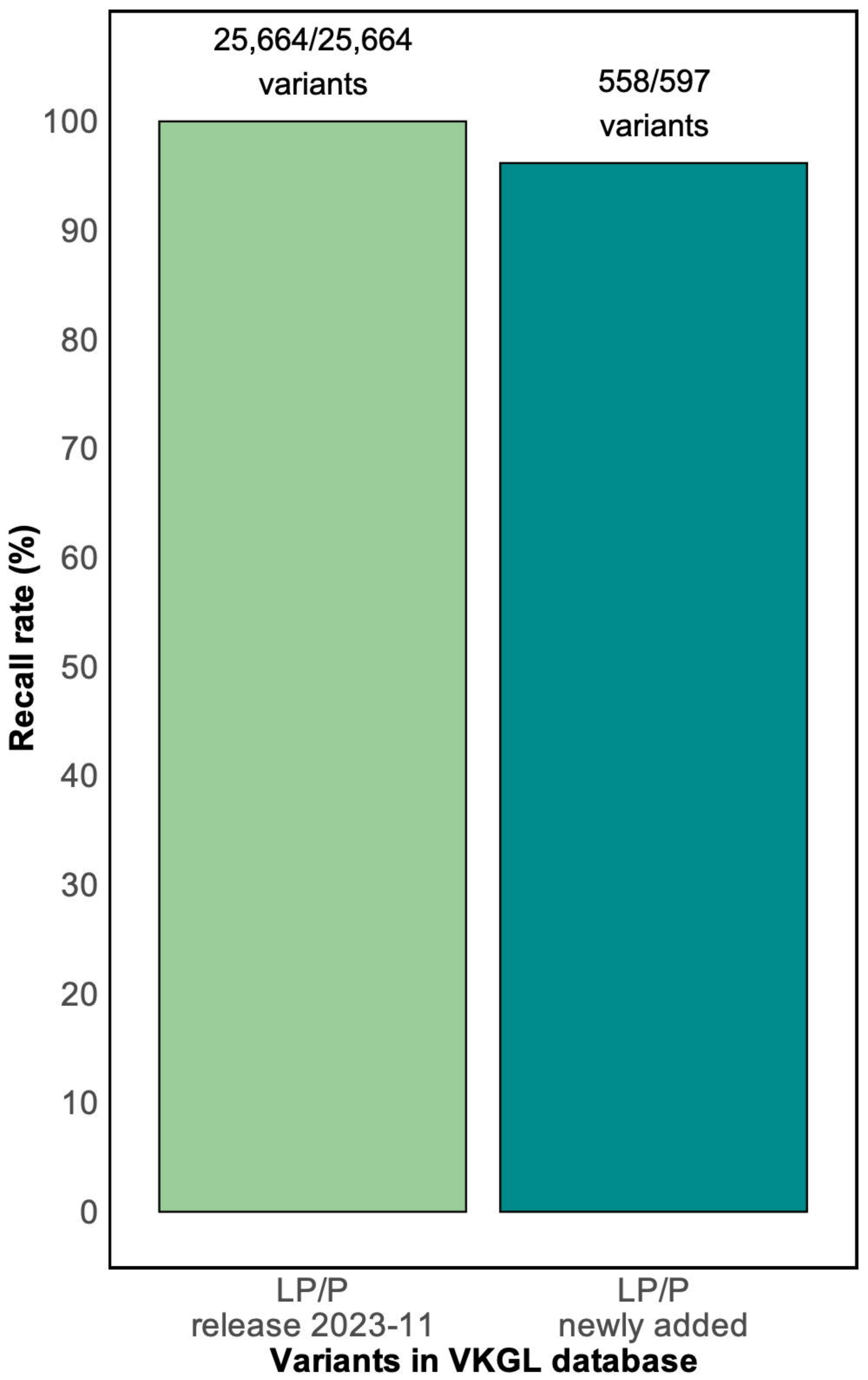
- [ ] HP:0000308
- [ ] HP:0000773
- [ ] HP:0001561
- [ ] HP:0002190
- [ ] HP:0002650
- [ ] HP:0010963
- [ ] HP:0011461
- [ ] GADO high
- [ ] GADO low

C. Variants

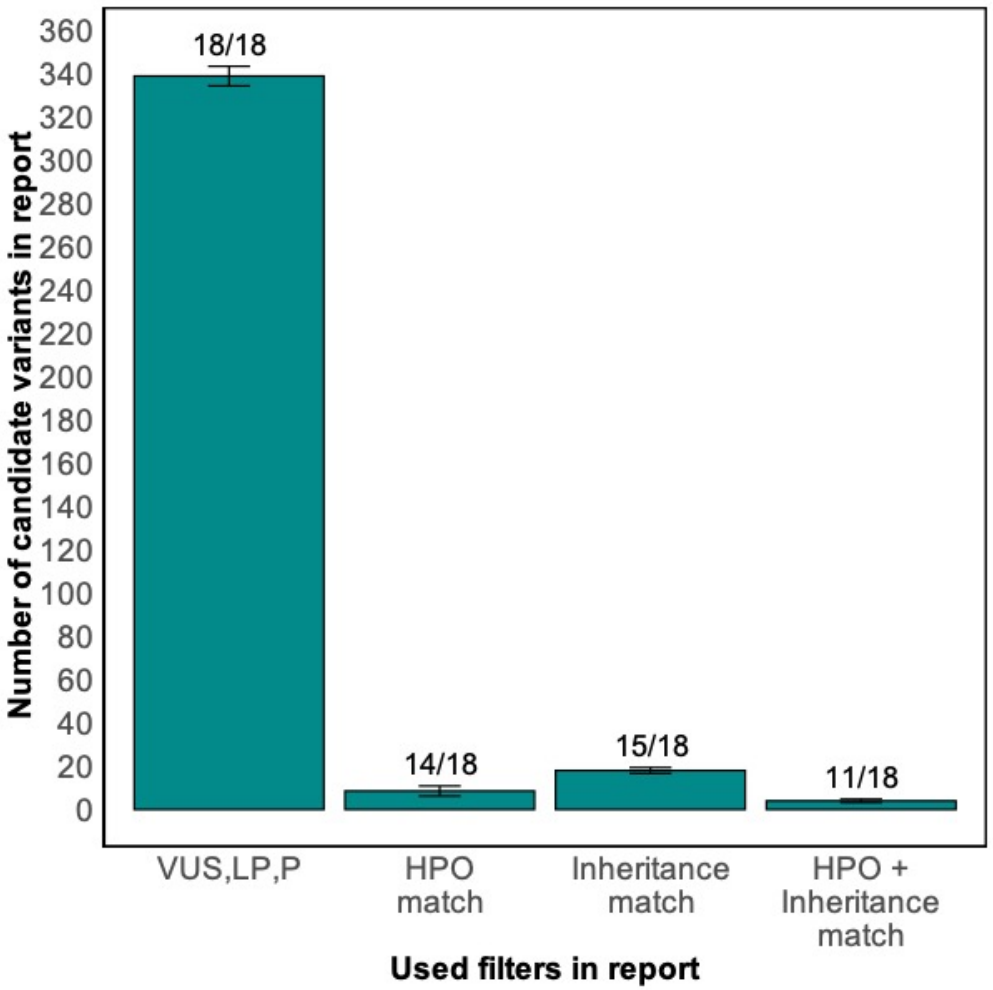
269 records

1 2 3 ... 14 Next

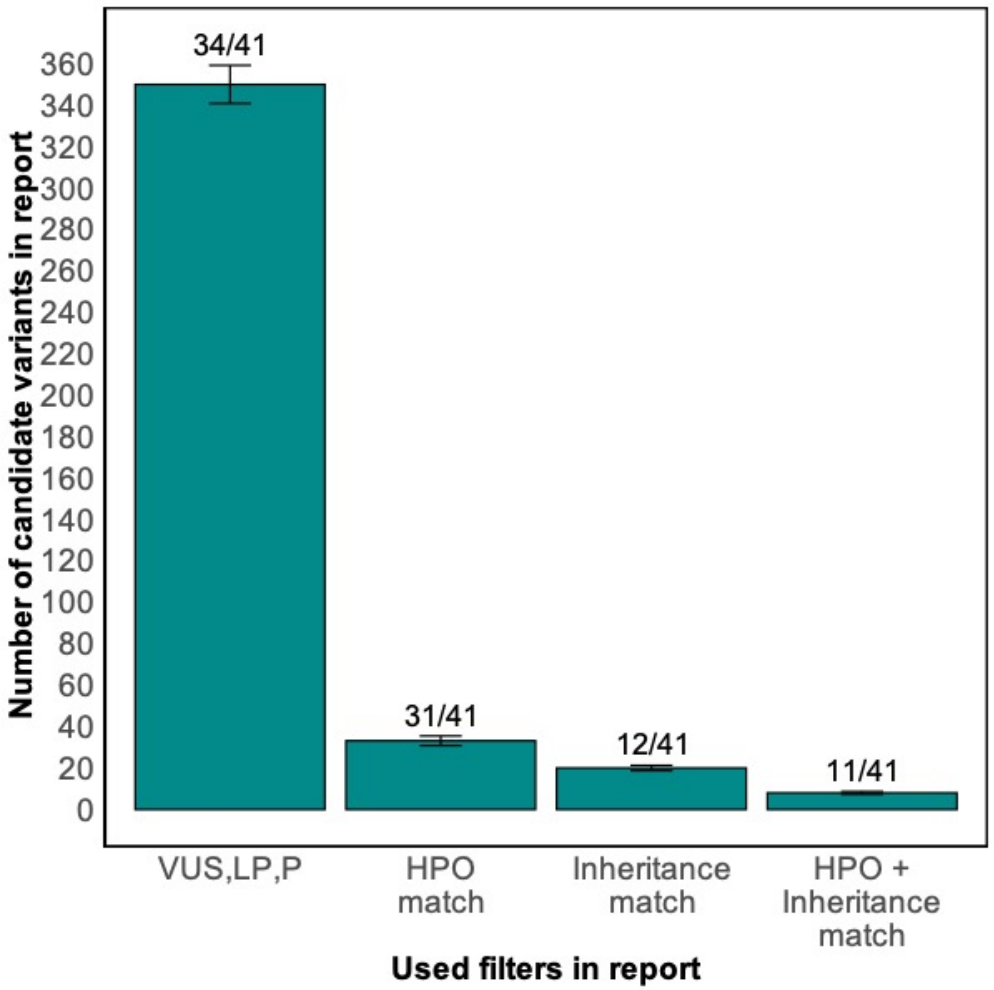
Position	Reference	Proband	Mother	Father	Effect	Gene	Inh.Pat.	HPO	HGVS.C	HGVS.P	CAPICE	VIP
19:40711875	GA	GA/G	GA/GA	GA/GA	frameshift_variant	MAP3K10			c.1247del	p.Glu416GlyfsTer30	0.9996	LP
1:144915623	AG	A/AG	?/A	A/AG	frameshift_variant	PDE4DIP			c.2212del	p.Leu738CysfsTer12	0.9993	LP
6:57398270	C	C/T	C/T	C/T	stop_gained	PRIM2		?	c.973C>T	p.Gln325Ter	0.9988	LP
17:16068343	G	G/A	G/A	G/A	stop_gained	NCOR1		i	c.568C>T	p.Arg190Ter	0.9988	LP
11:48347014	G	G/A	G/A	G/A	stop_gained	OR4C3			c.441G>A	p.Trp147Ter	0.9987	LP
1:144915561	G	G/A	G/A	G/A	stop_gained	PDE4DIP			c.2353C>T	p.Arg785Ter	0.9987	LP
13:41941576	A	A/AT	A/AT	A/A	frameshift_variant	NAA16			c.1548dup	p.Glu517Ter	0.9984	LP
17:45234406	CA	CA/C	CA/C	CA/C	frameshift_variant	CDC27			c.714del	p.Val239Ter	0.9983	LP
17:26708298	A	AGC/AGC	AGC/AGC	AGC/AGC	frameshift_variant	SARM1			c.542_543insCG	p.Ser181ArgfsTer65	0.9983	LP
13:21750538	G	G/A	G/A	G/A !	stop_gained	SKA3			c.79C>T	p.Arg27Ter	0.9982	LP
12:40882793	TAG	T/T	TAG/TAG	TAG/TAG	frameshift_variant	MUC19			c.16039_16040del	p.Arg5347GlyfsTer26	0.9979	LP
3:10088407	AG	AG/A	AG/A !	AG/A !	splice_donor_variant	FANCD2	AR	HP:0002650 ?	c.1278+1del		0.9971	LP
16:3119297	C	C/CG	C/CG	C/CG	frameshift_variant	IL32			c.454dup	p.Asp152GlyfsTer12	0.9969	LP
12:40881949	T	T/G	T/T	T/T	stop_gained	MUC19			c.15194T>G	p.Leu5065Ter	0.9967	LP
4:103822482	AAC	AAC/A	AAC/A	AAC/A	frameshift_variant	SLC9B1			c.1338_1339del	p.Leu447ArgfsTer47	0.9967	LP
1:12854090	T	T/A	T/A	T/A	stop_gained	PRAMEF1			c.314T>A	p.Leu105Ter	0.9964	LP
8:117779032	T	T/G	T/G	T/G	splice_donor_variant	UTP23			c.188+2T>G		0.9962	LP
7:72727122	T	T/TA	T/TA	T/T	frameshift_variant	TRIM50			c.1258dup	p.Tyr420LeufsTer2	0.9961	LP
1:144994658	C	C/A	C/A	C/A	missense_variant	PDE4DIP			c.83G>T	p.Arg28Leu	0.9958	LP
6:29910760	GGA	GGA/G	GGA/GGA !	GGA/GGA	frameshift_variant	HLA-A	AD		c.301_302del	p.Asp101ProfsTer75	0.9951	LP


25,664/25,664
variants
558/597
variants
100
90
80
70
60
50
40
30
20
10
0
Recall rate (%)
LP/P
release 2023-11
LP/P
newly added
Variants in VKGL database

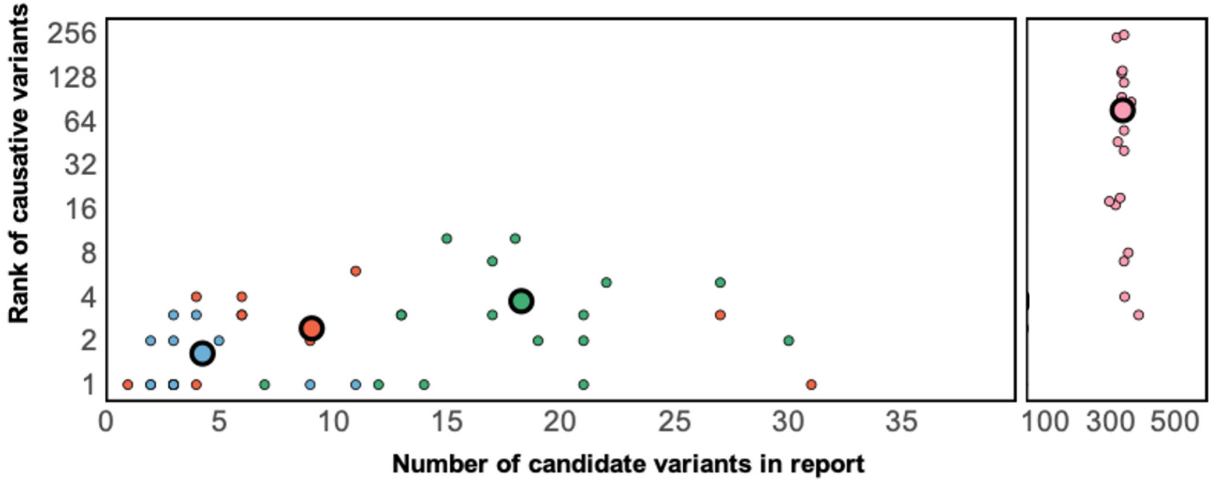
a


Number of candidate variants in report
360
340
320
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0
18/18
14/18
15/18
11/18
VUS,LP,P
HPO match
Inheritance match
HPO + Inheritance match
Used filters in report

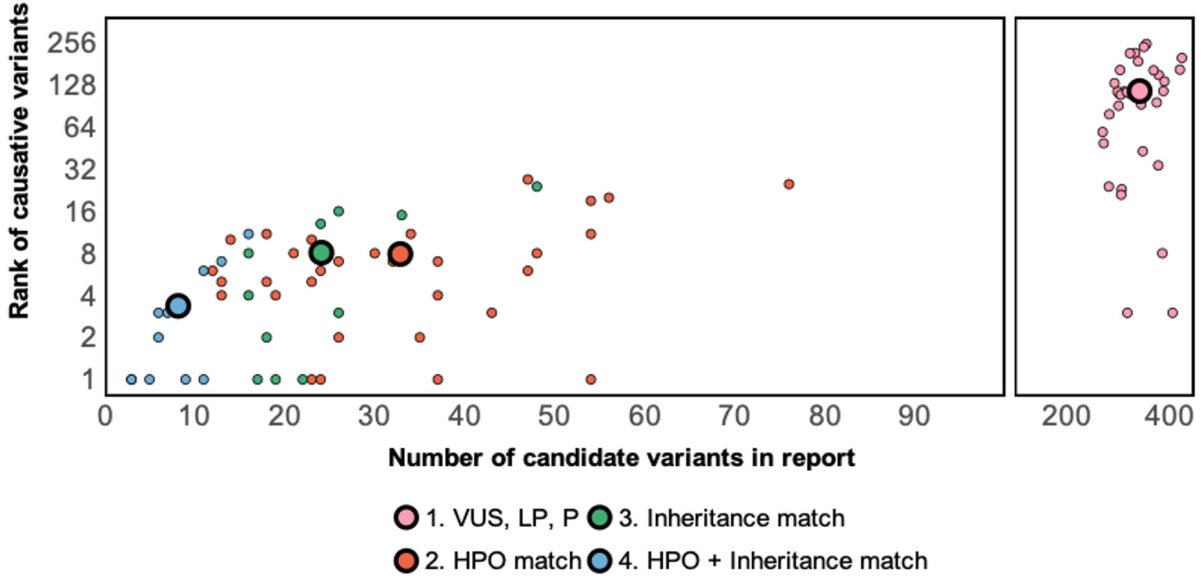
b


Number of candidate variants in report
360
340
320
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0
34/41
31/41
12/41
11/41
VUS,LP,P
HPO match
Inheritance match
HPO + Inheritance match
Used filters in report

a


Rank of causative variants
256
128
64
32
16
8
4
2
1
0
5
10
15
20
25
30
35
100
300
500
Number of candidate variants in report

b


Rank of causative variants
256
128
64
32
16
8
4
2
1
0
10
20
30
40
50
60
70
80
90
200
400
Number of candidate variants in report
1. VUS, LP, P
3. Inheritance match
2. HPO match
4. HPO + Inheritance match