doi: 10.1093/nar/gkx425

VCF.Filter: interactive prioritization of disease-linked genetic variants from sequencing data

Heiko Müller[1,2,*], Raul Jimenez-Heredia[1], Ana Krolo[1], Tatjana Hirschmugl[1], Jasmin Dmytrus[1], Kaan Boztug[1,3,4,5] and Christoph Bock[1,3,6,7,*]

[1]Ludwig Boltzmann Institute for Rare and Undiagnosed Diseases, 1090 Vienna, Austria, [2]Fondazione Istituto Italiano di Tecnologia, 16163 Genoa, Italy, [3]CeMM Research Center for Molecular Medicine of the Austrian Academy of Sciences, 1090 Vienna, Austria, [4]Department of Pediatrics and Adolescent Medicine, Medical University of Vienna, 1090 Vienna, Austria, [5]St. Anna Kinderspital and Children's Cancer Research Institute, Department of Pediatrics, Medical University of Vienna, 1090 Vienna, Austria, [6]Department of Laboratory Medicine, Medical University of Vienna, 1090 Vienna, Austria and [7]Max Planck Institute for Informatics, 66123 Saarbrücken, Germany

Received February 28, 2017; Revised May 01, 2017; Editorial Decision May 02, 2017; Accepted May 04, 2017

ABSTRACT

Next generation sequencing is widely used to link genetic variants to diseases, and it has massively accelerated the diagnosis and characterization of rare genetic diseases. After initial bioinformatic data processing, the interactive analysis of genome, exome, and panel sequencing data typically starts from lists of genetic variants in VCF format. Medical geneticists filter and annotate these lists to identify variants that may be relevant for the disease under investigation, or to select variants that are reported in a clinical diagnostics setting. We developed VCF.Filter to facilitate the search for disease-linked variants, providing a standalone Java program with a user-friendly interface for interactive variant filtering and annotation. VCF.Filter allows the user to define a broad range of filtering criteria through a graphical interface. Common workflows such as trio analysis and cohort-based filtering are pre-configured, and more complex analyses can be performed using VCF.Filter's support for custom annotations and filtering criteria. All filtering is documented in the results file, thus providing traceability of the interactive variant prioritization. VCF.Filter is an open source tool that is freely and openly available at http://vcffilter.rarediseases.at.

INTRODUCTION

Next generation sequencing facilitates the discovery of disease-linked genetic variants, and it is widely used for genetic diagnostics in routine clinical practice. The variant call format (VCF) has become the community standard for reporting genetic variant data from medical genetics research and diagnostics (1). VCF files describe each variant as a deviation from the indicated reference genome assembly. Standard columns include the variant's location in the reference genome, the bases that deviate from the reference, and the statistical confidence with which the variant was called. Additional annotations can be stored for each variant as key-value pairs in the INFO column, for example based on public databases such as dbSNP (2), ClinVar (3) and ExAC (4), or calculated using variant annotation tools such as SnpEff (5), ANNOVAR (6) and MutationTaster (7). Frequently used custom annotations include variant allele frequencies in healthy and diseased individuals as well as predictions of the deleterious impact of genetic variants, which can help decide whether or not a given variant may be pathogenic (8).

Medical geneticists use variant annotations to filter and prioritize the genetic variants observed in a patient, often taking additional data from other family members and from relevant cohorts into account. Although VCF files are text files that can in principle be viewed in a text editor or spreadsheet software, the VCF format defines a complex header and an internal data structure that requires dedicated tools for accurate and efficient variant filtering. Furthermore, VCF files can be quite large (millions of lines and hundreds of megabytes per sample when working with whole genome sequencing data), placing them beyond the limits of currentspreadsheet software including Microsoft Excel. There is thus a need for dedicated software tools that assist medical geneticists with VCF filtering and annotation, ideally under a user-friendly and efficient graphical interface.

We are currently aware of two non-commercial software tools that specialize on genetic variant filtering and prioritization using custom annotations, namely VCF-Miner (9) and BrowseVCF (10). Both tools have limitations when

[*]To whom correspondence should be addressed. Tel: +43 140 160 70018; Fax: +43 140 160 970000; Email: heiko.mueller@rud.lbg.ac.at
Correspondence may also be addressed to Christoph Bock. Tel: +43 140 160 70070; Fax: +43 140 160 970000; Email: christoph.bock@rud.lbg.ac.at

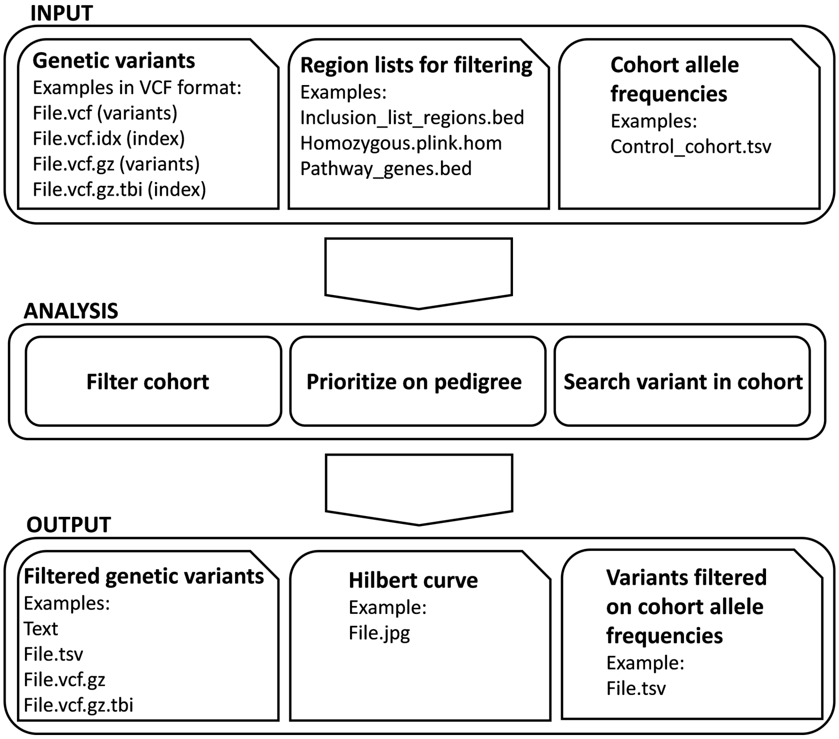


Figure 1. VCF.Filter overview and workflow for interactive prioritization of genetic variants. VCF.filter takes a VCF file as input, implements several filtering and analysis methods, and produces a filtered VCF file as output. The user can optionally provide additional input files for customized filtering, such as lists of genomic regions to include or exclude, and a table of allele frequencies in a reference cohort. VCF.Filter also produces a visualization of genetic variants using the Hilbert curve and tables of cohort allele frequencies for variant filtering.

working with families or cohorts that comprise a large number of samples. Additional tools exist for filtering VCF files based on a pre-defined set of annotations, including GEMINI (11), gNOME (12) or BiERapp (13), but these software packages are less flexible in the filtering rules and prioritization workflows provided. Finally, multiple command line tools provide VCF variant filtering (1,11,14–16), which is useful for automated workflows but less suitable for interactive analysis. A list of free academic software tools for VCF data filtering as well as some of their commercial alternatives is included in Supplementary Data S1.

We developed VCF.Filter as an easy-to-use, standalone, graphical software tool that is freely and openly available under the GNU GPL v3 open source license. VCF.Filter provides broad support for the VCF format (including the recently released VCF version 4.2), custom annotations, large VCF files, and flexible analysis types. VCF.Filter takes indexed VCF files as input and allows the user to interactively define, run, and save filter chains of any complexity using default and custom variant annotations. The output is a customizable list of fields in tab-separated value format that can be viewed, copied (e.g. to a spreadsheet), or saved as text files. Results can also be written directly to a new VCF file. VCF.Filter has been developed in close collaboration with medical geneticists and extensively tested. It is now routinely used for filtering VCF files obtained by bioinformatic processing pipelines built on the widely used GATK software (17).

Software input

VCF.Filter is a standalone application written in the Java programming language. It supports three types of input files (Figure 1). VCF files are the main input, providing sample-specific lists of genetic variants that are to be filtered. In addition, the user can also provide lists of genomic regions to help with the filtering, for example to focus the analysis on defined candidate genes (in clinical diagnostics), on homozygous regions (in consanguineous patients), or to exclude regions that are either irrelevant for the specific question or prone to false positive variant calls.

VCF.Filter accepts VCF files as indexed, non-indexed, compressed, and uncompressed files. Indexed and compressed VCF files are the preferred input of VCF.Filter because of their compactness and efficient computational access for filtering. Non-indexed VCF files can be indexed by VCF.Filter to prepare them for filtering. VCF files can contain a single sample or multiple samples, although single-sample VCF files are required for pedigree filtering. There is no size limit on VCF files, and files in the range of several gigabytes can be processed with VCF.Filter.

Lists of genomic regions can be used for inclusion (pass) and exclusion (non-pass) filtering. For example, parts of the genome that are identical by descent are often used in pedigree filtering of consanguineous families with a recessive mode of inheritance. This approach restricts the search space to those regions of the genome that most likely contain the disease-causing allele in a homozygous state. Such regions can be identified using genotyping microarray data analyzed with HomozygosityMapper or PLINK (18,19), or they can be inferred directly from sequencing data (20). Lists of genes that represent certain pathways or known disease-causing genes are further examples for using the list filtering functionality of VCF.Filter. The tool supports region lists in .bed format and in PLINK's .hom format (19).

Disease-causing genetic variants tend to be rare in a given population, and data on population allele frequencies is often used to exclude genetic variants that are most likely not disease-causing given large allele frequencies in the general population. Public resources such as 1000 Genomes (with 2,054 whole genomes) (21,22) and the Exome Aggregation Consortium (with 60,706 exomes) (23) provide extensive data on allele frequencies that are publicly available. Moreover, because the distribution of variants differs between countries and cohorts, it is advantageous to use custom allele frequency data for the studied population in addition to public databases, in order to discard variants linked to ethnicity or systematic errors. VCF.Filter can use—and also help generate—such cohort-specific tables of allele frequencies.

Variant filtering

VCF.Filter allows the user to design arbitrarily complex filtering rules ('filter chains') using a graphical interface and to apply them to any VCF file. For custom annotations in the VCF file, which can for example include clinical variant annotations or predictions of the deleterious impact, VCF.Filter interrogates the VCF header lines to read the ID, the Number, the Type, and the Description attributes that are reported for each annotation. Based on these values, VCF.Filter creates a dynamic filtering element on the graphical user interface that allows the user to define filtering rules for each annotation interactively. From a technical perspective, the Number and the Type attributes enable the Filter instance to apply the correct data types and operators, and to distinguish reference and alternative alleles. VCF.Filter can handle any combination of Number and Type attributes (https://samtools.github.io/hts-specs/VCFv4.2.pdf), thereby supporting complex filtering based on any custom annotations. During the filtering process, the VCF parser creates an instance of the 'VariantContext' class for each line in the VCF file. These instances are passed to the filter chain, and only variants that pass all filters are reported in the output. Parsing and manipulation of VCF data by VCF.Filter builds on two software packages developed by the Broad Institute, htsjdk (https://github.com/samtools/htsjdk) and Picard (http://broadinstitute.github.io/picard), which ensures full compatibility with variant call data produced by GATK-based pipelines (17).

The 'Filter variants' module is provided on the first tab of the VCF.Filter interface (Figure 2). Multiple VCF files can be loaded simultaneously and filtered for genetic variants that pass the defined filter chain, for example filtering for chromosome, position, and variant quality. Lists of genomic regions for inclusion (pass) and exclusion (non-pass) can be added to the analysis, and a table with cohort allele frequencies can also be included to filter out common genetic variants. In a typical application, the user loads one or more VCF files corresponding to patients with a rare genetic disease and then defines filters for tissue-specific expression patterns, predicted impact of the variant on protein function, and CADD score. The user also defines an upper limit on the cohort allele frequencies and perhaps restricts the results to variants located in regions of homozygosity. The output of the filtering is shown in the results area of VCF.Filter, from where it can be copied or saved, or it can be written directly to a VCF file. The chosen filter settings are documented in the VCF header to ensure traceability and reproducibility. If a visual representation of the filtering result is desired, an image of a Hilbert curve (24) can be generated where the variants are represented as colored dots organized according to their location in the genome.

The 'Filter variants' module is also used to generate tables of cohort allele frequencies based on a set of VCF files representing the cohort. Here, all VCF files of the cohort are loaded, and any desired filter chains are defined. VCF.Filter then counts the number of occurrences of each variant in the cohort that passes the filter chain and writes the cohort allele frequencies to the results area for visualization and saving.

The family analysis module of VCF.Filter (second tab on the graphical interface) facilitates variant filtering on pedigrees to identify genetic variants that segregate with a disease according to a defined mode of inheritance. The module requires that VCF files of affected and unaffected individuals are loaded separately. Where possible, parent–child relationships are defined, and the sex of affected individuals is indicated. When data from unaffected family members are not available, a cohort of healthy individuals may be used as controls. Filter chains, region inclusion/exclusion lists, and cohort allele frequencies can be loaded prior to starting the analysis. VCF.Filter can help select genetic variants according to the following modes of inheritance: dominant, recessive, compound heterozygous, X-linked, and *de novo* variants.

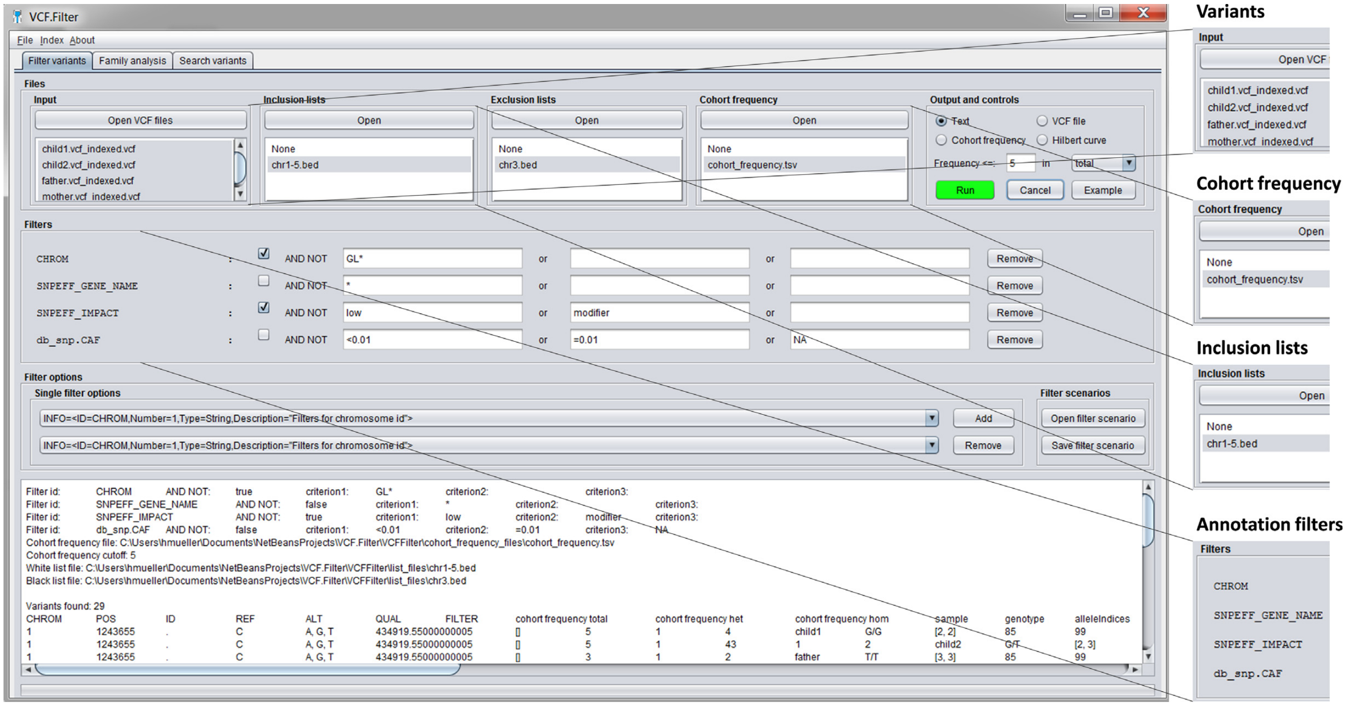
Dominant candidate variants are identified as variants that are present in all affected individuals in heterozygous or homozygous state, while being absent from all unaffected individuals.

Recessive candidate variants are reported when the candidate variant is present in homozygous state in all affected individuals and heterozygous or absent in all unaffected individuals.

Compound heterozygous candidate variants are reported when all affected individuals carry two different alleles mapping to the same gene while this is not the case for unaffected individuals and neither of the two alleles are in homozygous state in unaffected parents.

X-linked candidate variants are identified as those variants that are present in all affected males, heterozygous in their mothers and absent from all unaffected males.

De novo variants are reported when the child's genotype is inconsistent with the genotype of the parents, the variant is present in all affected individuals, and it is absent from all unaffected individuals. This variant set tends to include many sequencing errors and requires particularly careful filtering.



Input: child1.vcf_indexed.vcf, child2.vcf_indexed.vcf, father.vcf_indexed.vcf, mother.vcf_indexed.vcf

Inclusion lists: None, chr1-5.bed

Exclusion lists: None, chr3.bed

Cohort frequency: None, cohort_frequency.tsv

Output and controls: ● Text ○ VCF file ○ Cohort frequency ○ Hilbert curve; Frequency <=: 5 in total

CHROM	:	☑	AND NOT	GL*	or		or	
SNPEFF_GENE_NAME	:	☐	AND NOT	*	or		or	
SNPEFF_IMPACT	:	☑	AND NOT	low	or	modifier	or	
db_snp.CAF	:	☐	AND NOT	<0.01	or	=0.01	or	NA

INFO=<ID=CHROM,Number=1,Type=String,Description="Filters for chromosome id">

INFO=<ID=CHROM,Number=1,Type=String,Description="Filters for chromosome id">

```
Filter id:   CHROM              AND NOT:   true        criterion1:   GL*         criterion2:                 criterion3:
Filter id:   SNPEFF_GENE_NAME   AND NOT:   false       criterion1:   *           criterion2:                 criterion3:
Filter id:   SNPEFF_IMPACT      AND NOT:   true        criterion1:   low         criterion2:   modifier      criterion3:
Filter id:   db_snp.CAF   AND NOT:   false       criterion1:   <0.01       criterion2:   =0.01       criterion3:   NA
Cohort frequency file: C:\Users\hmueller\Documents\NetBeansProjects\VCF.Filter\VCFFilter\cohort_frequency_files\cohort_frequency.tsv
Cohort frequency cutoff: 5
White list file: C:\Users\hmueller\Documents\NetBeansProjects\VCF.Filter\VCFFilter\list_files\chr1-5.bed
Black list file: C:\Users\hmueller\Documents\NetBeansProjects\VCF.Filter\VCFFilter\list_files\chr3.bed

Variants found: 29
CHROM   POS       ID   REF   ALT       QUAL                 FILTER   cohort frequency total   cohort frequency het   cohort frequency hom   sample   genotype   alleleIndices
1       1243655   .    C     A, G, T   434919.55000000005   []       5                        1        4             child1      G/G        [2, 2]   85   99
1       1243655   .    C     A, G, T   434919.55000000005   []       5                        1        43            1           2          child2   G/T  [2, 3]
1       1243655   .    C     A, G, T   434919.55000000005   []       3                        1        2             father      T/T        [3, 3]   85   99
```

Input: child1.vcf_indexed.vcf, child2.vcf_indexed.vcf, father.vcf_indexed.vcf, mother.vcf_indexed.vcf

Cohort frequency: None, cohort_frequency.tsv

Inclusion lists: None, chr1-5.bed

Filters: CHROM, SNPEFF_GENE_NAME, SNPEFF_IMPACT, db_snp.CAF

Figure 2. Analysis example using the 'Filter variants' module of VCF.Filter. Once a VCF file has been loaded into VCF.Filter, the user can interactively define filtering rules, for example to identify disease-linked variants. Filtering can be based on annotations in the VCF file (e.g. variant type, location, or variant effect predictions). In addition, the user can upload a table of variant allele frequencies in a suitable reference cohort and filter potentially disease-linked variants by cohort allele frequencies. Furthermore, lists of genomic regions for inclusion (pass) and exclusion (non-pass) can be configured to restrict the analysis to certain genomic regions and/or to specifically exclude other regions. This screenshot shows an example with filtering on one standard VCF field (CHROM) and on three custom VCF annotations. The output is written to the text area in the bottom part of the user interface and can also be saved to a VCF file.

Unaffected carriers and incomplete penetrance are not considered in these analyses and will lead to loss of candidate variants. Where possible, child genotypes are tested for compatibility with parent genotypes, and inconsistent variants will be reported only when they fit the definition of a *de novo* variant.

Finally, the variant search module of VCF.Filter (third tab of the graphical interface) helps the user to find VCF files carrying a variant of interest, for example to assess if patients are present in a local cohort that carry a variant reported in a publication or by a collaborator. To start a search, all VCF files of the cohort are loaded, and the variant(s) of interest are provided in ExAC format (e.g. 14-106203380-C-T) or as a .bed file. When the search is started, the presence of the variant(s) in each VCF file of the cohort is checked with the help of the index file. If a variant is found, it is reported once for each VCF file it was found in.

Example analysis

To illustrate the analysis modes of VCF.Filter on a concrete example, we analyzed a whole genome sequenced trio (25) provided by the Genome in a Bottle Consortium (26). We chose this generic example for two reasons: first, although we routinely filter patient data, these data are subject to privacy restrictions and cannot be shared as raw, personally identifying, genetic variant information. Second, the whole genome sequence data reported for this trio illustrate the capability of VCF.Filter to analyze very large VCF files.

The analyzed trio represents a family of Ashkenazi origin and the data can be downloaded from ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/AshkenazimTrio/ using the identifiers HG002_NA24385_son, HG003_NA24149_father and HG004_NA24143_mother. We used the version 3.3.2 high-confidence SNP, small indel and homozygous reference calls. For each individual, a compressed indexed VCF file is provided, which contains 3–4 million variants on Chromosome 1–22. Variants on sex chromosomes are not listed. The variants are annotated with 16 different custom fields pertaining mainly to technical details regarding the sequencing on different platforms and the number of datasets that have been used. The full list of URLs to the data and the annotation fields used are shown in Supplementary Data S1.

We focused on those variants that have been interrogated in depth (platforms > 2, datasets > 4) to minimize the amount of experimental noise in the data. Filtering the file HG002_NA24385_son with these criteria returned 3,000,500 genetic variants that were written to an indexed VCF file in 73 seconds. For a more detailed analysis, we selected 90,039 filtered genetic variants located on Chromosome 22 (Supplementary Data S1). We then applied VCF.Filter's family analysis, pursuing three alternative scenarios:

son affected—mother and father unaffected
son and father affected—mother unaffected
son and mother affected—father unaffected

Since there is no clinical information, here the term 'affected' may refer to some unspecified monogenic phenotype shared between individuals that is absent from 'unaffected' individuals.

The results for the search of recessive, dominant, and *de novo* variants of high quality (platforms > 2, datasets > 4) located on Chromosome 22 are reported in Supplementary Data S1 (pedigree summary sheet). For each scenario, the analysis took less than 1 minute to complete on a laptop computer running Windows 7 Professional SP1 (64 bit) with four CPUs (Intel Core i7-5600U CPU@2.60Ghz) and 16 gigabytes of RAM.

The results confirmed that different sets of variants are returned depending on which individuals are defined as affected. In particular, when both parents are unaffected, zero dominant variants are found because at least one parent would have to be a carrier (incomplete penetrance is not considered by VCF.Filter). When one parent is affected along with the child, significantly fewer recessive variants are reported because recessive variants are required to be homozygous for all affected individuals, which is less likely in two individuals than in one. Finally, the number of *de novo* variants is relatively high notwithstanding the high quality variant calls in this curated dataset, which can be explained by sequencing errors and other technical issues related to variant calling. The high number of false positives could likely be reduced by variant quality score recalibration on a large cohort and by using genotype refinement approaches (27).

The results of all searches are reported in Supplementary Data S1. The output contains columns provided by VCF.Filter for all analyses. For convenient comparison of genotypes between children and parents, the genotypes of all individuals are reported along with the observed depth of sequencing coverage. The allele frequency of each genetic variant in the cohort is also reported where available. Furthermore, VCF.Filter makes it possible for the user to configure hyperlinks that make it easy to look up additional information for individual genetic variants, using databases such as dbSNP (2) or ExAC (4). For each configured hyperlink, an extra column is shown in the output.

DISCUSSION

VCF.Filter was developed in close collaboration with medical geneticists working on rare diseases of the immune system. The tool has been used to analyze 300 exomes and 500 gene panels sequenced at CeMM and the Ludwig Boltzmann Institute for Rare and Undiagnosed Diseases, and 20 whole genome datasets produced by the Genom Austria project (http://genomaustria.at/). Pathogenic variants in several families have been identified with the help of VCF.Filter and successfully validated, including cases that had previously gone unsolved using other tools and manual review.

The envisioned users of VCF.Filter are medical geneticists and biologists who analyze genetic variant data in the context of biomedical research and clinical diagnostics. Because VCF.Filter is a standalone tool, there is no upload of sensitive genetic data to external websites. Furthermore, VCF.Filter is open source, meaning that the user has full knowledge and control about the way in which the data are processed.

VCF.Filter can be customized in a number of ways to provide flexible support for complex analyses and user-specific workflows. For example, VCF.Filter permits filtering variants based on custom annotations, which is a non-trivial feature because it requires parsing of VCF headers and dynamic creation of adequate filtering modules and user interface components. The support for custom annotations also eliminates the need to preprocess and annotate VCF files in a tool-specific way prior to variant filtering.

User-defined filter chains are interactively designed with a graphical user interface. Filters can be saved and reloaded, allowing users to define and share standardized analysis workflows. Furthermore, VCF.Filter allows the user to configure variant-specific hyperlinks to external resources, which facilitates the task of interactively annotating disease-linked variants based on multiple public and private databases.

VCF.Filter can also be used to look up or filter variants in a large set of VCF files, to convert a VCF file into a spreadsheet compatible table format, to analyze variant frequency across a potentially large cohort, or to prioritize variants based on a pedigree. File size is not a limiting factor for these analyses because the data are accessed using an index file rather than loading the entire dataset into memory.

Comparing VCF.Filter to related software, its focus on variant filtering is complementary to existing variant annotation tools such as SnpEff (5) and Ensembl Variant Effect Predictor (28). To use annotations from such tools in VCF.Filter, the VCF files should be annotated with a suitable tool prior to loading them into VCF.Filter. The current version of VCF.Filter does not calculate linkage parameters, mutational burden tests or knowledge enrichment tests as is done in tools like KEGGSeq (16) and gNOME (12). Furthermore, it assumes complete penetrance and does not calculate segregation scores based on error models for nonconsistent genotype calls as is done in MendelScan (29). For family analysis, VCF.Filter takes advantage of the parent–child relationships to perform genotype consistency tests, while a more formal approach to prioritizing variants based on pedigree information exists (30). VCF.Filter does not currently provide dedicated support for somatic mutation analysis in cancer, although custom filter chains could be designed.

In summary, we have developed VCF.Filter as a user-friendly and effective tool for VCF filtering and variant gene prioritization in the context of medical genetics and rare diseases research.

AVAILABILITY

VCF.Filter is freely and openly available at http://vcffilter.rarediseases.at. This website provides a Java Web Start link for direct launch of the tool, a downloadable software archive for local installation, a detailed tutorial as well as other technical information, and a link to the source code hosted on Github.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We would like to thank all members of the Biomedical Sequencing Facility for processing the test data. We also thank all test users of VCF.Filter at CeMM and LBI-RUD for their valuable feedback.

FUNDING

Ludwig Boltzmann Institute for Rare and Undiagnosed Diseases; Austrian Academy of Sciences New Frontiers Group Award (to C.B.); ERC Starting Grant (European Union's Horizon 2020 research and innovation programme) [679146 to C.B. and 310857 to K.B.]. Funding for open access charge: CeMM Research Center for Molecular Medicine of the Austrian Academy of Sciences.

Conflict of interest statement. None declared.

REFERENCES

1. Danecek,P., Auton,A., Abecasis,G., Albers,C.A., Banks,E., DePristo,M.A., Handsaker,R.E., Lunter,G., Marth,G.T., Sherry,S.T. *et al.* (2011) The variant call format and VCFtools. *Bioinformatics*, **27**, 2156–2158.
2. Sherry,S.T., Ward,M. and Sirotkin,K. (1999) dbSNP-database for single nucleotide polymorphisms and other classes of minor genetic variation. *Genome Res.*, **9**, 677–679.
3. Landrum,M.J., Lee,J.M., Riley,G.R., Jang,W., Rubinstein,W.S., Church,D.M. and Maglott,D.R. (2014) ClinVar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Res.*, **42**, D980–D985.
4. Lek,M., Karczewski,K.J., Minikel,E.V., Samocha,K.E., Banks,E., Fennell,T., O'Donnell-Luria,A.H., Ware,J.S., Hill,A.J., Cummings,B.B. *et al.* (2016) Analysis of protein-coding genetic variation in 60,706 humans. *Nature*, **536**, 285–291.
5. Cingolani,P., Platts,A., Wang le,L., Coon,M., Nguyen,T., Wang,L., Land,S.J., Lu,X. and Ruden,D.M. (2012) A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of Drosophila melanogaster strain w1118; iso-2; iso-3. *Fly (Austin)*, **6**, 80–92.
6. Wang,K., Li,M. and Hakonarson,H. (2010) ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Res.*, **38**, e164.
7. Schwarz,J.M., Rodelsperger,C., Schuelke,M. and Seelow,D. (2010) MutationTaster evaluates disease-causing potential of sequence alterations. *Nat. Methods*, **7**, 575–576.
8. Shameer,K., Tripathi,L.P., Kalari,K.R., Dudley,J.T. and Sowdhamini,R. (2016) Interpreting functional effects of coding variants: challenges in proteome-scale prediction, annotation and assessment. *Brief. Bioinform.*, **17**, 841–862.
9. Hart,S.N., Duffy,P., Quest,D.J., Hossain,A., Meiners,M.A. and Kocher,J.P. (2016) VCF-Miner: GUI-based application for mining variants and annotations stored in VCF files. *Brief. Bioinform.*, **17**, 346–351.
10. Salatino,S. and Ramraj,V. (2016) BrowseVCF: a web-based application and workflow to quickly prioritize disease-causative variants in VCF files. *Brief. Bioinform.*, bbw054.
11. Paila,U., Chapman,B.A., Kirchner,R. and Quinlan,A.R. (2013) GEMINI: integrative exploration of genetic variation and genome annotations. *PLoS Comput. Biol.*, **9**, e1003153.
12. Lee,I.H., Lee,K., Hsing,M., Choe,Y., Park,J.H., Kim,S.H., Bohn,J.M., Neu,M.B., Hwang,K.B., Green,R.C. *et al.* (2014) Prioritizing disease-linked variants, genes, and pathways with an interactive whole-genome analysis pipeline. *Hum. Mutat.*, **35**, 537–547.
13. Aleman,A., Garcia-Garcia,F., Salavert,F., Medina,I. and Dopazo,J. (2014) A web-based interactive framework to assist in the prioritization of disease candidate genes in whole-exome sequencing studies. *Nucleic Acids Res.*, **42**, W88–W93.
14. Cingolani,P., Patel,V.M., Coon,M., Nguyen,T., Land,S.J., Ruden,D.M. and Lu,X. (2012) Using Drosophila melanogaster as a model for genotoxic chemical mutational studies with a new program, SnpSift. *Front. Genet.*, **3**, 35.
15. Maranhao,B., Biswas,P., Duncan,J.L., Branham,K.E., Silva,G.A., Naeem,M.A., Khan,S.N., Riazuddin,S., Hejtmancik,J.F., Heckenlively,J.R. *et al.* (2014) exomeSuite: whole exome sequence variant filtering tool for rapid identification of putative disease causing SNVs/indels. *Genomics*, **103**, 169–176.
16. Li,M.X., Gui,H.S., Kwan,J.S., Bao,S.Y. and Sham,P.C. (2012) A comprehensive framework for prioritizing variants in exome sequencing studies of Mendelian diseases. *Nucleic Acids Res.*, **40**, e53.
17. McKenna,A., Hanna,M., Banks,E., Sivachenko,A., Cibulskis,K., Kernytsky,A., Garimella,K., Altshuler,D., Gabriel,S., Daly,M. *et al.* (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.*, **20**, 1297–1303.
18. Seelow,D., Schuelke,M., Hildebrandt,F. and Nurnberg,P. (2009) HomozygosityMapper—an interactive approach to homozygosity mapping. *Nucleic Acids Res.*, **37**, W593–W599.
19. Purcell,S., Neale,B., Todd-Brown,K., Thomas,L., Ferreira,M.A., Bender,D., Maller,J., Sklar,P., de Bakker,P.I., Daly,M.J. *et al.* (2007) PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.*, **81**, 559–575.
20. Magi,A., Tattini,L., Palombo,F., Benelli,M., Gialluisi,A., Giusti,B., Abbate,R., Seri,M., Gensini,G.F., Romeo,G. *et al.* (2014) H3M2: detection of runs of homozygosity from whole-exome sequencing data. *Bioinformatics*, **30**, 2852–2859.
21. Auton,A., Brooks,L.D., Durbin,R.M., Garrison,E.P., Kang,H.M., Korbel,J.O., Marchini,J.L., McCarthy,S., McVean,G.A. and Abecasis,G.R. (2015) A global reference for human genetic variation. *Nature*, **526**, 68–74.
22. Sudmant,P.H., Rausch,T., Gardner,E.J., Handsaker,R.E., Abyzov,A., Huddleston,J., Zhang,Y., Ye,K., Jun,G., Hsi-Yang Fritz,M. *et al.* (2015) An integrated map of structural variation in 2,504 human genomes. *Nature*, **526**, 75–81.
23. Bahcall,O.G. (2016) Genetic variation: ExAC boosts clinical variant interpretation in rare diseases. *Nat. Rev. Genet.*, **17**, 584.
24. Anders,S. (2009) Visualization of genomic data with the Hilbert curve. *Bioinformatics*, **25**, 1231–1235.
25. Zook,J.M., Catoe,D., McDaniel,J., Vang,L., Spies,N., Sidow,A., Weng,Z., Liu,Y., Mason,C.E., Alexander,N. *et al.* (2016) Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Sci. Data*, **3**, 160025.
26. Zook,J.M., Chapman,B., Wang,J., Mittelman,D., Hofmann,O., Hide,W. and Salit,M. (2014) Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat. Biotechnol.*, **32**, 246–251.
27. do Valle,I.F., Giampieri,E., Simonetti,G., Padella,A., Manfrini,M., Ferrari,A., Papayannidis,C., Zironi,I., Garonzi,M., Bernardi,S. *et al.* (2016) Optimized pipeline of MuTect and GATK tools to improve the detection of somatic single nucleotide polymorphisms in whole-exome sequencing data. *BMC Bioinformatics*, **17**(Suppl. 12), 99–107.
28. McLaren,W., Gil,L., Hunt,S.E., Riat,H.S., Ritchie,G.R., Thormann,A., Flicek,P. and Cunningham,F. (2016) The Ensembl Variant Effect Predictor. *Genome Biol.*, **17**, 122.
29. Koboldt,D.C., Larson,D.E., Sullivan,L.S., Bowne,S.J., Steinberg,K.M., Churchill,J.D., Buhr,A.C., Nutter,N., Pierce,E.A., Blanton,S.H. *et al.* (2014) Exome-based mapping and variant prioritization for inherited Mendelian disorders. *Am. J. Hum. Genet.*, **94**, 373–384.
30. Peng,G., Fan,Y., Palculict,T.B., Shen,P., Ruteshouser,E.C., Chi,A.K., Davis,R.W., Huff,V., Scharfe,C. and Wang,W. (2013) Rare variant detection using family-based sequencing analysis. *Proc. Natl. Acad. Sci. U.S.A.*, **110**, 3985–3990.