AdaGenes: A streaming processor for high-throughput annotation and filtering of sequence variant data

Nadine S. Kurz[1,2*], Klara Drofenik[1,2], Kevin Kornrumpf[1], Kirsten Reuter-Jessen[3], Jürgen Dönitz[1,2,4]

[1*]Department of Medical Bioinformatics, University Medical Center Göttingen, Goldschmidtstraße 1, Göttingen, 37075, Niedersachsen, Germany.
[2]Comprehensive Cancer Center Göttingen (G-CCC), Organization, Street, City, 37075, State, Germany.
[3]Institute of Pathology, University Medical Center Göttingen, Robert-Koch-Str. 40, Göttingen, Robert-Koch-Str. 40, Niedersachsen, Germany.
[4]Campus-Institut Data Science (CIDAS), Section Medical Data Science (MeDaS), Göttingen, Niedersachsen, Germany.

*Corresponding author(s). E-mail(s): nadine.kurz@bioinf.med.uni-goettingen.de;
Contributing authors: klara.drofenik1@med.uni-goettingen.de; kevin.kornrumpf@bioinf.med.uni-goettingen.de; kirsten.reuter-jessen@med.uni-goettingen.de; juergen.doenitz@bioinf.med.uni-goettingen.de;

Abstract

The amount of sequencing data resulting from whole exome or genome sequencing (WES / WGS) presents challenges for annotation, filtering, and analysis. We introduce the Adaptive Genes processor (AdaGenes), a sequence variant streaming processor designed to efficiently annotate, filter, LiftOver and transform large-scale VCF files. AdaGenes provides a unified solution for researchers to streamline VCF processing workflows and address common challenges in genomic data processing, e.g. to filter out non-relevant variants to focus on further processing of the relevant positions. AdaGenes integrates genomic, transcript and protein data

annotations, while maintaining scalability and performance for high-throughput workflows. Leveraging a streaming architecture, AdaGenes processes variant data incrementally, enabling high-performance on large files due to low memory consumption and seamless handling of whole genome files. The interactive front end provides the user with the ability to dynamically filter variants based on user-defined criteria. It allows researchers and clinicians to efficiently analyze large genomic datasets, facilitating variant interpretation in diverse genomics applications, such as population studies, clinical diagnostics, and precision medicine. AdaGenes is able to parse and convert multiple file formats while preserving metadata, and provides a report of the changes made to the variant file. AdaGenes is available at https://mtb.bioinf.med.uni-goettingen.de/adagenes.

Keywords: Variant interpretation, Variant filtering, Somatic mutations, Whole exome sequencing, Whole genome sequencing

1 Introduction

As next-generation sequencing (NGS) data accumulate at an unprecedented rate, the task of identifying and interpreting disease-causing variants becomes increasingly complex. A crucial step in interpreting sequencing data is to reduce the large number of variants to those that are classified as pathogenic or oncogenic to investigate the origins of genome-related diseases. Annotation and filtering of genetic variants are thus essential processes that transform raw genetic variant data into interpretable data.

Command-line based annotation pipelines [1–4], in some cases also extended with a web server [5], offer robust solutions for variant annotation. Second, there are VCF (Variant Call Format) filtering tools [6–17] that allow variants to be filtered according to certain annotation criteria, such as population allele frequency or ClinVar reports. In addition, there are graphical tools for variant exploration and analysis of VCF files [18–20] that provide summary statistics and interactive visualizations of genetic variant data. Since, in addition to the definition of variants in VCF format, the description of variants in HGVS format is increasingly gaining attention, there are tools for normalizing and validating variants in HGVS format [21, 22]. However, most of the tools require users to install and run these pipelines manually, which requires a good experience in setting up the tools and defining complex filtering rules on the command line. While each tool has its own advantages, most of the tools focus either on LiftOver, annotation or filtering, requiring users to utilize multiple tools.

Here we present a fast and powerful tool, the Adaptive Genes processor (AdaGenes), for variant annotation as well as filtering genetic variant data enhanced by a versatile web interface. AdaGenes is capable of processing whole genome

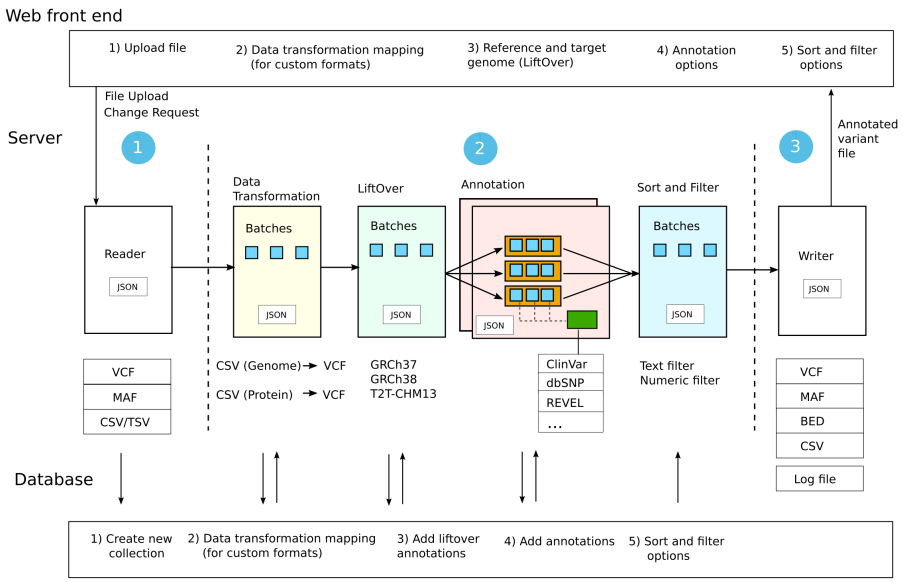


Fig. 1 The AdaGenes server workflow. AdaGenes supports 4 kinds of variant processing steps: The conversion of table-formatted data (e.g. CSV) at genome or protein level to VCF, LiftOver between reference genomes, the annotation of variants, and the sorting and filtering of variants. First, the AdaGenes server reads a variant file and converts it into an internal JSON-based representation (1). Next, the data is processed in several steps, utilizing modules (2). In all modules, large amounts of data are efficiently processed in small data blocks (batches, blue boxes). To accelerate annotation, large files are divided into smaller files (chunks, red boxes), which are then annotated in parallel. The chunk files' data is also divided into small batches, which are sent to the annotation modules (green box). The processed variants can then be written out again in one of the supported file formats, as well as a log file of all made modifications (3).

sequencing (WGS) variant files and provides an all-in-one solution for LiftOver, variant annotation, sorting, filtering, data transformations and interactive variant data exploration.

2 Results

We present AdaGenes, a high-throughput streaming processor for sequence variant annotation and filtering. AdaGenes' stream-based processing architecture allows it to annotate and filter large variant files. Its versatile web interface provides a straightforward application that allows the annotation, filtering and data transformations of genetic variant data into various formats. The interactive web application of AdaGenes consists of 3 main components: The server, the database, and the web front end (Figure 1). When a file is uploaded, AdaGenes converts the variant data into an internal, JSON-based representation and stores the variant data in the database. The server is able to perform different annotation and filtering steps, including the conversion of variant data at transcript and protein to genomic level, the LiftOver between

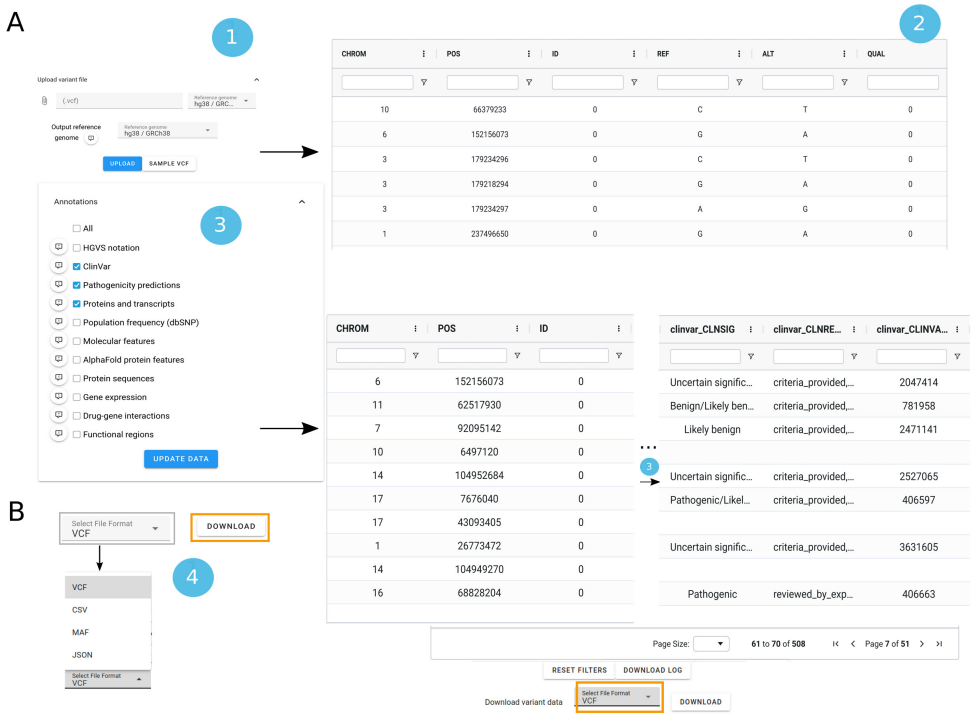


Fig. 2 Interactive annotation using AdaGenes. (**A**) The user selects a variant file to upload and specifies the file's reference genome, as well as the target reference genome (1). AdaGenes then generates an interactive table in the selected reference genome, where the VCF columns as well as each INFO annotation features is a separate column. After uploading the file, one or multiple annotation modules can be selected (3). After the annotation process, new features are added as columns that can be sorted and filtered (3). (**B**) The annotated variants can be downloaded multiple formats, including VCF, MAF, CSV and JSON (4).

genome assemblies (GRCH37, GRCH38, T2T-CHM13), annotation of variants, as well as sorting and filtering of variants.

2.1 Annotation of whole genome and exome sequencing data

AdaGenes provides a wide range of annotation features of genetic variants, including variant information at the genome and protein level, population allele frequency, ClinVar reports, pathogenicity predictions, functional regions, molecular characteristics, structural protein features, and drug-gene interactions. In order to annotate variants, the user first uploads a variant file in a standard format (VCF, MAF, CSV) and selects the corresponding reference genome. By selecting a different target genome, AdaGenes automatically performs a LiftOver to the selected target genome, while preserving the former genomic positions as annotations (POS_hg19, POS_hg38, POS_t2t). In order to

present variant annotations in a readily legible format, AdaGenes generates an interactive table that comprises the predefined columns of the VCF format - CHROM, POS, ID, REF, ALT, QUAL and FILTER - as well as a separate column for each annotation feature (Figure 2A). The user can now sort and filter the variant data, or annotate it with additional features. The annotation is possible by selecting one, multiple, or all modules for consultation from a list of available annotation modules. The annotation time per module depends on the database size and the complexity of the queries. The estimated expected processing speed can be viewed by clicking on the tool tips that are displayed next to each annotation module. Once the annotation process is complete, the variant table is automatically updated and the newly annotated properties are displayed as additional columns. Users can therefore immediately explore the data and filter variants according to the newly added features.

AdaGenes provides a comprehensive annotation of variants as well as new, unique annotation features related to the molecular and structure-specific feature of variants compared to existing variant annotation pipelines (Table 1). For example, AdaGenes annotates single nucleotide variants (SNVs) with the molecular characteristics of the reference and alternate amino acid, including the change in molecular weight, charge, hydropathy, flexibility, solubility, or ionization and phosphorylation potential. Structure-specific features are derived based on AlphaFold-predicted prtoein structures of the MANE Select transcripts, and include surface accessibility, secondary protein structure, Phi and Psi angles, as well as hydrogen bonds at the mutated site. Processed variants can be downloaded in multiple file formats, including VCF, MAF and CSV (Figure 2B).

Function	AdaGenes	ANNOVAR	SnpEff/ SnpSift	VEP
Sequence variant annotation				
HGVS	x	x	x	x
Functional regions	x	x	x	x
Protein and transcripts	x	x	x	x
Population allele frequency	x	x	x	x
Pathogenicity predictions	x	x	x	x
ClinVar	x	x	-	x
Splice variants	-	-	-	x
GO terms	-	-	-	x
Gene expression	x	-	-	-
Molecular features	x	-	-	-
Structural protein features	x	-	-	-
Protein sequences	x	-	-	-
Drug-gene interactions	x	-	-	-
Associated gene conditions	x	-	-	-

Table 1 Sequence variant annotation features of AdaGenes for the human genome compared to the variant annotation pipelines ANNOVAR, SnpEff and the Variant Effect Predictor (VEP).

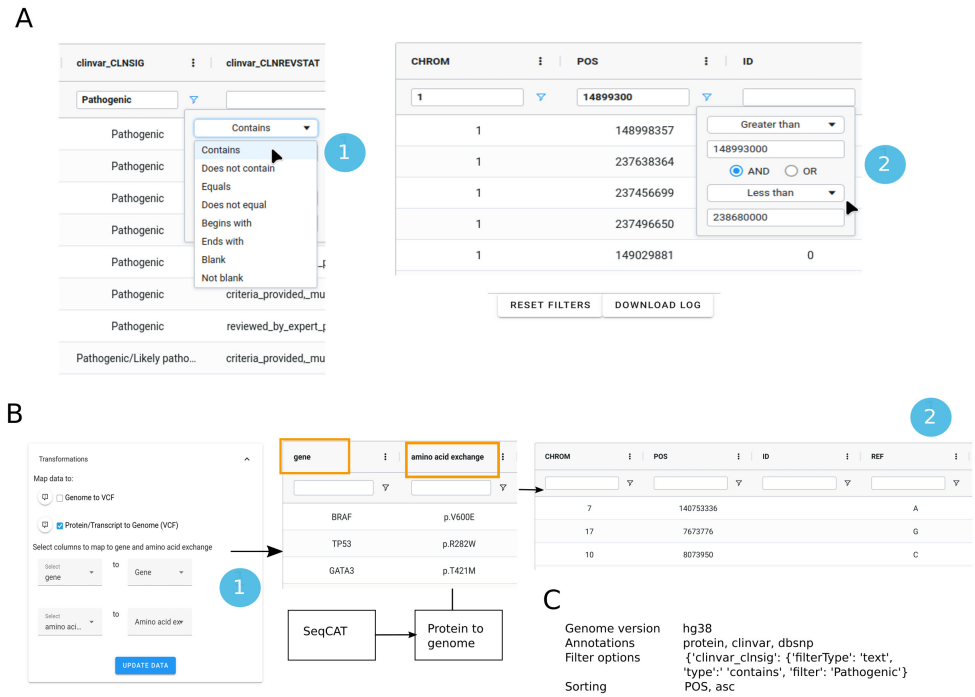


Fig. 3 Interactive filtering and protein to genome conversions. (**A**) Feature-specific variant filtering options in AdaGenes: Variants can be filtered with a free text search (1), including text-specific filtering options, as well as the generation of complex numeric queries (2) including the combination via AND/OR. (**B**) VCF data transformations in AdaGenes: Users can map custom data formats at transcript or protein level to VCF format by selecting columns from a table-formatted file (1), and the feature it should be mapped to. AdaGenes then converts variants from protein to genome level and generates a VCF representation (2). (**C**) AdaGenes log file entries, containing the reference genome, added annotations, as well as active filter and sort options.

2.2 Interactive variant filtering

After annotating the variants, the next step is to filter them using the added information to identify pathogenic variants. AdaGenes combines both the annotation and filtering of variants in one tool, whereby the filter criteria can be selected interactively in the result variant table. After a file has been uploaded, AdaGenes identifies the data type of each column based on the VCF metadata or the column data entries and automatically generates corresponding filtering options. It is possible to utilize a range of different filter options for text-based features (Figure 3A1), as well as numerical features (Figure 3A2) featuring the possibility to generate compound queries using AND or OR operators. AdaGenes generates a unique ID for each request, so that the annotation results for large files can be viewed again later. Using the link of the request page, the analysis can be accessed for 7 days before the data is deleted from

the server for data privacy reasons. Table 2 presents a comprehensive overview of the main available functions of AdaGenes, alongside a comparison with other variant annotation and filtering tools. All filter and sorting options can also be reset directly via a button in order to carry out a new filtering process.

2.3 Conversion from transcript and protein to genomic level

In addition to data from sequencing pipelines, biomedical research often includes data sets on genetic variation data that are available in their own data formats. For example, the CIViC database [23] is an important resource for expert-curated interpretation of cancer variants. However, the variants are available in the database at protein level (e.g. 'KRAS G12C'), which makes it difficult to annotate or analyze the variants with tools that require standardized data formats like VCF. For this purpose, AdaGenes uses SeqCAT [24] to convert own table-formatted data files into VCF format: Users can upload a file, whereupon the column names of the file are identified. Users can convert their own file formats to VCF by selecting the appropriate columns for genome positions or gene and variant identifiers (Figure 3B). To ensure reproducibility, AdaGenes also provides a downloadable log file, which records all modifications made to the original file, including annotations, LiftOver and applied filters (Figure 3C).

2.4 Command-line interface and local installation

In addition, the functions of AdaGenes are available as a Python package as well for usage from the command line. By using the command line based version, AdaGenes works without the need to install the database and the web front end by generating a new annotated or filtered file after each request. It is possible to install AdaGenes locally and use it in environments with restricted Internet access, e.g. clinical infrastructures. To perform the annotation completely locally, e.g. for sensitive patient data, Onkopus and SeqCAT can also be installed locally.

Function	AdaGenes	SnpEff/ SnpSift	VCFtools	BCFtools	GEMINI	BrowseVCF	VCF.Filter	VCF-Miner	VarFish
Web server	+	-	-	-	-	+[1]	-	-	+
GUI-based	+	-	-	-	+	-	+	+	-
Command-line interface	+	+	+	+	+	+	-	-	-
Input formats	VCF, MAF BED, CSV	VCF,BED	VCF,BCF	VCF,BCF	VCF	VCF	VCF	VCF	VCF
Output formats	VCF, MAF, CSV, JSON	VCF	VCF,BCF	VCF,BCF	Text	VCF	VCF,Text	tabular	tabular
LiftOver	+ (GRCh37, GRCh38, T2T)	-	-	-	-	-	-	-	-
Annotation	+	+	-	+	+	-	+	-	+
Filtering	+	+	+	+	+	+	+	+	+
Sorting	+	+	+	+	-	+	+	+	+
Transformations $^{\infty}$	+	-	-	+	-	-	-	-	-
Custom to VCF	+	-	-	-	-	-	-	-	-
VCF to custom	-	-	-	+	-	-	-	-	-
Docker	+	-	-	-	-	-	+	+	+

Table 2 Comparison of AdaGenes to other variant annotation and filtering tools

[1] No public web server instance available

3 Methods

3.1 The AdaGenes server

The AdaGenes server was implemented using Python (v3.11) and Flask (v2.2.5). A significant challenge in processing WGS files is that their size can exceed the available memory of the host system, making it impractical to read and process the entire file at once. We thus designed an architecture that reads, processes and writes files in batches, thereby minimizing memory requirements. The input file may be a variant file in VCF or MAF format, or a custom table-formatted format (e.g. CSV). The AdaGenes server can perform 4 different actions: It can 1) convert table-formatted data from custom file formats to VCF, 2) perform LiftOver between different reference genomes, 3) annotate variants, and 4) sort and filter variants (Figure 1). The annotation of variants is accelerated by splitting the amount of variants into chunks, which are processed in parallel and then merged again. As an internal variant data representation, AdaGenes is using a JSON-based structure, referring to each variant using a HGVS identifier. This structure can be annotated, processed, filtered or transformed into another format as required, and written out in different formats.

3.2 The AdaGenes database

The database storing the uploaded variant data is a MongoDB database, which enables the storage of dynamic, previously unknown annotation features due to the NoSQL approach [25]. In case of a liftover, AdaGenes adds the positions of other reference genomes as annotations. Each variant is identified by a unique HGVS identifier. Annotations are added by updating the respective entry of the variant

3.3 The AdaGenes web front end

The AdaGenes web front end was built using Vue.js (v3). Layout formatting was implemented using the Vuetify (v3) library. The interactive table for displaying sequence variants and filtering options was implemented using AG Grid. The AdaGenes server and web front end are both available containerized as Docker containers [26]. The display of large variant files, including whole exome and genome sequencing, is made possible by loading a small number of variants from the backend at a time.

3.4 Data preprocessing

To enable the processing of different input data formats, we implemented multiple custom data parsers for AdaGenes, including variant data parsers for VCF, MAF, BED, CSV and XLSX files. To process these different file formats, AdaGenes automatically identifies the file type and employs the associated reader. Each file reader parses the data in the same internal, JSON-based data structure, which allows further processing independent of the data structure of the original file. Similarly, we implemented various writers being able to export variants in different file formats, including VCF, MAF, CSV, and JSON. Liftover between genome assemblies was implemented using chain files of the UCSC FTP server [27] and the Python LiftOver package.

3.5 Annotation

To annotate variant data, Onkopus [28] was integrated for annotating variants by using the Python package. Onkopus queries by default the publicly available modules. By locally installing the Onkopus modules and adjusting the configuration environment variables, AdaGenes can also be used locally. Genomic locations were mapped to transcript and protein level using SeqCAT [24]. HGVS identifiers were generated based on the standardized nomenclature for genetic variants of the Human Genome Variety Society (HGVS) [29].

4 Discussion

We presented AdaGenes, a modular streaming processor for annotating and filtering genetic variant data. In comparison to other annotation pipelines, AdaGenes provides a broader range of annotations and uniquely offers the ability to annotate molecular and protein structure specific features of protein-coding variants. AdaGenes can parse and export multiple file formats and is so far the only tool capable of converting custom file formats at genome or protein-level to the VCF format. While existing tools provide a wealth of functionalities for variant annotation and filtering, their complexity and lack of user-friendliness may be a challenge for users. AdaGenes therefore offers an easy-to-use tool for annotating, filtering and exporting variants without prior knowledge of the annotation features. Another special feature is the ability to recognize the type of previously unknown annotations and filter them according to various combinable options. AdaGenes can be expanded and updated easily due to its modular structure. In addition, users can easily implement their own processing clients and integrate them in the AdaGenes workflow. AdaGenes is the only tool that combines all the functions mentioned (LiftOver, annotation, filtering, sorting, export) in one tool.

The option of defining filters in the graphical user interface provides a more intuitive and easier-to-use filtering of variants than the manual definition of filters via command-line-based tools. In addition, AdaGenes is a versatile tool for visual exploration and analysis of variant data. The recording of all modifications and the option to export them ensures the reproducibility of all results. The AdaGenes server and web front end as well as the annotation modules, including Onkopus and SeqCAT, are locally installable and can thus be utilized in clinical environments to process sensitive patient data, e.g. in molecular tumor boards. We are thus confident that AdaGenes provides an open-source platform that can significantly enhance the accessibility of genomic analysis, empowering more researchers and clinicians to support precision medicine and drive innovative biological insights.

Acknowledgements. We thank the UMG MTB for their support and valuable feedback. We also thank the International Max Planck Research School for Genome Science (IMPRS-GS). We are participating in the Lower Saxony Center for AI and Causal Methods in Medicine (CAIMED). We acknowledge support by the Open Access Publication Funds of the Göttingen University.

Funding. This work was supported by the Gemeinsamer Bundesauschuss (01NVF20006), the Volkswagen Foundation (11-76251-12-1/19), the Deutsche Krebshilfe (70114018), the Deutsche Forschungsgemeinschaft (KFO5002) and the Bundesministerium für Bildung und Forschung (BMBF) (01KD2437, 01KD2401B, 01KD2208A, 01KD2414A).

Conflict of interest. None declared.

Data availability. AdaGenes is freely available under Creative Commons License (CC BY-SA 4.0). No login is required. No data is transferred to third party partners. Uploaded data is stored anonymously. A public instance is available at https://mtb.bioinf.med.uni-goettingen.de/adagenes. AdaGenes and all annotation modules, including Onkopus and SeqCAT, use publicly available data sources.

Code availability. The source code of all AdaGenes modules is available at https://gitlab.gwdg.de/MedBioinf/mtb/adagenes, including the server, database and web front end, and instructions on how to install AdaGenes locally.

Author contributions. JD and NSK conceptualized the architecture and method. The software was implemented by NSK (Database, server, web front end) and KD (Annotation parallelization). KK optimized the annotation process. KRJ contributed test cases and evaluated the methodology. NSK wrote the manuscript. All authors proofread and approved the manuscript.

References

[1] Wang, K., Li, M., Hakonarson, H.: Annovar: functional annotation of genetic variants from high-throughput sequencing data. Nucleic acids research **38**(16), 164–164 (2010)

[2] Cingolani, P., Platts, A., Wang, L.L., Coon, M., Nguyen, T., Wang, L., Land, S.J., Lu, X., Ruden, D.M.: 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 **6**(2), 80–92 (2012)

[3] McLaren, W., Gil, L., Hunt, S.E., Riat, H.S., Ritchie, G.R., Thormann, A., Flicek, P., Cunningham, F.: The ensembl variant effect predictor. Genome biology **17**, 1–14 (2016)

[4] Desvignes, J.-P., Bartoli, M., Delague, V., Krahn, M., Miltgen, M., Béroud, C., Salgado, D.: Varaft: a variant annotation and filtration system for human next generation sequencing data. Nucleic acids research **46**(W1), 545–553 (2018)

[5] Yang, H., Wang, K.: Genomic variant annotation and prioritization with annovar and wannovar. Nature protocols **10**(10), 1556–1566 (2015)

[6] Cingolani, P., Patel, V.M., Coon, M., Nguyen, T., Land, S.J., Ruden, D.M., Lu, X.: Using drosophila melanogaster as a model for genotoxic chemical mutational studies with a new program, snpsift. Frontiers in genetics **3**, 35 (2012)

[7] 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.*: The variant call format and vcftools. Bioinformatics **27**(15), 2156–2158 (2011)

[8] Danecek, P., Bonfield, J.K., Liddle, J., Marshall, J., Ohan, V., Pollard, M.O., Whitwham, A., Keane, T., McCarthy, S.A., Davies, R.M., *et al.*: Twelve years of samtools and bcftools. Gigascience **10**(2), 008 (2021)

[9] Müller, H., Jimenez-Heredia, R., Krolo, A., Hirschmugl, T., Dmytrus, J., Boztug, K., Bock, C.: Vcf. filter: interactive prioritization of disease-linked genetic variants from sequencing data. Nucleic acids research **45**(W1), 567–572 (2017)

[10] Knaus, B.J., Grünwald, N.J.: vcfr: a package to manipulate and visualize variant call format data in r. Molecular ecology resources **17**(1), 44–53 (2017)

[11] Hart, S.N., Duffy, P., Quest, D.J., Hossain, A., Meiners, M.A., Kocher, J.-P.: Vcf-miner: Gui-based application for mining variants and annotations stored in vcf files. Briefings in bioinformatics **17**(2), 346–351 (2016)

[12] Salatino, S., Ramraj, V.: Browsevcf: a web-based application and workflow to quickly prioritize disease-causative variants in vcf files. Briefings in bioinformatics **18**(5), 774–779 (2017)

[13] Maranhao, B., Biswas, P., Duncan, J., Branham, K., Silva, G., Naeem, M., Khan, S., Riazuddin, S., Hejtmancik, J., Heckenlively, J., *et al.*: exomesuite: Whole exome sequence variant filtering tool for rapid identification of putative disease causing snvs/indels. Genomics **103**(2-3), 169–176 (2014)

[14] Tollefson, G.A., Schuster, J., Gelin, F., Agudelo, A., Ragavendran, A., Restrepo, I., Stey, P., Padbury, J., Uzun, A.: Viva (visualization of variants): a vcf file visualization tool. Scientific Reports **9**(1), 12648 (2019)

[15] Eidi, M., Abdolalizadeh, S., Moeini, S., Garshasbi, M., Zahiri, J.: 123vcf: an intuitive and efficient tool for filtering vcf files. BMC bioinformatics **25**(1), 68 (2024)

[16] Holtgrewe, M., Stolpe, O., Nieminen, M., Mundlos, S., Knaus, A., Kornak, U., Seelow, D., Segebrecht, L., Spielmann, M., Fischer-Zirnsak, B., *et al.*: Varfish: comprehensive dna variant analysis for diagnostics and research. Nucleic acids research **48**(W1), 162–169 (2020)

[17] Jiang, J., Gu, J., Zhao, T., Lu, H.: Vcf-server: A web-based visualization tool for high-throughput variant data mining and management. Molecular Genetics & Genomic Medicine **7**(7), 00641 (2019)

[18] Paila, U., Chapman, B.A., Kirchner, R., Quinlan, A.R.: Gemini: integrative exploration of genetic variation and genome annotations. PLoS computational

biology **9**(7), 1003153 (2013)

[19] Schutz, S., Monod-Broca, C., Bourneuf, L., Marijon, P., Montier, T.: Cutevariant: a standalone gui-based desktop application to explore genetic variations from an annotated vcf file. Bioinformatics Advances **2**(1), 028 (2022)

[20] Kamaraj, V., Sinha, H.: Sci-vcf: a cross-platform gui solution to summarize, compare, inspect and visualize the variant call format. NAR Genomics and Bioinformatics **6**(3), 083 (2024)

[21] Freeman, P.J., Hart, R.K., Gretton, L.J., Brookes, A.J., Dalgleish, R.: Variant-validator: Accurate validation, mapping, and formatting of sequence variation descriptions. Human mutation **39**(1), 61–68 (2018)

[22] Lefter, M., Vis, J.K., Vermaat, M., Dunnen, J.T., Taschner, P.E., Laros, J.F.: Mutalyzer 2: next generation hgvs nomenclature checker. Bioinformatics **37**(18), 2811–2817 (2021)

[23] Griffith, M., Spies, N.C., Krysiak, K., McMichael, J.F., Coffman, A.C., Danos, A.M., Ainscough, B.J., Ramirez, C.A., Rieke, D.T., Kujan, L., *et al.*: Civic is a community knowledgebase for expert crowdsourcing the clinical interpretation of variants in cancer. Nature genetics **49**(2), 170–174 (2017)

[24] Kornrumpf, K., Kurz, N.S., Drofenik, K., Krauß, L., Schneider, C., Koch, R., Beißbarth, T., Dönitz, J.: Seqcat: Sequence conversion and analysis toolbox. Nucleic Acids Research **52**(W1), 116–120 (2024)

[25] Győrödi, C., Győrödi, R., Pecherle, G., Olah, A.: A comparative study: Mongodb vs. mysql. In: 2015 13th International Conference on Engineering of Modern Electric Systems (EMES), pp. 1–6 (2015). IEEE

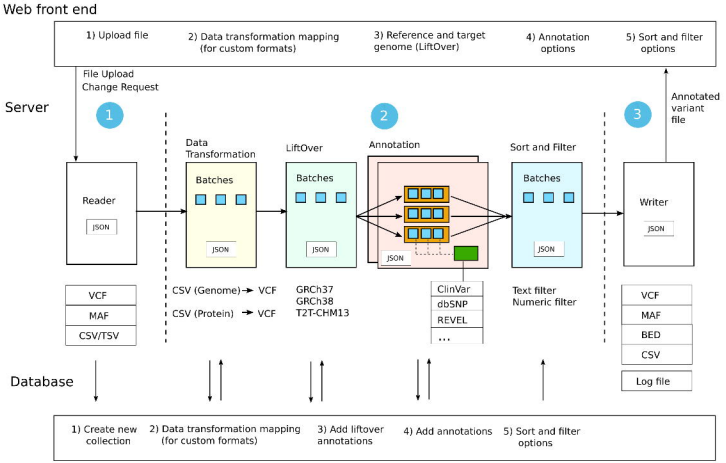
[26] Merkel, D., *et al.*: Docker: lightweight linux containers for consistent development and deployment. Linux j **239**(2), 2 (2014)

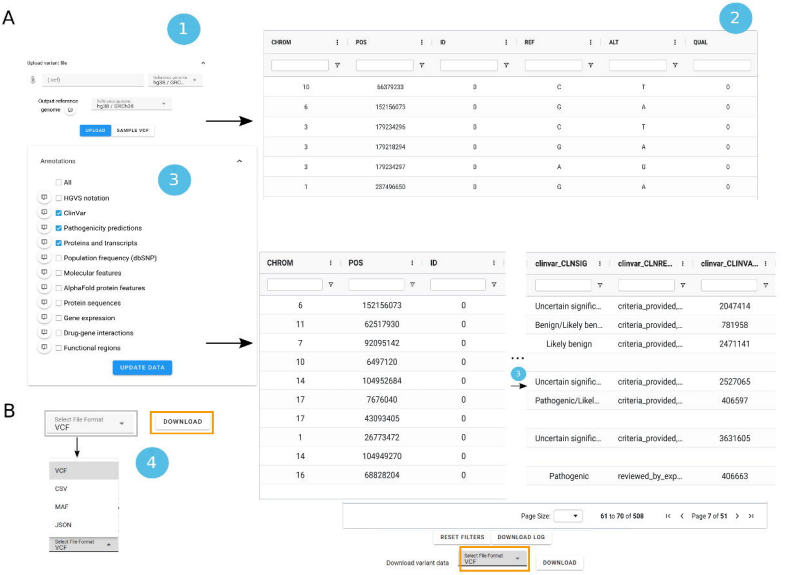
[27] Perez, G., Barber, G.P., Benet-Pages, A., Casper, J., Clawson, H., Diekhans, M., Fischer, C., Gonzalez, J.N., Hinrichs, A.S., Lee, C.M., *et al.*: The ucsc genome browser database: 2025 update. Nucleic Acids Research **53**(D1), 1243–1249 (2025)

[28] Kurz, N.S., Kornrumpf, K., Tucholski, T., Drofenik, K., König, A., Beißbarth, T., Dönitz, J.: Onkopus: precise interpretation and prioritization of sequence variants for biomedical research and precision medicine. Nucleic Acids Research, 376 (2025)

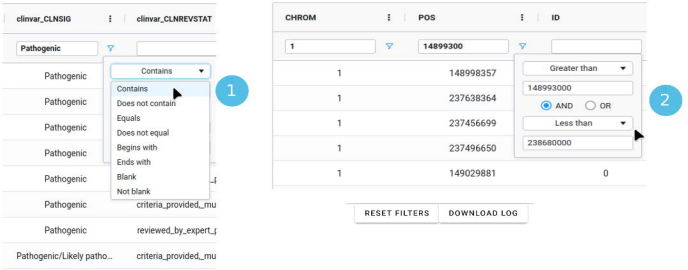
[29] Den Dunnen, J.T., Dalgleish, R., Maglott, D.R., Hart, R.K., Greenblatt, M.S., McGowan-Jordan, J., Roux, A.-F., Smith, T., Antonarakis, S.E., Taschner, P.E., *et al.*: Hgvs recommendations for the description of sequence variants: 2016

update. Human mutation **37**(6), 564–569 (2016)

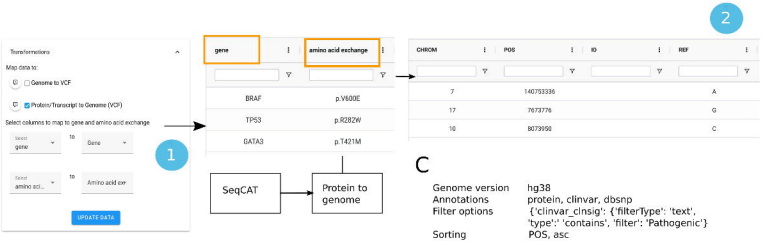

Web front end
1) Upload file
2) Data transformation mapping (for custom formats)
3) Reference and target genome (LiftOver)
4) Annotation options
5) Sort and filter options
File Upload
Change Request
Server
1
2
3
Annotated variant file
Data Transformation
LiftOver
Annotation
Sort and Filter
Reader
JSON
Batches
JSON
Batches
JSON
JSON
Batches
JSON
Writer
JSON
VCF
MAF
CSV/TSV
CSV (Genome) → VCF
CSV (Protein) → VCF
GRCh37
GRCh38
T2T-CHM13
ClinVar
dbSNP
REVEL
...
Text filter
Numeric filter
VCF
MAF
BED
CSV
Log file
Database
1) Create new collection
2) Data transformation mapping (for custom formats)
3) Add liftover annotations
4) Add annotations
5) Sort and filter options


A
1
Upload variant file
(vcf)
hg38 / GRCh38
Output reference genome
hg38 / GRCh38
UPLOAD
SAMPLE VCF
Annotations
All
HGVS notation
ClinVar
Pathogenicity predictions
Proteins and transcripts
Population frequency (dbSNP)
Molecular features
AlphaFold protein features
Protein sequences
Gene expression
Drug-gene interactions
Functional regions
UPDATE DATA
3
2
CHROM POS ID REF ALT QUAL
10 66379233 0 C T 0
6 152156673 0 G A 0
3 179234296 0 C T 0
3 179218294 0 G A 0
3 179234297 0 A G 0
1 237496650 0 G A 0
CHROM POS ID
6 152156073 0
11 62517930 0
7 92095142 0
10 6497120 0
14 104952684 0
17 7676040 0
17 43093405 0
1 26773472 0
14 104949270 0
16 68828204 0
clinvar_CLNSIG clinvar_CLNRE... clinvar_CLINVA...
Uncertain signific... criteria_provided,... 2047414
Benign/Likely ben... criteria_provided,... 781958
Likely benign criteria_provided,... 2471141
Uncertain signific... criteria_provided,... 2527065
Pathogenic/Likel... criteria_provided,... 406597
Uncertain signific... criteria_provided,... 3631605
Pathogenic reviewed_by_exp... 406663
Page Size:
61 to 70 of 508
Page 7 of 51
RESET FILTERS
DOWNLOAD LOG
Download variant data
Select File Format VCF
DOWNLOAD
B
Select File Format VCF
DOWNLOAD
4
VCF
CSV
MAF
JSON
Select File Format VCF

A



B



C

Genome version	hg38
Annotations	protein, clinvar, dbsnp
Filter options	{'clinvar_clnsig': {'filterType': 'text', 'type': 'contains', 'filter': 'Pathogenic'}
Sorting	POS, asc