





A Flexible and Open-Source Tool for Genetic Variant Annotation

Andrea Bombarda¹^a, Matteo Bellini²^b, Maria Iascone²^c, Domenico Fabio Savo¹^d

¹*Department of Management, Information, and Production Engineering, University of Bergamo, Bergamo, Italy*

²*Medical Genetics Lab, ASST Papa Giovanni XXIII, Bergamo, Italy*

{andrea.bombarda, domenicofabio.savo}@unibg.it, {m.bellini, miascone}@asst-pg23.it

Keywords: Medical Genetics, Variant Annotation, Rare Genetic Disease Research

Abstract: Advances in genomic research have significantly enhanced our understanding of the genetic factors influencing human health. A key output of this research are VCF (Variant Call Format) files, which document genetic variations detected through DNA sequencing. These files, however, provide limited information, making it challenging to interpret the biological significance of the variants without additional data. Annotation, the process of enriching VCF files with information from publicly available biomedical datasets, is essential for facilitating variant interpretation in research. In this paper, we present VCFAnnotator, a tool developed to adapt ANNOVAR software used in genetic research, enabling the annotation of entire directories with a single command and facilitating the use of any relevant external database. Additionally, VCFAnnotator offers the ability to scrape the various websites of the biomedical databases in use, ensuring that the researchers remain informed of any updates.

1 INTRODUCTION

Genomic research has transformed our understanding of human biology, providing crucial insights into the genetic factors that influence human health (Myers et al., 2001; Battista et al., 2011). The study of genetic variants is now pivotal in medical research, helping to identify new biological pathways, elucidate disease mechanisms, and inform the development of innovative therapeutic approaches. As the volume of genomic data continues to grow, the demand for efficient tools to process, analyze, and interpret this information has become increasingly critical to advancing research in medical genetics.


A crucial output of genetic analysis process are the VCF (Variant Call Format) files. These files, produced through DNA sequencing and alignment, document genetic variations in comparison to a reference genome. Some variants may be associated with the development of diseases, but not all are pathogenic. It is up to the geneticist, based on current medical research, to evaluate whether specific symptoms can be linked to these genetic variants.


To assist the genetics researchers in their work, the


data contained in the VCF file are enriched with additional information about each variant by using data from publicly available biomedical datasets. This process, called *annotation*, is considered crucial for interpreting genetic data and assessing the potential impact of variants on human health (Salgado et al., 2016).


In fact, apart from detailing the chromosome position and the nucleotides in a specific variant, VCF files provide very little information. As a result, without performing annotation, it becomes challenging to address key questions, such as whether a variant has already been identified in other studies, what its effects might be, or whether it is widespread in the population and therefore unlikely to be pathogenic. In the context of genetic research, where new correlations between variants and diseases are continuously discovered, leveraging up-to-date information is imperative for increasing the chances of achieving prompt and accurate results. For this reason, several tools that help researchers to annotate VCF files have been proposed (Wang et al., 2010; Yang and Wang, 2015; Pedersen et al., 2016; Cingolani et al., 2012; McLaren et al., 2010). However, despite their efficiency, most of them lack in providing information to the user about the up-to-date status of the used data sources and are limited to annotations performed using only a subset of the available biomedical databases.

In this paper, we present VCFAnnotator, a tool de-

^a <https://orcid.org/0000-0003-4244-9319>

^b <https://orcid.org/0009-0001-4297-9160>

^c <https://orcid.org/0000-0002-4707-212X>

^d <https://orcid.org/0000-0002-8391-8049>

veloped in collaboration with the medical genetics department of Papa Giovanni XXIII Hospital in Bergamo, to address the issues they observe in existing annotation tools. Specifically, VCFAnnotator acts as a wrapper for the ANNOVAR (Wang et al., 2010) tool, enabling the automatic verification of the up-to-date status of the data sources used to perform the annotation. Moreover, it allows for working with any database that is not natively compatible with ANNOVAR by automatically converting selected resources into a format accepted by the ANNOVAR annotation tool. With these features, VCFAnnotator integrates seamlessly with ANNOVAR, allowing researchers to perform VCF annotation more quickly and efficiently, both for new cases and for those previously analyzed but considered inconclusive.

The rest of the paper is structured as follows. Sect. 2 provides the background on the variant identification and annotation process, as well as the most used biomedical databases, and presents the ANNOVAR, which is at the base of VCFAnnotator. Then, Sect. 3 and 4, present the requirements and the design of VCFAnnotator and its implementation details and functionalities, respectively. Finally, Sect. 5 discusses related works, and Sect. 6 concludes the paper.

2 BACKGROUND

In this section, we describe the process of analysis and investigation that, from the biological material of a subject, allows for the extraction of his/her genetic variants, and the following annotation process. Moreover, we provide an overview of the software tools currently used to support the annotation phase and highlight their limitations.

2.1 VARIANT IDENTIFICATION

In Next Generation Sequencing (NGS), the journey from biological material to a VCF file is a complex, multistep process that integrates both wet-lab procedures and computational bioinformatics. This process aims to identify genetic variants, such as Single Nucleotide Variants (SNVs), insertions, deletions, Copy Number Variations (CNVs), and other mutations, which are fundamental for genetic analysis and therefore for formulating a diagnosis. In cases involving pediatric subjects, analyses typically involve data from the child, mother, and father, sometimes including other relatives.

Fig. 1 depicts a general overview of the process. In the first step, biological material (e.g., blood, saliva, or tissue) is collected, and DNA is extracted, frag-

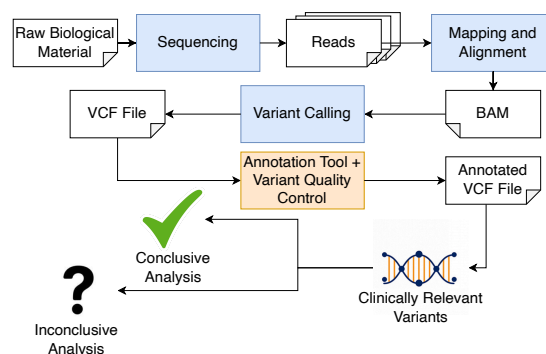


Figure 1: Overview of the Genetics Process

mented, and tagged with adapters for sequencing. The fragments are then processed through a sequencing platform, producing raw data (short or long reads) in FASTQ format with nucleotide sequences and quality scores. These reads are aligned to a reference genome, generating a binary alignment map (BAM) file with alignment details and quality metrics.

Then, *variant calling* is performed, and statistically significant differences (SNVs, insertions, deletions, CNVs, and structural variants) between the sample's DNA and the reference genome are identified. The result is a VCF¹ file (Danecek et al., 2011), listing all detected variants along with auxiliary information. An example of a VCF file is reported in Listing 1. It is a tab-separated file containing chromosome, variant position, ID, reference and altered nucleotides, quality score, filter info, read details (e.g., depth, genotype), field format, and case-specific data for each individual.

To aid researchers, VCF files undergo further processing, including *variant quality control* to remove low-quality variants and false positives, and *annotation* to enrich data with information about identified variants. During annotation, information taken from relevant biomedical databases (see Sect. 2.2) with the potential functional effects, known associations with diseases, or spread in the population for each variant is added. This phase is carried out through the use of annotation tools such as ANNOVAR (see Sect. 2.3). As a result, a VCF file similar to the one we report in Sect. 4 is produced. The annotation of VCF files is essential because it transforms a simple list of genetic variants into a tool rich in clinical and biological information, enabling their filtering and prioritization.

Given the importance of the annotation process in research, several biomedical databases have been released by the scientific community (see Sect. 2.2). Moreover, several tools are available to perform this activity. While most sequencing machine manufac-

¹The complete v4.3 specification of the VCF is available at <https://samtools.github.io/hts-specs/VCFv4.3.pdf>

Listing 1: Excerpt of a VCF file

```

1 #CHROM POS ID REF ALT QUAL FILTER INFO FORMAT W3HH-AFA-I
2 chr1 941119 . A G 568.23 PASS AC=2;AF=1;AN=2;DP=117;FS=0;MQ=250;QD=4.86;SOR=1.981 GT:AD:AF:DP:GQ:FT:F1R2:F2R1:PL
:GP:PP:DN 1/1:0,28:1:28:81:PASS:0,13:0,15:169,84,0:130.82,80.816,0:329,164,0:Inherited [...]
```

turers offer proprietary analysis tools, these often require subscriptions and incur per-use fees. Consequently, open-source and free alternatives, e.g., ANNOVAR (Sect. 2.3), have emerged as viable options.

2.2 GENETICS DATABASES

As discussed in Sect. 2.1, the usefulness and effectiveness of the annotation process depend on the databases used. Each of the available databases focuses on specific information, such as the frequency of a variant in the population or the predicted effect of the variant on the proteins. In the following, some of the most common databases are introduced.

OMIM: The Online Mendelian Inheritance in Man (OMIM) database (Hamosh, 2004) is a comprehensive resource that catalogs human genes and genetic disorders. It focuses on the relationships among genes and the diseases they cause, as well as their inheritance patterns. More specifically, it is used to individuate the inheritance patterns (dominant, recessive, etc.) thanks to the link of each genetic variant to the scientific literature, helping clinicians and researchers better understand the genetic basis of diseases.

GnomAD: The Genome Aggregation Database (GnomAD) (Chen et al., 2024) is a large public dataset cataloging human genetic variants. It is widely used in genomics research to help understand the spectrum of genetic variants across diverse populations and their frequency. Given that most genetics laboratories work on rare diseases, GnomAD is used to exclude all variants with high occurrence rates in the population, which are therefore not pathogenic.

Clinvar: ClinVar (Landrum et al., 2015) is a public, freely accessible database archiving reports on human genetic variants linked to diseases, along with supporting evidence. It enables easy access to information about the relationships between genetic variants and specific health conditions. ClinVar processes submissions including variants identified in patient samples, classifications related to diseases and drug responses, and additional supporting data, and provides a classification based on the predicted biological effect of a mutation (benign, pathogenic, etc.).

Gencode: The Gencode database (Pei et al., 2012) is a collection of annotations of human and mouse genomes. It provides detailed information on gene

structures, including protein-coding genes, noncoding RNA genes, and other functional elements. It is used for linking variants (both the position and the specific nucleotide alteration) to specific genes and regions.

RefSeq: RefSeq (Reference Sequence) (O’Leary et al., 2015) is a database curated by the National Center for Biotechnology Information (NCBI) that provides complete, annotated reference sequences for genomes, transcripts, and proteins. It is used as a standard for genomic research and comparative studies, offering accurate and non-redundant representations of genetic sequences from humans and other species.

HGMD: The Human Gene Mutation Database (HGMD) (Cooper, 1998) is a comprehensive resource collecting data on clinically relevant variants that are associated with genetic disorders. It can be used as a key reference for researchers by providing detailed information on variants that cause or may cause inherited diseases. The key advantage of this database is that it contains entries that come only from scientifically proven sources. Unlike the other previously presented databases, HGMD is not freely available.

SIFT: SIFT (Sorting Intolerant From Tolerant) (Ng, 2003) is a database used to determine whether an amino acid substitution in a protein will affect its function. It analyzes sequence homology and the physical properties of amino acids to assess whether a mutation is likely to be deleterious (damaging) or tolerated (neutral). It is commonly used to evaluate the potential impact of genetic variations.

2.3 ANNOVAR

ANNOVAR (Wang et al., 2010) is a tool that can be used to annotate Single Nucleotide Variants (SNVs) and insertions/deletions, such as examining their functional consequences on genes, reporting functional importance scores, or finding variants in conserved regions. It is available as a CLI software tool and can be used as a standalone application on systems in which standard Perl is supported. ANNOVAR works with text-based input VCF files (e.g., the example reported in Listing 1), whereby each line corresponds to a genetic variant and reports its characteristics. Then, to annotate variants, ANNOVAR needs to download gene annotation databases, such as those described in Sect. 2.2, and to save them to a local disk.

ANNOVAR is not the only tool available for annotating VCF files, and we describe most of the others in Sect. 5. However, being open-source and supported by the community makes ANNOVAR one of the most chosen options by researchers. Nevertheless, it still has some limitations, which we try to address with VCFAnnotator and discuss in the following section.

2.3.1 Limitations

Despite being very powerful, during our experiments, we discovered several limitations of ANNOVAR that our work aims to solve. First, the efficacy of the annotation process is closely linked to the use of up-to-date biomedical databases (see Sect. 2.2). Indeed, because new correlations between gene variants and pathologies are frequently discovered, the databases must be in their latest available versions for the annotation to be effective. However, ANNOVAR lacks a method to automatically check for newer database versions.

Additionally, ANNOVAR requires the databases used for annotating VCFs to be in a specific format, which is not always the one used by those databases' creators. To solve this issue, ANNOVAR provides a set of already adapted databases,² but most of these are not up-to-date and some required by the genetics laboratories are not available (e.g., OMIM; see Sect. 2.2). Thus, to get all needed and up-to-date information, working with "external" (and possibly in a different format) databases is required. Another similar limitation of ANNOVAR is that it can perform the annotation by exploiting only databases with the same format. This means that if one of the biomedical databases used for the annotation is stored in a `txt` file, all the other databases must be provided as `txt` as well. In practice, every biomedical database is provided in many possible formats and this limits the usability of ANNOVAR as it is. Finally, ANNOVAR was designed as a tool that is to annotate only a single VCF file per time, making it unsuitable for the re-analysis process, in which or a set of previously analyzed cases require reanalysis at the same time.

3 SOFTWARE DESIGN

In this section, we begin by introducing the requirements we identified for VCFAnnotator that were specifically tailored for overcoming the limitations identified for ANNOVAR. Then, we present the architecture we developed to allow the highest configurability and flexibility for the tool.

²<https://annovar.openbioinformatics.org/en/latest/user-guide/download/>

3.1 Software requirements

At the beginning of the project, thanks to meetings with genetics researchers we collaborated with, we identified several requirements for VCFAnnotator. More specifically, we identified some *general requirements* and three *modes*.

In terms of *general requirements*, VCFAnnotator should be as much configurable as possible because of the evolving environment in the genetics laboratory and scientific state-of-the-art. This means that the databases used for VCF annotation should be set by the user, allowing for different operations in different formats. Similarly, the paths in which the annotated VCF and biomedical databases are stored should be configurable to make VCFAnnotator usable even when additional storage is added and the file system structure changes. In terms of modes, we identified *scraping*, *DB preparation*, and *annotation* modes:

- When in *scraping* mode, VCFAnnotator should be able to automatically check the websites of the chosen biomedical databases, to discover whether new versions are available. This operation should be set as manual or automatic every time a new annotation is performed, but the update should always be manual to avoid unwanted overwrites or having databases in an inconsistent state, such as incomplete downloads.

- When in *DB preparation* mode, the downloaded databases should be adapted to make them suitable for use with ANNOVAR. This phase is pivotal, as biomedical databases are available in three different formats (`vcf`, `txt`, and `gff3`) and adopt different encoding. More specifically, VCFAnnotator should be able to remove all useless comments from `txt` files, convert some `vcf` file into a `txt` file, and substitute the `` string, used when the variant implies a deletion, with a dot. The user should be able to set for each database, the required preparation operations.

- When in *annotation* mode, VCFAnnotator should annotate the input VCF files using the information taken from selected databases. More specifically, VCFAnnotator should support working at least with the Gencode, Clinvar, GnomAD, OMIM and HGMD databases and, in general, with databases in the `vcf`, `gff3`, and `txt` format. Furthermore, VCFAnnotator should annotate a single file, a portion of a VCF file, or all files in a specified folder to make re-analyses possible on previously inconclusive cases. At the end of annotation, VCFAnnotator should produce a single file with database information stored in separate columns, enabling easier and up-to-date variant analysis.

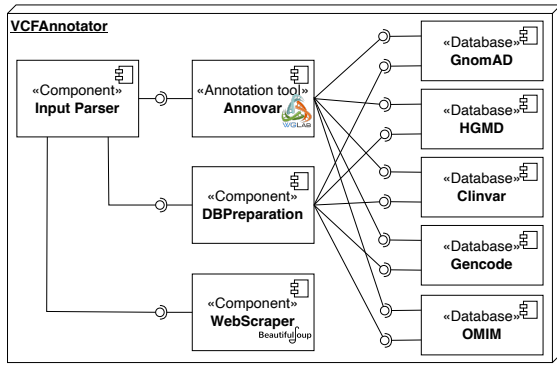


Figure 2: VCFAnnotator software architecture

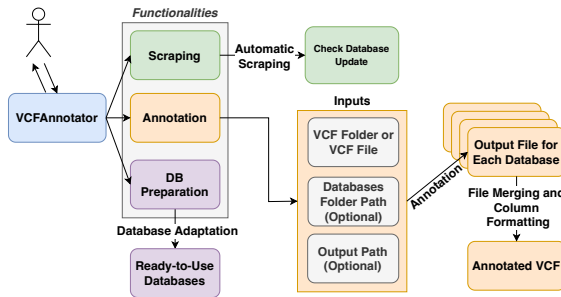


Figure 3: VCFAnnotator usage flow

3.2 Software architecture

Fig. 2 shows the VCFAnnotator software architecture. The tool, implemented in Python and available at <https://github.com/ANTARES-PRJ/VCFAnnotator>, provides an `Input Parser` component, which parses the input parameters and configuration files. Depending on the configuration and the user request, one of the three functionalities is started.

The annotation functionalities are carried out by ANNOVAR (see Sect. 2.3). This takes as input the VCF file (or files) to be annotated and, exploiting the selected biomedical databases, performs the annotation. Note that ANNOVAR is executed as a command-line tool. In this way, we kept the architecture as modular as possible, and substituting it with a different (and possibly more powerful or efficient) annotation tool is possible in the future. In Fig. 2, only five databases (i.e., those identified in the software requirement analysis and previously explained in Sect. 2.2) are reported, but the architecture is flexible, and new databases can be added on the fly.

The web scraping functionalities are carried out by the `WebScraper` component, which takes advantage of the functionalities offered by the `beautifulsoup` Python library (Hajba, 2018). It is devoted to fetching database websites and signaling whether new versions for each of the set databases are available.

Finally, the database preparation is performed by

the `DBPreparation` component. As for the ANNOVAR tool, despite only five databases being connected to the module in Fig. 2, the component can interact with all databases, depending on user configuration.

4 APP PROTOTYPE

In this section, we present the usage of VCFAnnotator, as shown in Fig. 3. VCFAnnotator supports three different types of operations, with each fulfilling one specific task - namely the *automatic scraping* of the database websites, the *preparation* of the databases, and the actual *annotation*.

Configuration The properties of VCFAnnotator and the configurations used in each of the three supported functionalities are set by using the `config.yaml` file reported in Listing 2. More specifically, the file contains two relevant paths: The `db_path`, where the databases used during the annotation phase are stored, and the `destination_path`, where the annotated files are saved (lines 1 and 2). Then, the configuration information for each database is described. As previously discussed, VCFAnnotator supports databases in the `txt`, `vcf`, and `gff3` formats. Thus, according to the format, each database is reported in a different list (`databasesTXT` at line 3, `databasesVCF` at line 10, and `databasesGFF3` at line 17). For each entry, the configuration file contains the `id`, `file name`, and the `operation type` to be used while annotating a VCF file with the ANNOVAR subcomponent.³

Automatic scraping The purpose of the automatic scraping procedure is to discover, for each of the databases used during the annotation phase, whether new versions are available. This is pivotal for the accuracy and effectiveness of the diagnosis provided by a genetics laboratory. The scraping operation can be performed automatically whenever a new annotation is launched (`autoCheck` parameter at line 27 in Listing 2) or launched manually. To manually execute this operation, VCFAnnotator can be called by using the `--checkDB` (or `-c`) option:

```
$ python vcf_annotator.py --checkDB
```

In this way, the tool automatically checks for database updates and prints a table in which updates are indicated with a `[!]`, as illustrated by the screenshot in Fig. 4. Note that at the moment, the update

³The following values can be used as operation: `g` for gene-based, `gx` for gene-based with cross-reference annotation, `r` for region-based, and `f` for filter-based.

Listing 2: Configuration file for VCFAnnotator

```

1 db_path: "humandb/"
2 destination_path: "result/"
3 databasesTXT: # Name of the DBs
4   - id: Clinvar
5     file: "clinvar"
6     operation: "f"
7   - id: OMIM
8     file: "omim"
9     operation: "f"
10 databasesVCF:
11   - id: gnomAD
12     file: "hg38_gnomad.vcf"
13     operation: "f"
14   - id: HGMD
15     file: "hg38_hgmd.vcf"
16     operation: "f"
17 databasesGFF3:
18   - id: Gencode
19     file: "hg38_gencode.gff3"
20     operation: "r"
21 convertFromVCFtoTxt: # convert from vcf to txt
22   - id: Clinvar
23   removeDEL: # substitute <DEL> with the .
24   - id: HGMD
25   clean: # remove the comments from txt files
26   - id: OMIM
27   autoCheck: false # Scraping
28   scraping:
29     - id: Gencode
30       release: "44"
31       website: "https://www.encodegenes.org/human"
32       textToSearch: "Release "
33       tag: "h1"
34     - id: Clinvar
35       date: "2024-04-22"
36       website: "https://ftp.ncbi.nlm.nih.gov/pub/clinvar"
37         /vcf_GRCh38/"
38       textToSearch: "clinvar_"
39       tag: "a"
40     - id: GnomAD
41       release: "4.1"
42       website: "https://gnomad.broadinstitute.org/news/"
43         category/release/"
44       textToSearch: "gnomAD"
45       tag: "h2"
46     - id: OMIM
47       date: "2024-09-20"
48       website: "https://omim.org"
49       textToSearch: "Updated "
50       tag: "h5"

```

of the databases is not automatic, and the user has to download the new files and manually substitute the old versions. More specifically, after having downloaded the new files and substituted them in the chosen `db_path` (see line 1 in Listing 2), the user may update the information in the `config.yaml` file (see Listing 2, starting from line 28) by adding for each database `id` the relevant version information (e.g., release number or release date), the website that has to be scraped to find whether new versions are available, and the text to be searched in a specific HTML tag. The update process was made manual to ensure researchers know the current database version and avoid annotations with inconsistently updated databases, which risk incomplete information. Furthermore, we emphasize that in the screen in Fig. 4, only four of the five databases used by VCFAnnotator are reported. Indeed, looking at the `scraping` section

Database	Release	Date	Update
Gencode	44	/	[!]
Clinvar	/	2024-04-22	[!]
GnomAD	4.1	/	
OMIM	/	2024-09-20	

Figure 4: Output table for the automatic scraping procedure

in Listing 2, the HGMD database is not set as one of those to be scraped because, as explained in Sect. 2.2, it is the only one not freely available.

Database preparation As introduced in Sect. 3, VCFAnnotator allows for annotating VCF files by using different databases. This operation is made possible by its integration with ANNOVAR. However, we have found that the formatting conventions used by ANNOVAR and those used by major databases may differ and, thus, a preparation of each database is needed. To execute this operation, VCFAnnotator can be called by using the `--prepare` (or `-p`) option:

```
$ python vcf_annotator.py --prepare
```

This operation encompasses three different steps and is performed depending on the configuration set in the `config.yaml` file reported in Listing 2 (from line 21 to line 26). The databases in the `convertFromVCFtoTxt` list are translated into the `.txt` format, which is supported by ANNOVAR. Despite ANNOVAR is supposed to work correctly with `vcf` files, we have found that for some of them, it does not. For example, in our experiments, we have seen that the Clinvar database is not supported by ANNOVAR if used as a `vcf` file, while the GnomAD one works fine. Thus, for those databases ANNOVAR does not work with, they are converted into `txt` files. Then, if some ``s are present, indicating a deletion in the databases in the `removeDEL` list, they are converted into a dot. Finally, all lines starting with `#` are removed from the databases in the `clean` list as they may be interpreted as the header by ANNOVAR.

VCF annotation This mode provides the core functionalities of VCFAnnotator- i.e., the annotation of VCF files. To execute this task, VCFAnnotator can be called using the `--annotateVCF` (or `-a`) option:

```
$ python vcf_annotator.py --annotateVCF input.vcf
```

This operation requires the user to specify an input `vcf` file or the path of a folder containing multiple files. In both cases, for each input file, the annotation is performed using all databases reported in the `config.yaml` file and, for each of

Listing 3: Example of an annotated VCF file

```

1  Chr Start End Ref Alt CLNALLEID CLNDN CLNDISDB CLNREVSTAT CLNSIG Id Qual Filter Info Format W3HH-AFA-I Gencode
   MIM Number Gene/Locus And Other Related Symbols Gene Name Approved Gene Symbol Entrez Gene ID Ensembl Gene ID
   Comments Phenotypes Mouse Gene Symbol/ID AC AN AF CLASS MUT GENE STRAND DNA PROT DB PHEN RANKSCORE SVTYPE END
   SVLEN
2  chr1 941119 941119 A G . . . . . 568.23 PASS AC=2;AF=1;AN=2;DP=117;FS=0;MQ=250;QD=4.86;SOR=1.981 GT:AD:AF:DP:GQ:
   FT:F1R2:F2R1:PL:GP:PP:DN 1/1:0,28:1:28:81:PASS:0,13:0,15:169,84,0:130.82,80.816,0:329,164,0:Inherited Name=
   ENSG00000187634.13,ENST00000618323.5,ENST00000474461.1,ENST00000478729.1,ENST00000618181.5,ENST00000618779.5,
   ENST00000622503.5,ENST00000342066.8,ENST00000616125.5,ENST00000341065.8,ENST00000616016.5,ENST00000455979.1,
   exon:ENST00000474461.1:1,ENST00000617307.5 . . . . . [...]

```

them, with the specific operation. We emphasize that we designed VCFAnnotator to work even with noncomplete vcf files, and with files composed by merging rows appertaining to multiple subjects. In such a way, researchers can also annotate a reduced set of variants, e.g., those under investigation for finding a specific pathology. To maintain the traceability of all operations, VCFAnnotator calls ANNOVAR iteratively on all databases, so that, at the end of the annotation process, multiple annotated vcf files are available with the name `DBName_VCFInputName_YYYY-mm-dd_HH_MM_SS`.

In addition, a merged file with the name `VCFInputName_YYYY-mm-dd_HH_MM_SS` is produced. It contains all annotations from all databases, each one in one or more columns with the same name (or starting with a prefix) as the used database.

During annotation, the databases are taken from the `db_path` folder (line 1 in Listing 2), and the results are stored in the `destination_path` folder (line 2 in Listing 2). However, different paths can be specified when executing VCFAnnotator from the command line. More specifically, the database path can be specified in the command after the option `--DBPath` (or `-db`), while destination path after the option `--DestinationPath` (or `-d`). An example of an annotated VCF obtained using VCFAnnotator and with the VCF previously shown in Listing 1 is reported in Listing 3. It can be seen that the annotation process adds several columns (e.g., all those starting with `CLN` for the Clinvar database and the one marked as `Gencode` for the gencode database).

5 RELATED WORK

Several attempts to provide efficient and effective tools are available in the literature. One of the most used is ANNOVAR (Wang et al., 2010), which is the tool underlying the annotation functionalities of VCFAnnotator. Over the years, it has been continuously updated and extended, including the addition of a web-based annotation environment, wANNOVAR (Yang and Wang, 2015). Similarly, in (Ped-

ersen et al., 2016), the Vcfanno tool was proposed. While powerful, its complexity makes it less suitable for non-expert users. Future work could explore automating its configuration as our architecture (Sect. 3) supports replacing the annotation component without affecting others. The SnpEff and VEP tools were proposed in (Cingolani et al., 2012) and (McLaren et al., 2010), respectively. They annotate variants by genomic location and predict coding effects but lack additional needed information, making SnpEff or VEP unsuitable replacements for ANNOVAR. Similarly, BCFTools (Danecek et al., 2021) can perform VCF annotation, but it works only by using a single reference database, and only in the GFF3 format. Instead, VCFAnnotator allows users to use more databases with multiple input formats. Finally, most companies producing sequencers provide their own annotation tools, such as for Illumina.⁴ However, in most cases, the tools are not open-source and are expensive and inflexible. Other tools, such as VCF-Miner (Hart et al., 2015), was proposed in the literature, but they allow only for mining information into already annotated VCFs.

In addition to what mentioned in Sect. 2.2, other tools can enhance the annotated VCF and help prioritize variants. GERP (Genomic Evolutionary Rate Profiling) (Davydov et al., 2010) detects conserved regions by analyzing evolutionary constraints across species. CADD (Combined Annotation-Dependent Depletion) (Kircher et al., 2014) predicts variant impact by combining annotations into a score, identifying those likely to affect gene function. PolyPhen-2 (Polymorphism Phenotyping v2) (Adzhubei et al., 2013) assesses the impact of amino acid substitutions on protein structure and function, aiding in evaluating genetic variant pathogenicity.

6 CONCLUSIONS

Genetic variant annotation plays a crucial role in genetics-related diseases research. This process en-

⁴<https://emea.illumina.com>

ables genetics laboratory researchers to filter variants and assess their potential impact on human beings. However, the analysis is not always definitive due to the evolving nature of scientific knowledge, making the use of up-to-date databases critically important.

Various tools have been introduced in the literature to annotate VCF files, with ANNOVAR being one of the most widely used. Despite its popularity, ANNOVAR has certain limitations regarding the range of databases it supports. In this paper, we propose VCFAnnotator to address these limitations. It facilitates the use of external databases, automatically preparing them for compatibility with ANNOVAR, and includes a scraping feature to detect updated databases.

This study marks an initial step towards developing an automated system for streamlining various tasks in genetic research processes. As future work, we aim to enable researchers to selectively re-annotate specific entries of the VCF file based on pre-defined criteria and to facilitate comparisons between these new annotations and previous ones. Additionally, we aim to explore steps required for VCFAnnotator's certification for diagnostic use (Bombarda et al., 2022; Bombarda et al., 2021).

ACKNOWLEDGEMENTS

This work was funded by PNRR - ANTHEM (Advanced Technologies for Human-centred Medicine) - Grant PNC0000003 – CUP: B53C22006700001 - Spoke 1 - Pilot 1.4. We would like to thank Fabio As-solari and Simone Ronzoni for the preliminary work that they did for this project during their B.Sc. theses.

REFERENCES

- Adzhubei, I. et al. (2013). Predicting functional effect of human missense mutations using polyphen-2. *Current Protocols in Human Genetics*, 76(1).
- Battista, R., Blancquaert, I., et al. (2011). Genetics in health care: an overview of current and emerging models. *Public health genomics*, 15(1):34–45.
- Bombarda, A., Bonfanti, S., et al. (2021). Lessons learned from the development of a mechanical ventilator for covid-19. In *2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE)*, page 24–35. IEEE.
- Bombarda, A., Bonfanti, S., et al. (2022). Guidelines for the development of a critical software under emergency. *Information and Software Technology*, 152:107061.
- Chen, S., Francioli, L. C., Goodrich, J. K., et al. (2024). A genomic mutational constraint map using variation in 76,156 human genomes. *Nature*, 625(7993):92–100.
- Cingolani, P., Platts, A., et al. (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*, 6(2):80–92.
- Cooper, D. (1998). The human gene mutation database. *Nucleic Acids Research*, 26(1):285–287.
- Danecek, P., Auton, A., et al. (2011). The variant call format and vcftools. *Bioinformatics*, 27(15):2156–2158.
- Danecek, P., Bonfield, J. K., et al. (2021). Twelve years of samtools and bcftools. *GigaScience*, 10(2).
- Davydov, E. V., Goode, D. L., Sirota, M., et al. (2010). Identifying a high fraction of the human genome to be under selective constraint using gerp++. *PLoS Computational Biology*, 6(12):e1001025.
- Hajba, G. L. (2018). *Using Beautiful Soup*, page 41–96. Apress.
- Hamosh, A. (2004). Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research*, 33(Database issue):D514–D517.
- Hart, S. N., Duffy, P., et al. (2015). Vcf-miner: Gui-based application for mining variants and annotations stored in vcf files. *Briefings in Bioinformatics*, 17(2):346–351.
- Kircher, M. et al. (2014). A general framework for estimating the relative pathogenicity of human genetic variants. *Nature Genetics*, 46(3):310–315.
- Landrum, M. J. et al. (2015). Clinvar: public archive of interpretations of clinically relevant variants. *Nucleic Acids Research*, 44(D1):D862–D868.
- McLaren, W. et al. (2010). Deriving the consequences of genomic variants with the ensembl API and SNP effect predictor. *Bioinformatics*, 26(16):2069–2070.
- Myers, C., Paulk, N., and Dudlak, C. (2001). Genomics: implications for health systems. *Frontiers of Health Services Management*, 17(3):3–16.
- Ng, P. C. (2003). Sift: predicting amino acid changes that affect protein function. *Nucleic Acids Research*, 31(13):3812–3814.
- O’Leary, N. A., Wright, M. W., et al. (2015). Reference sequence (refseq) database at ncbi: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research*, 44(D1):D733–D745.
- Pedersen, B. S., Layer, R. M., and Quinlan, A. R. (2016). Vcfanno: fast, flexible annotation of genetic variants. *Genome Biology*, 17(1).
- Pei, B., Sisu, C., Frankish, A., et al. (2012). The gencode pseudogene resource. *Genome Biology*, 13(9):R51.
- Salgado, D., Bellgard, M. I., et al. (2016). How to identify pathogenic mutations among all those variations: variant annotation and filtration in the genome sequencing era. *Human mutation*, 37(12):1272–1282.
- Wang, K., Li, M., et al. (2010). ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Res.*, 38(16):e164.
- Yang, H. and Wang, K. (2015). Genomic variant annotation and prioritization with annovar and wannovar. *Nature Protocols*, 10(10):1556–1566.