



# re-Searcher: GUI-based bioinformatics tool for simplified genomics data mining of VCF files

Daniyar Karabayev<sup>1,\*</sup>, Askhat Molkenov<sup>1</sup>, Kaiyrgali Yerulanuly<sup>1,2</sup>, Ilyas Kabimoldayev<sup>1</sup>, Asset Daniyarov<sup>1</sup>, Aigul Sharip<sup>1</sup>, Ainur Ashenova<sup>1</sup>, Zhaxybay Zhumadilov<sup>1,3</sup> and Ulykbek Kairov<sup>1,\*</sup>

<sup>1</sup>Laboratory of Bioinformatics and Systems Biology, Center for Life Sciences, National Laboratory Astana, Nazarbayev University, Nur-Sultan, Kazakhstan

<sup>2</sup>L.N. Gumilyov Eurasian National University, Nur-Sultan, Kazakhstan

<sup>3</sup>School of Medicine, Nazarbayev University, Nur-Sultan, Kazakhstan

\*These authors contributed equally to this work.

## ABSTRACT

**Background.** High-throughput sequencing platforms generate a massive amount of high-dimensional genomic datasets that are available for analysis. Modern and user-friendly bioinformatics tools for analysis and interpretation of genomics data becomes essential during the analysis of sequencing data. Different standard data types and file formats have been developed to store and analyze sequence and genomics data. Variant Call Format (VCF) is the most widespread genomics file type and standard format containing genomic information and variants of sequenced samples.

**Results.** Existing tools for processing VCF files don't usually have an intuitive graphical interface, but instead have just a command-line interface that may be challenging to use for the broader biomedical community interested in genomics data analysis. re-Searcher solves this problem by pre-processing VCF files by chunks to not load RAM of computer. The tool can be used as standalone user-friendly multiplatform GUI application as well as web application (<https://nla-lbsb.nu.edu.kz>). The software including source code as well as tested VCF files and additional information are publicly available on the GitHub repository (<https://github.com/LabBandSB/re-Searcher>).

**Subjects** Bioinformatics, Genetics, Genomics, Molecular Biology, Medical Genetics

**Keywords** Genome analysis, Genomics, Bioinformatics, Genomics data mining, NGS data analysis, Variant call format, VCF

## INTRODUCTION

Recent achievements in high-throughput sequencing technologies (*Goodwin, McPherson & McCombie, 2016; van Dijk et al., 2018*) have led to the generation of massive amounts of genomic data (*Gao et al., 2019; The ICGC/TCGA Pan-Cancer Analysis of Whole Genomes Consortium, 2020*) available for the research community. Many omics databases have been developed and have collected freely accessible datasets (*Molkenov et al., 2019; Rigden & Fernández, 2020*) for analysis by the bioinformatics community. Modern bioinformatics tools and methods are in high demand for analyzing and interpreting the big omics data generated by the different types of multi-omics platforms available. Different

Submitted 17 September 2020

Accepted 1 April 2021

Published 3 May 2021

Corresponding author

Ulykbek Kairov,  
ulykbek.kairov@nu.edu.kz

Academic editor

Hossein Khiabani

Additional Information and  
Declarations can be found on  
page 8

DOI 10.7717/peerj.11333

© Copyright  
2021 Karabayev et al.

Distributed under  
Creative Commons CC-BY 4.0

OPEN ACCESS

standard data types and file formats have been developed to store and analyze sequence and genomics data. Variant Call Format (VCF) ([Danecek et al., 2011](#)) is a tab-delimited text file format that is often used in bioinformatics to store genomic variants. A VCF file consists of the header, including meta-information lines and field definition lines (column names), and the body (data section). An arbitrary number of meta-information lines start with '##' and provide a description of the VCF file. The body of the file consists of eight mandatory columns: chromosome (CHROM), starting position of a variant (POS), variant identifiers (ID), the reference allele (REF), a list of alternate alleles (ALT), a PHRED-scaled quality score (QUAL), filter information regarding variant validity (FILTER), and annotation information (INFO). Additional columns describing samples can also be added. Each row of the file describes specific genomic variants (SNVs, INDELS, CNVs, and other structural variants) at the given chromosome and genomic position.

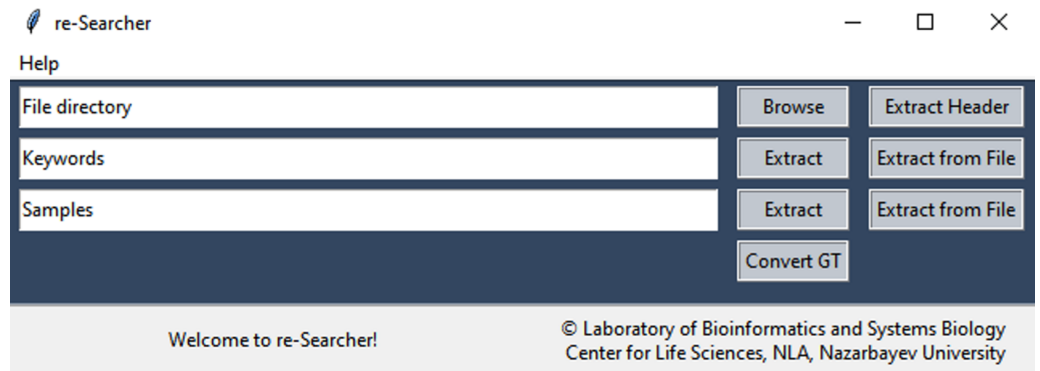
VCF files often store information about numerous samples and can therefore reach huge sizes—gigabytes, or sometimes terabytes. This creates an issue for the readability of VCF files and further analysis for non-programmers, as manual data extraction and analysis using Microsoft Excel or other table processing software may not be possible due to the RAM capacity limitation of standard computers. We introduce re-Searcher, bioinformatics tool specifically developed for simplified mining and analysis of big-size VCF files. We developed a multi-platform user-friendly graphical user interface (GUI) tool for offline access, while re-Searcher web application can be used online via web browser. re-Searcher has been developed for the broader biomedical community and solves the problem of working and analyzing genomics data stored in VCF format.

## MATERIALS & METHODS

### Implementation

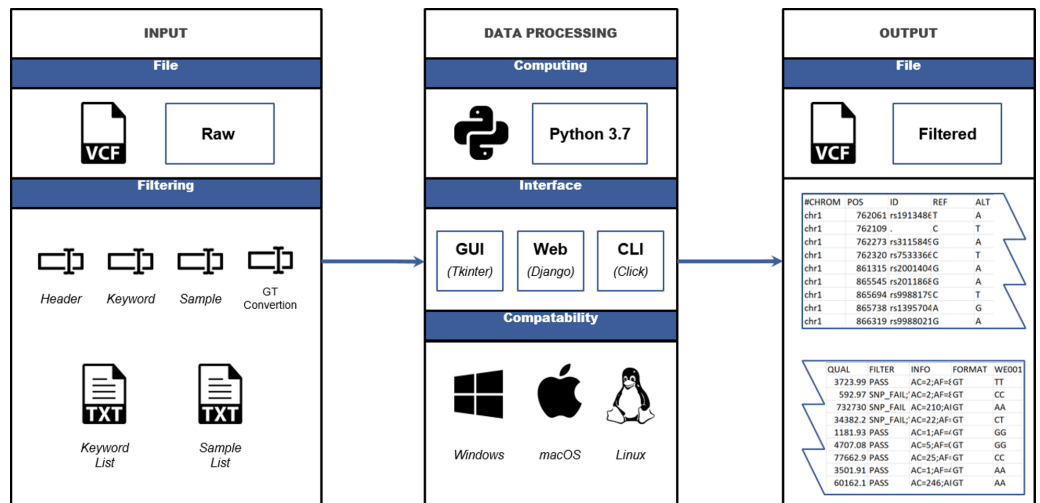
The re-Searcher application was written in Python 3 ([Van Rossum & Drake, 2010](#)) with the implementation of the Tkinter ([Python Software Foundation, 2020](#)) package to build the GUI, and the Pandas ([McKinney, 2010](#)) library to extract columns. For convenience of users who would like to build re-Searcher into their pipelines command line interface (CLI) is available as python script (see Availability). CLI mirrors functionality of GUI regarding files processing. We developed web version of re-Searcher for users to manipulate VCF files without downloading CLI or GUI versions of re-Searcher. The web application was developed using Django web framework ([Django Software Foundation, 2013](#)), to run python script via WSGI on Apache web server ([Fielding & Kaiser, 1997](#)). The web version runs re-Searcher scripts on server and takes input files from website and returns processed files to user to download.

re-Searcher solves the problem of analyzing large VCF files by not loading the whole file directly into RAM, but instead pre-processing it in chunks and utilizing a simple and intuitive interface ([Fig. 1](#)). The main advantage of re-Searcher in comparison with other tools is the presence of a simple and user-friendly interface, GUI and web interface, instead of a CLI, as well as a lack of confused installation procedures typical for existing tools. The generalized workflow of re-Searcher consists of several steps: selecting an input file, setting



**Figure 1** Main window of re-Searcher GUI.

Full-size DOI: 10.7717/peerj.11333/fig-1



**Figure 2** Data processing workflow.

Full-size DOI: 10.7717/peerj.11333/fig-2

up necessary filtering parameters, data processing, and exporting a filtered output VCF file (Fig. 2). re-Searcher browses and opens VCF files with extensions “.txt” or “.vcf”, before performing the following filtering and extraction options:

### Header extraction

VCF files can be large and, if a user needs to know only certain information in a file header (e.g., a particular meta-line or sample ID), the software can extract only the header from the original VCF file and save it into a new file.

### Keyword search

If genomic variants need to be filtered according to the presence of a keyword, the software can find these rows and extract them into a new VCF file. Users may input multiple keywords by accessing the entry field or by uploading a .txt file with keywords.

**A**

#	A	B	C	D	E	F	G	H	I	J	K
1	#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	HG00157	HG00150
2	chr10	60494	rs568182	A	G	100	PASS	AC=27;AF=GT		0 0	0 0
3	chr10	390318	rs4881153	A	G	100	PASS	AC=2549;AF=GT		1 1	1 1
4	chr10	614952	rs3680122	G	A	100	PASS	AC=1;AF=GT		numeric GT	
5	chr10	866075	rs2020461	A	A	100	PASS	AC=54;AF=GT		0/0	0/0
6	chr1	151015299	rs1206706	G	A,GA,GAA	100	PASS	AC=2;AF=GT		0 2	2 2
7	chr1	151386711	rs5307160	T	G	100	PASS	AC=1;AF=GT		0/0	0/0
8	chr1	151776687	rs5572095	G	A	100	PASS	AC=1;AF=GT		0/0	0/0
9	chr1	152164508	rs1121642	G	A	100	PASS	AC=16;AF=GT		0/0	0/0
10	chr1	152466592	rs6668271	G	A	100	PASS	AC=212;AF=GT		0/0	0/0
11	chr1	152799940	rs2001293	A	G	100	PASS	AC=303;AF=GT		0/0	0/0
12	chr1	153095488	rs1824412	C	T	100	PASS	AC=1;AF=GT		0/0	0/0

**B**

#	A	B	C	D	E	F	G	H	I	J	K
1	#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	HG00157	HG00150
2	chr10	60494	rs568182	A	G	100	PASS	AC=27;AF=GT		A/A	A/A
3	chr10	390318	rs4881153	A	G	100	PASS	AC=2549;AF=GT		G/G	G/G
4	chr10	614952	rs3680122	G	A	100	PASS	AC=1;AF=GT		letter GT	
5	chr10	866075	rs2020461	A	A	100	PASS	AC=54;AF=GT		A/A	A/A
6	chr1	151015299	rs1206706	G	A,GA,GAA	100	PASS	AC=2;AF=GT		G/GA	GA/GA
7	chr1	151386711	rs5307160	T	G	100	PASS	AC=1;AF=GT		T/T	T/T
8	chr1	151776687	rs5572095	G	A	100	PASS	AC=1;AF=GT		G/G	G/G
9	chr1	152164508	rs1121642	G	A	100	PASS	AC=16;AF=GT		G/G	G/G
10	chr1	152466592	rs6668271	G	A	100	PASS	AC=212;AF=GT		G/G	G/G
11	chr1	152799940	rs2001293	A	G	100	PASS	AC=303;AF=GT		A/A	A/A
12	chr1	153095488	rs1824412	C	T	100	PASS	AC=1;AF=GT		C/C	C/C

**Figure 3** Genotype conversion example. (A) The numeric genotype of biallelic and multiallelic variants before conversion, and (B) letter genotype of the same variants after conversion.

Full-size [DOI: 10.7717/peerj.11333/fig-3](https://doi.org/10.7717/peerj.11333/fig-3)

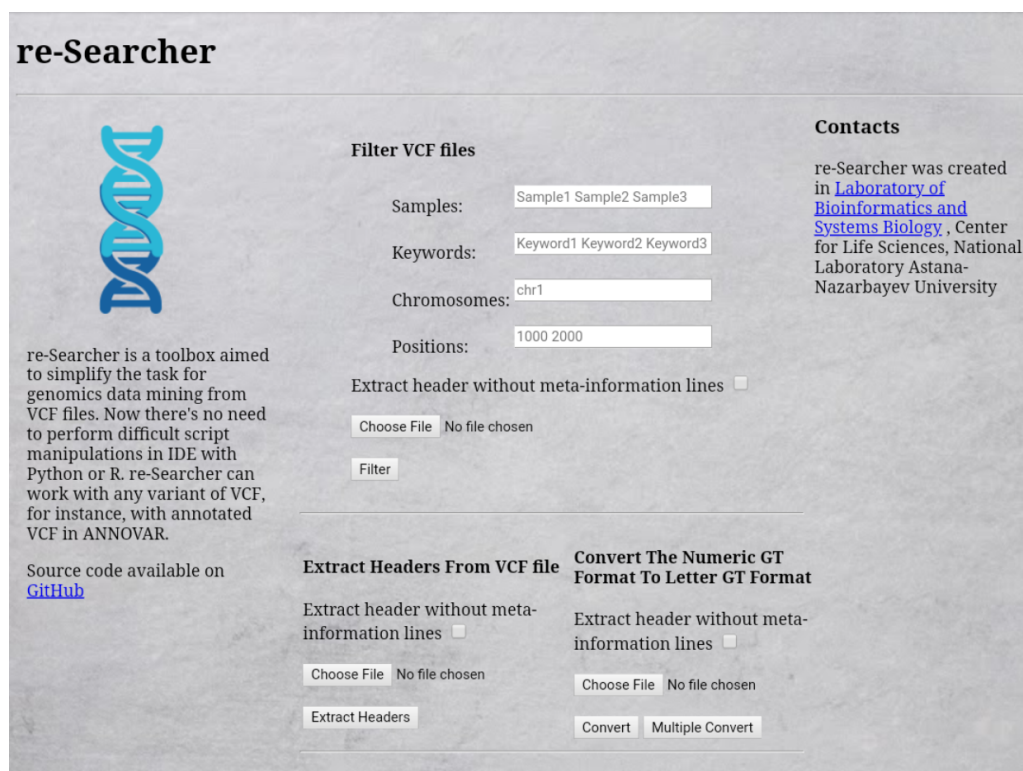
## Sample extraction

If the user needs only particular sample IDs in a VCF file, the software can extract the necessary sample columns into a new file. Similar to a keyword search, users may input multiple sample IDs by accessing the entry field or by uploading a .txt file with the IDs.

## Genotype format conversion

re-Searcher can convert numeric genotype (GT) format into letter format. The conversion option is one of the most used operations when working with VCF files, for example, in further comparative analysis of genetic variants or SNPs. The original GT format in VCF files is numeric (0/0, 0/1, 1/1 for biallelic sites or 1/2, 2/3, etc. for multiallelic sites), where 0 is a reference (REF) allele, 1 is a first alternative (ALT) allele, 2 is a second ALT allele and so on (Danecek et al., 2011; (Campbell et al., 2016)). After GT format conversion REF number is replaced with REF letter and ALT number with corresponding ALT letter (Fig. 3). For instance, if GT of first sample is 0/1 and GT of second sample is 1/2 in numeric format, while REF and two ALT are GC, T and CAA respectively, then after conversion first sample's letter GT becomes GC, T and second sample's letter GT becomes T, CAA.

The final output file is a filtered and processed VCF file and is generated with a complement log file containing the file processing information, name of specified file and work directory with outputs.



**Figure 4** Web interface of re-Searcher available via browser at <https://nla-lbsb.nu.edu.kz>.

Full-size  DOI: 10.7717/peerj.11333/fig-4

## RESULTS AND DISCUSSION

We have compared the main features of re-Searcher with other existing and open source tools VIVA (Tollefson *et al.*, 2019), VCFtools (Danecek *et al.*, 2011), GEMINI (Paila *et al.*, 2013), BrowseVCF (Salatino & Ramraj, 2016), VCF.Filter (Müller *et al.*, 2017), VCF-Miner (Hart *et al.*, 2016) and prepared a detailed table (Table 1). The compared tools have been developed and implemented in different programming languages (Julia, C++, Perl, Java and Python) and dedicated libraries. VCFtools is one of the most cited and advanced tools for processing VCF files, but it requires additional computational skills for effective usage. VIVA is the only tool that provides the possibility for advanced visualization and plotting figures. In addition to re-Searcher, other tools with a GUI available for users are BrowseVCF, VCF.Filter, and VCF-Miner, while web interface is available in only in re-Searcher (Fig. 4) and GEMINI. From these, only re-Searcher, BrowseVCF and VCF.Filter tools support multiple operational systems (Windows, MacOS, Linux). Searching the whole VCF file by keyword and the corresponding extraction of data based on keywords are features available on re-Searcher and BrowseVCF, whereas genotype conversion is a unique feature of re-Searcher.

re-Searcher is a multi-platform tool and can be run on MacOS, Windows and Linux operating systems. Performance of re-Searcher has been evaluated on these PC platforms (Windows 10 Pro OS and Linux Mint 17 OS-based PC: CPU Intel Core i5-8250U 1.80

**Table 1** Comparison of re-Searchers features with similar tools.

Categories	Features	re-Searcher	VIVA	VCFtools	GEMINI	BrowseVCF	VCF.Filter	VCF-Miner
Technical Aspects	Compatibility with operation system	Windows, MacOS, Linux	Windows, MacOS, Linux	Windows, MacOS, Linux	Windows, MacOS, Linux	Windows, MacOS, Linux	Windows, MacOS, Linux	Windows
	Language	Python	Julia	C++, Perl	Python	Python, JavaScript, CSS, HTML5	Java	Java
	Interface	GUI, Web Browser, CLI	CLI, Jupyter Notebook	CLI	Web Browser, CLI	GUI, CLI	GUI	GUI
	Works offline	V	V	V	X	X	V	V
	Portable launcher	V	X	X	X	V	X	X
	Search by keyword	V	X	X	X	V	X	X
	Sample selection	V	V	V	V	V	V	V
Functionality	Genotype format conversion	V	X	X	X	X	X	X
	Visualization	X	V	X	X	X	X	X
	Export filtered VCF file	V	X	V	X	X	V	V

**Table 2** re-Searcher multi-platform run time comparison.

OS	VCF file size (Gb)	Execution Time (sec)			
		Extract header	Keyword extraction	Sample ID extraction	GT conversion
Linux	0.081	2.227	9.995	15.243	38.375
	0.814	2.497	38.106	21.334	117.186
	1.320	2.462	67.260	61.159	206.868
	1.980	2.115	75.331	104.167	330.527
	7.950	6.145	366.137	200.347	1117.168
Windows	0.081	14.865	22.482	4.785	49.642
	0.814	18.898	48.820	21.398	139.641
	1.320	21.054	44.669	59.329	238.192
	1.980	10.919	53.958	90.446	339.275
	7.950	16.308	502.996	309.177	1320.197
MacOS	0.081	9.627	9.297	10.423	16.298
	0.814	5.262	20.916	19.705	116.82
	1.320	6.286	35.433	53.247	186.181
	1.980	3.231	37.923	119.136	286.544
	7.950	5.457	148.612	254.128	1130.225

**Notes.**

OS, operational system; GT, genotype; sec, seconds; Gb, gigabyte.

GHz, RAM 4Gb and MacOS Catalina based PC: CPU Intel Core i7, 3.2 GHz, RAM 8 Gb) with different VCF file sizes. Different size VCF files (0.081 Gb, 0.814 Gb, 1.320 Gb, 1.980 Gb and 7.950 Gb) were used as input datasets for evaluating re-Searcher performance. The results of the performance benchmarking are shown in [Table 2](#).

We have used big VCF files from the 1000 Genomes Project (*The 1000 Genomes Project Consortium, 2015*) and then generated the different sized testing VCF files from this dataset. The Linux-based systems had the fastest execution time for different operations in comparison with Windows and MacOS systems.

**Availability**

re-Searcher executable software including source code, tested VCF files and additional information are publicly available on the GitHub repository <https://github.com/LabBandSB/re-Searcher>. re-Searcher is free bioinformatics tool and open to all users without login and registration requirements and do not require an installation of additional tools. CLI version of re-Searcher is also available on the GitHub repository for incorporation into other pipelines. In addition, web version of re-Searcher is available at <https://mla-lbsb.nu.edu.kz>.

**CONCLUSIONS**

Exploring and analyzing VCF files generated after the bioinformatics processing of sequencing data is one of the important steps performed by researchers during analysis and meta-analysis of genotype/phenotype associations. We have developed and introduced an easy-to-use bioinformatics tool, re-Searcher, with several unique features for mining

big VCF files and realized with a simple graphical user interface and web interface that makes it easily available for clinicians and researchers without any computational skills. Several improvements such as visualization options (clustering and plotting functions) with Principal Component Analysis and heatmap methodologies are under future development of re-Searcher.

## ACKNOWLEDGEMENTS

This work is dedicated to the blessed memory of Dr. Vasily Ogryzko.

## ADDITIONAL INFORMATION AND DECLARATIONS

### Funding

This work has been supported by grant projects (AP08855353, AP05135430 and AP05136106) and program-targeted funding (PTF No. BR05236508) from the Committee of Science, Ministry of Education and Science of the Republic of Kazakhstan. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

### Grant Disclosures

The following grant information was disclosed by the authors:

Committee of Science, Ministry of Education and Science of the Republic of Kazakhstan: AP08855353, AP05135430, AP05136106, PTF No. BR05236508.

### Competing Interests

The authors declare there are no competing interests.

### Author Contributions

- Daniyar Karabayev conceived and designed the experiments, performed the experiments, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Askhat Molkenov conceived and designed the experiments, analyzed the data, prepared figures and/or tables, and approved the final draft.
- Kaiyrgali Yerulanuly, Ilyas Kabimoldayev, Asset Daniyarov, Aigul Sharip and Ainur Ashenova performed the experiments, prepared figures and/or tables, and approved the final draft.
- Zhaxybay Zhumadilov conceived and designed the experiments, authored or reviewed drafts of the paper, supervised the study and were involved in funding acquisition, and approved the final draft.
- Ulykbek Kairov conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the paper, supervised the study and were involved in funding acquisition, and approved the final draft.



## Data Availability

The following information was supplied regarding data availability:

The software including source code and tested VCF files and additional information are available at GitHub: <https://github.com/LabBandSB/re-Searcher>.

## REFERENCES

- Campbell IM, Gambin T, Jhangiani SN, Grove ML, Veeraraghavan N, Muzny DM, Shaw CA, Gibbs RA, Boerwinkle E, Yu F, Lupski JR. 2016. Multiallelic positions in the human genome: challenges for genetic analyses. *Human Mutation* 37:231–234 DOI 10.1002/humu.22944.
- Django Software Foundation. 2013. Django. Available at <https://djangoproject.com> (accessed on 20 April 2021).
- Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, Handsaker RE, Lunter G, Marth GT, Sherry ST, McVean G, Durbin R, 1000 Genomes Project Analysis Group. 2011. The variant call format and VCFtools. *Bioinformatics* 27:2156–2158 DOI 10.1093/bioinformatics/btr330.
- Fielding RT, Kaiser G. 1997. The Apache HTTP Server Project. *IEEE Internet Computing* 1:88–90 DOI 10.1109/4236.612229.
- Gao GF, Parker JS, Reynolds SM, Silva TC, Wang L-B, Zhou W, Akbani R, Bailey M, Balu S, Berman BP, Brooks D, Chen H, Cherniack AD, Demchok JA, Ding L, Felau I, Gaheen S, Gerhard DS, Heiman DI, Hernandez KM, Hoadley KA, Jayasinghe R, Kemal A, Knijnenburg TA, Laird PW, Mensah MKA, Mungall AJ, Robertson AG, Shen H, Tarnuzzer R, Wang Z, Wyczalkowski M, Yang L, Zenklusen JC, Zhang Z, Liang H, Noble MS. 2019. Before and after: comparison of legacy and harmonized TCGA genomic data commons' data. *Cell Systems* 9:24–34 DOI 10.1016/j.cels.2019.06.006.
- Goodwin S, McPherson JD, McCombie WR. 2016. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics* 17:333–351 DOI 10.1038/nrg.2016.49.
- Hart SN, Duffy P, Quest DJ, Hossain A, Meiners MA, Kocher J-P. 2016. VCF-Miner: GUI-based application for mining variants and annotations stored in VCF files. *Briefings in Bioinformatics* 17:346–351 DOI 10.1093/bib/bbv051.
- McKinney W. 2010. In: *Data Structures for Statistical Computing in Python*, Austin, Texas. 56–61 DOI 10.25080/Majora-92bf1922-00a.
- Molkenov A, Zhelambayeva A, Yermekov A, Mussurova S, Sarkytbayeva A, Kalykbergenov Y, Zhumadilov Z, Kairov U. 2019. Transcriptomic satabases. In: Ranganathan S, Gribskov M, Nakai K, Schönbach C, eds. *Encyclopedia of bioinformatics and computational biology*. Amsterdam, Netherlands: Elsevier, 341–351 DOI 10.1016/B978-0-12-809633-8.20208-2.
- Müller H, Jimenez-Heredia R, Krolo A, Hirschmugl T, Dmytrus J, Boztug K, Bock C. 2017. VCF.Filter: interactive prioritization of disease-linked genetic variants from sequencing data. *Nucleic Acids Research* 45:W567–W572 DOI 10.1093/nar/gkx425.

- Paila U, Chapman BA, Kirchner R, Quinlan AR. 2013.** GEMINI: integrative exploration of genetic variation and genome annotations. *PLOS Computational Biology* **9**:e1003153 DOI [10.1371/journal.pcbi.1003153](https://doi.org/10.1371/journal.pcbi.1003153).
- Python Software Foundation. 1999.** tkinter—Python interface to Tcl/Tk. Available at <https://docs.python.org/3/library/tkinter.html> (accessed on 20 April 2021).
- Rigden DJ, Fernández XM. 2020.** The 27th annual nucleic acids research database issue and molecular biology database collection. *Nucleic Acids Research* **48**:D1–D8 DOI [10.1093/nar/gkz1161](https://doi.org/10.1093/nar/gkz1161).
- Van Rossum G, Drake FL. 2010.** The Python language reference. Hampton: Python Software Foundation.
- Salatino S, Ramraj V. 2016.** BrowseVCF: a web-based application and workflow to quickly prioritize disease-causative variants in VCF files. *Briefings in Bioinformatics* **bbw054** DOI [10.1093/bib/bbw054](https://doi.org/10.1093/bib/bbw054).
- The ICGC/TCGA Pan-Cancer Analysis of Whole Genomes Consortium. 2020.** Pan-cancer analysis of whole genomes. *Nature* **578**:82–93 DOI [10.1038/s41586-020-1969-6](https://doi.org/10.1038/s41586-020-1969-6).
- The 1000 Genomes Project Consortium. 2015.** A global reference for human genetic variation. *Nature* **526**:68–74 DOI [10.1038/nature15393](https://doi.org/10.1038/nature15393).
- Tollefson GA, Schuster J, Gelin F, Agudelo A, Ragavendran A, Restrepo I, Stey P, Padbury J, Uzun A. 2019.** VIVA (Visualization of VARIants): a VCF file visualization tool. *Scientific Reports* **9**:12648 DOI [10.1038/s41598-019-49114-z](https://doi.org/10.1038/s41598-019-49114-z).
- Van Dijk EL, Jaszczyszyn Y, Naquin D, Thermes C. 2018.** The third revolution in sequencing technology. *Trends in Genetics* **34**:666–681 DOI [10.1016/j.tig.2018.05.008](https://doi.org/10.1016/j.tig.2018.05.008).