Viewing and manipulating Variant Call Format files

Elliott Magnuson

December 12th, 2022

Contents

1 Introduction

1.1 1000s genome project

The 1000s Genomes Project, started in 2008, was a large effort by researchers to catalog the genetic variation of different people all over the world via the usage of whole genome sequencing. Its goal was to create a reference of human genetic variation globally. The project was completed in 2015 after characterizing the genomes of 2,504 people from 26 different populations [1]. While the vast majority of the 88 million variants they discovered were single nucleotide polymorphism (SNPs)(84.7 million), the researchers also discovered short insertions/deletions (indels)(3.6 million) and structural variants (60,000). Figure 1 below shows the locations that were collected by the 1000s Genomes Project.

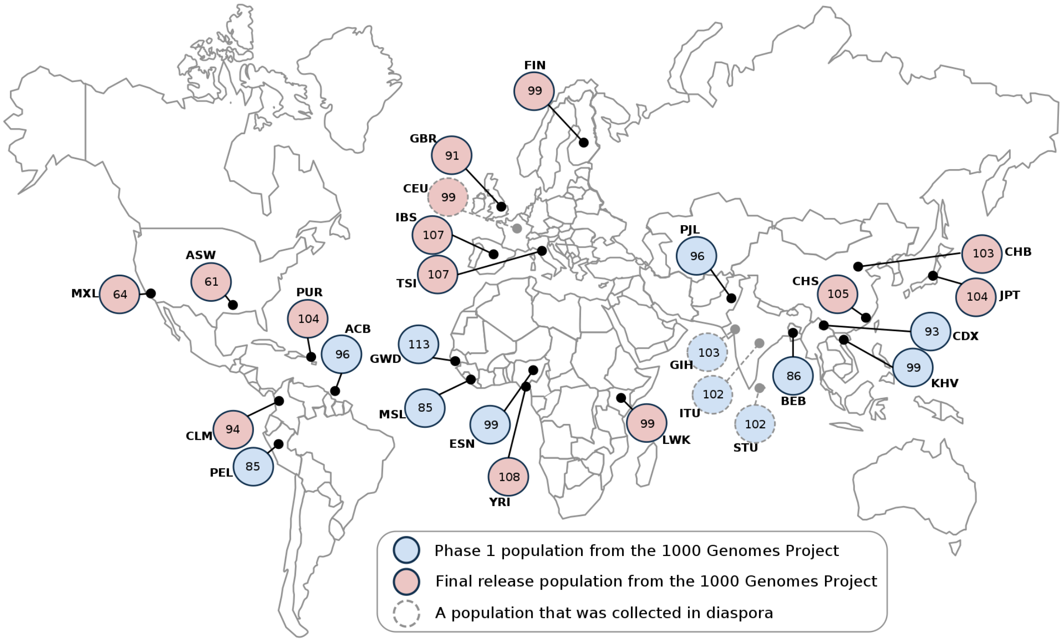


Figure 1: Locations of volunteers for the 1000s Genomes Project. It is noticeable that the project did not collect from various countries around the world, notably among them Russia. Researchers have created the "Genome Russia Project" to fix this [2].

1.2 Variant Call Format

The Variant Call Format (VCF) file type was created by the 1000s Genomes Project so they could store genetic variation data collected in an organized and meaningful way [3]. Using this format, the only necessary thing that needs to be stored is the reference genome and the variations themselves. The VCF file extension is ".vcf", but the file itself is just text, so any text editor can open and display the data inside.

There are a number of steps that are required to go through to end up with a meaningful VCF file. If you send in your samples for sequencing, many companies will complete these steps for you through what is called a pipeline, and they will send you your results of SNPs and indels in VCF form. For example, a sample of DNA is sequenced, which produces a raw FASTA DNA sequence. The sequence is aligned with a reference creating a Sequence Alignment/Map (SAM) file. Processing the SAM file involves IDing where the alignment is different between the reference and the target sequence, and the VCF is built. The term for this is "variant calling" - whereby we are able to ultimately ID the variants from the input of the sequence data. This process has become a standardized way for bioinformaticians to tidy and handle massive amounts of genomic data into parasable files that contain only the variation data. The Global Alliance for Genomics & Health (GA4H) has laid out what they call the Large Scale Genomics work stream, the process mentioned above acting as a pipeline of genomic formatting.

1.3 Anatomy of a VCF file

While the text in a VCF file in itself is not immediately very helpful to a human reader, I will go over the basics of the format itself before we learn how to use tools that have been created to parse and manipulate them. The anatomy of a VCF file is extremely well documented, and the full specification of the newest standard (VCFv4.3) is freely available online.

Figure 2 below shows the standardized anatomy of a VCF file. It can be a lot to parse if you have never seen one before, so I will go over the most important parts below.

There are three standard parts of a VCF File:

1. Meta data rows that begin with "##"
2. Field definition line that begins with "#" and is always the last row of the header. This row has 8 mandatory fields, which contain further information.
3. Variation data itself, with each row representing one variant. Data is filled according to the column.

```
"." is used in place of known data
```

VCF example (From Figure 2a) with only one variant row:

```
##fileformat=VCFv4.1
##fileDate=20110413
##source=VCFtools
##reference=file:///refs/human_NCBI36.fasta
#CHROM  POS ID    REF   ALT QUAL  FILTER    INFO
 1        5  rs12 A     G    67   PASS      .
```

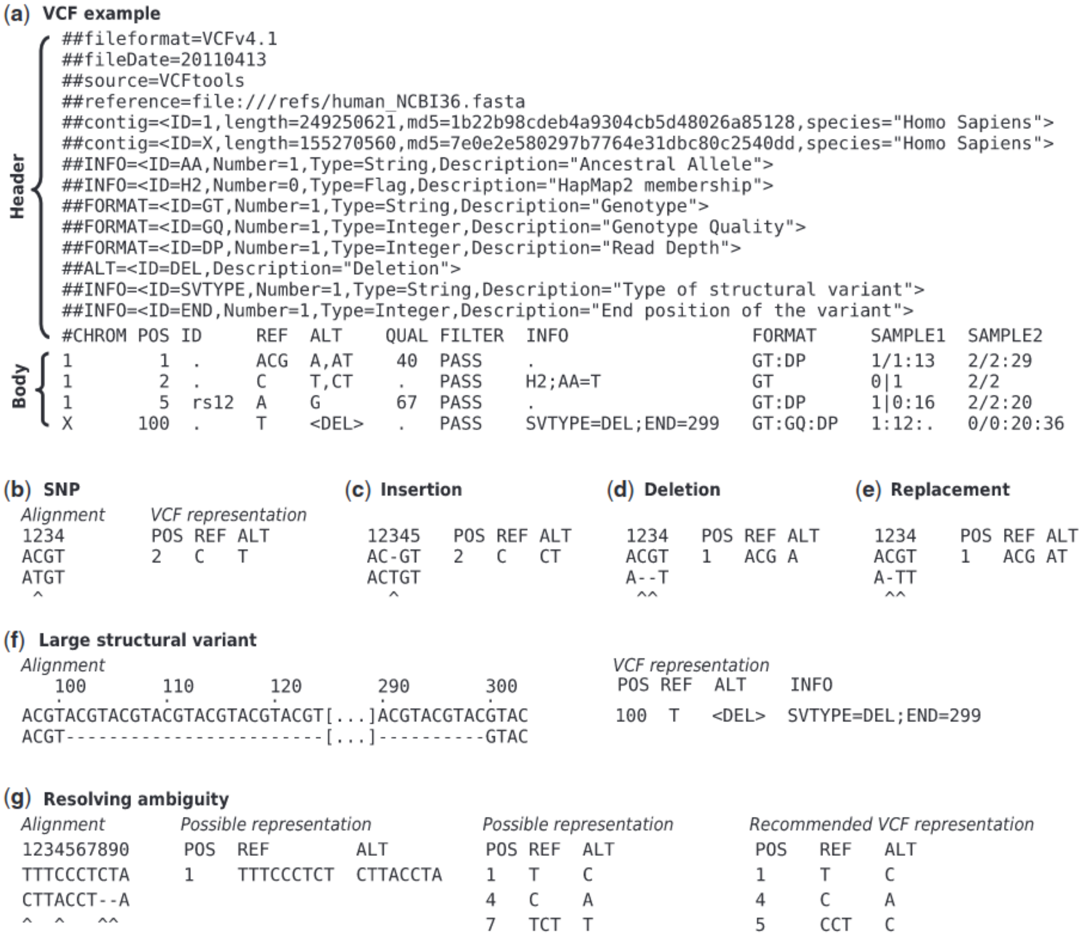
This small VCF example, taken from Figure 2a, show the first four rows of the VCFfile, the 8 mandatory fields, and one variant. While `fileformat` is mandatory as the first header row, the other rows are not. Yet, it is clear that presenting data such as the reference genome is useful. This metadata provides a standardized way to share information in the VCF file about various things like version of reference sequence, software that has already been used on the file, and is featured to fit the dataset itself. In this VCF you can see the reference human_NCBI36.fasta was used.

The field definition lines contains the following fields:

Field name	VCF Example value	Description
CHROM	1	Chromosome (in order)
POS	5	Position of the starting base pair variant
ID	rs12	Unique variant identifier
REF	A	reference allele (one BP or sequence) at the same position
ALT	G	alleles from the sample

Field name	VCF Example value	Description
QUAL	67	quality score that is phred-scaled
FILTER	PASS	information of site filtering
INFO	.	semicolon separated and has user annotations that correspond to meta data above

Each row below the headers represents one variant. As a reminder, VCF files exist to show the variants and this is represented by SNPs (Figure 2b), indels (Figure 2c,d), and structural variants (Figure 2e,f).



(a) VCF example

Header
Body

```
##fileformat=VCFv4.1
##fileDate=20110413
##source=VCFtools
##reference=file:///refs/human_NCBI36.fasta
##contig=<ID=1,length=249250621,md5=1b22b98cdeb4a9304cb5d48026a85128,species="Homo Sapiens">
##contig=<ID=X,length=155270560,md5=7e0e2e580297b7764e31dbc80c2540dd,species="Homo Sapiens">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##ALT=<ID=DEL,Description="Deletion">
##INFO=<ID=SVTYPE,Number=1,Type=String,Description="Type of structural variant">
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the variant">
#CHROM POS ID     REF  ALT    QUAL FILTER  INFO                 FORMAT     SAMPLE1   SAMPLE2
1        1 .      ACG  A,AT    40  PASS    .                    GT:DP      1/1:13    2/2:29
1        2 .      C    T,CT    .   PASS    H2;AA=T              GT         0|1       2/2
1        5 rs12   A    G       67  PASS    .                    GT:DP      1|0:16    2/2:20
X      100 .      T    <DEL>   .   PASS    SVTYPE=DEL;END=299   GT:GQ:DP   1:12:.    0/0:20:36
```

(b) SNP

```
Alignment    VCF representation
1234         POS REF ALT
ACGT         2   C   T
ATGT
 ^
```

(c) Insertion

```
12345   POS REF ALT
AC-GT   2   C   CT
ACTGT
  ^
```

(d) Deletion

```
1234   POS REF ALT
ACGT   1   ACG A
A--T
 ^^
```

(e) Replacement

```
1234   POS REF ALT
ACGT   1   ACG AT
A-TT
 ^^
```

(f) Large structural variant

```
Alignment                                                      VCF representation
   100       110       120       290       300                 POS REF  ALT    INFO
    .         .         .         .         .
ACGTACGTACGTACGTACGTACGTACGT[...]ACGTACGTACGTAC                100  T   <DEL>  SVTYPE=DEL;END=299
ACGT------------------------[...]----------GTAC
```

(g) Resolving ambiguity

```
Alignment     Possible representation       Possible representation    Recommended VCF representation
1234567890    POS  REF        ALT           POS REF  ALT               POS   REF   ALT
TTTCCCTCTA    1    TTTCCCTCT  CTTACCTA      1   T    C                 1     T     C
CTTACCT--A                                  4   C    A                 4     C     A
^  ^   ^^                                   7   TCT  T                 5     CCT   C
```

Figure 2: a) Anatomy of a VCF file. b-f) Further explanation of the BAM alignment representation of a genome and its VCF representation. g) Recommended VCF representation of alignment data that could be ambiguous [1].

The variant ID is a unique value that is useful when looking to search for information about the variant. For example, in our VCF example above, the one row that exists has an ID of `rs12`. This SNP ID is assigned by the NCBI's database of SNPs, called dbSNP or the European Variation Archive (EVA). The European Bioinformatics Institute (EMBL-EBI) has a great resource listing all of the varying ID values and where they come from.

You may have noticed that there is some meta data information in Figure 2a that I did not introduce. The specifics of what each line means can become quite convoluted. The NIH has an excellent resource on INFO tags and their meaning. Likewise, the 1000s Genome Project created a poster with a VCF example as well.

2 Tutorial

We will be using the clinvar_20221211.vcf.gz VCF file from the NIH repo here. It is a VCF file that represents all human variants that are of clinical significance and has been mapped to GRCh37 assemblies. The clinvar_20221211.vcf.gz file is inside of the VCF folder in the same directory as this PDF, so there is no need to download it. While the `.gz` extension is a zipped version of the file, IGV will still be able to view the file, so there is no need to unzip the file either.

Running the Unix command "`wc -l clinvar_20221211.vcf`" on the file shows that this it is 240,695 lines long, so it will be a mistake to open this in a text file or in Excel. Luckily, there are many freely available programs that can open and parse this data in an efficient manner. As is common in Computer Science, and perhaps an example of not invented here syndrome, there are about a million different tools that various people and organizations have created to deal with managing VCF files.

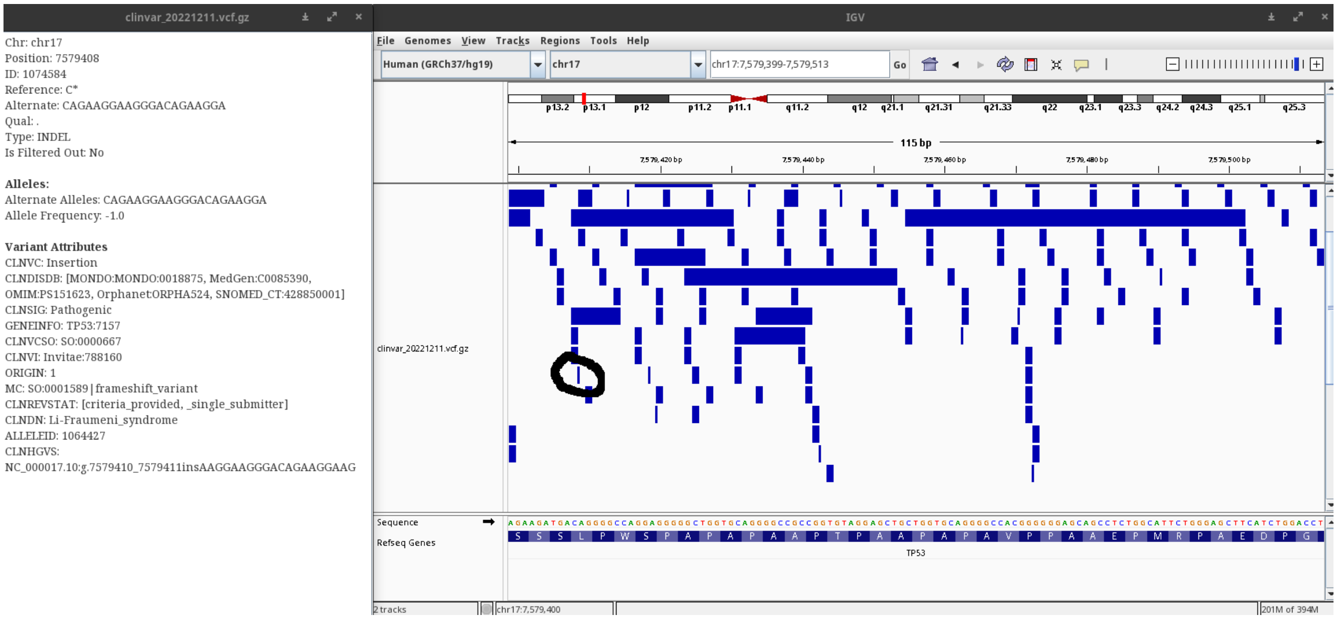
2.1 Viewing a VCF file with IGV

So, we have a VCF file. The variants in this file can be viewed in many different biologically meaningful ways. It would be nice to see a visual representation of all of the genetic variants on each chromosome. Fortunately, the Broad Institute has created a program, Integrated Genomics Viewer (IGV) that can do exactly that.

Steps for IGV

1. IGV can be installed here. There is also a web app, but it is not as functional as the desktop app.
2. If you are familiar with Java and know you have Java Runtime Environment 11 (JRE) install on your computer, you can install the app that says "Separate Java 11 required", otherwise install the "Java included" version of your OS.
3. Unzip the file installed onto your computer. Open a terminal and `cd` to the directory and execute `\.igv-launcher.bat` if you are on Windows or `./igv.sh` if you are on Linux or Mac OS.
4. In the upper left use `File` -> `Load from File` -> and point to `clinvar_20221211.vcf.gz`. In order for this file to load correctly, IGV also needs the `clinvar_20221211.vcf.gz.tbi` file in the directory (it is there already).

5. You'll still see an empty screen. This VCF file contains variants across all chromosomes, so it can be difficult to zoom into find one of the variants.
6. In the search bar type in `chr17:7,579,399-7,579,513`. Chromosome 17 p13.1 is the location of the tp53 gene. Mutations in this gene happens in many types of cancers, so we would expect many different variants to be of clinical significance.



clinvar_20221211.vcf.gz

Chr: chr17
Position: 7579408
ID: 1074584
Reference: C*
Alternate: CAGAAGGAAGGGACAGAAGGA
Qual: .
Type: INDEL
Is Filtered Out: No

Alleles:
Alternate Alleles: CAGAAGGAAGGGACAGAAGGA
Allele Frequency: -1.0

Variant Attributes
CLNVC: Insertion
CLNDISDB: [MONDO:MONDO:0018875, MedGen:C0085390, OMIM:PS151623, Orphanet:ORPHA524, SNOMED_CT:428850001]
CLNSIG: Pathogenic
GENEINFO: TP53:7157
CLNVCSO: SO:0000667
CLNVI: Invitae:788160
ORIGIN: 1
MC: SO:0001589|frameshift_variant
CLNREVSTAT: [criteria_provided, _single_submitter]
CLNDN: Li-Fraumeni_syndrome
ALLELEID: 1064427
CLNHGVS:
NC_000017.10:g.7579410_7579411insAAGGAAGGGACAGAAGGAAG

IGV
File Genomes View Tracks Regions Tools Help
Human (GRCh37/hg19)
chr17
chr17:7,579,399-7,579,513
Go
p13.2 p13.1 p12 p11.2 p11.1 q11.2 q12 q21.1 q21.31 q21.33 q22 q23.1 q23.3 q24.2 q24.3 q25.1 q25.3
115 bp
7,579,420 bp
7,579,440 bp
7,579,460 bp
7,579,480 bp
7,579,500 bp
clinvar_20221211.vcf.gz
Sequence
AGAAGATGACAGGGGCCAGGAGGGGGCTGGTGCAGGGGCCGCCGGTGTAGGAGCTGCTGGTGCAGGGGCCACGGGGGGAGCAGCCTCTGGCATTCTGGGAGCTTCATCTGGACCT
Refseq Genes
S S S L P W S P A P A P A A P T P A A P A P A V P P A A E P M R P A E D P G
TP53
2 tracks
chr17:7,579,400
201M of 394M

Figure 3: IGV showing the variants of the clinvar_20221211.vcf file compared to the reference GRCh37. The section with all of the blue bars is the variants. The depth has no meaning merely helps find variants

Figure 3 above shows the variants present in the clinvar_20221211.vcf file for the tp53 gene at position `chr17:7,579,399-7,579,513`. The blue lines are the "Variant Information Track" and each blue bar is a single variant. The depth has no meaning and is present to help see each variant. I have circled one of the variants I clicked, and the pop up menu to the left shows information about this variant, as well as the 8 fields of a VCF file. As you can see, at chromosome 17 position 7579408 the reference shows a single `C` base pair, but the sample has a large indel alternative value of `CAGAAGGAAGGGACAGAAGGA`. Further information shows this is clinically significant and related to Li-Fraumeni syndrome, a rare genetic disorder that causes a nearly 100% chance of breast cancer in women.

We can go further as well. Searching the ID `1074584` on NIH's ClinVar website, we can see further information, as well as the publication that found the indel [4].

As we can see, IGV is an extremely powerful program. It is one of the many classic free/open source programs that accomplishes it's goal with wonderful efficiency but just also has a really bland UI. That is opposed to paid and closed source software that looks beautiful and sucks. Or God forbid a paid closed source software that also looks terrible.. shudders.

2.2 Filtering VCF files with VCF Tools

This VCF files we used, clinvar_20221211.vcf.gz, has hundreds of thousands of variants present, spread out across every chromosome. Unless you know exactly where to look, it can be difficult to find the variant you are looking for in a specific chromosome. This can be achieved with BCF Tools created by researchers affiliated with the 1000s genome project, an incredibly powerful command line program built for manipulating VCF files [2].

2.2.1 Installing Windows Subsystem for Linux

Unfortunately, the really good programs that let you parse and manipulate VCF files are for Unix like operating systems. This includes VCF Tools. As such, if you are on Windows you will need to install the Windows Subsystem for Linux (WSL).

Full documentation is here, but the steps are as follows:

1. Open PowerShell in administrator mode.
2. Enter `wsl --install` into PowerShell.
3. Reboot your computer.
4. Ubuntu, a distribution of Linux is installed on your computer now.
5. Open the start menu, search Ubuntu, and running it as an App will open up and instance of Ubuntu.
6. Follow the prompt to create a username and password.
7. Run `sudo apt update` to update your system.
8. VCF Tools will require a C compiler, make, and a `gz` zipper, so run `sudo apt install gcc make gzip`.

2.2.2 Installing VCF Tools

1. We can install VCF Tools from here. Click the download button and it will install `vcftools_0.1.13.gz`.
2. You can unzip this file with `gzip -d vcftools_0.1.13.gz`.
3. Enter `cd vcftools_0.1.13`.
4. Enter `make` and it will compile VCF Tools to a file in the current directory called `bin`.
5. After compiling, enter `cd bin`
6. Enter `pwd` and copy the output, this is the current directory you are in
7. To make the VCF Tools binary file run as a command line program from any directory, you then need to enter `export PATH=$PATH:paste/path/here/`
8. my result for pwd is `/home/elliott/.sourcecode/vcftools_0.1.13/bin` so I would enter `export PATH=$PATH:/home/elliott/.sourcecode/vcftools_0.1.13/bin`.

This was a lot of work, but after closing and reopening your Ubuntu instance, now the VCF Tools binary can be run from any directory you want!

2.2.3 Using VCF Tools

1. Open up a terminal in the directory where you have `clinvar_20221211.vcf.gz`.

Side note: with Windows, here's where things get even more weird. You are running two different operating systems at the same time, and they each have their own file system. Microsoft has documentation on how to access files between them here, but it can be boiled down to use `explorer.exe .` in the terminal to open the currently directory of your Linux system. You can paste `clinvar_20221211.vcf.gz` in this directory and it should be accessible after that.

VCF Tools is well documented, but I will go over a few commands that may be useful below.

2. VCF Tools requires your `.vcf` file to unzipped, so you must first run `gzip clinvar_20221211.vcf.gz` and this will create `gzip -d clinvar_20221211.vcf` in the same directory.
3. `vcftools --vcf clinvar_20221211.vcf` will give you the amount of variants in the file, 1572295.
4. You can specify what chromosome to look at with `vcftools --vcf clinvar_20221211.vcf --chr 17 --from-bp 0 --to-bp 2000000 --recode -c | gzip -c > Chromosome17Subset.vcf.gz` and this command with parse out only the base pairs between 0 and 20 million on chromosome 17 and place it into a file called `Chromosome17Subset.vcf.gz`.

Further documentation can be found here, here, and here.

3 Other available programs

This tutorial has examined two programs, IGV and VCFTools to visualize and parse data respectively. However, there are a large number of other programs that have been created to do this as well. I have listed them here as a resource should you wish to view them.

- sgkit provides a Python API, with parsing performed by cyvcf2
- The authors at samtools created another very powerful command line program called BCF Tools[5].

4 Conclusion

The pipeline from sequencing DNA to determining the variation in the DNA sample compared to a reference involves a lot of steps and a massive amount of data. There exists a standardized way in which bioinformaticians use to process this data. One of the steps is the output, a VCF file containing the variation. Yet, this file still contains a large amount of data that can make it difficult to immediately be meaningful. There are programs and tools that we can use to help parse the information out of these VCF files into a more human readable format, so that we can extract biologically meaningful conclusions, and results, etc.

For example, IGV allows us to interactively see the variants in your VCF files. From the example above, we were even able to determine what the possible effect would be from this variant, as well as the exact location on the genome.

5 References

1. 1000 Genomes Project Consortium, Auton A, Brooks LD, Durbin RM, Garrison EP, Kang HM, Korbel JO, Marchini JL, McCarthy S, McVean GA, Abecasis GR. A global reference for human genetic variation. Nature. 2015 Oct 1;526(7571):68-74. doi: 10.1038/nature15393. PMID: 26432245; PMCID: PMC4750478.
2. Oleksyk TK, Brukhin V, O'Brien SJ. The Genome Russia project: closing the largest remaining omission on the world Genome map. Gigascience. 2015 Nov 13;4:53. doi: 10.1186/s13742-015-0095-0. PMID: 26568821; PMCID: PMC4644275.
3. 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. The variant call format and VCFtools. Bioinformatics. 2011 Aug 1;27(15):2156-8. doi: 10.1093/bioinformatics/btr330. Epub 2011 Jun 7. PMID: 21653522; PMCID: PMC3137218.
4. Ruijs MW, Verhoef S, Rookus MA, Pruntel R, van der Hout AH, Hogervorst FB, Kluijt I, Sijmons RH, Aalfs CM, Wagner A, Ausems MG, Hoogerbrugge N, van Asperen CJ, Gomez Garcia EB, Meijers-Heijboer H, Ten Kate LP, Menko FH, van 't Veer LJ. TP53 germline mutation testing in 180 families suspected of Li-Fraumeni syndrome: mutation detection rate and relative frequency of cancers in different familial phenotypes. J Med Genet. 2010 Jun;47(6):421-8. doi: 10.1136/jmg.2009.073429. PMID: 20522432.
5. Petr Danecek, James K Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O Pollard, Andrew Whitwham, Thomas Keane, Shane A McCarthy, Robert M Davies, Heng Li, Twelve years of SAMtools and BCFtools, GigaScience, Volume 10, Issue 2, February 2021, giab008, https://doi.org/10.1093/gigascience/giab008