GenepoweRx™
Genetic Science powered by K&H

GenepoweRx® Bioinformatics for Comprehensive testing

White Paper | Version 1.0 | 3-March-2023

Authored by: K. Srinivas

Reviewed and supported by

Dr. Kalyan Ram Uppaluri
Managing Director

Vamsi Mohan Challa
CTO

Uppaluri K&H Personalized Medicine Clinic Pvt. Ltd. Hyderabad

Dr. Hima J Challa
Director

Dr. Kalyani P
CSO

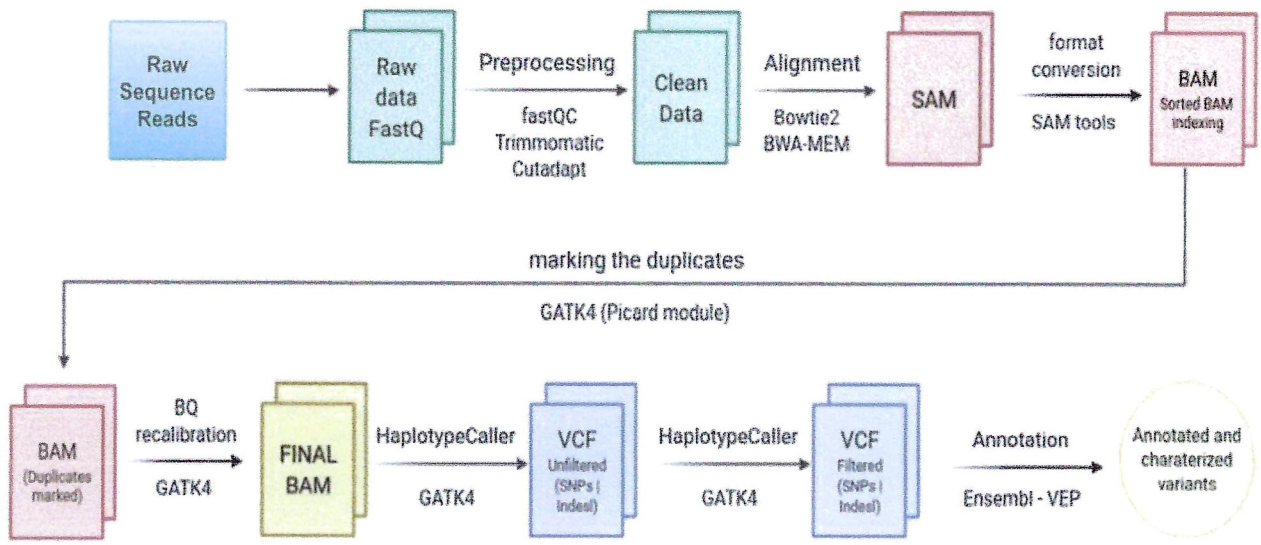
GenepoweRx™
Genetic Science powered by K&H

Title: GenepoweRx® Bioinformatics for Comprehensive testing

Authors: Kalyan Ram Uppaluri, Hima J. Challa, Kalyani P, Ramya Gadicherla, Krishna Vardhini.K, Anusha. G, Natya.K, Aswini.K, Shyam Sundar, Srinivas K,

Bioinformatics (biological informatics), is a interdisciplinary field of biology and computer science that involves using computer technology to collect, store, analyze and disseminate biological data and information, such as DNA and amino acid sequences or annotations about those sequences. The huge amount of biological data has been generated by the Human Genome Project (HGP). Bioinformatics uses its computational applications and analytical tools to capture and interpret the data. It also uses various database that organize and index such biological information to increase our understanding of health and disease and, in medical practices.

variant detection in Human Whole Exome Sequencing Samples:



FastQC:

To perform the quality control of raw reads we adopted an open source tool called FastQC. FastQC tool is most popular for quality check of high-throughput sequencing data. To get quality check report for FASTQ files, we run FastQC workflow. Quality control can be performed for raw data, before and after trimming of the raw reads to observe the changes which are occurred after trimming in FastQ files. FastQC tool outputs a html file that can be viewed directly in internet browser.

After saving the fastqc report. We will observe the bellow listed parameters summary

1. Basic statistics
2. Per Base Sequence Quality
3. Per Sequence Quality Scores
4. **Per base sequence content**
5. Per base GC content
6. Per Sequence GC content
7. Per base N content
8. Sequence Length Distribution
9. **Sequence Duplication Levels**
10. **Overrepresented Sequences**
11. Adapter content
12. kmer content

From above mentioned parameters result file shows the accepted signal with green color, warning message and failure report with red color.

Functions of FastQC are listed below:

Import Raw data FastQ files, SAM or BAM files (any variant) Quickly provides overview to tell us problematic area in the sequence for import data tables and summary graphs can be quickly accessed Offline operation to allow automated generation of reports without running the interactive application. After quality check satisfactory report we will proceede further steps in our analysis pipeline

Trimmomatic:

Trimmomatic (version 0.36) tool used for trim and crop the illumina raw reads (FASTQ format) and to remove adapter sequences from 5' end and 3' end. We adopted paired end mode from trimmomatic tool to maintain read pairs. using phred+33 or phred + 64 sequence quality scores will be considered depending on the Illumina pipeline used. The trimming steps and their parameters are given on the command line.

trimming steps:

ILLUMINACLIP: remove adapter and other illumina-specific sequences from the reads.

LEADING: cut the bases from the start of a read, if below a threshold quality.

TRAILING: cut the bases off the end of a read, if below a threshold quality.

CROP: Cut the read to a specified length by removing bases from the end

HEADCROP: Cut the specified number of bases from the start of the read

MINLEN: Drop the read if it is below a specified length

AVGQUAL: Drop the read if the average quality is below the specified level

Multiple parameters can be specified based on requirement, using additional arguments.

Input/Output files required for trimmomatic:

paired end data of forward and reverse reads, and 4 output files should be specified in command line, 2 for paired output where both the pairs survived the processing, and 2 for corresponding unpaired output where a read survived, but the partner read did not.

Trimmomatic running command:

java -jar trimmomatic-0.36.jar PE -threads 8 (No of threads based on system configuration) -trimlog outputlog (creates a log of all read trimmings) forward_reads.fastq reverse_reads.fastq forward_trim.fq forward_untrim.fq reverse_trim.fq reverse_untrim.fq CROP:91 (CROP will remove bases from the end reads)

Cutadapt:

Before we start the alignment and analysis processes, it is useful to perform some initial quality checks on your raw data. We may also need to pre-process the sequences to trim them or remove adapters. Cutadapt is a tool for removing adapter sequences from DNA sequencing data.

Although most of the adapters are located at the 3' end of the sequencing read, Cutadapt allows multiple adapter removal from both 3' and 5' ends.

Cutadapt finds and removes adapter sequences, primers, poly-A tails and other types of unwanted sequence from your high-throughput sequencing reads. Cleaning your data in this way is often required: Reads from small-RNA sequencing contain the 3' sequencing adapter because the read is longer than the molecule that is sequenced. Amplicon reads start with a primer sequence. Poly-A tails are useful for pulling out RNA from your sample, but often you don't want them to be in your reads.

Cutadapt helps with these trimming tasks by finding the adapter or primer sequences in an error-tolerant way. It can also modify and filter single-end and paired-end reads in various ways. Adapter sequences can contain IUPAC wildcard characters. Cutadapt can also demultiplex your reads.

The basic usage of Cutadapt:

Where <adapter_sequence> is the nucleotide sequence of the actual adapter, input reads.[fasta|fastq] is the input file with sequencing data in fasta/fastq format, and respectively, output reads.[fasta|fastq] is the final trimmed file in fasta/fastq format.

Cutadapt searches for the adapter in all reads and removes it when it finds it. All reads that were present in the input file will also be present in the output file, some of them trimmed, some of them not. Even reads that were trimmed entirely (because the adapter was found in the very beginning) are output. All of this can be changed with command-line options.

Cutadapt is available under the terms of the MIT license.

Cutadapt development was started at TU Dortmund University in the group of Prof. Dr. Sven Rahmann. It is currently being developed within NBIS (National Bioinformatics Infrastructure Sweden).

Alignment of raw reads to reference genome:

Tools: 1. BWA, 2. BWA-MEME, 3. STAR 4. Bowtie

Bowtie2 is an open source fast and memory-efficient tool for mapping the trimmed/smoothed reads to reference genome. Bowtie2 runs on the command line under Windows, Mac OS X and Linux. Bowtie2 allows gapped, local, and paired-end alignment modes. Multiple processors can be used for parallel mapping to reduce alignment time and increase alignment speed. Bowtie2 gives output of aligned reads in SAM format, enabling the operations with large number of other tools (e.g. SAMtools, GATK) which uses SAM files as input.

Bowtie2-build index:

Bowtie2 in build function bowtie2-build used to build the index files for Human whole genome (GRCh38.fa) and outputs of 6 files suffixes .1.bt2, .2.bt2, .3.bt2, .4.bt2, .rev.1.bt2, and .rev.2.bt2. These files constitute the index: they all are required to align reads to the reference genome. once the index is built the original FASTA files are no longer used by Bowtie2.

index inspector bowtie2-inspect:

For created indexes from above step will be validated or cross checked using bowtie2-inspect built-in function to extract information about what kind of index it is and what reference sequences were used to build the index. it can also use to extract just the reference sequence names using -n/--names option.

bowtie2 aligner:
Paired input:

Alignment of raw reads to reference genome can be done using bowtie2 index files created in the previous step. Alignment process enables us to discover how and where the read sequences are similar to the reference sequence (where a read is originated with respect to the reference genome).

Paired SAM output:

Bowtie2 gives/prints a SAM format output file for aligned read pairs, it prints two records for each paired read. First one describes the alignment for forward read and second read describes the alignment for reverse read.

Alignment summary:

When bowtie2 completes the alignment, it prints summarizing the process on console for paired end reads it will give how many reads were paired, number of reads aligned concordantly 0 times, aligned concordantly exactly 1 time, aligned concordantly more than 1 time.

mapping command line:

if computer has multiple processors/cores, use -p option to define the cores for this job/task

bowtie2 -p (No. of threads) -x index files path with prefix -1 forward_file.fastq -2 reverse_file.fastq -S output.sam

Samtools:

Samtools is an open source suite for dealing with high-throughput sequencing data. It has three repositories.

Samtools: viewing/indexing/writing/reading/editing of SAM/BAM/CRAM format files

BCFtools: writing/reading BCF2/VCF/gVCF files and filtering/calling/summarizing SNP and short indel sequence variants

(VarScan2)

HTSlib: it is a C library for writing/reading high-throughput sequence data

SAM Sequence Alignment/Map:

BAM Binary Alignment/Map

CRAM compressed columnar file format for storing biological sequence aligned to a reference sequence

Set of utilities for interacting with high-throughput sequencing data with processing of short DNA sequence read alignment in SAM. SAM files are generated using read aligners like Bowtie2, Bowtie, BWA, STAR, HISAT, HISAT2 etc. In our pipeline we have used Bowtie2 to align the reads to reference genome and generated the SAM files in previous step. Major applications of samtools like variant calling and alignment viewing as well as sorting,

indexing, data extraction and format conversion. SAM files are very large in size and difficult to handle with normal computers so compression is used to save space and analysis time.

Samtools makes it possible to work directly with BAM files, so using samtools view -bS input.sam -o output.sam command we compressed SAM files into BAM files. BAM files are binary equiuvalent of SAM (human-readable text file) files.

sort BAM file:

samtools sort function sorts BAM file by aligned read coordinates, or by read name. Sort order header tag will be added or existing one updated if necessary.

Samtools sorted output is written into sorted BAM file or the specified file with -o option.

Index sorted BAM file:

samtools index function index to a coordinate-sorted BAM file for fast access. Index is required when region arguments are used to limit samtools view and command to a particular region of interest.

Samtools pileup:

Samtools mpileup utility provides a summary coverage of aligned reads to reference genome at a single base pair resolution, the output from samtools mpileup can be given to BCFtools or VarScan tool to call genomic variants

VarScan:

VarScan is a command line tool for variant calling, it will take a pileup file generated using samtools. VarScan employs a robust heuristic/statistic approach to call variants that meet desired thresholds for read depth, base quality, variant allele frequency, and statistical significance.

Usage to call variants from samtools pileup

Varscan considers/takes as input of samtools generated pileup file (recent version of pileup is mpileup)

varscan2 functions are listed below, and can be reviewed using -h option in command line.

pileup2snp: Identify SNPs from a pileup file

pileup2indel: Identify indels a pileup file

pileup2cns: Call consensus and variants from a pileup file

mpileup2snp: Identify SNPs from an mpileup file

mpileup2indel: Identify indels an mpileup file

mpileup2cns: Call consensus and variants from an mpileup file

somatic: Call germline/somatic variants from tumor-normal pileups

copynumber: Determine relative tumor copy number from tumor-normal pileups

readcounts: Obtain read counts for a list of variants from a pileup file

filter: Filter SNPs by coverage, frequency, p-value, etc.

somaticFilter: Filter somatic variants for clusters/indels

fpfilter: Apply the false-positive filter

processSomatic: Isolate Germline/LOH/Somatic calls from output

copyCaller: GC-adjust and process copy number changes from VarScan copynumber output

compare: Compare two lists of positions/variants

limit: Restrict pileup/snps/indels to ROI positions

calling SNVs:

To identify SNPs from samples SNP calls, with options allow settings the stringency of the varscan2 predictions are listed below.

VarScan mpileup2snp [mpileup file] OPTIONS > SNP output file included vcf format

For SNP call OPTIONS:

--min-coverage: Minimum read depth at a position to make a call [8]

--min-reads2: Minimum supporting reads at a position to call variants [2]

--min-avg-qual: Minimum base quality at a position to count a read [15]

--min-var-freq: Minimum variant allele frequency threshold [0.01]

--min-freq-for-hom: Minimum frequency to call homozygote [0.75]

--p-value: Default p-value threshold for calling variants [99e-02]

--strand-filter: Ignore variants with >90% support on one strand [1]

--output-vcf: If set to 1, outputs in VCF format

--variants: Report only variant (SNP/indel) positions (mpileup2cns only) [0]

calling small InDels:

For calling indels, the following command along with options can used

VarScan mpileup2indel [mpileup file] OPTIONS > indel output file

For indel call OPTIONS:

--min-coverage: Minimum read depth at a position to make a call [8]

--min-reads2: Minimum supporting reads at a position to call variants [2]

--min-avg-qual: Minimum base quality at a position to count a read [15]

--min-var-freq: Minimum variant allele frequency threshold [0.01]

--min-freq-for-hom: Minimum frequency to call homozygote [0.75]

--p-value: Default p-value threshold for calling variants [99e-02]

--strand-filter: Ignore variants with >90% support on one strand [1]

--output-vcf: If set to 1, outputs in VCF format

--variants: Report only variant (SNP/indel) positions (mpileup2cns only) [0]

calling for both SNPs and Indels together also can be done using varscan tool.

filtering results:

Filtering is necessary for identified SNPs and as well as indels. In this step low coverage supporting reads, variant frequency, and or average base quality reads will be discorded. Filetering option applied for output from mpileup2snp or mpileup2indel files.

VarScan filter [variants file] OPTIONS variants file - A file of SNP or indel calls from VarScan pileup2snp or pileup2indel

OPTIONS to filter SNPs and indels:

--min-coverage: Minimum read depth at a position to make a call [10]

--min-reads2: Minimum supporting reads at a position to call variants [2]

--min-strands2: Minimum # of strands on which variant observed (1 or 2) [1]

--min-avg-qual: Minimum average base quality for variant-supporting reads [20]

--min-var-freq: Minimum variant allele frequency threshold [0.20]

--p-value: Default p-value threshold for calling variants [1e-01]

--indel-file: File of indels for filtering nearby SNPs, from pileup2indel command

--output-file: File to contain variants passing filters

VCF Annotation:

dbSNP:

dbSNP (Single Nucleotide Polymorphism Database) is a part of NCBI. it's a major source of molecular biology information. It contains more than **64 million distinct non-redundant** variants including Homo sapiens, Mus musculus, Oryza sativa, and many more other species. dbSNP is a open source public repository database, stores genetic variation data submitted from research laboratories, and this database also considered as source for information using tools we can retrieve the existing information from these database. The dbSNP database accepts only neutral polymorphisms, polymorphisms corresponding to known phenotypes, and region of no variation.

Purpose of dbSNP:

Biology researchers consider dbSNP is a major resource for variants. dbSNP contains all identified genetic variation, which can be used for further investigate for a wide variety of genetically based natural phenomenon. dbSNP provides/guides to applied researchers in pharmacogenomics and association with phenotypic traits.

So we are using SnpSift database along with filtered vatiants generated through VarScan were used and retriving the existing information for our identified SNPs and indels from dbSNP database. dbSNP gives rsIDs for indels as well as SNPs.

Ensembl Variant Effect Predictor (VEP):

VEP is a tool in Ensembl and Ensembl Genomes. This tool allows us to annotate the predicted variants. VEP determines the associated effect of corresponding Ensembl transcripts and proteins. It works by considering the input coordinates of alleles (SNPs,CNVs, indels or structural variations) which we have identified on a particular gene, sequence, protein, transcript or transcription factor. If a input variant causes a change in the protein sequence, the Ensembl VEP will calculate the possible amino acids at that position and the variant would be given a consequence type of missense. VEP can be accessed in two ways. also accessed through online. The first form is online-based which includes the user selection the parameters:

Which species to be compared. And The database for comparison of Ensembl Transcripts. Uploaded data name, input format for the data and fields for data upload. Data can be uploaded from user computer, from an URL-based location or copying their content into an input box.

The second option for user to use VEP is by downloading or cloning the source code in UNIX environments. The features are equal between online and offline downloaded script versions.

File format supports by VEP includes VCF, pileup, HGVS notations and a default format. The default format is a whitespace-separated file that contains the data in columns. The first five columns of input file indicate the chromosome, start position of the allele, end position, allele name, and the strand. The rest of the columns are variation identifiers and are optional. If these columns left in blank, VEP will assign an identifier to in output file.

VEP also provides additional identifier options to the users, extra options to complement the output and filtering. The filtering options allow features like removal of known variants from results, returning variants in exons only, and restriction of results to specific consequences of the variants.

VEP generates the following results by considering the location of variants and the nucleotide variations.

1. Uploaded variation - as chromosome_start_alleles

2. Location - in standard coordinate format (chr:start or chr:start-end)

3. Allele - the variant allele used to calculate the consequence

4. Gene - Ensembl stable ID of affected gene

5. Feature - Ensembl stable ID of feature

6. Feature type - type of feature. Currently one of Transcript, Regulatory Feature, Motif Feature.

7. Consequence - consequence type of this variation

8. Position in cDNA - relative position of base pair in cDNA sequence

9. Position in CDS - relative position of base pair in coding sequence

10. Position in protein - relative position of amino acid in protein

11. Amino acid change - only given if the variation affects the protein-coding sequence

12. Codon change - the alternative codons with the variant base in upper case

13. Co-located variation - known identifier of existing variation

14. Extra - this column contains extra information as key=value pairs separated by ";". Displays extra identifiers

15. Comparison with other databases to find equal known variants

ClinVar Database:

Introduction

Clinvar is an archived public database which is maintained by NCBI. All the genetic variants and information which is of clinical significance is present in ClinVar which is collected or gathered by various laboratories and research groups. It is widely used in the medical genetic field. Around the world, various laboratories are working on medical genetics, each laboratory encounters few diseases and related genetic variants. All this information is gathered at one place. ClinVar uses data standards, such as HGVS nomenclature for variants and MedGen identifiers for conditions. The data are available on the web as variant specific views. The entire data can be downloaded via FTP.

In ClinVar, there is three ID classes VCV (Variation in ClinVar), RCV(Reference ClinVar) and SCV(Submitted record ClinVar) If there are multiple submitted records (SCV IDs) about the same variation / condition pair, they are aggregated within ClinVar's data flow and reported as reference accession with RCV IDs.. Because one variant may include in multiple RCV accessions whenever different conditions are reported for that variants Flow of the modal:

ClinVar provides data in various formats:

XML Format: there are two types of XML files: 1) ClinVarVariationRelease (carries VCV-centric data of CLinVar). 2) ClinVarFullRelease (carries RCV-centric data of ClinVar). These files are released on a monthly basis. VCF Format: This is a type of text file of variants used in bioinformatics for storing gene information. This file carries additional information columns with extra information such as Clinical information, Allele Frequency data, Allele Identifiers, HGVS expression, Variants in other databases, types of variation, Molecular consequences and Allele origin.

Database Scope:

ClinVar accepts variant information from any part of the genome and interpreted for any type of condition. Variants identified through GWAS studies are individually curated and provide an interpretation of clinical significance. ClinVar has descriptive information of associated traits form Human Phenotype Ontology(HPO), OMIM and other sources are trackable and can be used in queries.

1. It supports the clinical validation of human variation.

It helps us to understand the relationship between the genotype and medically important phenotype.

dbVAR Database:

Dbvar is a human genomic structural variant(SV) database from which users can easily access data for their study / research. Structural variation affecting 50bp or more of this in DNA that includes genomic imbalances (insertion, deletion) referred to as copy number variants(CNVs), duplication, inversion, translocation etc. It provides access to the raw data based on availability. dbVar is a free resource that is developed and maintained by NCBI. Its

collection of about 60,000 clinical Structural Variants including clinical assertions (like Benign, likely benign, Uncertain significance, Likely pathogenic, pathogenic). dbVar operates in cooperation with the Database of Genomic Variants Archive (DGVa), a sister database at EBI. DGVa and dbVar both accept submissions, and use similar data models and submission templates. After syncing, dbVar and DGVa contain the same data.

Significance of dbVar:

1. Determine the structural variant(SV) regions to determine whether they occur due to biologically or genomic imbalanced regions.
2. Get information for rare SVs or common SVs.
3. Determine high-priority SVCs with significant functional impact and effects.
4. Identification of evidence of variations in all public SRA data.
5. Identification of population-specific SVCs to gain insight into the functional significance of structural variants and their evolution.
6. Easier aggregation of annotations such as disease and phenotype, frequency, and genomic features that co-locate with SVs.
7. Better searching and matching of genomic coordinates across studies.

References:

1. Andrews S. (2010). FastQC: a quality control tool for high throughput sequence data. Available online at: http://www.bioinformatics.babraham.ac.uk/projects/fastqc
2. http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf
3. MARTIN, Marcel. Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet.journal, [S.l.], v. 17, n. 1, p. pp. 10-12, may 2011. ISSN 2226-6089. Available at: <http://journal.embnet.org/index.php/embnetjournal/article/view/200>. Date accessed: 08 july 2021. doi:https://doi.org/10.14806/ej.17.1.200.
4. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3322381/pdf/nihms-366740.pdf
5. Daniel C Koboldt, Qunyuan Zhang, David E Larson, Dong Shen, Michael D McLellan, Ling Lin, Christopher A Miller, Elaine R Mardis, Li Ding, Richard K Wilson VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. Genome Res.: 2012, 22(3);568-76 [PubMed:22300766].
6. Daniel C Koboldt, Ken Chen, Todd Wylie, David E Larson, Michael D McLellan, Elaine R Mardis, George M Weinstock, Richard K Wilson, Li Ding VarScan: variant detection in massively parallel sequencing of individual and pooled samples. Bioinformatics: 2009, 25(17);2283-5 [PubMed:19542151].
7. Sherry, S.T., Ward, M.H., Kholodov, M., Baker, J., Phan, L., Smigielski, E.M., Sirotkin, K. (2001). dbSNP: the NCBI database of genetic variation. Nucleic Acids Research, 29: 308-311

8. McLaren, W., Gil, L., Hunt, S.E. et al. The Ensembl Variant Effect Predictor. Genome Biol 17, 122 (2016) doi:10.1186/s13059-016-0974-4.
9. Landrum, M. J., Chitipiralla, S., Brown, G. R., Chen, C., Gu, B., Hart, J., Hoffman, D., Jang, W., Kaur, K., Liu, C., Lyoshin, V., Maddipatla, Z., Maiti, R., Mitchell, J., O'Leary, N., Riley, G. R., Shi, W., Zhou, G., Schneider, V., Maglott, D., Holmes, J.B., Kattman, B. L. ClinVar: improvements to accessing data. Nucleic Acids Res. 2020;48(D1):D835-D844.
10. Landrum MJ, Lee JM, Benson M, Brown GR, Chao C, Chitipiralla S, Gu B, Hart J, Hoffman D, Jang W, Karapetyan K, Katz K, Liu C, Maddipatla Z, Malheiro A, McDaniel K, Ovetsky M, Riley G, Zhou G, Holmes JB, Kattman BL, Maglott DR. ClinVar: improving access to variant interpretations and supporting evidence. Nucleic Acids Res. 2018 Jan 4;46(D1):D1062-D1067
11. https://www.ncbi.nlm.nih.gov/dbvar/content/walkthrough/
12. https://www.ncbi.nlm.nih.gov/dbvar/content/help/
13. https://www.ncbi.nlm.nih.gov/dbvar/content/overview/

