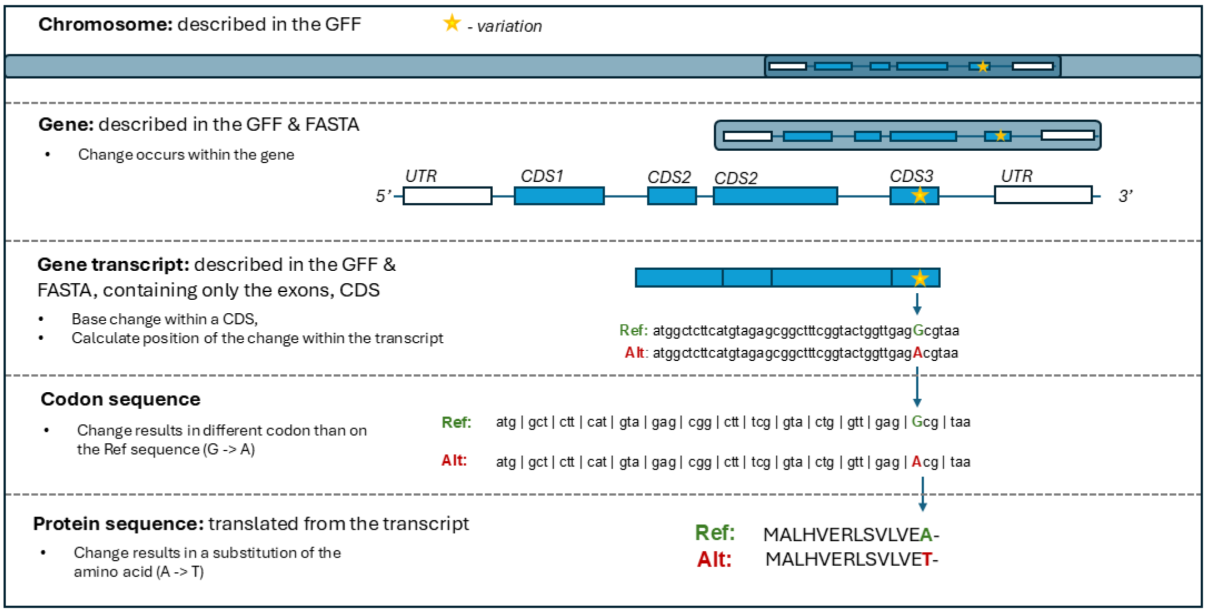
Report

Program Summary & Aim

Whole genome analysis with variant calling software outputs a Variant Call Format (VCF) file, containing the location of the sequence change within the chromosome. It does not, however, state whether the change is within the protein-coding region (A CoDing Sequence, CDS). In order to establish that, the following script tracks the location of variation from the chromosome to the transcript and to the protein (*Figure 1*). Furthermore, it classifies the mutation as: 1) non-coding – within non-coding sequence, 2) Synonymous – not resulting in the change of codon, 3) Non-Synonymous – resulting in the change of codon.

The script untangles the information on the variants from VCF with sequences from the FASTA and genomic features from the general feature format (GFF). Ultimately, it establishes the consequences of the variations on the proteome.



Chromosome: described in the GFF
★ - variation
Gene: described in the GFF & FASTA
• Change occurs within the gene
UTR
CDS1
CDS2
CDS2
CDS3
UTR
5'
3'
Gene transcript: described in the GFF & FASTA, containing only the exons, CDS
• Base change within a CDS,
• Calculate position of the change within the transcript
Ref: atggctcttcatgtagagcggctttcggtactggttgagGcgtaa
Alt: atggctcttcatgtagagcggctttcggtactggttgagAcgtaa
Codon sequence
• Change results in different codon than on the Ref sequence (G -> A)
Ref: atg | gct | ctt | cat | gta | gag | cgg | ctt | tcg | gta | ctg | gtt | gag | Gcg | taa
Alt: atg | gct | ctt | cat | gta | gag | cgg | ctt | tcg | gta | ctg | gtt | gag | Acg | taa
Protein sequence: translated from the transcript
• Change results in a substitution of the amino acid (A -> T)
Ref: MALHVERLSVLVEA-
Alt: MALHVERLSVLVET-

Figure 1. Establishing the downstream effects of the variation in set chromosomal location to the protein. *If the variation is within the coding region of the gene, it is traced back to the specific CDS and its location within the transcript, not the chromosome is calculated. Then, the change of base pair is transferred onto codon, and the ultimate protein sequences from the reference (Ref) and altered (Alt) transcript are compared.*

Input files

All input files should be entered in the command line command in the order assigned by arguments. The exemplary command line for the script would be as follows: *python Assignment.py --Input_vcf assessmentData.vcf.gz --Input_gff PlasmoDB-54_Pfalciparum3D7.gff --Input_fasta PlasmoDB-54_Pfalciparum3D7_Genome.fasta.* The arguments in the command are:

1) Path to the python script – positional argument, here as: *"Assignm2.py"*, if it is located in the working directory, the name of the file is sufficient. Mandatory argument.
2) "—Input_VCF": path to the VCF file, here as *"assessmentData.vcf.gz"*. Mandatory argument. VCF file supplies the information about detected variants between the reference and altered genome.
3) "—Input_GFF": path to the GFF file, here as "*PlasmoDB-54_Pfalciparum3D7.gff"*. Mandatory argument. GFF supplies information on the genomic features, eg. Exons, mRNA, CDS, within the genome.

4) "—Input_fasta": path to the fasta file, here as "*PlasmoDB-54_Pfalciparum3D7_Genome.fasta*". Mandatory argument. Fasta file supplies the sequences for the gene ID.

Output file

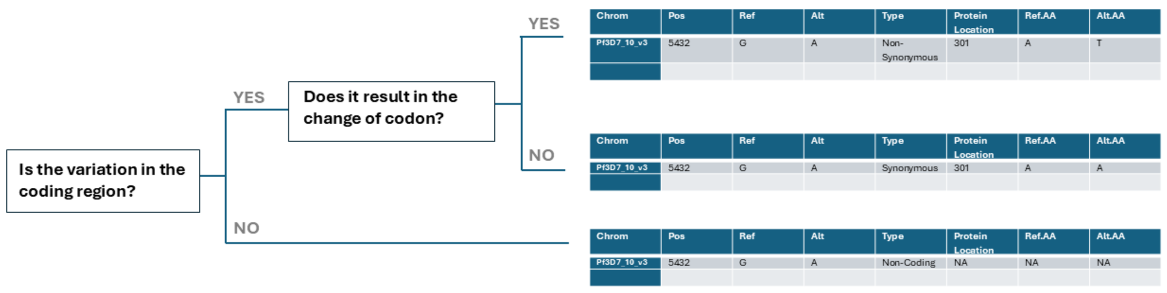
The script produces three outputs: 1) a tab-separated table with data per variation about the amino acid and nucleotide change , 2) a bar-graph with the number of variations grouped by type (non-coding/synonymous/non-synonymous), 3) a log file with pertinent information on the running on the script.

Firstly, the output table has information on the type, location (on the chromosome and the protein), the amino acid and nucleotide change in per chromosome transcript (*Figure 2*). Variations not located in the coding region are included in the table with type "Non-Coding" and NA per protein coordinates (*Figure2*). The type is used to group the variations, what is illustrated with a bar graph (Matplotlib package).

A.

Chrom	Pos	Ref	Alt	Type	Protein Location	Ref.AA	Alt.AA
Pf3D7_10_v3	5432	G	A	Non-Coding	NA	NA	NA
Pf3D7_11_v3	5432	G	A	Synonymous	301	A	A
Pf3D7_11_v3	5432	G	A	Non-Synonymous	301	A	T

B.


Is the variation in the coding region?
YES
NO
Does it result in the change of codon?
YES
NO

Chrom	Pos	Ref	Alt	Type	Protein Location	Ref.AA	Alt.AA
Pf3D7_10_v3	5432	G	A	Non-Synonymous	301	A	T

Chrom	Pos	Ref	Alt	Type	Protein Location	Ref.AA	Alt.AA
Pf3D7_10_v3	5432	G	A	Synonymous	301	A	A

Chrom	Pos	Ref	Alt	Type	Protein Location	Ref.AA	Alt.AA
Pf3D7_10_v3	5432	G	A	Non-Coding	NA	NA	NA

Figure2. The tab-separated output with gathered data on per variation. A. *Three possible outputs for the same variation on exemplary chromosome Pf3D7_10_v3 (imaginary data). The logic of filling the table is illustrated on* ***Fig.2B****. If the change is within the non-coding region, it is inputted as the type. With no protein produced, the columns referring to the protein location and amino acids are filled with "NA". If the change of base between reference (col. "Ref") and the observed sequence (col. "alt") results in the change of the amino acid (Alt.AA vs Ref.AA), the type is synonymous.*

Secondly, the log file contains messages from all errors and warnings, as well as information on the file name of the input files, number of variation recordings with quality below 20 Phred excluded from the output and the path directory of the output.

Packages

Script uses **BioPython Seq module** to manipulate sequences and translate them into protein, **PyVCF** to handle VCF file, **gffUtils** and **SQlite3** to create read GFF file as a database, **Matplotlb** to create a graph, **Arg-Parse** to use the command line arguments and **logging** for error handling.

Script pipeline

- ### Assembling transcript sequence

Firstly, the list of variations is filtered based on the quality of their reading with a threshold of 20. The number of readings below the threshold is noted and they are excluded from further analysis. The chromosomal position of each of the remaining readings is compared against the GFF database, checking whether it matches any of the locations of the coding regions (***Fig.3***). If so, the transcripts for that region are identified. The remaining participating CDS are located to construct a transcript sequence. The transcript is then translated to a protein and compared between reference and the reading.

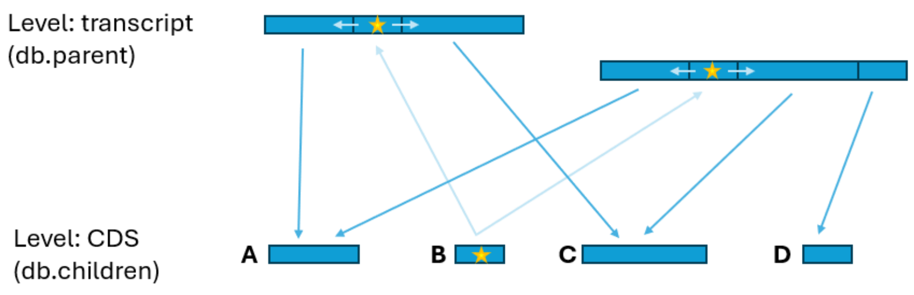


Figure 3. Visual representation of the recursive relationship between the CDS and transcript. *Each CDS is part of at least one transcript. After a variation is detected within the CDS (not yet mRNA), it is matched with its corresponding transcripts and the remaining compound coding regions are identified to find the protein sequence. In the illustrated example, the change is identified within CDS B, which is part of two transcripts: one consisting of A,B and C and the other - of A, B, C and D.*

- Differential treatment of the positive and negative strands for calculating the protein location

In the positive strand, the position of the change within the protein is calculated by summing the lengths of the proceeding CDS and the distance between the start of the affected CDS and the change position from the VCF file. However, for the negative strands, the order of the CDS is reversed. Thus, the position of the change within the protein is calculated by adding all proceeding CDS (as seen from right to left) and the distance between the end of the affected CDS and change position.

- Handling pseudogene transcripts and pseudogenes

The script is focused on the protein focussing genes, so the pseudogenes are handled with an exception that is noted in the log file: "{record.id} is not bound to a mRNA, but to a pseudogene and will not be included in the final tsv output".

Known limitations

The script is able to handle the substitutions and insertions, but it does not have a clause for the deletion, if it is inputted in a different way than an empty string instead of the base letter. Handling of the pseudogenes could be built into an optional argument, to limit the number of warnings that is returned.

The biggest limitation of the script is the dysfunctional part calculating the position of the base change on the negative strand - I have not managed to fix it within the time constrains. Even though I follow the same logic as for the positive strands, reversing the CDS order, adding the preceding CDS and the distance between the base change and the end of the CDS – the base at my calculated position within the transcript is not the same as it should be based on the “Ref” value from the VCF file.