HerediCaRe_VCF_Upload

The target of this program is to parse any VCF file to get a text file with variants informations ready to upload to the HerediCaRe database.
The programm expect valid files corresponding to the newest VCF specification v4.3

Usage

```
C:\HerediCaRe_VCF_Upload>java -jar HerediCaRe_VCF_Upload.jar -h

usage: ParseVCF [-h] [-o <output>] [-sp <snpEff>] [-genome <database>]
                [-jp <java>] [-d <debug>] [-ram <value>] [-t <transcript>]
                <Input>

Parse VCF to txt for database upload.

positional arguments:
  <Input>                Folder with VCF files to parse

named arguments:
  -h, --help             show this help message and exit
  -o <output>            Output  folder  for  final  .txt  files  (default:
                         Output)
  -sp <snpEff>           Path    to    snpEff    JAR    file    (default:
                         resources\snpEff\snpEff.jar)
  -genome <database>     snpEff database  name  (default:  GRCh38.mane.1.0.
                         refseq)
  -jp <java>             Path to the java executable.  Note that Java v>=12
                         is required to run snpEff! (default: java)
  -d <debug>             Debug  Folder.  Contains  processed,  erroneous  &
                         normalized   VCFs   +   Rejected   Variants(.tsv)
                         (default: Debug)
  -ram <value>           Accessible  RAM  (GB)  for  java  virtual  machine
                         (default: 3)
  -t <transcript>        Transcript  file  (default:  resources\transcript.
                         tsv)
```

Example:

`java -jar HerediCaRe_VCF_Upload.jar myInputFolder`

Requirements

JAVA

For the execution of this program java is mandatory. As snpEff required java 12 is is recommended to use at least this version or other newest versions (current tests were done with JDK 17 for windows)

Folder structure

The file and folder names must be respected (excepted *myFolder*) for automatic execution without optional arguments.

```
myFolder\
|   HerediCaRe_VCF_Upload.jar
|
|__ resources\
    |   hg19ToHg38.over.chain
    |   hg38ToHg19.over.chain
    |   transcript.tsv
    |
    |__ ChromFa\
    |   |
    |   |__ hg19\
    |   |      chr1.fa
    |   |      ...
    |   |
    |   |__ hg38\
    |           chr1.fa
    |           ...
    |
    |__ snpEff\
        |   snpEff.config
        |   snpEff.jar
        |
        |__ data\
            |
            |__ GRCh38.mane.1.0.refseq\
                    sequence.1.bin
                    ...
```

snpEff

- Download snpEff (current version 5.1)
- Unzip file and store in *resources* as shown here
- (Optionally) download the snpEff database (Currently "GRCh38.mane.1.0.refseq") needed to the annotation of the VCFs. If not downloaded, snpEff will do it automatically during the first run.

Command to download the snpEff database:
`java -jar resources\snpEff\snpEff.jar download GRCh38.mane.1.0.refseq`

Transcript

A transcript file can be submitted to the program, either throw the programm argument *-t* or stored in the *ressources* folder (see).
The file must be filled as a 3 columns tab seperated values (tsv).

Example:

Gene	TranscriptID	MANE Select
BRCA1	NM_007294	1
SMARCA4	NM_001387283	0

As with the outcome of MANE, which is recommended by the VUS task force, some variants can't be represented only by the main transcript set MANE SELECT, it is also important to take the MANE plus clinical set into consideration (ex. SMARCA4) while **priorizing** the MANE SELECT. So the third column will contain **1** if the transcript is MANE SELECT and **0** if it is MANE plus clinical.

LiftOver Chain

To convert the genomic positions from a reference genome to another it is necessary to have two chain files.

- Download hg19ToHg38.over.chain
- Download hg38ToHg19.over.chain
- Unzip and save in *ressource* (see)

Reference genome FASTA

For the normalization step it is necessary to have the reference genome stored chromosome by chromosome(chr1-chr22, chrX, chrY)

- Download hg19 reference
- Download hg38 reference
- Unzip and save in *ressource* (see)

Workflow

- Set default values & Handle Arguments with argparse
- check the *resources* folder (see)
- create a hashmap with the transcript file (see)
- create debug & output directory (Debug, Output)
- For every vcf file in the Input folder:
  - Parse & save filename informations: reference, patientID, mguNr, emplID, timeStamp
  - Read and save the vcf content [1]
  - Normalize and write the output into *Debug*.
  - LiftOver corresponding to the given reference in the input file
  - Run snpEff with variants corresponding to the hg38 reference genome
  - create a file for rejected Variants in the *Debug\rejected Variants* folder.
  - For each line in snpEff output:
    - If the Variant don't have a snpEff annotation, report it with the failure `#No snpEff Annotation`
    - For each snpEff annotation:
      - if the transcriptID is available in the transcript file retrieve more informations like gene, effect, HGVSc, HGVSp, CLASS [2], Genotype, POS, REF, ALT (for both references hg19 & hg38) and save [3]
      - report variants with invalid transcriptID as failure `#Invalid TranscriptID`

        - report variants with class *artefact* or *false reference* with failure `#Invalid CLASS`
    - write the output file (.txt) with the retained variants
    - Move succefully processed files to the *Debug\Processed* folderor erroneous to *Debug\Error*

[1] Mitochondrial variants are ignored & prefix *chr* are removed (chr1 --> 1)
[2] in the following order: first "MutDB:Classification", after "CLASS", then "MT"
[3] In case of a simultanous annotation with MANE select and MANE plus clinical, Mane select is prioritized and the other one is rejected