VaRank Manual

Version 1.4
VaRank is a program for genetic Variant Ranking from NGS data

Please feel free to contact us for any suggestions or bug reports
email: veronique.geoffroy@inserm.fr; jeanmuller@unistra.fr

==

TABLE OF CONTENTS

==

==

1. INTRODUCTION

===============

VaRank is a program designed for variant ranking from next generation sequencing data. It provides a comprehensive workflow for annotating and ranking SNVs and indels.

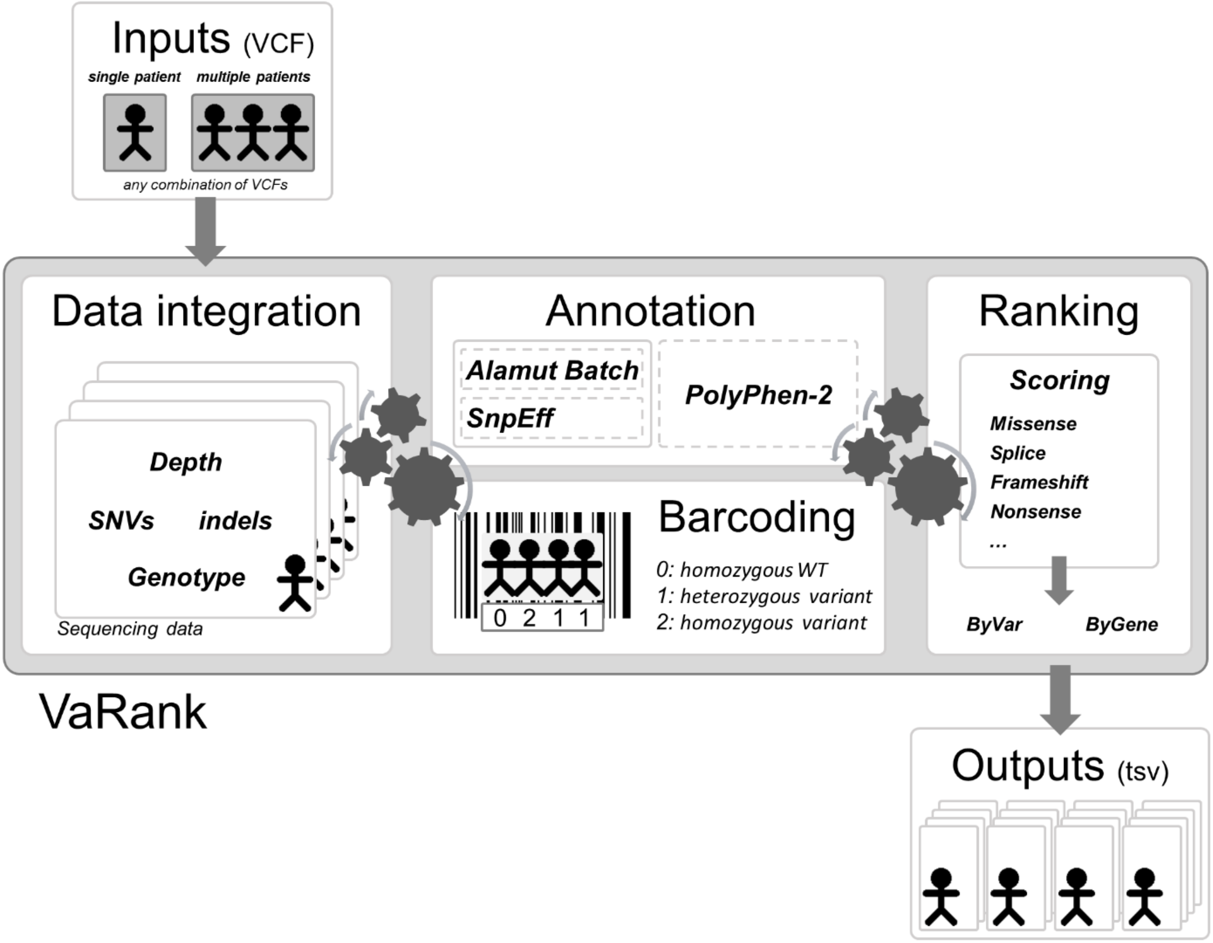
Four modules create the strength of this workflow:

(i) Integration of the sequencing data: variant call quality summary (total and variant depth of coverage, phred like information), to filter out false positive calls.

(ii) Alamut Batch or SnpEff annotations, to integrate genetic and predictive information (functional impact, putative effects in the protein coding regions, population frequency...) from different sources, using HGVS nomenclature.

(iii) Barcode representing the presence/absence of variants (with homozygote/heterozygote status), to search for recurrence between families or group of individuals.

(iv) Prioritization score to rank variants according to their predicted pathogenic status.



2. INSTALLATION/REQUIREMENTS

=============================

The VaRank program is written in the Tcl/Tk language. Modern Unix systems have this scripting language already installed (otherwise it can be downloaded from http://www.tcl.tk/).This tool is composed of different other programs and databases:

- VaRank sources can be downloaded from http://www.lbgi.fr/VaRank under the GNU GPL license.

VaRank sources installation

The source .tar.gz should be extracted and uncompressed to any directory.

```
mkdir VaRank
cd VaRank
wget http://www.lbgi.fr/VaRank/Sources/VaRank_latest_core.tar.gz
tar -xvf VaRank_latest_core.tar.gz
```

The installation requires simply to set the following environment variables:
- $VARANK : VaRank installation directory

Depending on the selected annotation engine:
- $ALAMUT : Alamut Batch installation directory
- $SNPEFF : SnpEff and SnpSift installation directory

- The annotation engine could be from either:
  - Alamut Batch developed and commercialized by Interactive Biosoftware (Rouen, France). If you do not own a license, a 30-day free trial can be requested here (http://www.interactive-biosoftware.com/request-trial-alamut/).
  - SnpEff and SnpSift developed by Pablo Cingolani (http://snpeff.sourceforge.net/).

VaRank version	Compatibility	Release Date
1.4	Alamut Batch 1.6-1.9 SnpEff V4.2-4.3	2018/04/20
1.3.5	Alamut Batch 1.5.0-1.5.2 SnpEff V4.2	2016/08/04
1.3.3->1.3.4	Alamut Batch 1.4.3-1.4.4 SnpEff V4.2	2016/01/18->2016/03/11
1.3->1.3.2	Alamut Batch 1.4.2 SnpEff V4.1L	2015/11/20->2015/12/18
1.2.3->1.2.5	Alamut Batch 1.4.0 SnpEff V4.1a/V4.1b	2015/02/26->2015/07/24
1.1 -> 1.2.2	Alamut Batch 1.3 SnpEff V4.1	2014/11/12->2015/02/02
1.0	Alamut Batch 1.2	2014/07/27

Optional:
- PolyPhen-2 (PPH2) provides prediction of functional effects of human nsSNPs (Adzhubei IA *et al* Nature Methods 2010). It needs to be locally installed to be used. You can freely download it from http://genetics.bwh.harvard.edu/pph2/dokuwiki/downloads.
- Protein databases can be used to connect to PPH2 by helping extracting the protein sequences, checking the amino acid change to be tested and then submit this sequence if no precomputed data exists. UniProt and RefSeq can respectively be downloaded using the following procedures and should be placed in the “pph2Databases” directory.

- <u>UniProt:</u>

The human reference protein file can be downloaded and renamed using the following commands:

```
cd $VARANK/pph2Databases
wget
ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/reference_p
roteomes/Eukaryota/UP000005640_9606_fasta.gz
mv UP000005640_9606.fasta.gz HUMAN.fasta.gz
```

- <u>RefSeq</u>

Please run these commands to download and prepare the data for VaRank and copy them in the the "pph2Databases" directory:

```
cd $VARANK/pph2Databases
wget -rnd -A 'human.*.protein.faa.gz'
ftp://ftp.ncbi.nlm.nih.gov/refseq/H_sapiens/mRNA_Prot/
foreach file (*.protein.faa.gz)
echo "reading $file"
cat $file >> human.protein.faa.gz
chmod 755 $file
rm $file
end
```

Alamut Batch installation

Reagrding Alamut Batch, we recommend using the standalone version which is very handy to install with a single tar.gz file and a single database file, and more efficient for the annotation. The first use of Alamut Batch requires the user license agreement, <u>we recommend to do it right after the installation</u>.

SnpEff and SnpSift installation

The installation of SnpEff and SnpSift is also well described on its website. Apart from the programs additional databases should be downloaded (the human reference genome, dbSNP, dbNFSP, phastCons). Make sure you are running the required java version (i.e. SnpEff 4.3 requires Java 1.8). You should check the following sections and download the corresponding files:

```
cd /path/to/SnpEff/dir/
wget http://sourceforge.net/projects/snpeff/files/snpEff_latest_core.zip
unzip snpEff_latest_core.zip
cd SnpEff

#Define the SNPEFF environment variable
setenv SNPEFF /path/to/SnpEff/dir/snpEff/
mkdir Test_VaRank
#Copy one vcf example file from the $SNPEFF/examples directory
cp examples/test.vcf VaRank

#To check if the SnpEff is running properly
#In this example the '-v GRCh37.75' will automatically download the human reference
#genome if not yet downloaded. You can change the value to match another version.
java -Xmx4g -jar $SNPEFF/snpEff.jar eff -c $SNPEFF/snpEff.config -v GRCh37.75
$SNPEFF/Test_VaRank/test.vcf > $SNPEFF/Test_VaRank/file.eff.vcf
#No error message validate this step, one can still have a look at the output file

#To check if the SnpSift is running properly
java -Xmx4g -jar $SNPEFF/SnpSift.jar varType $SNPEFF/Test_VaRank/file.eff.vcf >
$SNPEFF/Test_VaRank/file.eff.varType.vcf
#No error message validate this step, one can still have a look at the output file
```

- <u>Additional databases</u>

The followings commands and links will guide you through the installation of SnpEff:
http://snpeff.sourceforge.net/SnpEff_manual.html
http://snpeff.sourceforge.net/SnpSift.html#VariantType

To install **dbSNP** (http://snpeff.sourceforge.net/SnpSift.html#annotate)

```
cd $SNPEFF
mkdir -p db/dbSNP
cd db/dbSNP
```

To download dbSNP in VCF format (GRCh37 coordinates):

```
wget ftp://ftp.ncbi.nlm.nih.gov/snp/organisms/human_9606_b146_GRCh37p13/VCF/00-
All.vcf.gz
```

To download dbSNP in VCF format (GRCh38 coordinates):

```
wget ftp://ftp.ncbi.nih.gov/snp/organisms/human_9606/VCF/00-All.vcf.gz

gunzip 00-All.vcf.gz
#To check if SnpSift is properly installed using dbSNP data
java -Xmx4g -jar $SNPEFF/SnpSift.jar annotate $SNPEFF/db/dbSNP/00-All.vcf
$SNPEFF/Test_VaRank/file.eff.varType.vcf>
$SNPEFF/Test_VaRank/file.eff.varType.dnsnp.vcf
#No error message validate this step, one can still have a look at the output file
```

To install **dbNSFP** (http://snpeff.sourceforge.net/SnpSift.html#dbNSFP)

```
cd $SNPEFF
mkdir -p db/dbNSFP
cd db/dbNSFP
```

Download the files from SnpEff's site (remember that you need both the database and the index file).

```
#To check if SnpSift is properly using dbNSFP data
java -Xmx4g -jar $SNPEFF/SnpSift.jar dbnsfp -db $SNPEFF/db/dbNSFP/”XXX”
$SNPEFF/Test_VaRank/file.eff.varType.vcf >
$SNPEFF/Test_VaRank/file.eff.varType.dbnsfp.vcf
#No error message validate this step, one can still have a look at the output file
```

To install **phastCons** (http://snpeff.sourceforge.net/SnpSift.html#phastCons)

```
cd $SNPEFF
mkdir -p db/phastCons/
cd db/phastCons
foreach c (1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 M X Y)
wget
http://hgdownload.soe.ucsc.edu/goldenPath/hg19/phastCons100way/hg19.100way.phastCons/
chr$c.phastCons100way.wigFix.gz
end
wget http://snpeff.sourceforge.net/genome.fai
cd $SNPEFF
#To check if SnpSift is properly using phastCons data
java -Xmx4g -jar $SNPEFF/SnpSift.jar phastCons $SNPEFF/db/phastCons
$SNPEFF/Test_VaRank/file.eff.varType.vcf>
$SNPEFF/Test_VaRank/file.eff.varType.phastCons.vcf
#No error message validate this step, one can still have a look at the output file
```

SnpEff annotation of some multiple alleles (at the same position), especially deletions, is not clear enough, thus these variations are split into multiple variant/lines while creating the non-redundant input vcf files. During this process the genotype are stored in memory but modified in the novel vcf files to “0/1” by default (VcfDirectory/SnpEff/Input). These specific vcf files should then not be used.

While running VaRank with SnpEff some parameters should be used including: -snpeffHumanDB, -dbSNP, -dbNSFP and -phastConsDB (see section 6 for more details).

The following environment variable is optional:

- $PPH : PolyPhen-2 installation directory

By default the VaRank installation directory looks like this:

```
VaRank                          #the program installation directory
 |
 |----- bin/                    #where an alias is set to the main .tcl script
 |
 |----- ExtAnn/                 #used to add external annotations
 |
 |----- pph2Databases/          #where to store the UniProt and RefSeq fasta files
 |
 |----- SnpEffTests/            #contains an example to check SnpEff execution
 |
 |----- sources/                #where the source .tcl files are stored
 |
 |----- changeLogs.txt          #description of VaRank changes
 |
 |----- configfile              #a configfile example that can be copied to any analysis directory
 |                              #for modification purpose
 |
 |----- License.txt             #GNU GPL license
 |
 |----- README.VaRank.*.pdf     #this file
```

To make sure the Tcl script finds correctly the Tcl interpreter, the best way is to put, as the first line of the main script (which is already done in VaRank-main.tcl), the following command:

```
#!/usr/bin/env tclsh
```

It can also be changed to any other path like:

```
#!/usr/local/ActiveTcl/bin tclsh
```

Typically, you can create an alias of the main Tcl script “sources/VaRank-main.tcl” for example to “VaRank”, place it in the “/bin” directory”(this is done be default already) and add the path to this in your $PATH.

3. INPUT

=======

VaRank supports the commonly used VCF (Variant Call Format, https://github.com/samtools/hts-specs) input format for variants analysis that allows the program to be easily integrated into NGS bioinformatics analysis pipelines.

Since version 1.3, VaRank is compatible with the VCF version 4.2 specification (26 Jan 2015). One major addition is the use of the ‘*’ allele. The ‘*’ is used to indicate that one allele is missing due to a upstream deletion. This implies that the variant calling is aware of false homozygous status for variants in *trans* of deletions. As in the following example, the variant genotype is now properly annotated in the VCF version 4.2:

```
#CHROM   POS   ID   REF  ALT  ...  FORMAT     Sample1
1        172   .    A    G    ...  GT:AD:DP   1/1:0,20:20     (VCF up to version 4.1)
1        172   .    A    G,*  ...  GT:AD:DP   1/2:0,20,9:29   (VCF since version 4.2)
```

Limitations of the VCF support in VaRank:

- Analysis of structural variants (ALT=<ID=type,Description=description>) not supported.
- Genotype phase/unphase status (‘|’ vs ’/’). Phase not analysed.

In order not to miss any variant and given that structural variations are not always reported and not analyzed by VaRank, we have decided to systematically report a variant for each '*'. Nevertheless, given the unknown positions of the deletion no score is attributed to the variant.

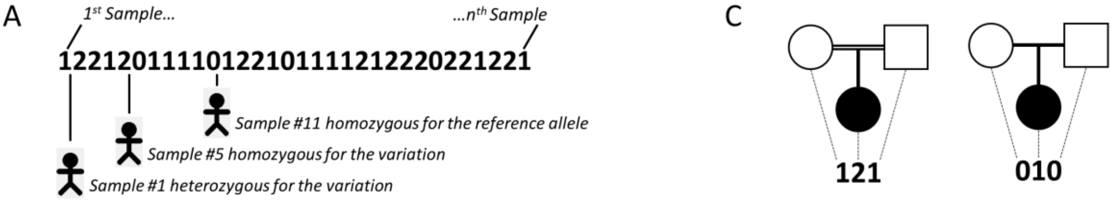
Depending on the VCF file origin (different variant callers), the variant read depth information can be stored in many different fields (NR, AD, AC, AO…). Moreover in certain situation these fields can be present together (merged VCF file). We have therefore decided to select the first information (not empty) available within the following ordered fields: NR, AD, AC, AO and DV.
Variant that do not have a depth of coverage (AD="." or AD="NA") and wild type variants (GT="0/0") are not considered (i.e. not annotated) by VaRank and will thus absent from the output files.

It is to notice that Gzip VCF files are supported.

VaRank takes also several argument as options to the command line that are detailed in section 6 ("USAGE / OPTIONS"). The different arguments can be passed either on the command line or using a specific file named "configfile" that needs to be put in the same directory as the input VCF files. An example of configfile is provided in the VaRank installation directory.

a. Family Barcode

The barcode in VaRank allows a quick overview of the presence/absence status of each variant and their zygosity status within the analyzed individuals ("0" representing homozygous wild type, "1" heterozygous and "2" homozygous for the variant, see the figure below Panel A). Panel B displays 3 variants example and 32 patients analyzed together. Together with the barcode, simple counts on the individuals (homozygous, heterozygous and total allelic counts) are also available as well as an estimate of the allele frequency in the user cohort.


A
1st Sample…
…nth Sample
12212011110122101111212220221221
Sample #11 homozygous for the reference allele
Sample #5 homozygous for the variation
Sample #1 heterozygous for the variation
C
121
010

B

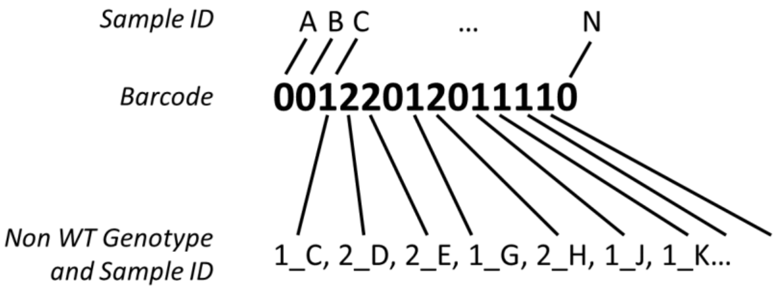
Gene	Chr	Start	Ref	Mut	Zygosity	TotalRead Depth	VarRead Depth	cNomen	pNomen	familyBarcode	Barcode	#Hom	#Het	#Allele	#Sample
BBS2	16	56548501	C	T	hom	142	142	c.209G>A	p.=	**222**	**22222222202222222222222222222222**	31	0	62	32
ALMS1	2	73716993	-	C	hom	143	126	c.7911dup	p.Asn2638Glnfs*24	**121**	**00000000000000000000000000200000**	1	0	2	32
TTC21B	2	166797646	C	T	het	144	80	c.601G>A	p.Val201Met	**011**	**12211011112122101111212220121221**	12	17	41	32

Together with the main barcode describing all the patients analyzed together in one VaRank run, one can define a second barcode. This second barcode named "*familyBarcode*" can be configured by the user to group selected samples (e.g. trios where affected child and parents could be analyzed together). This can be done using the configfile by simply grouping sample names together. As an example, 2 families where the fam1 corresponds to a trio sequencing (proband and parents, see Fig. C) and fam2 with 2 affected child:

fam1: Sample1 Sample2 Sample3
fam2: Sample4 Sample5

Grouping sample names together allows also to follow the same naming convention for the files with the same prefix (fam1_ for all family members).

In order to have more details about which samples are different from the reference within the barcode (values 1 for heterozygote and 2 for homozygote alternative base), the “samVa” output column (for Samples containing genotyped Variant) gives access to the list of the 10 first “Sample ID” and their associated status that are different from the reference.


Sample ID
A B C
...
N
Barcode
00122012011110
Non WT Genotype and Sample ID
1_C, 2_D, 2_E, 1_G, 2_H, 1_J, 1_K...

As an example, this allow users to quickly know in which samples a single variation belongs. This can be useful in case of rare variants and potential pathogenic mutations.

b. External Gene annotation

In order to further enrich the annotation for each variant and each gene, VaRank can integrate (using the option -extann) external annotations imported from a tab separated values file into the output files (gzip files are supported). The file format is easy and should look like this: the first line is a header including a column entitled "genes" that should be the first column too.

The following example (provided in the VaRank sources) has been set to provide annotation for the gene including the MIM number, the phenotypes as well as the transmission mode of the gene (here AR means “autosomic recessive”, AD means “autosomic dominant”).

genes	Mim Number	Phenotypes	Inheritance
ACTG2	102545	Visceral myopathy, 155310 (3)	AD
ADAMTSL2	612277	Geleophysic dysplasia 1, 231050 (3)	AR

4. OUTPUT
=========

VaRank provides 4 .tsv (TAB separated values) output files divided into 2 categories:

-Files named with "**ByVar**" contains variations sorted from the most to the least pathogenic (according to the VaRank score.

In some cases, one variation can be annotated using several genes. This happens when overlapping genes exists. VaRank selects the most pathogenic annotation and thus the first gene described is the one corresponding to this situation. VaRank keep all the other gene names. In the following example, *BBS1* is the gene retained for the annotation of the considered variation but as indicated in the column “Gene” there is a second overlapping gene (e.g. *DPP3*):
11_66277969_C_T   BBS1/DPP3

-Files named with "**ByGene**" contains variations classified by gene ("ByGene") where the list is sorted using the gene as a proxy to the score. Each gene is scored according to most pathogenic variant (homozygous) or the first two most pathogenic variants. In order to make sure that no variants are missed all gene variation are reported also below the variant(s) used to score the gene. This file is more suitable when dealing with a recessive mode of inheritance.
It is to notice that given the focus on genes in those output files, variants that could be attributed to several genes are duplicated and associated to each gene individually.

A part from these 2 categories, each file is also available in 2 versions:

-Raw file ("**allVariants**") with no variants filtered out.
-Already prefiltered files ("**filteredVariants**") with variants filtered out using the following criteria:

- with a total depth of coverage <=10x
- with a supporting reads count <=10x
- with a percent of supporting reads <=15%
- with validated annotation in the dbSNP database (i.e. at least with 2 evidences from the ClinVar field) that are not pathogenic (from the ClinicalSignificance field in dbSNP and from ClinVar)
- with an allele frequency >1% (extracted from the dbSNP, 1000Genomes, gnomAD…)

Remarks:
-A high false positive variant calls has been observed for indels in the EVS dataset, leading to unexpectedly highly frequent (>1%) frameshifts in gene known to cause rare diseases. Therefore, we have decided to remove the EVS dataset from the outputs starting from version 1.4 and from the automatic filtering steps.

-Since the minor allele frequency described in the dbSNP database (MAF, see http://www.ncbi.nlm.nih.gov/projects/SNP/docs/rs_attributes.html) does not necessary represent the variation that is being observed in the vcf file being annotated but rather one at the position, we only filter on this if the allele is the same.

-Using the gnomAD database, when available we only filter variation when marked “PASS” in the “gnomadFilter” field.

-The “filtered” files can be considered as very stringent filtering step to ensure a very quick first analysis of the data. Users can always modify the default parameters.

The output organization can be described as follows:

```
VcfDirectory
 |
 |-- Alamut/                                      #Contains all Alamut Batch related files
 |    |--AlamutInputFile_all.txt                  #Alamut input file generated from the vcf(s) files
 |    |--AlamutAnnotations_all.txt                #Alamut output file with annotated variants
 |    |--AlamutUnnanotated_all.txt                #Alamut output file with unannotated variants
 |    |--AlamutOutput_all.txt                     #Alamut log file
 |
 |-- SnpEff/                                      #Contains all SnpEff and SnpSift related files
 |    |-- Input/
 |         |--*.vcf                               #Non redundant variant input vcf files
 |    |-- Output/
 |    |--*.varType.vcf.log                        #SnpSift varType annotation log
 |    |--*.varType.dbsnp.vcf.log                  #SnpSift dbSNP annotation log
 |    |--*.varType.dbsnp.dbnsfp.vcf.log           #SnpSift dbsnfp annotation log
 |    |--*.varType.dbsnp.dbnsfp.phastCons.vcf.log #SnpSift phastCons annotation log
 |    |--*.varType.dbsnp.dbnsfp.phastCons.vcf     #Final annotation file
 |
 |-- PPH2/                                        #(option) Contains all PolyPhen-2 related files
 |    |-- PPH2input_all.txt                       #PPH2 input file
 |    |-- PPH2features_all.txt                    #PPH2 output file
 |    |-- PPH2humvar_all.txt                      #PPH2 output file
 |    |-- PPH2errors_all.txt                      #PPH2 log file
 |
 |----- fam#_SampleName_allVariants.rankingByVar.tsv
 |----- fam#_SampleName_filteredVariants.rankingByVar.tsv
 |----- fam#_SampleName_allVariants.rankingByGene.tsv
 |----- fam#_SampleName_filteredVariants.rankingByGene.tsv
 |
 |----- fam#_SampleName_statistics.tsv            #Short counts report (e.g. homozygous, heterozygous
 |                                                #and total counts) for each of the variant categories
 |
 |----- SNV_global_statistics.tsv                 #Contains the same counts as defined for each patient
 |                                                #but for the whole analyzed cohort
 |
 |----- VCF_Coordinates_Conversion.tsv            #List of VariantID and corresponding VCF positions
 |
 |----- configfile                                #To define sample group and set running options
 |
 |----- *InputFile.vcf or *.vcf.gz                #Input files
 |
```

a. VariantID

The output files contains a columns named *VariantID* which is a variation identifier meant to be unique. The format is described as follows:

[#chr]_[genomicposition]_[RefBase]_[VarBase]

[RefBase] being the nucleotide sequence in the reference genome
[VarBase] being the alternate nucleotide sequence.

Ex1: 16_56548501_C_T describes the change of C to T on chromosome 16 at position 56548501.

In order to optimize the description of this identifier for larger indels, the [RefBase] and [VarBase] values are restricted to 50bp by default.
Ex2:21_9448722_330bp_- describes the deletion on chromosome 21 of 330pb.

In case of redundancy (e.g. insertion of different sequences at the same position of the same size) in order to keep non-redundant identifiers a versioning is applied.
Ex3:21_9448722_-_89bp and 21_9448722_-_89bp.1 correspond to the insertion of 2 different sequences of the same length on chromosome 21.

The "VCF_Coordinates_Conversion.tsv" is a tab separated output file containing for each *VariantID* the corresponding VCF positions ([#chr] [genomicposition] [RefBase] [VarBase]).

b. Absence of annotations

It is to notice that when no annotation is available for a specific column, the empty value is set to "NA". Exception is made for several numerical columns (including the frequency, read depth or count related columns: *rsMAF, alleleFrequency, 1000g_AF, gnomadAltFreq_all, gnomadReadDepth, gnomadHomCount…*) where "-1" is used that allows the user to further filter information without losing data.

5. SCORING
==========

VaRank uses the variation type (i.e. substitution, deletion, insertion, duplication) and the coding effect to score. The VaRank scoring is categorized from the most likely to the less likely pathogenic state as follows (score into parenthesis): known mutation (110), stop gain (100), frameshift (100), essential splice site (2 first bases before and after the exon) (90), start loss (80), stop loss (80), intron-exon boundary (donor site is -3 to +6, acceptor site -12 to +2) (70), missense (50), splice site creation (40), strong or weak splice site activation (40, 35), in-frame (30), deep intronic changes (25), synonymous coding (10), exonic or intronic changes (2) and 3'/5' UTR (1). Each category is further described in the USAGE/OPTIONS section and each score can be changed.

Each specific variant score is further adjusted using additional information. For this, variants are assessed at the genomic level (phastCons) and at the protein level (SIFT and if installed PolyPhen-2), and an adjustment score (0 or +5) is added to the relevant category. The adjustment score can be changed by the user.
To ensure the best use of SIFT predictions, the deleterious status is only taken if the SIFT median value is comprised between [2.75-3.5].

Scores in bold reflect score values after the adjustment score is applied. 1/ Each variant score is adjusted (+5) if high conservation at the genomic level is observed (phastCons cutoff >0.95). 2/ Missense scores are adjusted (+5) for each deleterious prediction (SIFT and/or PPH2).

Variant Category	Option name	VaRank Score	Description
Known mutation	*S_Known*	110	Known mutation as annotated by HGMD and/or dbSNP (rsClinicalSignificance or clinVarClinSignifs="pathogenic/likely pathogenic")
Stop gain	*S_StopGain*[1]	100, **105**	A single-base substitution in DNA resulting in a STOP codon (TGA, TAA or TAG).
Frameshift	*S_Fs*	100	Exonic insertion/deletion of a non-multiple of 3bp resulting often in a premature stop in the reading frame of the gene.
Essential splice site	*S_EssentialSplice*[1]	90, **95**	Variation in one of the canonical splice sites resulting in a significant effect on splicing.
Start loss	*S_StartLoss*[1]	80, **85**	Variation leading to the loss of the initiation codon (Met).
Stop loss	*S_StopLoss*[1]	80, **85**	Variation leading to the loss of the STOP codon.
Intron-exon boundary	*S_CloseSplice*[1]	70, **75**	Variation outside of the canonical/essential splice sites (donor site is -3 to -1, +3 to +6, acceptor site is -12 to +2).
Missense	*S_Missense*[1,2]	50, **55, 60, 65**	A single-base substitution in DNA not resulting in a change in the amino acid.
Local Splice Effect	*S_LSEstrong* *S_LSEweak*	40 35	LocalSpliceEffect field is used to score splice site creation (40, based on New Donor Site, New Acceptor Site), strong splice site activation (40, based Cryptic Donor Strongly Activated, Cryptic Acceptor Strongly Activated) or weak splice site activation (35, based on Cryptic Donor Weakly Activated, Cryptic Acceptor Weakly Activated).
Indel in-frame	*S_Inframe*	30	Exonic insertion/deletion of a multiple of 3bp.
Deep intron-exon boundary	*S_DeepSplice*[1]	25, **30**	Intronic variation resulting in a significant effect on splicing.
Synonymous coding	*S_Synonymous*[1]	10, **15**	A single-base substitution in DNA not resulting in a change in the amino acid.
Exonic or intronic	*S_ExonIntron*	2	Exonic or intronic variant without any predicted effect
3'/5' UTR	*S_UTR*	1	3'/5' UTR variant without any predicted effect

6. USAGE / OPTIONS

==================

A tutorial together with examples are available on the website to further describe the use of VaRank. To run VaRank, the default command line is the following:

```
$VARANK/bin/VaRank -vcfdir '/Path/To/The/Directory/Containing/vcf/files' >& log.log &
```

The command line can be completed by the list of options described below or modified in the configfile. To show the options simply type:

```
$VARANK/bin/VaRank –help or $VARANK/bin/VaRank
```

OPTIONS:

-help More information on the arguments.

-vcfDir Path of your study directory containing your vcf input file.

-vcfInfo To extract the info column from the .vcf file and insert the data in the output file (last columns).
Range values: yes or no (default)

-vcfFields “List of fields”. To extract selected fields from the VCF “INFO” column and insert these data in the output files (last columns). Only if -vcfInfo is set to “yes”.

-rsfromvcf To extract the rsID and validation status from the .vcf file and insert this in the output files.
Range values: yes or no (default)

-Homstatus To force the determination of the homozygous or heterozygous state of one variation. If set to yes it will use the Homcutoff value to decide.
Range values: yes or no (default)

-Homcutoff To determine the homozygous or heterozygous state of one variation. If set to some value it will force to reconsider the data provided.
Range values: [0,100] default: 80 (active only if Homstatus=yes or when no status is given)

-MEScutoff MaxEntScan cutoff, to determine the impact of the variant on splicing. Expressed as the % difference between the variant and the WT score.
Range values: [-100,0], default: -15

-SSFcutoff Splice Site Finder cutoff, to determine the impact of the variant on splicing. Expressed as the % difference between the variant and the WT score.
Range values: [-100,0], default: -5

-NNScutoff NNSplice cutoff, to determine the impact of the variant on splicing. Expressed as the % difference between the variant and the WT score.
Range values: [-100,0], default: -10

-phastConsCutoff To determine when a genomic position is conserved or not. Above the cutoff is considered as conserved.
Range values: [0,1], default: 0.95

-readFilter Minimum number of reads for the variants.
Range values: [0,-], default: 10

-depthFilter Minimum depth for the variants.
Range values: [0,-], default: 10

-readPercentFilter Minimum percent of variant reads for considering a variant.
Range values: [0,100], default: 15

-freqFilter Filtering variants based on their MAF in the SNV databases (dbsnp, 1000Genome, EVS, gnomAD, …).
Range values: [0.0,1.0], default: 0.01

-rsFilter	Filtering variants on the SNP information.
Values: removeNonPathoRS (remove variants without "probable-pathogenic" or "pathogenic" annotation, see clinical significance field in dbSNP website. Filtering only for variants with at least 2 validations).
none = keep all variants, no filtering on rsID.
Default: removeNonPathoRS

-extann	Tab separated file containing annotation to add to the final output files. Restrictions for the format are: first line is a header, first column is the gene name. Typical use would be a gene file containing specific annotations such as transmission mode, disease, expression...

-metrics	Changing numerical values from frequencies to us or fr metrics (e.g. 0.2 or 0,2).
Range values: us (default) or fr

-alamutHumanDB	Alamut Batch specific option to select the reference human genome version.

-javaPath	To explicitly set the Java path.

-snpeffHumanDB	SnpEff specific option to select the reference human genome version.
Ex: “GRCh37.75”

-dbSNP	SnpEff specific option to describe the full path to the location of the dbSNPvcf file used by SnpSift.
Ex: “$SNPEFF/db/dbSNP.2015-01-09_00-All.vcf"

-dbNSFP	SnpEff specific option to describe the full path to the location of the dbNSFPvcf file used by SnpSift.
Ex: "$SNPEFF/db/dbNSFP/dbNSFP2.4.txt.gz"

-phastConsDB	SnpEff specific option to describe the full path to the location phastConsdirectory used by SnpSift.
Ex: “$SNPEFF/db/phastCons"

-uniprot	Name of the UniProt sequence file (optional, only use if PPH2 is installed).
Ex: HUMAN.fasta.gz (default)

-refseq	Name of the RefSeq sequence file (optional, only use if PPH2 is installed).
Ex: human.protein.faa.gz (default)

-hgmdUser	HGMD User login (optional, only use if you have an HGMD license).
-hgmdPasswd	HGMD User password (optional, only use if you have an HGMD license).

-proxyUser	Proxy User login.
-proxyPasswd	Proxy User password (optional, default is “password”).
-proxyServer	The name of the proxy.
-proxyPort	The port used by the proxy to communicate (default is “8080”).

-SamVa	To add a new column with the sample ID of the 10 first non homozygous WT variants, with their homozygote/heterozygote status.
Range values: yes or no (default)

-SamOut	A “Sample list”. To write output files only for selected samples.
Ex: “sample1 sample2”.

- AlamutProcesses	#processes (Alamut batch Standalone version only). Alamut Annotation jobs are split among multiple processes on the same computer.
Range values: Integer (default=0, no multi-process)

- AlamutAlltrans	Annotate variants on all transcripts with Alamut (rather than on the longest transcript).
Range values: yes (default) or no

The following options are provided to adapt the scoring scheme to the users:

-S_Known	Known mutation as annotated by HGMD and/or dbSNP (rsClinicalSignificance or clinVarClinSignifs="pathogenic/probable-pathogenic").

Default: 110

-S_StopGain
A single-base substitution in DNA resulting in a STOP codon (TGA, TAA or TAG).
default: 100

-S_Fs
Exonic insertion/deletion of a non-multiple of 3bp resulting often in a premature stop in the reading frame of the gene.
default: 100

-S_EssentialSplice
Mutation in one of the canonical splice sites resulting in a significant effect on splicing (at least 2 out of the 3 programs indicate a relative variation in their score compared to the wild type sequence).
default: 90

-S_StartLoss
Mutation leading to the loss of the initiation codon (Met).
default: 80

-S_StopLoss
Mutation leading to the loss of the STOP codon.
default: 80

-S_CloseSplice
Mutation outside of the canonical splice sites (donor site is -3 to +6', acceptor site -12 to +2) resulting in a significant effect on splicing (at least 2 out of the 3 programs indicate a relative variation in their score compared to the wild type sequence).
default: 70

-S_Missense
A single-base substitution in DNA not resulting in a change in the amino acid.
default: 50

-S_LSEstrong
Strong local splice effect (splice site creation or strong activation).
default: 40

-S_LSEweak
Weak local splice activation.
default: 35

-S_Inframe
Exonic insertion/deletion of a multiple of 3bp.
default: 30

-S_DeepSplice
Intronic mutation resulting in a significant effect on splicing (at least 2 out of the 3 programs indicate a relative variation in their score compared to the wild type sequence).
default: 25

-S_Synonymous
A single-base substitution in DNA not resulting in a change in the amino acid.
default: 10

-B_phastCons
Each variant score is adjusted if a conservation at the genomic level is observed (PhastCons cutoff >0.95).
default: 5

-S_ExonIntron
Exonic or intronic variations without any predicted effect
default: 2

-S_UTR
3'/5' UTR variant without any predicted effect
default: 1

-B_SIFT
Missenses scores are adjusted for each SIFT deleterious prediction.
default: 5

-B_PPH2
Missenses scores are adjusted for each PPH2 deleterious prediction status.
default: 5

7. Annotations columns available in the output files

==

In the following table, we describe the annotations that are available in the VaRank output files. It is to notice that, since VaRank can be configured using 2 different annotation engines, in some cases specific annotations are only present while using one annotations engine and in some cases the values for the same type of information are slightly different.

Moreover, in order to add more flexibility to the output files, the user can select the annotation columns and their order using the configfile.

Column name	Annotation	Alamut Batch	SnpEff
variantID	Variant identifier [#chr]_[genomicposition]_[RefBase]_[VarBase]	X	X
gene	Gene symbol	X	X
geneDesc	Gene description	X	
omimId	OMIM® id	X	
transcript	RefSeq transcript id	X	
strand	Transcript strand	X	
transLen	Length of transcript (full cDNA length)	X	
cdsLen	CDS length	X	
chr	Chromosome of variant	X	X
start	Start position of variant	X	X
end	End position of variant	X	X
ref	Nucleotide sequence in the reference genome (restricted to 50bp)	X	X
alt	Alternate nucleotide sequence (restricted to 50bp)	X	X
protein	Protein ID (NCBI)	X	
Uniprot	UniProt ID	X	X
assembly	Genome assembly	X	
posAA	Amino acid position	X	
wtAA_1	Reference codon	X	X
wtCodon	WT codon	X	X
varAA_1	Alternate codon	X	X
zygosity	Homozygote or heterozygote status	X	X
totalReadDepth	Total number of reads covering the position	X	X
varReadDepth	Number of reads supporting the variant	X	X
readsVariationPercent	Percent of reads supporting variant over those supporting reference sequence/base	X	X
QUALphred	**QUAL:** The Phred scaled probability that a REF/ALT polymorphism exists at this site given sequencing data. Because the Phred scale is -10*log(1-p), a value of 10 indicates a 1 in 10 chance of error, while a 100 indicates a 1 in 10^10 chance. These values can grow very large when a large amount of NGS data is used for variant calling.	X	X
varType	Variant Type (substitution, deletion, insertion, duplication, delins)	X	X
codingEffect	Variant Coding effect (synonymous, missense, stop gain, in-frame, frameshift, start loss, stop loss)	X	X
VarLocation	Variant location (upstream, 5'UTR, exon, intron, 3'UTR, downstream)	X	X
exon	Exon number (nearest exon if intronic variant)	X	X
intron	Intron number	X	X
gDNAstart	gDNA start	X	
gDNAend	gDNA end	X	

gNomen	Genomic-level nomenclature	X	
cDNAstart	cDNA start	X	
cDNAend	cDNA end	X	
cNomen	cDNA-level nomenclature	X	X
pNomen	Protein-level nomenclature	X	X
clinVarIds	List of ClinVar Ids separated by "|"	X	
clinVarOrigins	List of ClinVar origins separated by "|", values: germline, somatic, de novo, maternal…	X	
clinVarMethods	List of ClinVar methods separated by "|", values: clinical testing, research, literature only…	X	
clinVarClinSignifs	List of ClinVar clinical significances separated by "|"	X	
clinVarReviewStatus	List of ClinVar reviews separated by "|", number of stars (0-4)	X	
clinVarPhenotypes	List of ClinVar phenotypes Ids separated by "|"	X	
rsID	dbSNP variation	X	X
rsValidation	dbSNP validated status	X	
rsValidationNumber	dbSNP number of validation categories	X	
rsClinicalSignificance	dbSNP variation clinical significance	X	
rsAncestralAllele	dbSNP ancestral allele	X	
rsHeterozygosity	dbSNP variation average heterozygosity	X	
rsMAF	dbSNP variation global Minor Allele	X	
rsMAFAllele	dbSNP variation global minor allele	X	
rsMAFCount	dbSNP variation sample size	X	
1000g_AF	1000 genomes global allele frequency	X	
1000g_AFR_AF	1000 genomes allele frequency in African population	X	
1000g_SAS_AF	1000 genomes allele frequency in South Asian population	X	
1000g_EAS_AF	1000 genomes allele frequency in East Asian population	X	
1000g_EUR_AF	1000 genomes allele frequency in European population	X	
1000g_AMR_AF	1000 genomes allele frequency in American population	X	
gnomadAltFreq_[2]	gnomAD alternate allele frequency [2]	X	
gnomadHomCount_[2]	gnomAD homozygous genotype count [2]	X	
gnomadHetCount_[2]	gnomAD heterozygous genotype count [2]	X	
gnomadHemCount_[2]	gnomAD hemizygous genotype count [2]	X	
gnomadFilter	gnomAD VCF filter value (AC0, PASS, RF…)	X	
gnomadReadDepth	gnomAD read depth	X	
gnomadOrigin	gnomAD variant origin (Exomes, genomes, or both)	X	
deltaMaxEntSscorePercent	% difference between the splice score of variant with the score of the reference base	X	
wtMaxEntScore	WT seq. MaxEntScan score	X	
varMaxEntScore	Variant seq. MaxEntScan score	X	
deltaSSFscorePercent	% difference between the splice score of variant with the score of the reference base	X	
wtSSFScore	WT seq. SpliceSiteFinder score	X	
varSSFScore	Variant seq. SpliceSiteFinder score	X	
deltaNNSscorePercent	% difference between the splice score of variant with the score of the reference base	X	
wtNNSScore	WT seq. NNSPLICE score	X	
varNNSScore	Variant seq. NNSPLICE score	X	
distNearestSS	Distance to Nearest splice site	X	

nearestSSChange	Nearest splice site change	X	
nearestSSType	Nearest splice site type	X	
localSpliceEffect	Splicing effect in variation vicinity (New donor Site, New Acceptor Site, Cryptic Donor Strongly Activated, Cryptic Donor Weakly Activated, Cryptic Acceptor Strongly Activated, Cryptic Acceptor Weakly Activated)	X	
localSpliceSiteAnnotation	Splice site annotation: -"essential splice donor": variant in 5' SS at intronic position +1 or +2 -"essential splice acceptor": variant in 3' SS at intronic position -1 or -2 -"close splice donor": variant in 5' SS at position -3 to -1, +3 to +6 -"close splice acceptor": variant in 3' SS at position -12 to -2, and 0 to +2	X	
localSS_pos	Genomic position of predicted new splice site or activated cryptic site	X	
localSS_wtMaxEntScore	WT seq. MaxEntScan score used for local splice effect detection	X	
localSS_wtNNSScore	WT seq. NNSPLICE score used for local splice effect detection	X	
localSS_wtHSFScore	WT seq. HSF score used for local splice effect detection	X	
localSS_varMaxEntScore	Variant seq. MaxEntScan score used for local splice effect detection	X	
localSS_varNNSScore	Variant seq. NNSPLICE score used for local splice effect detection	X	
localSS_varHSFScore	Variant seq. HSF score used for local splice effect detection	X	
branchPointPos	Genomic position of affected putative branch point	X	
branchPointChange	Change between WT and variant BP scores (range -5 to 100)	X	
proteinDomain1	Protein domain 1	X	
proteinDomain2	Protein domain 2	X	
proteinDomain3	Protein domain 3	X	
proteinDomain4	Protein domain 4	X	
SIFTprediction	SIFT prediction	X	X
SIFTweight	SIFT score ranges from 0 to 1. The amino acid substitution is predicted damaging is the score is <= 0.05, and tolerated if the score is > 0.05.	X	
SIFTmedian	SIFT median ranges from 0 to 4.32. This is used to measure the diversity of the sequences used for prediction. A warning will occur if this is greater than 3.25 because this indicates that the prediction was based on closely related sequences. The number should be between 2.75 and 3.5	X	
PPH2pred	PolyPhen-2 prediction using HumVar model are either "neutral, possibly damaging, probably damaging" or "neutral, deleterious" depending on the annotation engine.	X[1]	
PPH2class			X
phyloP	phyloP score	X	
PhastCons	phastCons score	X	X
GranthamDist	Grantham distance	X	
AGVGDclass	AlignGVGD class	X	
AGVGDgv	AlignGVGD: variation (GV)	X	
AGVGDgd	AlignGVGD: deviation (GD)	X	
vaRank_VarScore	Prioritization score according to VaRank	X	X
annotationAnalysis	Indicates the annotation status (value: Yes or No)	X	X
avgTotalDepth	Total read depth average at the variant position for all samples analyzed that have the variation	X	X
sdTotalDepth	Standard deviation associated with Avg_TotalDepth	X	X
countTotalDepth	Number of samples considered for the average total read depth	X	X
avgVariantDepth	Variation read depth average at the variant position for all samples analyzed that have the variation	X	X
sdVariantDepth	Standard deviation associated with Avg_SNVDepth	X	X
countVariantDepth	Number of samples considered for the average SNV read depth	X	X
familyBarcode	Homozygote or heterozygote status for the sample of interest and its associated samples	X	X
barcode	Homozygote or heterozygote status for all sample analyzed together (Hom: 2 ; Het: 1; Sample name is	X	X

	given at the first line of the file: ## Barcode)		
homCount	Number of homozygote over all samples analyzed together	X	X
hetCount	Number of heterozygote over all samples analyzed together	X	X
alleleCount	Number of alleles supporting the variant	X	X
sampleCount	Total number of samples	X	X
alleleFrequency	Allele frequency in all samples analyzed (with 4 decimals)	X	X
samVa	Sample ID of the 10 first non WT variants, with their homozygote/heterozygote status	X	X
Annotation_Impact	Estimation of putative impact / deleteriousness : {HIGH, MODERATE, LOW, MODIFIER}		X
Gene_ID	Gene identifier		X
Feature_Type	Which type of feature is in the next field (e.g. transcript, motif, miRNA, …)		X
Feature_ID	Transcript ID (preferably using version number), Motif ID, miRNA, ...)		X
Transcript_BioType	The bare minimum is at least a description on whether the transcript is {"Coding", "Noncoding"}		X
cDNA.pos	Position in cDNA		X
cDNA.length	Transcript's cDNA length		X
CDS.pos	Position of coding bases		X
CDS.length	Number of coding bases		X
AA.pos	Position of AA		X
AA.length	Number of AA		X
Distance	Up/Downstream: Distance to first / last codon Intergenic: Distance to closest gene Distance to closest Intron boundary in exon (+/- up/downstream)		X
LOF	Loss of function prediction		X
NMD	Nonsense-mediated decay prediction		X
dbNSFP_1000Gp1_...	1000 genomes allele frequency in ... population		X
dbNSFP_CADD_phred	CADD phred score		X
dbNSFP_ExAC_..._AC	ExAC allele count in ... population		X
dbNSFP_ExAC_..._AF	ExAC allele frequency in ... population		X
dbNSFP_FATHMM_pred	FATHMM prediction scores		X
dbNSFP_GERP___NR	GERP conservation NR scores		X
dbNSFP_GERP___RS	GERP conservation RS scores		X
dbNSFP_LRT_pred	LRT conservation scores		X
dbNSFP_MetaSVM_pred	dbNSFP scores based on 10 component scores (SIFT, PolyPhen-2 HDIV and HVAR, GERP++, MutationTaster, Mutation Assessor, FATHMM, LRT, SiPhy, PhyloP) and the maximum frequency observed (1000 genomes)		X
dbNSFP_MutationAssessor_pred	MutationAssessor prediction scores		X
dbNSFP_MutationTaster_pred	MutationTaster prediction scores		X
dbNSFP_PROVEAN_pred	Provean prediction scores		X
dbNSFP_Polyphen2_HDIV_pred	PolyPhen-2 HDIV prediction scores		X
dbNSFP_Polyphen2_HVAR_pred	PolyPhen-2 HVAR prediction scores		X
dbNSFP_SIFT_pred	SIFT prediction scores		X

[1] if PPH2 is installed separately.

[2] all = All populations, afr = African, amr = Latino, asj = Ashkenazy Jewish, eas = East Asian, sas = South Asian, nfe = Non-Finnish European, fin = Finnish European, oth = Other populations, popmax = Maximum Allele Frequency across populations (excluding OTH)

When –vcfinfo is set to "yes", all the vcf annotations are reported in separate columns after the last columns described here.

8. FAQ
======

Q: How to cite VaRank in your work?
A: If you are using VaRank, please cite our work using the following reference:

Geoffroy V.*, Pizot C.*, Redin C., Piton A., Vasli N., Stoetzel C., Blavier A., Laporte J. and Muller J.
VaRank: a simple and powerful tool for ranking genetic variants.
PeerJ. 2015. (10.7717/peerj.796)

Q: What are the WARNINGs that VaRank mention while running?
A: VaRank writes to the standard output the progress of the analysis including warnings about issues or missing information that can be either blocking or simply informative. More specifically while loading the VCF file(s) specific information are under survey such as VCF format consistency, patient redundancy, the total and variant read depth, the genotype, the indels. Any surveyed default will be reported to the user.

Q: I want to run a VaRank analysis again, what shall I do?
A: Simply remove all output files (*.tsv) and type the new command line. All annotations will be kept and the analysis should be done very quickly.

Q: I have already computed 5 samples in my analysis and I want to add 10 more, what should I do?
A: Considering no updated version of any annotation source or VaRank available, you can simply add the new vcf files to the already computed ones, remove all output files (*.tsv) and rerun VaRank. VaRank will only recompute the missing annotations and will save you the computation time of reannotating multiple times the same variants.

Q: How are the variant homozygous or heterozygous status reported?
A: VaRank trust by default the zygosity status provided by the vcf and report this in the column "*Zigosity*" in the output files. Nonetheless, in the case when no data is provided but total and variant depth of coverage is available, VaRank recompute this by applying the simple rule everything >= Homcutoff (default 80% see options) is homozygous and the rest is heterozygous. In order to clearly show difference with other variants those recomputed will be noted "hom?" or "het?". The same rule is applied when using the option "-Homstatus" except that variant are noted "hom" or "het".

Q: In the output files, some values are set to "NA"?
A: When for a specific type of annotation no information is available then the empty value is set to "NA" (e.g. Not Available). Exception is made for several numerical columns (including *rsMAF, gnomAD frequencies…*) where "-1" is used that allows the user to further filter information without losing data.

Q: When PolyPhen-2 prediction are running?
A: Depending on the annotation engine PPH2 either needs to be installed separately (Alamut Batch) or is already integrated (SnpEff). Nevertheless one can still have SnpEff installed and a local installation of PPH2. If the 2 programs are installed and properly setup for the use in VaRank, despite the fact that SnpEff annotations might already contain PPH2 predictions, the local PPH2 installation will be used. If this is not your intention simply unset PPH2 environment variable

Q: If SNPEFF and ALAMUT environment variables are both set, what annotation engine is running?
A: By default, the Alamut annotation engine is used.

Q: I do have a proxy, how can deal with this?
A: if you use Alamut Batch, 4 parameters can be used to allow the program to connect to the internet via the proxy. Depending on your setup (white list or not), you can add the servers name (proxyServer), the port (proxyPort), the user (proxyUser) and corresponding password (proxyPasswd). Please be aware that the command line will be launched with these informations. If your configuration allows the use of only you user ID, the password will be set by default to the following value “default”.

Q: Why can we have several genes in the annotation of one variation?
In some cases, one variation can be annotated using several genes. This happens when overlapping genes exist. VaRank selects the most pathogenic annotation and thus the first gene described is the one corresponding to this situation. VaRank keep all the other gene names. In the following example, BBS1 is the gene in consideration for the annotation of the considered variation but as indicated in the column “Gene” there is a second gene that is DPP3:11_66277969_C_T BBS1/DPP3