

## About

- Programs
- Input, Output
- Options
- Tutorial

## Input files

- Common format
- Required
  - . Variant File
- Provided
  - . Annotation databases
  - . Gene database
  - . Protein domain
  - . Ancestral variant
  - . Pre-annotated variant
  - . Pre-annotated site
  - . Pre-annotated region

## Programs

- vPHASE
- vMNP
- vAnnGene
- vAnnDel
- vAnnBase
- vAnnDomain

## Topics

- Gene database

## 1. About

### 1.1. Programs

[\[Back to top\]](#)

This software package includes the following command-line programs:

- vPHASE -- gets phase data from ShapeIt output and rewrites a VCF file with phase
- vMNP -- merges multiple nucleotide polymorphisms if they are in complete linkage disequilibrium
- vAnnDel -- annotates deleteriousness scores (PolyPhen, SIFT, MetaLR, MetaSVM, and so on)
- vAnnGene -- annotates functional consequence
- vAnnBase -- annotates from a database containing scores for all possible SNVs in the entire genome

It is important to protect the privacy of participants in a research. Therefore, this software suite is not a web-based application, but downloadable programs that run in your local computer. No information will be collected, stored, or sent out from your computer. Once installed, it does not retrieve any data from the Internet except for version checking.

The version of this software package is represented by a version number and a build date. If it is a beta version, the version number is followed by the word "beta". To help you avoid program bugs, beta release always checks for new versions through the Internet. Please note that they do not send out any information; they only retrieve the latest version number from this website.

### 1.2. Input, output and Unix pipe

[\[Back to top\]](#)

These programs read a Genotype File from the standard input, analyze and modify it, then write to the standard output. Therefore, you can use the Unix pipe ("|") to run multiple programs sequentially. Most programs also accept a Genotype File filename in the command line (any string that is not recognized as a program option or argument), so that they provide the flexibility as to where to start a chain of analysis. At the end of the chain, you can redirect the standard output to a file ("| gzip -c > results.gz") or request the program to directly write to a file (-o results.gz). Unix pipe can save time and disk space. However, if it is your first time to run these programs, it is better to execute one program at a time and redirect the intermediate output to a file. That way it is easier to locate a problem if there's any.

### 1.3. Options

[\[Back to top\]](#)

This User Manual tells the purpose of each program option. For more details about their usage, such as the number of arguments, data types, and default values, please use the (-h) or (--help) option to print the program help text. For the general rules in the usage of program options, please be referred to the Manual page of VICTOR.

### 1.4. Tutorial

[\[Back to top\]](#)

Please see the Tutorial page of VICTOR.

## 2. Input Files

[\[Back to top\]](#)

### 2.1. Common format

Unless otherwise stated, input files contain columns divided by a space or a tab. Multiple successive delimiters are not treated as one. Lines starting with a '#' are comments and will be ignored. The reading of input files is robust to Linux/Mac/Windows/mixed line breaks. It is also robust to the missing of a line break after the last line. Programs will stop reading a file at the first blank line. The input file does not need to be uncompressed if it is a .gz or .bz2 because the programs can read them directly. You don't even need to type ".gz" or ".bz2" in the command, as the programs will first look for the uncompressed file, then file.gz, followed by file.bz2. Below are the descriptions of each input file. You can also search for "Xxxxx File" in this manual to find all related information about each file type.

## 2.2. Basic input files

[\[Back to top\]](#)

A **Variant File** for annotation is a VCF file, with or without the genotype columns.

## 2.3. Other input files that are already provided

[\[Back to top\]](#)

An **Annotation File** is a database file for variant annotation by vAnnDel. The information for annotation could be deleteriousness scores, dbSNP IDs, allele frequency, etc. The format of this file is:

- 1) Lines starting with ## are comments;
- 2) The first row is the header row. Contents of this row will be used as headers in the output;
- 3) The first 4 columns must be #CHROM, POS, REF, and ALT;
- 4) There is no limit to the number of columns;
- 5) Variants with multiple alternative alleles are split into separate lines;
- 6) Lines are sorted by #CHROM, POS, REF and ALT;
- 7) The file is compressed by bgzip and indexed by tabix.

A **Gene File** is a gene database file that will be used by vAnnGene. The format is the same as the refGene.txt.gz file from genome.ucsc.edu.

A **Domain File** contains the definition of protein domains. It has 4 columns for transcript ID, start position, end position, and domain name. It should have a header row. The headers of the 2nd & 3rd columns could start with "AA" for amino acid positions, or "CDS" for nucleotide positions relative to translation start site, or "RNA" for nucleotide positions relative to transcription start site, or "DNA" for genomic nucleotide positions.

An **Ancestral Variant File** contains ancestral variants. The format is similar to the first 5 columns of a VCF file, except that the ID columns should be the HGVS r. or n. nomenclature of the variant.

A **Pre-Annotated Variant File** contains pre-annotated variants. It has 6 columns for CHR, POS, REF, ALT, Score and GeneSymbol, respectively. Header row is allowed. Positions are 1-based. To save disk space, a blank field represents the value in the previous row.

A **Pre-Annotated Site File** contains pre-annotated locus. It has 5 columns with a header row of #TxID, xxx\_begin, xxx\_end, Annotation, and Score, respectively. Here xxx is "DNA" for genomic location, "RNA" for nucleotide position relative to the transcription start site, "CDS" for nucleotide position relative to the translation start site, "AA" for amino acid position, "UTR3" for nucleotide position relative to the begin of UTR3, or "DOWN" for nucleotide position relative to the transcription end site. All positions are 1-based and should be positive. For "RNA", negative position is allowed, which means upstream to the transcription start site.

A **Pre-Annotated Region File** is similar to the Pre-Annotated Site File, except that it has only 4 columns, and the content of the 4th column should be one of the pre-defined strings, which include histone\_acetylation, histone\_methylation, histone\_binding, ChIP\_seq, TF\_binding, open\_chromatin, methylation, acetylation, and acylation.

A **VKS File** contains variants of known significance (as oppose to VUS, variants of unknown significance). This file has 9 columns: #CHROM,POS,REF,ALT,Symbol,Type,HGVS,BayesDel,ClinSig. An HGVS example is NM\_000546:c.188C>T:p.A63V. ClinSig should be either 1 for pathogenic or 0 for benign. Type is the functional consequence of the variant. It should start with "Missense" for missense changes. BayesDel is BayesDel score.

## 3. Programs

### 3.1. vPHASE

[\[Back to top\]](#)

vPHASE reads two files: the .phased.haps file output from ShapeIt and a [Genotype File](#). It will integrate the phase information into the genotypes, then output a modified [Genotype File](#).

### 3.2. vMNP

[\[Back to top\]](#)

vPHASE merges consecutive variants into a multiple nucleotide polymorphism (MNP) if they are in complete linkage disequilibrium. It reads a phased VCF file; outputs a modified VCF file.

### 3.3. vAnnGene

[\[Back to top\]](#)

vAnnGene determines the functional consequence of a genetic variant, i.e., whether it is synonymous, missense, stop-gain, stop-loss, splice altering, or located in a miRNA binding site or transcription factor binding site, etc. It reads a VCF-like genotype file and write a modified file with 3 extra columns, Func\_Gene, Func\_Type, and Func\_Detail, corresponding to gene symbol, functional consequence type, and functional consequence details, respectively.

By default, this program annotates against a modified RefSeq gene database. This database includes only the most frequently occurring transcript for each gene, whenever such information is available. For the genes that different transcripts are expressed in different organs, all transcripts are included. The annotated functional consequence of a variant for each gene is the most severe one among all transcripts. The order of consequence from the most benign to the most severe is Intergenic, Intronic, Synonymous, Downstream, Upstream, UTR3, UTR5, miRNA-Bind, Missense, In-frame, StopLoss, StopGain, Frameshift, SpliceAltering, No-translation, and No-transcription. Optionally, you can annotate against canonical transcripts only (--canonical), which is the transcript that has the longest CDS or the longest RNA (when all transcripts have the same CDS length or the gene is a non-coding RNA gene) for each gene.

You can pre-annotate a set of variants and store the results in a database file. Later, when you annotate a VCF file using this program, it can match the variants with those in the database and annotate them with the correct information (--pav, the acronym for Pre-Annotated Variants). An example is splice site variants. This package is shipped with a database of splice site variants extracted from dbNSFP. In dbNSFP, all possible SNVs within -3 to +8 at the 5' splice site or -12 to +2 at the 3' splice site were analyzed by a Random Forest model to predict splice-altering. Only those with a score greater than or equal to 0.6 were extracted. The default argument of (--pav) uses this database to annotate splice site variants. This should be more accurate than simply calling all variant within these regions a splice site variant. In addition, variants within 2 bp in an intron (--splice-5e, --splice-5i, --splice-3i, --splice-3e) are always annotated as splice site variants, no matter whether they are in the dbNSFP. The (--pav) option can be used multiple times to annotate different functional consequence types.

Besides pre-annotated variants, you can also define pre-annotated sites (--pas). A site is a location relative to a gene transcription, such as UTR3. An example is miRNA binding sites. The default argument of (--pas) uses the TargetScan version 7.1 to annotate miRNA binding sites. The format of the argument for (--pas) is the same as that for (--pav), while the format of the file is different. Please see the help text for the file format. In short, this file has a header row to indicate the site location type, which could be relative to genomic coordinate, transcription start site, translation start site, UTR3 start site, or transcription end site. Same as (--pav), the (--pas) option can be used multiple times to annotate different functional consequence types.

In addition, this program can annotate regulatory regions (--par). The default argument uses the ENSEMBL's Regulatory Build to annotate transcription factor binding sites (--par=ensembl\_mf.bed), which was extracted from ENSEMBLE's MotifFeatures.gff.gz. The Regulatory Build has another file (AnnotatedFeatures.gff.gz) that contains ChIPseq regions, open chromatin regions, acetylation sites, acylation sites, methylation sites, histone binding sites, and histone modification sites. Please build your own (--par) file to annotate these regions for the cell types that are relevant to your study.

Intronic, Synonymous, Upstream, Downstream, UTR5, UTR3 variants are filtered out in the output (--filter), except for those affecting translational efficiency, affecting splicing (--pav), overlapping a regulatory region (--par), overlapping a miRNA binding site (--pas). Ancestral and Intergenic variants are filtered out anyway. Use the (--filter) option to change the variant types to be filtered. Some variants will be omitted from annotation and output. The reasons could be 1) REF doesn't match with the reference genome, or 2) either REF or ALT has

an N. The program will report these issues to the standard error output. Structural variations are not annotated by default (--sv) because the vSVA program annotates them at the time of performing an association analysis.

If an InDel has different 3' or 5' representations, vAnnGene chooses the most biologically relevant one for annotation and writes the corresponding results to Func\_Type and Func\_Detail, while the alternative representation is within brackets in the Func\_Detail field. In addition, the POS, REF, ALT columns in the output will be changed based on the biological relevance. This is a convenient feature for subsequent deleteriousness annotation, because most annotation programs do not consider an alternative representations (such as CADD), thus using the biological irrelevant representation will result in a wrong deleteriousness score. It has been argued to left-normalized an InDel so that they can be easily matched, but a correct deleteriousness prediction is far more important than variant matching. In addition, VICTOR already provides a solution for variant matching even when InDels are not always left-normalized, which is to create variant databases that contain both the 3' and 5' most representations for each InDel. The MaxAF database provided by VICTOR is already created in this way.

In the column Func\_Type, loss-of-function (LoF) variants will be labeled "(LoF)", for example, "SpliceSite(LoF)". This text, "LoF", is useful for the PERCH:vDEL program to assign deleteriousness scores, where LoF variants will have the highest score among all possible non-synonymous variants. In this program, LoF variants are defined as 1) StopGain or FrameShift variants, excluding those affecting only the last 5% of CDS; 2) SpliceSites variants that involve coding regions; 3) variants that inhibit transcription or translation. If a variant is a SpliceSite in one transcript but not in another functional transcript of the same gene, this variant is not a LoF. Besides LoF, variants that affect translational efficiency will be labeled "(TrEff)". StopGain variants that may cause non-sense mediated decay will be labeled "(NMD)". Ancestral variants are filtered out in the output. But if they were not filtered, they will be labeled "(Ancestral)". If a variant match with a VKS (variant of known significance), it will be labeled "(knClinSig=#.\$)", where # is '1' for pathogenic and '0' for benign, \$ is 'a' for genomic sequence match, 'b' for protein sequence match, and 'c' for missense of the same amino acid as a VKS but with a different substitution, for which the BayesDel score is higher than the minimum score among pathogenic VKS or lower than the maximum score among benign VKS.

If a variant has functional consequences in multiple overlapping genes, the variant will be written in multiple lines, where one line belongs to one gene and lines are sorted by gene symbol. This feature is recommended because it helps the downstream analysis by PERCH. vAnnGene also annotates Grantham score, Blosum62 score, exon number, and whether the variant is inside the last exon, in the INFO field only when this feature is turned on. Optionally, you could turned it off by (--no-split), which will also demands that lines be sorted by CHR and POS. Please note that line order may still be different from the input because POS may change for some InDels.

By default three columns named Func\_Gene, Func\_Type, and Func\_Detail will be added to the output file. The program can also write the annotations to INFO instead of the additional columns (--add-info). In INFO, Annotations are comma-divided strings for gene symbol, functional type and details, respectively. If there are annotations for more than one transcript or gene, they will be listed one after another starting from the most severe consequence, i.e., "vAnnGene=gene1,function1,detail1,gene2,function2,detail2".

The default input file for this program is VCF. Sometimes people have special data format that is not exactly VCF. This program can reformat into VCF. A supported format include data about chromosome, start and end position in 1-based basepair, and an alternative allele. Reference allele is not provided because it can be obtained from the position and is deemed redundant. It is hard to convert this format into VCF manually since it involves getting the reference allele. To use vAnnGene to convert this format, you still need to make a "VCF" file that POS is either a 1-based basepair or a 1-based range, and REF is always the string FromPOS. You don't need a special option to let vAnnGene do the format conversion. vAnnGene identify this format by the string FromPOS and does the conversion automatically.

### 3.4. vAnnDel [\[Back to top\]](#)

This program annotates a Genotype File with information obtained from an Annotation File (--ann). If there are more than one columns in the Annotation File to be added to the Genotype File, then the program will add new columns to the Genotype File. If there is only one column for annotation, it will either add a new column, or update an existing column, or add a sub-field to INFO (--add-info). In updating annotations, the annotations for unmatched variants will be kept unchanged. The columns in the Annotation File to be added to the Genotype File can be specified by the option (-f). By default all columns starting from the 5th are for annotations.

Variants are matched by the fields #CHROM, POS, REF, and ALT. It is required that both the [Genotype File](#) and the [Annotation File](#) are sorted by #CHROM, POS, REF, and ALT, and have been left-normalized. Most VCF files are already sorted by #CHROM and POS, and this program is robust to a small proportion of unordered lines, so it will run properly in most cases even if lines are not sorted by the four columns. Nevertheless, this program also provides an option (`--use-tabix`) for the situation when the input file is completely scrambled. This option forces the program to search for variants in the [Annotation File](#) by tabix, which may be slow for a large input file. So I hope you don't have to use it. On the other hand, if your input file has only a few hundreds variants, this option would make the annotation faster.

The most common usage of this program is to annotate deleteriousness scores, such as PolyPhen, SIFT, MetaLR, MetaSVM, CADD, etc. By default the program annotates deleteriousness scores in dbNSFP version 3.3a (`--ann=nsfp33a`). However, the file nsfp33a is not provided anymore because it is no longer necessary to annotate these scores due to the vAnnBase program. To annotate other information you need the (`--ann`) option, for which the argument is the path to the [Annotation File](#). The ".gz" at the end may be omitted. The prefix can be omitted too if the file is located in the default data folder. For example, `--ann=~/local/VICTOR/data/hg19/snp147.gz` can be written as `--ann=snp147`. This option makes the program to annotate dbSNP147 rs#. Note that the annotation will be written in the ID column of a VCF file because the 5th column of the file snp147.gz is named "ID". The option (`--missing`) determines what to annotate when a variant has no match in the [Annotation File](#). "`--missing=index`" has a special meaning, which refers to Chr\_Pos\_Ref\_Alt. This is useful when applied together with `--ann=snp147`, so that the ID column will contain either rs# or Chr\_Pos\_Ref\_Alt.

This software package includes an [Annotation File](#) with MaxAF calculated from the 1000 Genomes phase III v5b, UK10K whole genome sequencing cohorts (ALSPAC and TWINSUK) v20131101, and the gnomAD r2.0.1, and Genomes of the Netherlands GoNL (MaxAF\_gUGN.gz). In making this file, variants were removed if Hardy-Weinberg Equilibrium test yielded a p-value<0.000001, the FILTER field is not PASS, the VQSLOD is <-5.368 for SNVs and <-4.208 for InDels, or the VQSLOD is missing. Allele frequencies were calculated only when AD is >=200. You can annotate this MaxAF by `--ann=MaxAF_gUGN`.

### 3.5. vAnnBase

[\[Back to top\]](#)

This program is very fast in annotating numerical scores at the cost of a lower resolution. The idea is to store a double-precision number in a fixed length of bytes (default is 1) in a file so that it is easy to find the annotation for a chromosomal position. VICTOR provides pre-computed BayesDel scores and uses this program to annotate variants. This is straightforward for single-nucleotide variants (SNV). vAnnBase also has a function to annotate InDels too. It will take the maximum or minimum scores (`--indel=max` or `--indel=min`) among all possible SNVs in the affected nucleotides with a padding of 10 basepairs (`--padding`). If the annotated score is BayesDel it can also integrate MaxAF (`--maxaf-pr`).

### 3.6. vAnnDomain

[\[Back to top\]](#)

This program annotates protein domains for a [Variant File](#). Domain definitions are obtained from a [Domain File](#) (`--ann`). If [Domain File](#) name (argument for `--ann`) contains the string "@GDB", the string will be replaced with either "refGene" or "ensGene", depending on the gene database used by the program (`--gdb`).

## 4. Topics

### 4.1. Gene database

[\[Back to top\]](#)

The package provides database files for the NCBI RefSeq genes and the Ensembl genes, each of which has two versions -- the Lite and the Full. The Lite version includes only the principle transcripts for a gene whenever such information is available, and excludes read-through transcripts and pseudogenes. The default is refGeneLite, which is recommended for a clean annotation of variants. On the other hand, the ensGeneLite includes more transcripts and is more suitable for an exploratory research. These files were slightly modified from genome.ucsc.edu's refGene.txt or ensGene.txt by replacing the first column with the official gene symbol.