

Use the VEP online to analyse your variants through a simple point-and-click interface.

The web interface allows you to access the key features of the VEP without using the command line. Interactively filter your results to find the data you want. Download your results in multiple data formats, easily share your results with others, and integrate your variation data with the powerful Ensembl web browser.

If you use the VEP in your work, please cite **McLaren et. al.** ([doi:10.1093/bioinformatics/btq330](https://doi.org/10.1093/bioinformatics/btq330))

Any questions? Send an email to the Ensembl developer's mailing list, dev@ensembl.org or contact the Ensembl Helpdesk at helpdesk@ensembl.org.



Documentation contents

 [Download documentation in PDF format](#)

Input form

- [Data input](#)
- [Identifiers](#)
- [Variants and frequency data](#)
- [Additional annotations](#)
- [Predictions](#)
- [Filtering options](#)
- [Advanced options](#)
- [Jobs](#)

Results

- [Results summary](#)
- [Results preview table](#)
- [Navigating results](#)
- [Filtering results](#)
- [Downloading results](#)

Data formats

- [Input](#)
- [Output](#)

FAQ

- [General questions](#)
- [Web VEP questions](#)
- [VEP script questions](#)

When you reach the VEP web interface, you will be presented with a form to enter your data and alter various options.

Data input

1. First select the correct species for your data. Ensembl hosts many vertebrate genomes; genomes for plants, protists and fungi can be found at [Ensembl Genomes](#).
2. You can optionally choose a name for the data you upload - this can make it easier for you to identify jobs and files that you have uploaded to the VEP at a later point.
3. You have three options for uploading your data:

- **File upload** - click the "Choose file" button and locate the file on your system
- **Paste file** - simply copy and paste the contents of your file into the large text box
- **File URL** - point the VEP to a file hosted on a publically accessible address. This can be either a **http://** or **ftp://** address.

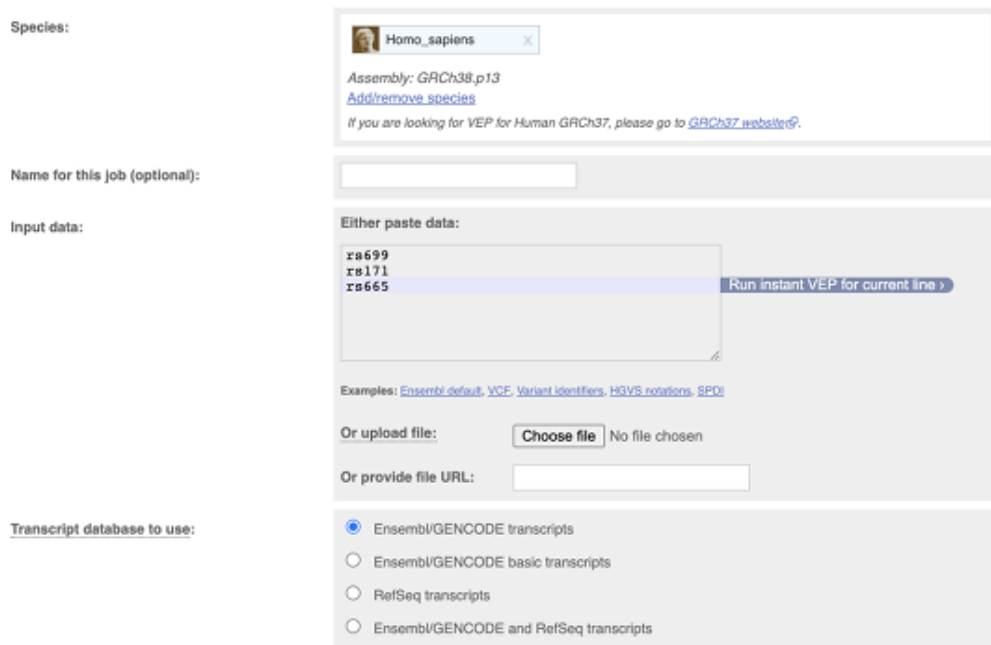
Once you have uploaded some data, you can select it as the input for future jobs by choosing the data from the drop down menu.

The format of your data is automatically detected; see the examples or the [input format](#) documentation.

4. For pasted data you can get an instant preview of the results of your first variant by clicking the button that appears when you paste your data. This quickly shows you the consequence type, the IDs of any overlapping variants, genes, transcripts and regulatory features, as well as SIFT and PolyPhen predictions. To see the full results set submit your job as normal.
5. For some species you can select which transcript database to use. The default is to use Ensembl transcripts, which offer the most rich annotation through VEP.

GENCODE Basic is a subset of the GENCODE gene set, and is intended to provide a simplified, high-quality subset of the GENCODE transcript annotations that will be useful to the majority of users. GENCODE Basic includes all genes in the GENCODE gene set, with a representative subset of the transcripts (splice variants).

You can also select to use RefSeq transcripts from the [otherfeatures database](#); note though that these transcripts are simply aligned to the reference genome and the database is missing much of the annotation found when using the main Ensembl database (e.g. protein domains, CCDS identifiers).



The screenshot shows the VEP web interface input form. It includes a 'Species:' dropdown menu with 'Homo_sapiens' selected, a 'Name for this job (optional):' text box, and an 'Input data:' section. The 'Input data:' section has a 'Either paste data:' text box containing three lines of variant data (rs699, rs171, rs665) and a 'Run instant VEP for current line' button. Below this is a 'Transcript database to use:' section with four radio button options: 'Ensembl/GENCODE transcripts' (selected), 'Ensembl/GENCODE basic transcripts', 'RefSeq transcripts', and 'Ensembl/GENCODE and RefSeq transcripts'. There are also links for 'Add/remove species' and 'Examples'.

Identifiers

VEP can provide additional identifiers for genes, transcripts, proteins and variants.

- **Gene symbol**

Add the gene symbol for the gene to the output. This will typically be, for example, the [HGNC](#) identifier for genes in human. Equivalent to `--symbol` in the VEP script.

- **Transcript version**

Add the transcript version to the transcript identifier. Equivalent to `--transcript_version`.

- **CCDS**

Add the [Consensus CDS](#) transcript identifier where available. Equivalent to `--ccds`.

- **Protein**

Add the Ensembl protein identifier (ENSP). Equivalent to `--protein`.

- **UniProt**

Add identifiers for translated protein products from three [UniProt](#)-related databases (SWISSPROT, TrEMBL and UniParc). Equivalent to `--uniprot`.

- **HGVS**

Generate [HGVS](#) identifiers for your input variants relative to the transcript coding sequence (HGVS_c) and the protein sequence (HGVS_p). Equivalent to `--hgvs`.

The screenshot shows a web interface for the VEP (Variant Effect Predictor) tool. At the top, there is a tab labeled 'Identifiers' with a dropdown arrow, and a subtitle 'Additional identifiers for genes, transcripts and variants'. Below this, the 'Identifiers' section is displayed. It contains a list of identifiers with corresponding checkboxes: 'Gene symbol:' (checked), 'Transcript version:' (checked), 'CCDS:' (unchecked), 'Protein:' (unchecked), 'UniProt:' (unchecked), and 'HGVS:' (unchecked). Each identifier is followed by a light gray rectangular button containing the checkbox.

Variants and frequency data

VEP can also search the Ensembl database for known variants that are co-located with variants from your input data.

- **Find co-located known variants** - report known variants from the Ensembl Variation database that overlap with your input. A list of variant sources imported can be viewed [here](#). Note that this feature is only available for species with an Ensembl Variation database. Equivalent to `--check_existing`.

VEP will by default compares the alleles of your input variant to that of the existing variant; VEP will only report the existing variant ID if none of the alleles in your input variant are novel.

For example, if your input variant has alleles A/G, and the existing variant has alleles A/T, then the existing variant will not be reported. If instead your input variant has alleles A/T, then the existing variant will be reported.

To disable this allele matching, select the option "Yes but don't compare alleles" for the option "Find co-located known variants".

For known variants VEP can also provide PubMed IDs of publications citing the variant (equivalent to `--pubmed`).

- **Frequency data for co-located variants**

VEP can also report allele frequency (AF) data for existing variants from several major genotyping projects, the [1000 Genomes Project](#), the [NHLBI-ESP](#) and [gnomAD](#); this only applies when you have selected human as your species.

- **1000 Genomes global** - the combined phase 3 population (i.e. all individuals from all populations). Equivalent to `--af`
- **1000 Genomes continental** - the four continent-level populations - AFR (African), AMR (American), ASN (Asian) and EUR (European). Equivalent to `--af 1kg`

- **ESP** - AA (African American) and EA (European American) populations. Equivalent to [--af_esp](#)
- **gnomAD** - combined, AFR, AMR, ASJ, EAS, FIN, NFE, OTH, SAS populations. Equivalent to [--af_gnomad](#)
- **PubMed IDs for citations of co-located variants**
Report the PubMed IDs of any publications that cite the co-located variant(s).
- **Include flagged variants**
Variants flagged as failed by the Ensembl Variation quality control.

Variants and frequency data
Co-located variants and frequency data

Variants and frequency data

Find co-located known variants:

Frequency data for co-located variants:

- ☒ [1000 Genomes global minor allele frequency](#)
- ☐ [1000 Genomes continental allele frequencies](#)
- ☐ [ESP allele frequencies](#)
- ☐ [gnomAD \(exomes\) allele frequencies](#)

PubMed IDs for citations of co-located variants:
☒

Include flagged variants:
☐

Additional annotations

- **Transcript biotype**
Add the [transcript biotype](#) to the output. Equivalent to [--biotype](#) in the VEP script.
 - **Exon and intron numbers**
Report the exon or intron number that a variant falls in as NUMBER / TOTAL, i.e. exon 2/5 means the variant falls in the 2nd of 5 exons in the transcript. Equivalent to [--numbers](#).
 - **Transcript support level**
Report the [transcript support level](#) of the overlapped transcript. Equivalent to [--tsl](#).
 - **APPRIS**
Report the [APPRIS](#) score of the overlapped transcript. Equivalent to [--appris](#).
 - **Identify canonical transcripts**
Add a flag to the output indicating if the reported transcript is the [canonical transcript](#) for the gene. Equivalent to [--canonical](#).
 - **Upstream/Downstream distance (bp)**
Change the distance to assign the upstream and downstream consequences. Equivalent to [--distance](#).
 - **miRNA structure**
Determines where in the secondary structure of a miRNA a variant falls. Equivalent to the VEP plugin [miRNA](#).
-
- **Protein domains**
Report [protein domains](#) from [PDBe](#), [Pfam](#), [Prosit](#) and [InterPro](#) that overlap input variants. Equivalent to [--domains](#).
-
- **Get regulatory region consequences**
In addition to predicting consequences with overlapping transcripts, VEP can find overlaps with known regulatory regions as determined in the [Ensembl Regulatory build](#).
Using this option, VEP will also report if a variant falls in a transcription factor binding motif, and give a score that reflects whether the altered motif sequence is more or less similar to the consensus.
Get regulatory consequences is equivalent to [--regulatory](#).
-

- **Phenotypes**

Report the phenotypic data overlapping the genomic features. This functionality is provided by the [Phenotypes](#) plugin. For more information on the imported phenotypic data for genes, variation and QTLs see [our phenotype documentation](#).

Note: This web functionality is not reporting cancer phenotypic data this release. However the cancer phenotypic data is available in the command line version.

- **DisGeNET**

Report Variant-Disease-PMID associations from the [DisGeNET](#) database. This functionality is provided by the [DisGeNET](#) plugin.

Note: This web functionality is reporting the unique disease names.

Additional annotations
Additional transcript, protein and regulatory annotations

Transcript annotation

Transcript biotype: ☒

Exon and intron numbers: ☐

Transcript support level: ☒

APPRIS: ☒

MANE: ☒

Identify canonical transcripts: ☐

Upstream/Downstream distance (bp):

miRNA structure: ☐

Protein annotation

Protein domains: ☐

Regulatory data

Get regulatory region consequences:

Phenotype data

Phenotypes: ☐

DisGeNET: ☐

Predictions

- **SIFT predictions**

[SIFT](#) predicts whether an amino acid substitution affects protein function based on sequence homology and the physical properties of amino acids. Only available in popular species. For both SIFT and PolyPhen VEP can report either a score between 0 and 1, a prediction in words, or both. Equivalent to [--sift](#).

- **PolyPhen predictions**

[PolyPhen](#) is a tool which predicts possible impact of an amino acid substitution on the structure and function of a human protein using straightforward physical and comparative considerations. Equivalent to [--polyphen](#).

- **dbNSFP**

Retrieves data for missense variants from [dbNSFP](#). Equivalent to the VEP plugin [dbNSFP](#).

- **Condel**

Calculates the Consensus Deleteriousness ([Condel](#)) score for a missense mutation based on the pre-calculated SIFT and PolyPhen-2 scores. Equivalent to the VEP plugin [Condel](#).

- **LoFtool**

Provides a rank of genic intolerance and consequent susceptibility to disease based on the ratio of Loss-of-function (LoF) to synonymous mutations. Equivalent to the VEP plugin [LoFtool](#).

- **dbscSNV**

Retrieves data for splicing variants from [dbscSNV](#). Equivalent to the VEP plugin [dbscSNV](#).

- **MaxEntScan**

Get splice site predictions from [MaxEntScan](#). Equivalent to the VEP plugin [MaxEntScan](#).

- **SpliceAI**

Pre-calculated annotations from [SpliceAI](#) a deep neural network, developed by Illumina, Inc that predicts splice junctions from an arbitrary pre-mRNA transcript sequence. Used for non-commercial purposes. This functionality is provided by the [SpliceAI](#) plugin.

- **BLOSUM62**

Looks up the BLOSUM 62 substitution matrix score for the reference and alternative amino acids predicted for a missense mutation. Equivalent to the VEP plugin [Blosum62](#).

- **Ancestral allele**

Retrieves ancestral allele sequences from a FASTA file. Ensembl produces [FASTA file dumps](#) of the ancestral sequences of key species. Equivalent to the VEP plugin [AncestralAllele](#).

Predictions ▾ Variant predictions, e.g. SIFT, PolyPhen

Pathogenicity predictions

SIFT:

Prediction and score ▾

PolyPhen:

Prediction and score ▾

dbNSFP:

☒ Disabled
☐ Enabled

CADD:

☐

Condel:

☒ Disabled
☐ Enabled

LoFtool:

☐

Splicing predictions

dbscSNV:

☐

MaxEntScan:

☐

SpliceAI:

☐

Conservation

BLOSUM62:

☐

Ancestral allele:

☐

VEP allows you to pre-filter your results e.g. by MAF or consequence type. Note that it is also possible to perform equivalent operations on the results page for VEP, so if you aren't sure, don't use any of these options!

- **By frequency**

Filter variants by minor allele frequency (MAF). Two options are provided:

- **Exclude common variants**

Filter out variants that are co-located with an existing variant that has a frequency greater than 0.01 (1%) in the 1000 Genomes global population. Equivalent to `--filter common` in the VEP script.

- **Advanced filtering**

Enabling this option allows you to specify a population and frequency to compare to, as well whether matching variants should be included or excluded from the results.

- **Return results for variants in coding regions only**

Exclude variants that don't fall in a coding region of a transcript. Equivalent to `--coding_only`.

- **Restrict results**

For many variants VEP will report multiple consequence types - typically this is because the variant overlaps more than one transcript. For each of these options VEP uses consequence ranks that are subjectively determined by Ensembl. [This table](#) gives all of the consequence types predicted by Ensembl, ordered by rank. Note that enabling one of these options not only loses potentially relevant data, but in some cases may be scientifically misleading. Options:

- **Show one selected consequence**

Pick one consequence type across all those predicted for the variant; the output will include transcript- or feature-specific information. Consequences are chosen by the canonical, biotype status and length of the transcript, along with the ranking of the consequence type according to [this table](#). This is the best method to use if you are interested only in one consequence per variant. Equivalent to `--pick`.

- **Show one selected consequence per gene**

Pick one consequence type for each gene using the same criteria as above. Note that if a variant overlaps more than one gene, output for each gene will be reported. Equivalent to `--per_gene`.

- **Show only list of consequences per variant**

Give a comma-separated list of all observed consequence types for each variant. No transcript-specific or gene-specific output will be given. Equivalent to `--summary`.

- **Show most severe per variant**

Only the most severe of all observed consequence types is reported for each variant. No transcript-specific or gene-specific output will be given. Equivalent to `--most_severe`.

The screenshot shows the 'Filtering options' section of the VEP web interface. It has a title 'Pre-filter results by frequency or consequence type'. Under the 'Filters' heading, there are three main sections: 'Filter by frequency:', 'Return results for variants in coding regions only:', and 'Restrict results:'. In the 'Filter by frequency:' section, three radio buttons are present: 'No filtering' (unselected), 'Exclude common variants' (unselected), and 'Advanced filtering' (selected). Below these, there are two input fields: 'Exclude' (with a dropdown arrow) and 'variants with MAF greater than' (with a dropdown arrow), followed by a text input containing '0.01'. Below that is another dropdown menu with 'in 1000 genomes (1KG) combined population' selected. In the 'Return results for variants in coding regions only:' section, there is an unchecked checkbox. In the 'Restrict results:' section, there is a dropdown menu with 'Show all results' selected. At the bottom of the form, there is a note: 'NB: Restricting results may exclude biologically important data!'.

Advanced options

The VEP web interface allows you to use/setup advanced options:

- **Buffer size**

By default VEP process the variants by blocks of 5000 (i.e. what we call "buffer size").

In some cases, reducing the size of the blocks (buffer size) could prevent memory issues for large VEP queries (e.g. use of

regulatory data, many plugins or custom annotations).

This is why the maximum buffer size is automatically set to 500 on the VEP Web interface when the "Regulatory data" option is selected.

- **Right align variants prior to consequence calculation**

By default VEP performs consequence calculation at the given input coordinates.

Optionally, VEP can shift insertions and deletions found within repeated regions as far as possible in the 3' direction, normalising output.

Advanced options ▾ Additional enhancements

Advanced options

Buffer size: 5000 ▾

NB: When the **Regulatory data** option is selected then due to the large amount of regulatory data available, the **maximum buffer size** is automatically reduced from the default value of 5000 to 500. This reduces the memory requirement but might increase the run time. If you find that your jobs are still failing due to memory limitations then you can select a value **lower than 500**.

Right align variants prior to consequence calculation: Right align relative to transcript ▾

Jobs

Once you have clicked "Run", your input will be checked and submitted to the VEP as a job. All jobs associated with your session or account are shown in the "Recent Tickets" table. You may submit multiple jobs simultaneously.

The "Jobs" column of the table shows the current status of the job.

- **Queued** - your job is waiting to be submitted to the system
- **Running** - your job is currently running
- **Done** - your job is finished - click the [View results] link to be taken to the results page
- **Failed** - there is a problem with your job - click the magnifying glass icon 🔍 to see more details

You may delete a job by clicking the trash can icon 🗑️. If you are logged in to Ensembl, you can save the job by clicking the save icon 💾.

You may also resubmit a job (for example, to re-run with the same data but change some parameters) by clicking the edit icon ✎.

You can see a summary of the options that you selected for your VEP job by clicking on the magnifying glass icon 🔍.

Show	All ▾ entries	Show/hide columns (1 hidden)	Filter
Analysis	Jobs	Submitted at	
Variant Effect Predictor	VEP analysis of pasted data in Bos_taurus Done [View results] 🔍 ✎ 🗑️	13/07/2015, 09:44	💾 🔍 🗑️
Variant Effect Predictor	VEP analysis of pasted data in Ovis_aries Done [View results]	08/07/2015, 13:19	💾 🔍 🗑️
Variant Effect Predictor	VEP analysis of pasted data in Ovis_aries Failed	07/07/2015, 16:51	💾 🔍 🗑️

The VEP presents a [summary](#) and a detailed [results preview](#) on its results page.

Summary

The summary panel on the VEP results page gives a brief overview of the VEP job, along with some basic statistics about the results.

Statistics

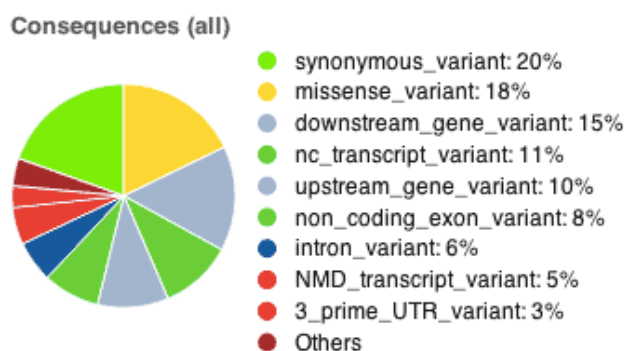
Various statistics are listed in a table, including:

- Variants processed - any variants not parsed by the VEP are not included in this count
- Variants remaining after [filtering](#)
- Novel / known variants - the number and percentage of novel variants vs existing variants in the input (see [input page documentation](#))
- Number of overlapped genes, transcripts and regulatory features

Category	Count
Variants processed	498955
Variants remaining after filtering	498955
Novel / known variants	-
Overlapped genes	825
Overlapped transcripts	2888
Overlapped regulatory features	7309

Pie charts

Pie charts are shown detailing the proportion of consequence types called across all variants in the results. The colour scheme of the pie chart matches the colours used to draw variants on the Ensembl region in detail view.



Results preview table

The results table shows one row per transcript and variant. By default all of the columns are shown; to temporarily hide columns, click the blue "Show/hide columns" button and select or deselect the columns you wish to view. The columns you select will be recalled when viewing other jobs.

Hover over a column title to see a description. See the [VEP output format documentation](#) for more details on each of the results columns.

The table can be sorted by any column - click the column header to toggle sorting behaviour.

To download what you see in the table, hover over the spreadsheet icon in the top right corner of the table.

Several columns have special features for the data they contain:

- **Location** - click the link to navigate to the region in detail view for the region surrounding this variant

- **Gene, Feature and Existing Variation** - click the link to bring up a summary view of the gene, transcript, regulatory feature or variation, from which you can navigate to the main Ensembl page for it
- **Consequence** - hover over the consequence name to see the [Sequence Ontology](#) definition. See the [Ensembl Variation documentation](#) for a full list of consequence types used by the VEP and their definitions
- **SIFT and PolyPhen** - predictions and scores are coloured according to the nature of the prediction, with red indicating deleterious or damaging

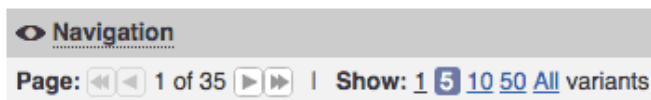
Uploaded variation	Location	Feature	Feature type	Consequence	CDS position	Protein position	Amino acids	Codons	SIFT	GMAF
rs116383664	1:1115461	ENST00000528923	RegulatoryFeature	regulatory_region_variant	-	-	-	-	-	T:0.0137
rs116383664	1:1115461	ENST00000379317	Transcript	upstream_gene_variant	-	-	-	-	-	T:0.0137
rs116383664	1:1115461	ENST00000486379	Transcript	upstream_gene_variant	-	-	-	-	-	T:0.0137
rs116383664	1:1115461	ENST00000379289	Transcript	missense_variant	247	83	R/W	Cgg/Tgg	tolerated(0.06)	T:0.0137
rs116383664	1:1115461	ENST00000460998	Transcript	upstream_gene_variant	-	-	-	-	-	T:0.0137
rs116383664	1:1115461	ENST00000514695	Transcript	upstream_gene_variant	-	-	-	-	-	T:0.0137
rs116383664	1:1115461	ENST00000379290	Transcript	missense_variant	247	83	R/W	Cgg/Tgg	tolerated(0.06)	T:0.0137
rs116383664	1:1115461	ENST00000379288	Transcript	missense_variant	28	10	R/W	Cgg/Tgg	deleterious(0.03)	T:0.0137

Navigating results

The navigation panel can be used to scroll through pages of results.

By default, the results for five variants are shown. Note that since a variant can overlap multiple transcripts, the table will often show **more than** five rows. To change the number shown, click the appropriate link. Be warned that if your input file is large, it is inadvisable to show all results unless you are sure you have applied sufficient filters - your browser may become unresponsive if it tries to display many thousands of rows in the table.

To navigate between pages of results, use the four arrow icons. Note that when any filters are enabled, it is not possible to navigate to the last page of results as the total number of results cannot be calculated.



Filtering results

You can apply any combination of filters to your results in order to identify interesting data. This is equivalent to using the [VEP filtering script](#) on the command line.

To add a filter, simply select the column you wish to filter on, select an "operator", and input a value for the filter to compare to.

To edit a filter, click the pencil icon . To remove a filter, click the cross icon .

When you have added more than one filter, you are given the option to match any or all of the rules shown; click the "Update" button once you have made your selection.

Certain columns when selected have special features:

- **Location** - for this column you may enter genomic coordinates in the format "chromosome:start-end". It is also possible to enter just a chromosome, e.g. enter "12" to show only variants on chromosome 12.

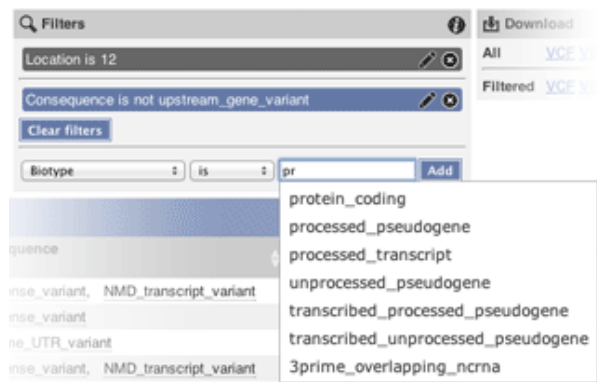
Adding multiple location filters allows you to select multiple regions - location filters are not affected by whether you select "Match all" or "Match any" (see above).

Users should note that enabling at least one location filter will greatly speed up the return of results (this is because [tabix](#) is used behind the scenes).

Location filters are not affected by the operator selected.

- **Allele, Feature type, Consequence, SIFT, PolyPhen and Biotype** - for these columns, autocomplete will help you fill in the value when you start typing
- **SIFT and PolyPhen** - these columns can contain both text (e.g. a SIFT prediction) and a number (e.g. a frequency value). The VEP allows you to filter on either part of this.

For example, you may enter "is" and "deleterious" for SIFT to return deleterious predictions, or "<" and "0.1" to find results with a SIFT score less than 0.1.

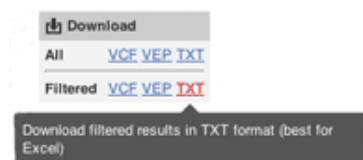


Downloading results

The VEP allows you to download either your full or filtered results set in a choice of data formats.

- **VCF** - [VCF](#) is a portable format for variant data. Consequence data is encoded as a series of delimited strings under the "CSQ" key in the VCF INFO field.
- **VEP** - The default [VEP output format](#) gives one row per variant and transcript overlap.
- **TXT** - Text format is a tab-delimited format, equivalent to what can be seen in the results table. Note that the columns you select to be visible in the table do not affect the downloaded file - all columns are outputted. This format is best if you intend to import the results into a spreadsheet program such as Microsoft Excel.

You can also send the genes or known variants in your current preview to BioMart. This allows you to easily retrieve any of BioMart's rich data associated with these genes (other database references, GO terms, orthologues/paralogues) and variants (phenotype annotations, synonyms, citations).



Input

Both the web and script version of VEP can use the same input formats. Formats can be auto-detected by the VEP script, but must be manually selected when using the web interface.

VEP can use different input formats:

- [Default VEP input](#)
- [VCF](#)
- [VCF - Structural variants](#)
- [HGVS identifiers](#)
- [Variant identifiers](#)
- [Genomic SPDI notation](#)
- [REST-style regions](#)

Default VEP input

The default format is a simple **whitespace-separated** format (columns may be separated by space or tab characters), containing five required columns plus an optional identifier column:

1. **chromosome** - just the name or number, with no 'chr' prefix
2. **start**
3. **end**
4. **allele** - pair of alleles separated by a '/', with the reference allele first
5. **strand** - defined as + (forward) or - (reverse).
6. **identifier** - this identifier will be used in VEP's output. If not provided, VEP will construct an identifier from the given coordinates and alleles.

1	881907	881906	-/C	+	
5	140532	140532	T/C	+	
12	1017956	1017956	T/A	+	
2	946507	946507	G/C	+	
14	19584687	19584687	C/T	-	
19	66520	66520	G/A	+	var1
8	150029	150029	A/T	+	var2

An insertion (of any size) is indicated by start coordinate = end coordinate + 1. For example, an insertion of 'C' between nucleotides 12600 and 12601 on the forward strand of chromosome 8 is indicated as follows:

8	12601	12600	-/C	+	
---	-------	-------	-----	---	--

A deletion is indicated by the exact nucleotide coordinates. For example, a three base pair deletion of nucleotides 12600, 12601, and 12602 of the reverse strand of chromosome 8 will be:

8	12600	12602	CGT/-	-	
---	-------	-------	-------	---	--

VCF

VEP also supports using [VCF \(Variant Call Format\) version 4.0](#). This is a common format used by the 1000 genomes project, and can be produced as an output format by many variant calling tools.

Users using VCF should note a peculiarity in the difference between how Ensembl and VCF describe unbalanced variants. For any unbalanced variant (i.e. insertion, deletion or unbalanced substitution), the VCF specification requires that the base immediately before the variant should be included in both the reference and variant alleles. This also affects the reported position i.e. the reported position will be one base before the actual site of the variant.

In order to parse this correctly, VEP needs to convert such variants into Ensembl-type coordinates, and it does this by removing the additional base and adjusting the coordinates accordingly. This means that if an identifier is not supplied for a variant (in the 3rd column of the VCF), then the identifier constructed and the position reported in VEP's output file will differ from the input.

This problem can be overcome with the following:

1. ensuring each variant has a unique identifier specified in the 3rd column of the VCF
2. using VCF format as output (`--vcf`) - this preserves the formatting of your input coordinates and alleles
3. using `--minimal` and `--allele_number` (see [Complex VCF entries](#)).

The following examples illustrate how VCF describes a variant and how it is handled internally by VEP. Consider the following aligned sequences (for the purposes of discussion on chromosome 20):

```
Ref: a t C g a // C is the reference base
1 : a t G g a // C base is a G in individual 1
2 : a t - g a // C base is deleted w.r.t. the reference in individual 2
3 : a t CAg a // A base is inserted w.r.t. the reference sequence in individual 3
```

Individual 1

The first individual shows a simple balanced substitution of G for C at base 3. This is described in a compatible manner in VCF and Ensembl styles. Firstly, in VCF:

```
20 3 . C G . PASS .
```

And in Ensembl format:

```
20 3 3 C/G +
```

Individual 2

The second individual has the 3rd base deleted relative to the reference. In VCF, both the reference and variant allele columns must include the preceding base (T) and the reported position is that of the preceding base:

```
20 2 . TC T . PASS .
```

In Ensembl format, the preceding base is not included, and the start/end coordinates represent the region of the sequence deleted. A "-" character is used to indicate that the base is deleted in the variant sequence:

```
20 3 3 C/- +
```

The upshot of this is that while in the VCF input file the position of the variant is reported as 2, in the output file from VEP the position will be reported as 3. If no identifier is provided in the third column of the VCF, then the constructed identifier will be:

```
20_3_C/-
```

Individual 3

The third individual has an "A" inserted between the 3rd and 4th bases of the sequence relative to the reference. In VCF, as for the deletion, the base before the insertion is included in both the reference and variant allele columns, and the reported position is that of the preceding base:

```
20 3 . C CA . PASS .
```

In Ensembl format, again the preceding base is not included, and the start/end positions are "swapped" to indicate that this is an insertion. Similarly to a deletion, a "-" is used to indicate no sequence in the reference:

```
20 4 3 -/A +
```

Again, the output will appear different, and the constructed identifier may not be what is expected:

```
20_3_-/A
```

Using VCF format output, or adding unique identifiers to the input (in the third VCF column), can mitigate this issue.

Complex VCF entries

For VCF entries with multiple alternate alleles, VEP will only trim the leading base from alleles if **all** REF and ALT alleles start with the same base:

```
20 3 . C CAAG,CAAGAAG . PASS .
```

This will be considered internally by VEP as equivalent to:

```
20 4 3 -/AAG/AAGAAG +
```

Now consider the case where a single VCF line contains a representation of both a SNV and an insertion:

```
20 3 . C CAAAG,G . PASS .
```

Here the input alleles will remain unchanged, and VEP will consider the first REF/ALT pair as a substitution of C for CAAG, and the second as a C/G SNV:

```
20 3 3 C/CAAG/G +
```

To modify this behaviour, VEP script users may use `--minimal`. This flag forces VEP to consider each REF/ALT pair independently, trimming identical leading and trailing bases from each as appropriate. Since this can lead to confusing output regarding coordinates etc, it is not the default behaviour. It is recommended to use the `--allele_number` flag to track the correspondence between alleles as input and how they appear in the output.

VCF - Structural variants

VEP can also call consequences on structural variants encoded in tab-delimited or VCF format. To recognise a variant as a structural variant, the allele string (or "SVTYPE" INFO field in VCF) must be set to one of the currently recognised values:

- **INS** - insertion
- **DEL** - deletion
- **DUP** - duplication
- **TDUP** - tandem duplication

Examples of structural variants encoded in tab-delimited format:

```
1 160283 471362 DUP + sv1
1 1385015 1387562 DEL + sv2
```

Examples of structural variants encoded in VCF format:

```
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT
1 160283 sv1 . <DUP> . . SVTYPE=DUP;END=471362 .
1 1385015 sv2 . <DEL> . . SVTYPE=DEL;END=1387562 .
```

See the [VCF definition document](#) for more detail on how to describe structural variants in VCF format.

HGVS identifiers

See <https://varnomen.hgvs.org> for details. These must be relative to genomic or Ensembl transcript coordinates.

It also is possible to use RefSeq transcripts in both the web interface and the VEP script (see [script documentation](#)): this works for RefSeq transcripts that align to the genome correctly.

Examples:

```
ENST00000207771.3:c.344+626A>T
ENST00000471631.1:c.28_33delTCGCGG
ENST00000285667.3:c.1047_1048insC
5:g.140532T>C
```

Examples using RefSeq identifiers (using `--refseq` in the VEP script, or select the other features transcript database on the web interface and input type of HGVS):

```
NM_153681.2:c.7C>T
NM_005239.4:c.190G>A
NM_001025204.1:c.336G>A
```

HGVS protein notations may also be used, provided that they unambiguously map to a single genomic change. Due to redundancy in the amino acid code, it is not always possible to work out the corresponding genomic sequence change for a given protein sequence change. The following example is for a permissible protein notation in dog (*Canis familiaris*):

```
ENSCAFP00000040171.1:p.Thr92Asn
```

HGVS notations may also be given in [LRG](#) coordinates:

```
LRG_1t1:c.841G>T
LRG_1:g.10006G>T
```

Variant identifiers

These should be e.g. dbSNP rsIDs, or any synonym for a variant present in the Ensembl Variation database. See [here](#) for a list of identifier sources in Ensembl.

Genomic SPDI notation

VEP can also support genomic SPDI notation which uses four fields delimited by colons S:P:D:I (Sequence:Position:Deletion:Insertion). See [here](#) for details.

Examples:

```
NC_000016.10:68684738:G:A
NC_000017.11:43092199:GCTTTT:
NC_000013.11:32315789::C
NC_000016.10:68644746:AA:GTA
16:68684738:2:AC
```

REST-style regions

VEP's region REST endpoint requires variants are described as `[chr]:[start]-[end]:[strand]/[allele]`. This follows the same conventions as the [default input format](#) described above, with the key difference being that this format does not require the reference (REF) allele to be included; VEP will look up the reference allele using either a provided FASTA file (preferred) or Ensembl core database. Strand is optional and defaults to 1 (forward strand).

```
# SNP
5:140532-140532:1/C

# SNP (reverse strand)
14:19584687-19584687:-1/T

# insertion
1:881907-881906:1/C

# 5bp deletion
2:946507-946511:1/-
```

Output

VEP can return the results in different formats:

- [Default VEP output](#)
- [Tab-delimited output](#)
- [VCF](#)
- [JSON output](#)

Along with the results VEP computes and returns some [statistics](#).

Default VEP output

The default output format ("VEP" format when downloading from the web interface) is a 14 column tab-delimited file. Empty values are denoted by '-'. The output columns are:

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, RegulatoryFeature, MotifFeature.
7. **Consequence** - [consequence type](#) of this variant
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 variant 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 variant
14. **Extra** - this column contains extra information as key=value pairs separated by ";", see below.

Other output fields:

- **REF_ALLELE** - the reference allele
- **IMPACT** - the impact modifier for the consequence type
- **VARIANT_CLASS** - Sequence Ontology [variant class](#)
- **SYMBOL** - the gene symbol
- **SYMBOL_SOURCE** - the source of the gene symbol
- **STRAND** - the DNA strand (1 or -1) on which the transcript/feature lies

- **ENSP** - the Ensembl protein identifier of the affected transcript
- **FLAGS** - transcript quality flags:
 - *cds_start_NF*: CDS 5' incomplete
 - *cds_end_NF*: CDS 3' incomplete
- **SWISSPROT** - Best match UniProtKB/Swiss-Prot accession of protein product
- **TREMBL** - Best match UniProtKB/TrEMBL accession of protein product
- **UNIPARC** - Best match UniParc accession of protein product
- **HGVSc** - the HGVS coding sequence name
- **HGVSp** - the HGVS protein sequence name
- **HGVSG** - the HGVS genomic sequence name
- **HGVS_OFFSET** - Indicates by how many bases the HGVS notations for this variant have been [shifted](#)
- **NEAREST** - Identifier(s) of nearest transcription start site
- **SIFT** - the SIFT prediction and/or score, with both given as prediction(score)
- **PolyPhen** - the PolyPhen prediction and/or score
- **MOTIF_NAME** - the source and identifier of a transcription factor binding profile aligned at this position
- **MOTIF_POS** - The relative position of the variation in the aligned TFBP
- **HIGH_INF_POS** - a flag indicating if the variant falls in a high information position of a transcription factor binding profile (TFBP)
- **MOTIF_SCORE_CHANGE** - The difference in motif score of the reference and variant sequences for the TFBP
- **CELL_TYPE** - List of cell types and classifications for regulatory feature
- **CANONICAL** - a flag indicating if the transcript is denoted as the canonical transcript for this gene
- **CCDS** - the CCDS identifier for this transcript, where applicable
- **INTRON** - the intron number (out of total number)
- **EXON** - the exon number (out of total number)
- **DOMAINS** - the source and identifier of any overlapping protein domains
- **DISTANCE** - Shortest distance from variant to transcript
- **IND** - individual name
- **ZYG** - zygosity of individual genotype at this locus
- **SV** - IDs of overlapping structural variants
- **FREQS** - Frequencies of overlapping variants used in filtering
- **AF** - Frequency of existing variant in 1000 Genomes
- **AFR_AF** - Frequency of existing variant in 1000 Genomes combined African population
- **AMR_AF** - Frequency of existing variant in 1000 Genomes combined American population
- **ASN_AF** - Frequency of existing variant in 1000 Genomes combined Asian population
- **EUR_AF** - Frequency of existing variant in 1000 Genomes combined European population
- **EAS_AF** - Frequency of existing variant in 1000 Genomes combined East Asian population
- **SAS_AF** - Frequency of existing variant in 1000 Genomes combined South Asian population
- **AA_AF** - Frequency of existing variant in NHLBI-ESP African American population
- **EA_AF** - Frequency of existing variant in NHLBI-ESP European American population
- **gnomAD_AF** - Frequency of existing variant in gnomAD exomes combined population
- **gnomAD_AFR_AF** - Frequency of existing variant in gnomAD exomes African/American population
- **gnomAD_AMR_AF** - Frequency of existing variant in gnomAD exomes American population
- **gnomAD_ASJ_AF** - Frequency of existing variant in gnomAD exomes Ashkenazi Jewish population
- **gnomAD_EAS_AF** - Frequency of existing variant in gnomAD exomes East Asian population
- **gnomAD_FIN_AF** - Frequency of existing variant in gnomAD exomes Finnish population
- **gnomAD_NFE_AF** - Frequency of existing variant in gnomAD exomes Non-Finnish European population
- **gnomAD_OTH_AF** - Frequency of existing variant in gnomAD exomes combined other combined populations

- **gnomAD_SAS_AF** - Frequency of existing variant in gnomAD exomes South Asian population
- **MAX_AF** - Maximum observed allele frequency in 1000 Genomes, ESP and gnomAD
- **MAX_AF_POPS** - Populations in which maximum allele frequency was observed
- **CLIN_SIG** - ClinVar clinical significance of the dbSNP variant
- **BIOTYPE** - Biotype of transcript or regulatory feature
- **APPRIS** - Annotates alternatively spliced transcripts as primary or alternate based on a range of computational methods. NB: not available for GRCh37
- **TSL** - Transcript support level. NB: not available for GRCh37
- **PUBMED** - Pubmed ID(s) of publications that cite existing variant
- **SOMATIC** - Somatic status of existing variant(s); multiple values correspond to multiple values in the Existing_variation field
- **PHENO** - Indicates if existing variant is associated with a phenotype, disease or trait; multiple values correspond to multiple values in the Existing_variation field
- **GENE_PHENO** - Indicates if overlapped gene is associated with a phenotype, disease or trait
- **ALLELE_NUM** - Allele number from input; 0 is reference, 1 is first alternate etc
- **MINIMISED** - Alleles in this variant have been converted to minimal representation before consequence calculation
- **PICK** - indicates if this block of consequence data was picked by [--flag_pick](#) or [--flag_pick allele](#)
- **BAM_EDIT** - Indicates success or failure of edit using BAM file
- **GIVEN_REF** - Reference allele from input
- **USED_REF** - Reference allele as used to get consequences
- **REFSEQ_MATCH** - the RefSeq transcript match status; contains a number of flags indicating whether this RefSeq transcript matches the underlying reference sequence and/or an Ensembl transcript ([more information](#)).
 - *rseq_3p_mismatch*: signifies a mismatch between the RefSeq transcript and the underlying primary genome assembly sequence. Specifically, there is a mismatch in the 3' UTR of the RefSeq model with respect to the primary genome assembly (e.g. GRCh37/GRCh38).
 - *rseq_5p_mismatch*: signifies a mismatch between the RefSeq transcript and the underlying primary genome assembly sequence. Specifically, there is a mismatch in the 5' UTR of the RefSeq model with respect to the primary genome assembly.
 - *rseq_cds_mismatch*: signifies a mismatch between the RefSeq transcript and the underlying primary genome assembly sequence. Specifically, there is a mismatch in the CDS of the RefSeq model with respect to the primary genome assembly.
 - *rseq_ens_match_cds*: signifies that for the RefSeq transcript there is an overlapping Ensembl model that is identical across the CDS region only. A CDS match is defined as follows: the CDS and peptide sequences are identical and the genomic coordinates of every translatable exon match. Useful related attributes are: *rseq_ens_match_wt* and *rseq_ens_no_match*.
 - *rseq_ens_match_wt*: signifies that for the RefSeq transcript there is an overlapping Ensembl model that is identical across the whole transcript. A whole transcript match is defined as follows: 1) In the case that both models are coding, the transcript, CDS and peptide sequences are all identical and the genomic coordinates of every exon match. 2) In the case that both transcripts are non-coding the transcript sequences and the genomic coordinates of every exon are identical. No comparison is made between a coding and a non-coding transcript. Useful related attributes are: *rseq_ens_match_cds* and *rseq_ens_no_match*.
 - *rseq_ens_no_match*: signifies that for the RefSeq transcript there is no overlapping Ensembl model that is identical across either the whole transcript or the CDS. This is caused by differences between the transcript, CDS or peptide sequences or between the exon genomic coordinates. Useful related attributes are: *rseq_ens_match_wt* and *rseq_ens_match_cds*.
 - *rseq_mrna_match*: signifies an exact match between the RefSeq transcript and the underlying primary genome assembly sequence (based on a match between the transcript stable id and an accession in the RefSeq mRNA file). An exact match occurs when the underlying genomic sequence of the model can be perfectly aligned to the mRNA sequence post polyA clipping.
 - *rseq_mrna_nonmatch*: signifies a non-match between the RefSeq transcript and the underlying primary genome assembly sequence. A non-match is deemed to have occurred if the underlying genomic sequence does not have a perfect alignment to the mRNA sequence post polyA clipping. It can also signify that no comparison was possible as the model stable id may not have had a corresponding entry in the RefSeq mRNA file (sometimes happens when accessions are retired or changed). When a non-match occurs one or several of the following transcript attributes will also be present to provide more detail on the nature of the non-match: *rseq_5p_mismatch*, *rseq_cds_mismatch*, *rseq_3p_mismatch*, *rseq_nctran_mismatch*, *rseq_no_comparison*
 - *rseq_nctran_mismatch*: signifies a mismatch between the RefSeq transcript and the underlying primary genome assembly sequence. This is a comparison between the entire underlying genomic sequence of the RefSeq model to the mRNA in the case of RefSeq models that are non-coding.

- **rseq_no_comparison**: signifies that no alignment was carried out between the underlying primary genome assembly sequence and a corresponding RefSeq mRNA. The reason for this is generally that no corresponding, unversioned accession was found in the RefSeq mRNA file for the transcript stable id. This sometimes happens when accessions are retired or replaced. A second possibility is that the sequences were too long and problematic to align (though this is rare).
- **OverlapBP** - Number of base pairs overlapping with the corresponding structural variation feature
- **OverlapPC** - Percentage of corresponding structural variation feature overlapped by the given input
- **CHECK_REF** - Reports variants where the input reference does not match the expected reference
- **AMBIGUITY** - IUPAC allele ambiguity code

Example of VEP default output format:

11_224088_C/A	11:224088	A	ENSG00000142082	ENST00000525319	Transcript	missense_va
11_224088_C/A	11:224088	A	ENSG00000142082	ENST00000534381	Transcript	5_prime_UTR
11_224088_C/A	11:224088	A	ENSG00000142082	ENST00000529055	Transcript	downstream
11_224585_G/A	11:224585	A	ENSG00000142082	ENST00000529937	Transcript	intron_vari
22_16084370_G/A	22:16084370	A	-	ENSR00000615113	RegulatoryFeature	regulatory

The VEP script will also add a header to the output file. This contains information about the databases connected to, and also a key describing the key/value pairs used in the extra column.

```
## ENSEMBL VARIANT EFFECT PREDICTOR v102.0
## Output produced at 2017-03-21 14:51:27
## Connected to homo_sapiens_core_102_38 on ensembl.db.ensembl.org
## Using cache in /homes/user/.vep/homo_sapiens/102_GRCh38
## Using API version 102, DB version 102
## polyphen version 2.2.2
## sift version sift5.2.2
## COSMIC version 78
## ESP version 20141103
## gencode version GENCODE 25
## genebuild version 2014-07
## HGMD-PUBLIC version 20162
## regbuild version 16
## assembly version GRCh38.p7
## ClinVar version 201610
## dbSNP version 147
## Column descriptions:
## Uploaded_variation : Identifier of uploaded variant
## Location : Location of variant in standard coordinate format (chr:start or chr:start-end)
## Allele : The variant allele used to calculate the consequence
## Gene : Stable ID of affected gene
## Feature : Stable ID of feature
## Feature_type : Type of feature - Transcript, RegulatoryFeature or MotifFeature
## Consequence : Consequence type
## cDNA_position : Relative position of base pair in cDNA sequence
## CDS_position : Relative position of base pair in coding sequence
## Protein_position : Relative position of amino acid in protein
## Amino_acids : Reference and variant amino acids
## Codons : Reference and variant codon sequence
## Existing_variation : Identifier(s) of co-located known variants
## Extra column keys:
## IMPACT : Subjective impact classification of consequence type
## DISTANCE : Shortest distance from variant to transcript
## STRAND : Strand of the feature (1/-1)
## FLAGS : Transcript quality flags
```

Tab-delimited output

The [--tab](#) flag instructs VEP to write output as a tab-delimited table.

This differs from the default output format in that each individual field from the "Extra" field is written to a separate tab-delimited column.

This makes the output more suitable for import into spreadsheet programs such as Excel.

Furthermore the header is the same as the one for the VEP default output format and this is also the format used when selecting the "TXT" option on the VEP web interface.

Example of VEP tab-delimited output format:

<i>#Uploaded_variation</i>	<i>Location</i>	<i>Allele</i>	<i>Gene</i>	<i>Feature</i>	<i>Feature_type</i>	<i>Consequence</i>
11_224088_C/A	11:224088	A	ENSG00000142082	ENST00000525319	Transcript	missense
11_224088_C/A	11:224088	A	ENSG00000142082	ENST00000534381	Transcript	downstre
11_224088_C/A	11:224088	A	ENSG00000142082	ENST00000529055	Transcript	downstre
11_224585_G/A	11:224585	A	ENSG00000142082	ENST00000529937	Transcript	intron_v

The choice and order of columns in the output may be configured using [--fields](#). For instance:

```
./vep -i examples/homo_sapiens_GRCh38.vcf --cache --force_overwrite --tab --fields "Uploaded_variation"
```

VCF output

The VEP script can also generate VCF output using the [--vcf](#) flag.

Main information about the specificity of the VEP VCF output format:

- Consequences are added in the INFO field of the VCF file, using the key "**CSQ**" (you can change it using [--vcf_info_field](#)).
- Data fields are encoded separated by the character "|" (pipe). The order of fields is written in the VCF header. Unpopulated fields are represented by an empty string.
- Output fields in the "CSQ" INFO field can be configured by using [--fields](#).
- Each prediction, for a given variant, is separated by the character "," in the CSQ INFO field (e.g. when a variant overlaps more than 1 transcript)

Here is a list of the (default) fields you can find within the CSQ field:

```
Allele|Consequence|IMPACT|SYMBOL|Gene|Feature_type|Feature|BIOTYPE|EXON|INTRON|HGVSc|HGVSp|cDNA|
```

Example of VEP command using the [--vcf](#) and [--fields](#) options:

```
./vep -i examples/homo_sapiens_GRCh38.vcf --cache --force_overwrite --vcf --fields "Allele,Consequence"
```

VCFs produced by VEP can be filtered by [filter_vep.pl](#) in the same way as standard format output files.

If the input format was VCF, the file will remain unchanged save for the addition of the CSQ field and the header (unless using any filtering). If an existing CSQ field is found, it will be replaced by the one added by the VEP (use [--keep_csq](#) to preserve it).

Custom data added with [--custom](#) are added as separate fields, using the key specified for each data file.

Commas in fields are replaced with ampersands (&) to preserve VCF format.

```
##INFO=<ID=CSQ,Number=.,Type=String,Description="Consequence annotations from Ensembl VEP. Format: Allele|Consequence|IMPACT|SYMBOL|Gene|Feature_type|Feature|BIOTYPE|EXON|INTRON|HGVSc|HGVSp|cDNA|REF|ALT|QUAL|FILTER|INFO|CSQ=">
#CHROM POS ID REF ALT QUAL FILTER INFO
21 26978790 rs75377686 T C . . CSQ=C|missense_variant|MODERATE|MRPL39|ENSG00000142082
```

JSON output

VEP can produce output in the form of serialised [JSON](#) objects using the [--json](#) flag. JSON is a serialisation format that can be parsed and processed easily by many packages and programming languages; it is used as the default output format for [Ensembl's REST server](#).

Each input variant is reported as a single JSON object which constitutes one line of the output file. The JSON object is structured somewhat differently to the other VEP output formats, in that per-variant fields (e.g. co-located existing variant details) are reported

only once. Consequences are grouped under the feature type that they affect (Transcript, Regulatory Feature, etc). The original input line (e.g. from VCF input) is reported under the "input" key in order to aid aligning input with output.

Here follows an example of JSON output (prettified and redacted for display here):

```
{
  "input": "1 230845794 test1 A G . . .",
  "id": "test1",
  "seq_region_name": "1",
  "start": 230845794,
  "end": 230845794,
  "strand": 1,
  "allele_string": "A/G",
  "most_severe_consequence": "missense_variant",
  "colocated_variants": [
    {
      "id": "rs699",
      "seq_region_name": "1",
      "start": 230845794,
      "end": 230845794,
      "strand": 1,
      "allele_string": "A/G",
      "minor_allele": "A",
      "minor_allele_freq": 0.3384,
      "afr_allele": "A",
      "afr_maf": 0.13,
      "amr_allele": "A",
      "amr_maf": 0.36,
      "asn_allele": "A",
      "asn_maf": 0.16,
      "eur_allele": "A",
      "eur_maf": 0.41,
      "pubmed": [
        18513389,
        23716723
      ]
    }
  ],
  {
    "seq_region_name": "1",
    "strand": 1,
    "id": "COSM425562",
    "allele_string": "A/G",
    "start": 230845794,
    "end": 230845794
  }
],
  "transcript_consequences": [
    {
      "variant_allele": "G",
      "consequence_terms": [
        "missense_variant"
      ],
      "gene_id": "ENSG00000135744",
      "gene_symbol": "AGT",
      "gene_symbol_source": "HGNC",
      "transcript_id": "ENST00000366667",
      "biotype": "protein_coding",
      "strand": -1,
      "cdna_start": 1018,
      "cdna_end": 1018,
      "cds_start": 803,
      "cds_end": 803,
      "protein_start": 268,
      "protein_end": 268,
      "codons": "aTg/aCg",
      "amino_acids": "M/T",
      "polyphen_prediction": "benign",
      "polyphen_score": 0,
      "sift_prediction": "tolerated",
      "sift_score": 1,
    }
  ]
}
```

```

    "hgvcsc": "ENST00000366667.4:c.803T>C",
    "hgvsp": "ENSP00000355627.4:p.Met268Thr"
  },
  "regulatory_feature_consequences": [
    {
      "variant_allele": "G",
      "consequence_terms": [
        "regulatory_region_variant"
      ],
      "regulatory_feature_id": "ENSR00001529861"
    }
  ]
}

```

In accordance with JSON conventions, all keys are lower-case. Some keys also have different names and structures to those found in the other VEP output formats:

Key	JSON equivalent(s)	Notes
Consequence	consequence_terms	
Gene	gene_id	
Feature	transcript_id, regulatory_feature_id, motif_feature_id	Consequences are grouped under the feature type they affect
ALLELE	variant_allele	
SYMBOL	gene_symbol	
SYMBOL_SOURCE	gene_symbol_source	
ENSP	protein_id	
OverlapBP	bp_overlap	
OverlapPC	percentage_overlap	
Uploaded_variation	id	
Location	seq_region_name, start, end, strand	The variant's location field is broken down into constituent coordinate parts for clarity. "seq_region_name" is used in place of "chr" or "chromosome" for consistency with other parts of Ensembl's REST API
GMAF	minor_allele, minor_allele_freq	
*_maf	*_allele, *_maf	
cDNA_position	cdna_start, cdna_end	
CDS_position	cds_start, cds_end	
Protein_position	protein_start, protein_end	
SIFT	sift_prediction, sift_score	
PolyPhen	polyphen_prediction, polyphen_score	

Statistics



VEP writes an HTML file containing statistics pertaining to the results of your job; it is named **[output_file]_summary.html** (with the default options the file will be named **variant_effect_output.txt_summary.html**). To view it you should open the file in your web browser.

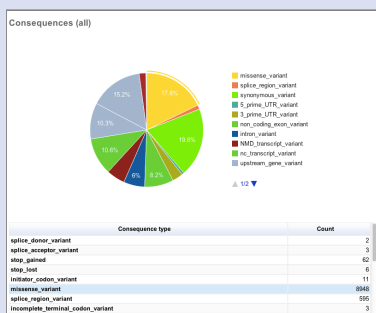
To prevent VEP writing a stats file, use the flag [--no_stats](#). To have VEP write a machine-readable text file in place of the HTML, use [--stats_text](#). To change the name of the stats file from the default, use [--stats_file \[file\]](#).

The page contains several sections:

General statistics

Charts and tables

	
Links <ul style="list-style-type: none"> Top of page View on Ensembl General details Genes and proteins Transcript variants Comparisons with other species Annotations and tracks Genomic context Search by description Search by ID number 	VEP core statistics VEP version: 72 (72) Cache/Database: infoensembl, ensembl, ensembl_havana, ensembl72 Species: human_hg38 Chromosomal line options: chr1 chr2 chr3 chr4 chr5 chr6 chr7 chr8 chr9 chr10 chr11 chr12 chr13 chr14 chr15 chr16 chr17 chr18 chr19 chr20 chr21 chr22 chr23 chr24 chr25 chr26 chr27 chr28 chr29 chr30 chr31 chr32 chr33 chr34 chr35 chr36 chr37 chr38 chr39 chr40 chr41 chr42 chr43 chr44 chr45 chr46 chr47 chr48 chr49 chr50 chr51 chr52 chr53 chr54 chr55 chr56 chr57 chr58 chr59 chr60 chr61 chr62 chr63 chr64 chr65 chr66 chr67 chr68 chr69 chr70 chr71 chr72 chr73 chr74 chr75 chr76 chr77 chr78 chr79 chr80 chr81 chr82 chr83 chr84 chr85 chr86 chr87 chr88 chr89 chr90 chr91 chr92 chr93 chr94 chr95 chr96 chr97 chr98 chr99 chr100 chr101 chr102 chr103 chr104 chr105 chr106 chr107 chr108 chr109 chr110 chr111 chr112 chr113 chr114 chr115 chr116 chr117 chr118 chr119 chr120 chr121 chr122 chr123 chr124 chr125 chr126 chr127 chr128 chr129 chr130 chr131 chr132 chr133 chr134 chr135 chr136 chr137 chr138 chr139 chr140 chr141 chr142 chr143 chr144 chr145 chr146 chr147 chr148 chr149 chr150 chr151 chr152 chr153 chr154 chr155 chr156 chr157 chr158 chr159 chr160 chr161 chr162 chr163 chr164 chr165 chr166 chr167 chr168 chr169 chr170 chr171 chr172 chr173 chr174 chr175 chr176 chr177 chr178 chr179 chr180 chr181 chr182 chr183 chr184 chr185 chr186 chr187 chr188 chr189 chr190 chr191 chr192 chr193 chr194 chr195 chr196 chr197 chr198 chr199 chr200 chr201 chr202 chr203 chr204 chr205 chr206 chr207 chr208 chr209 chr210 chr211 chr212 chr213 chr214 chr215 chr216 chr217 chr218 chr219 chr220 chr221 chr222 chr223 chr224 chr225 chr226 chr227 chr228 chr229 chr230 chr231 chr232 chr233 chr234 chr235 chr236 chr237 chr238 chr239 chr240 chr241 chr242 chr243 chr244 chr245 chr246 chr247 chr248 chr249 chr250 chr251 chr252 chr253 chr254 chr255 chr256 chr257 chr258 chr259 chr260 chr261 chr262 chr263 chr264 chr265 chr266 chr267 chr268 chr269 chr270 chr271 chr272 chr273 chr274 chr275 chr276 chr277 chr278 chr279 chr280 chr281 chr282 chr283 chr284 chr285 chr286 chr287 chr288 chr289 chr290 chr291 chr292 chr293 chr294 chr295 chr296 chr297 chr298 chr299 chr300 chr301 chr302 chr303 chr304 chr305 chr306 chr307 chr308 chr309 chr310 chr311 chr312 chr313 chr314 chr315 chr316 chr317 chr318 chr319 chr320 chr321 chr322 chr323 chr324 chr325 chr326 chr327 chr328 chr329 chr330 chr331 chr332 chr333 chr334 chr335 chr336 chr337 chr338 chr339 chr340 chr341 chr342 chr343 chr344 chr345 chr346 chr347 chr348 chr349 chr350 chr351 chr352 chr353 chr354 chr355 chr356 chr357 chr358 chr359 chr360 chr361 chr362 chr363 chr364 chr365 chr366 chr367 chr368 chr369 chr370 chr371 chr372 chr373 chr374 chr375 chr376 chr377 chr378 chr379 chr380 chr381 chr382 chr383 chr384 chr385 chr386 chr387 chr388 chr389 chr390 chr391 chr392 chr393 chr394 chr395 chr396 chr397 chr398 chr399 chr400 chr401 chr402 chr403 chr404 chr405 chr406 chr407 chr408 chr409 chr410 chr411 chr412 chr413 chr414 chr415 chr416 chr417 chr418 chr419 chr420 chr421 chr422 chr423 chr424 chr425 chr426 chr427 chr428 chr429 chr430 chr431 chr432 chr433 chr434 chr435 chr436 chr437 chr438 chr439 chr440 chr441 chr442 chr443 chr444 chr445 chr446 chr447 chr448 chr449 chr450 chr451 chr452 chr453 chr454 chr455 chr456 chr457 chr458 chr459 chr460 chr461 chr462 chr463 chr464 chr465 chr466 chr467 chr468 chr469 chr470 chr471 chr472 chr473 chr474 chr475 chr476 chr477 chr478 chr479 chr480 chr481 chr482 chr483 chr484 chr485 chr486 chr487 chr488 chr489 chr490 chr491 chr492 chr493 chr494 chr495 chr496 chr497 chr498 chr499 chr500 chr501 chr502 chr503 chr504 chr505 chr506 chr507 chr508 chr509 chr510 chr511 chr512 chr513 chr514 chr515 chr516 chr517 chr518 chr519 chr520 chr521 chr522 chr523 chr524 chr525 chr526 chr527 chr528 chr529 chr530 chr531 chr532 chr533 chr534 chr535 chr536 chr537 chr538 chr539 chr540 chr541 chr542 chr543 chr544 chr545 chr546 chr547 chr548 chr549 chr550 chr551 chr552 chr553 chr554 chr555 chr556 chr557 chr558 chr559 chr560 chr561 chr562 chr563 chr564 chr565 chr566 chr567 chr568 chr569 chr570 chr571 chr572 chr573 chr574 chr575 chr576 chr577 chr578 chr579 chr580 chr581 chr582 chr583 chr584 chr585 chr586 chr587 chr588 chr589 chr590 chr591 chr592 chr593 chr594 chr595 chr596 chr597 chr598 chr599 chr600 chr601 chr602 chr603 chr604 chr605 chr606 chr607 chr608 chr609 chr610 chr611 chr612 chr613 chr614 chr615 chr616 chr617 chr618 chr619 chr620 chr621 chr622 chr623 chr624 chr625 chr626 chr627 chr628 chr629 chr630 chr631 chr632 chr633 chr634 chr635 chr636 chr637 chr638 chr639 chr640 chr641 chr642 chr643 chr644 chr645 chr646 chr647 chr648 chr649 chr650 chr651 chr652 chr653 chr654 chr655 chr656 chr657 chr658 chr659 chr660 chr661 chr662 chr663 chr664 chr665 chr666 chr667 chr668 chr669 chr670 chr671 chr672 chr673 chr674 chr675 chr676 chr677 chr678 chr679 chr680 chr681 chr682 chr683 chr684 chr685 chr686 chr687 chr688 chr689 chr690 chr691 chr692 chr693 chr694 chr695 chr696 chr697 chr698 chr699 chr700 chr701 chr702 chr703 chr704 chr705 chr706 chr707 chr708 chr709 chr710 chr711 chr712 chr713 chr714 chr715 chr716 chr717 chr718 chr719 chr720 chr721 chr722 chr723 chr724 chr725 chr726 chr727 chr728 chr729 chr730 chr731 chr732 chr733 chr734 chr735 chr736 chr737 chr738 chr739 chr740 chr741 chr742 chr743 chr744 chr745 chr746 chr747 chr748 chr749 chr750 chr751 chr752 chr753 chr754 chr755 chr756 chr757 chr758 chr759 chr760 chr761 chr762 chr763 chr764 chr765 chr766 chr767 chr768 chr769 chr770 chr771 chr772 chr773 chr774 chr775 chr776 chr777 chr778 chr779 chr780 chr781 chr782 chr783 chr784 chr785 chr786 chr787 chr788 chr789 chr790 chr791 chr792 chr793 chr794 chr795 chr796 chr797 chr798 chr799 chr800 chr801 chr802 chr803 chr804 chr805 chr806 chr807 chr808 chr809 chr810 chr811 chr812 chr813 chr814 chr815 chr816 chr817 chr818 chr819 chr820 chr821 chr822 chr823 chr824 chr825 chr826 chr827 chr828 chr829 chr830 chr831 chr832 chr833 chr834 chr835 chr836 chr837 chr838 chr839 chr840 chr841 chr842 chr843 chr844 chr845 chr846 chr847 chr848 chr849 chr850 chr851 chr852 chr853 chr854 chr855 chr856 chr857 chr858 chr859 chr860 chr861 chr862 chr863 chr864 chr865 chr866 chr867 chr868 chr869 chr870 chr871 chr872 chr873 chr874 chr875 chr876 chr877 chr878 chr879 chr880 chr881 chr882 chr883 chr884 chr885 chr886 chr887 chr888 chr889 chr890 chr891 chr892 chr893 chr894 chr895 chr896 chr897 chr898 chr899 chr900 chr901 chr902 chr903 chr904 chr905 chr906 chr907 chr908 chr909 chr910 chr911 chr912 chr913 chr914 chr915 chr916 chr917 chr918 chr919 chr920 chr921 chr922 chr923 chr924 chr925 chr926 chr927 chr928 chr929 chr930 chr931 chr932 chr933 chr934 chr935 chr936 chr937 chr938 chr939 chr940 chr941 chr942 chr943 chr944 chr945 chr946 chr947 chr948 chr949 chr950 chr951 chr952 chr953 chr954 chr955 chr956 chr957 chr958 chr959



Varaints by chromosome

Chromosome	Count
1	450
2	350
3	250
4	250
5	300
6	250
7	250
8	250
9	250
10	250
11	300
12	300
13	50
14	250
15	250
16	250
17	250
18	250
19	250
20	250
21	150
22	150

Distribution of variants on chromosome 1

Distribution of variants across chromosomes

For any questions not covered here, please send an email to the Ensembl [developer's mailing list](#) (public) or contact the [Ensembl Helpdesk](#) (private).

General questions

Q: Why has my insertion/deletion variant encoded in VCF disappeared from the VEP output?

Ensembl treats unbalanced variants differently to VCF - your variant hasn't disappeared, it may have just changed slightly! You can solve this by giving your variants a unique identifier in the third column of the VCF file. See [here](#) for a full discussion.

Q: Why don't I see any co-located variants when using species X?

Ensembl only has variation databases for a subset of all Ensembl species - see [this document](#) for details.

Q: Why do I see multiple known variants mapped to my input variant?

VEP compares you input to known variants from the Ensembl variation database. In some cases one input variant can match multiple known variants:

- Germline variants from dbSNP and somatic mutations from COSMIC may be found at the same locus
- Some sources, e.g. HGMD, do not provide public access to allele-specific data, so an HGMD variant with unknown alleles may colocate with one from dbSNP with known alleles
- Multiple alternate alleles from your input may match different variants as they are described in dbSNP

See [here](#) for a full discussion.

Q: VEP is not assigning a frequency to my input variant - why?

VEP's cache contains frequency data only for variants and alleles imported into Ensembl's variation database. See [here](#) for a full discussion.

Q: Why do I see so many lines of output for each variant in my input?

While it would be convenient to have a simple, one word answer to the question "What is the consequence of this variant?", in reality biology is not this simple! Many genes have more than one transcript, so VEP provides a prediction for each transcript that a variant overlaps. VEP has options to help select results according to your requirements; the [--canonical](#) and [--ccds](#) options indicate which transcripts are canonical and belong to the CCDS set respectively, while [--pick](#), [--per_gene](#), [--summary](#) and [--most_severe](#) allow you to give a more summary level assessment per variant.

Furthermore, several "compound" consequences are also possible - if, for example, a variant falls in the final few bases of an exon, it may be considered to affect a splicing site, in addition to possibly affecting the coding sequence.

Q: How do I reduce VEP's memory requirement?

There are a number of ways to do this-

1. Ensure your input file is sorted by location. This can greatly reduce memory requirements and runtime
2. Consider reducing the buffer size. This reduces the number of variants annotated together in a batch and can be modified in both command line and web interfaces. Reducing buffer size may increase run time.
3. Ensure you are only using the options you need, rather than `--everything`. Some data-rich options, such as regulatory annotation have an impact on memory use

Web VEP questions

Q: How do I access the web version of the Variant Effect Predictor?

You can find the web VEP on the [Tools](#) page.

Q: Why is the output I get for my input file different when I use the web VEP and command line VEP?

Ensure that you are passing equivalent arguments to the script that you are using in the web version. If you are sure this is still a problem, please report it on the [ensembl-dev](#) mailing list.

Command line VEP questions

Q: How can I make VEP run faster?

There are a number of factors that influence how fast VEP runs. Have a look at our [handy guide](#) for tips on improving VEP runtime.

Q: Why do I see "N" as the reference allele in my HGVS strings?

Q: Why do I see the following error (or similar) in my VEP output?

```
substr outside of string at /nfs/users/nfs_w/wm2/Perl/ensembl-variation/modules/Bio/Ensembl/Vari
Use of uninitialized value $ref_allele in string eq at /nfs/users/nfs_w/wm2/Perl/ensembl-variati
Use of uninitialized value in concatenation (.) or string at /nfs/users/nfs_w/wm2/Perl/ensembl-v
```

Both of these error types are usually seen when using a [FASTA file](#) for retrieving sequence. There are a couple of steps you can take to try to remedy them:

1. The index alongside the FASTA can become corrupted. Delete [fastafilename].index and re-run VEP to regenerate it. By default this file is located in your \$HOME/.vep/[species]/[version]_[assembly] directory.
2. The FASTA file itself may have been corrupted during download; delete the fasta file and the index and re-download (you can use the [VEP installer](#) to do this).
3. Older versions of BioPerl (1.2.3 in particular is known to have this) cannot properly index large FASTA files. Make sure you are using a later (>=1.6) version of BioPerl. The [VEP installer](#) installs 1.6.924 for you.

If you still see problems after taking these steps, or if you were not using a FASTA file in the first place, please [contact us](#).

Q: Why do I see the following warning?

```
WARNING: Chromosome 21 not found in annotation sources or synonyms on line 160
```

This can occur if the chromosome names differ between your input variant and any annotation source that you are using (cache, database, GFF/GTF file, FASTA file, custom annotation file). To circumvent this you may provide VEP with a [synonyms file](#). A synonym file is included in VEP's cache files, so if you have one of these for your species you can use it as follows:

```
./vep -i input.vcf -cache -synonyms ~/.vep/homo_sapiens/102_GRCh38/chr_synonyms.txt
```

The file consists of lines containing pairs of tab-separated synonyms. Order is not important as synonyms can be used in both "directions".

Q: Can I get gnomAD or ExAC allele frequencies in VEP?

Yes, see [this guide](#).

Q: Why do I see the following error?

```
Could not connect to database homo_sapiens_core_63_37 as user anonymous using [DBI:mysql:database=
Unknown MySQL server host 'ensembldb.ensembl.org' (2) at $HOME/src/ensembl/modules/Bio/Ensembl/I

----- EXCEPTION -----
MSG: Could not connect to database homo_sapiens_core_63_37 as user anonymous using [DBI:mysql:da
Unknown MySQL server host 'ensembldb.ensembl.org' (2)
```

By default VEP is configured to connect to the public MySQL server at ensembl.mysql.ensembl.org. Occasionally the server may break connection with your process, which causes this error. This can happen when the server is busy, or due to various network issues. Consider using a [local copy of the database](#), or the [caching system](#).

Q: Can I use VEP on Windows?

Yes - see the [documentation](#) for a few different ways to get the VEP running on Windows.

Q: Can I download all of the SIFT and/or PolyPhen predictions?

The Ensembl Variation database and the human VEP cache file contain precalculated SIFT and PolyPhen-2 predictions for every possible amino acid change in every translated protein product in Ensembl. Since these data are huge, we store them in a compressed format. The best approach to extract them is to use our Perl API.

The format in which the data are stored in our database is described [here](#)

The simplest way to access these matrices is to use an API script to fetch a ProteinFunctionPredictionMatrix for your protein of interest and then call its 'get_prediction' method to get the score for a particular position and amino acid, looping over all possible amino acids for your position. There is some detailed documentation on this class in the API documentation [here](#).

You would need to work out which peptide position your codon maps to, but there are methods in the [TranscriptVariationAllele](#) class that should help you (probably translation_start and translation_end).