Variant Effect Predictor (VEP)

A Practical Introduction

Flavio Lombardo

Introduction

Somatic vs Germline

- A germline mutation is inherited by the individual from birth. They can be oncogenic (ie Rb = retinoblastoma)
- A somatic mutation, or acquired mutation, happens in somatic cells instead of germ cells and will not pass to offspring (ie TP53).

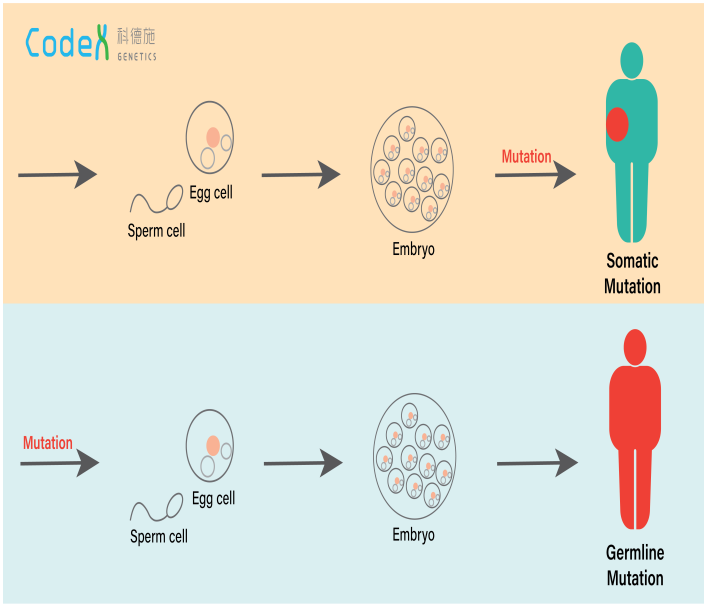

Figure 1: Germline vs somatic

How do mutations appear?

- Mutations occur due to replication errors (1 nucleotide per ~10^4) but with reparation mechanisms (1 error per $10^{9\text{-}10}10$)
- As a consequence of DNA damage (~70,000 nucleotide lesions or modifications per day)
  - Exposure to mutagens such as UV, smoking increases the mutations' frequency

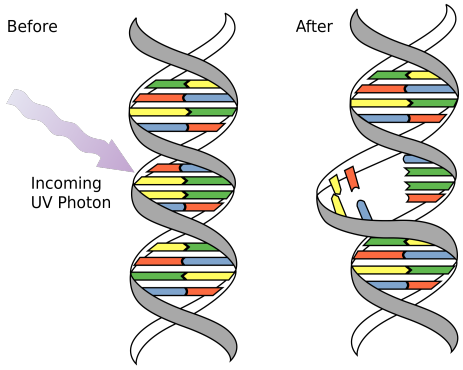

Figure 2: Example UV damage

Key points: - https://www.nature.com/articles/s41576-021-00376-2

Can the cell repair them?

Yes, there are many mechanisms of DNA repair. Some of the most common are:

- **NER (Nucleotide Excision Repair)**: Repairs bulky lesions such as thymine dimers caused by UV radiation. This mechanism involves the removal of a short single-stranded DNA segment containing the damage, followed by DNA synthesis using the complementary strand as a template.
- **MMR (Mismatch Repair)**: Fixes replication errors such as base mismatches or insertion/deletion loops. MMR recognizes the newly synthesized strand and corrects errors by removing the incorrect nucleotides and replacing them.
- **BER (Base Excision Repair)**: Repairs single-base lesions such as oxidative damage, alkylation, or deamination. This involves the removal of damaged bases by specific glycosylases, followed by the excision of the resulting abasic site.
- **HR (Homologous Recombination)**: Repairs double-strand breaks using a homologous sequence as a template, typically from a sister chromatid. This is an error-free repair mechanism.

- **NHEJ (Non-Homologous End Joining)**: Repairs double-strand breaks without the need for a homologous template. While quicker, this method is error-prone and can lead to insertions or deletions.

Visualizing DNA Repair Mechanisms

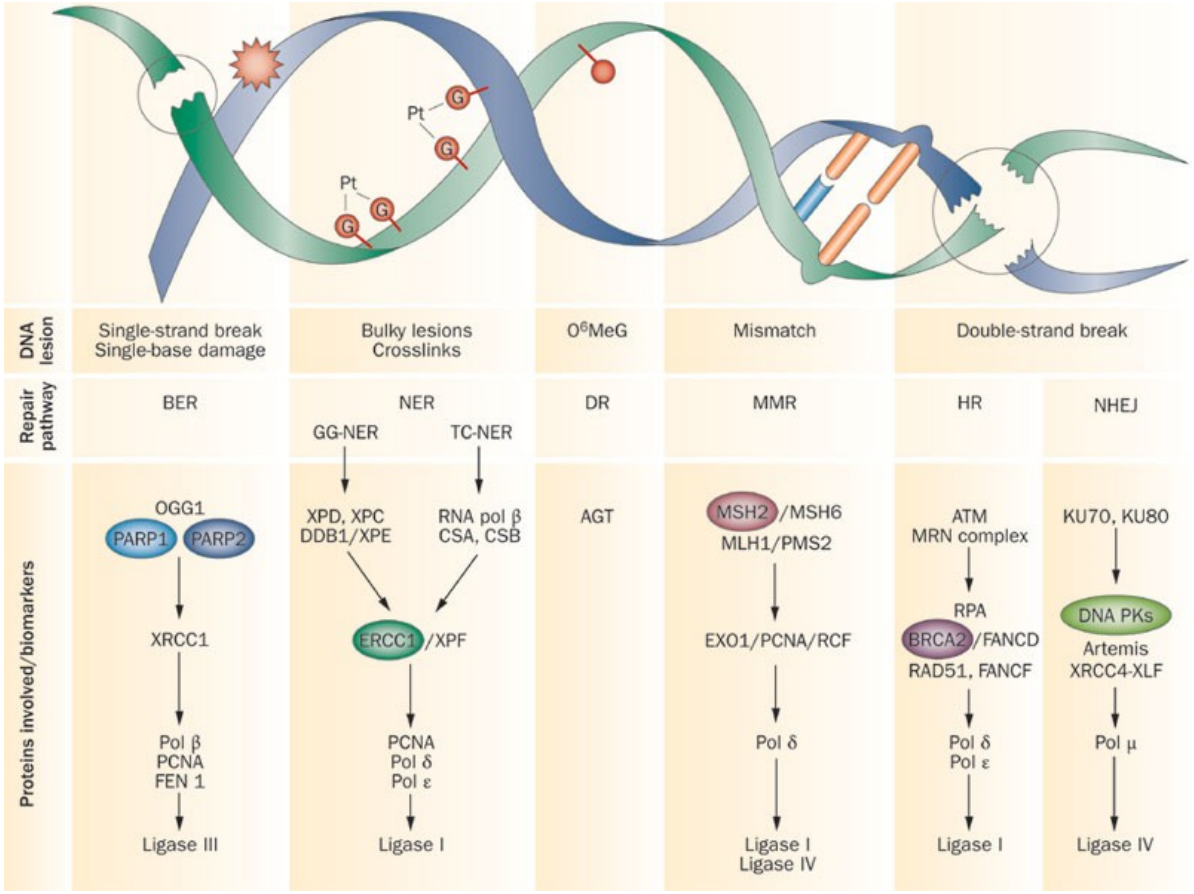

Figure 3: DNA Repair Mechanisms

Key points: - https://www.nature.com/articles/nrclinonc.2012.3/figures/1

Acquisition of mutation in cancer

Key points: - https://www.nature.com/articles/nature07943

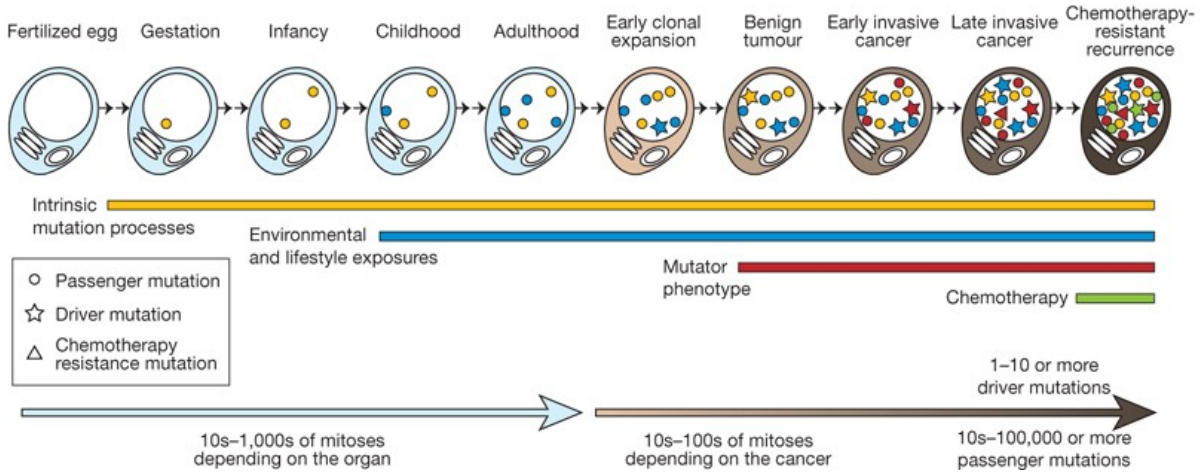

Figure 4: Cancer somatic mut

Concept of driver mutation

Some mutations are more important than others for tumor progression. Perhaps they are more disruptive and detrimental for the cell to harbor. Those can be observed in multiple cancer types (think of TP53 or BRCA1/2).

> Not all the driver mutations are known and not all the driver mutations are really always drivers.

What Defines a Driver Mutation?

> Not all mutations are driving the tumorigenesis!

- Functional Impact:
  - Driver mutations confer a selective growth advantage to cells, promoting tumor development and progression. These mutations often affect genes regulating cell cycle, apoptosis, DNA repair, and immune evasion.
  - Example pathways: MAPK, PI3K/AKT, and WNT signaling.
- Recurrent Patterns Across Tumors:
  - Frequently observed across different cancer types (e.g., TP53 in over 50% of cancers).
  - Some drivers are specific to tissue types (e.g., EGFR in lung cancer, KIT in gastrointestinal stromal tumors, IDH1 in gliomas).

Key points: - Not all HIGH impact variants from VEP are driver mutations, integration of evidence-based databases such as COSMIC, ClinVar and OncoKB (and literature) is required. However HIGH and MODERATE impact mutations can be driver mutations

Types of Driver Mutations

- Oncogenes:
  - Gain-of-function mutations in genes like KRAS, BRAF, and EGFR drive tumor growth by promoting uncontrolled cell division or survival.
- Tumor Suppressors:
  - Loss-of-function mutations in genes like TP53, RB1, and PTEN impair cellular mechanisms that prevent tumor formation.
- Mutator Genes:
  - Mutations in genes like MLH1 or MSH2 lead to genomic instability, enabling the accumulation of additional mutations.

How can we identify what's a mutation's role?

- Can we somehow quantify its importance?

VEP (Variant Effect Predictor)

(a) Ensembl-VEP

Figure 5

What is VEP?

- 2010: The first version of VEP was introduced as part of the Ensembl project.
- 2012-2015 Functional prediction scores were introduced.
- 2017 Other databases are now part of the VEP, such as COSMIC, ClinVar, gnomAD. The tool was made more generalizable and personalizable with the addition of VEP's plugin feature.
- VEP is 100% written in perl VEP Github

Key points: - VEP is widely used in research and clinical settings - Provides comprehensive variant annotation - Especially useful for large-scale analysis

Important databases

- COSMIC What it tells us: Somatic mutations found in cancer Example output: COSV59384583; OCCURENCE=1(skin) Interpretation: Mutation seen in skin cancer; frequency helps assess if likely driver
- ClinVar What it tells us: Clinical significance of variants Example output: Pathogenic/Likely_pathogenic Interpretation: Strong evidence for disease causation
- gnomAD What it tells us: Population frequency Example output: AF=0.00001 Interpretation: Very rare variant (<0.01% frequency)
- AlphaMissense What it tells us: Predicted functional impact Example output: likely_pathogenic Interpretation: AI model predicts damaging effect

What does VEP do?

Imagine thousands of variants to annotate manually to understand what each of them does, searching in different databases, searching in the literature. It would take weeks and it would be error prone. Automation of these tasks reduces the time-to- discovery

- Powerful tool for annotating genomic variants (both somatic and germline)
- Essential in cancer genomics and clinical applications
- Helps understand functional impact of mutations

What is VEP NOT doing for you?

VEP is a great tool, but it does not remove the work from the scientists. Automation simplifies the process but it can give false positives and false negatives. If the experimental setup is poor, the results most likely will be poor quality too.

- Calling variants
- Magic bullet
- Knowing everything

- Brainstorming, reiteration and analysis
- Discovery phase

What is not well understood?

There are no databases investigating those aspects of biology that might play an important role in the cancer development. After all it is estimated that >97% of all the mutations are "passenger events" and they do not have direct impact on the tumor growth.

- The effects of epigenetic changes
- Glycosilations
- Transposomes (like LINE-1)
- SV (inversion for example)
- Alternative splicing impacts
- Non-coding RNAs and their alterations
- Tumor Microenvironment
- Microbiome
- Etc.

In summary: Why do we use VEP?

We can answer questions like:

- How damaging is a certain somatic mutation?
- What is the impact in a particular cancer type?
- Is the mutation known for that cancer?
- Are there therapeutic implications?

Installation & Setup

1. Installer Script
2. Manual Installation
3. Docker
4. Conda/Bioconda

Installation using conda

- Conda/Bioconda

```
# Installing Conda #https://docs.anaconda.com/miniconda/
mkdir -p ~/miniconda3
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda3/m
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
rm ~/miniconda3/miniconda.sh
source ~/miniconda3/bin/activate
conda init --all

# Using Conda
conda create -n vep samtools python=3.10 ensembl-vep=113
# conda install -c bioconda ensembl-vep # if installing in current env

# Activate VEP's env
conda activate vep
```

```
mamba create -n vep samtools python=3.10 ensembl-vep=113 -c bioconda -c
conda-forge
```

VEP's Basic Configuration

```
# Test installation
vep --help

# Download cache files, it takes a long time and ~/.vep (~25GB)
vep_install -a cf -s homo_sapiens -y GRCh38 # download precomputed human data

# if install official plugins
vep_install -a p --PLUGINS list # a for action LIST akk plugins
```

```
# vep_install -a p --PLUGINS all # install all plugins
# vep_install -a p --PLUGINS dbNSFP,... # install specific plugins
```

if you want to specify a specific folder to store the VEP's cache: `-c ~/vep_cache`

Example plugin installation

SubsetVCF for example does not need additional data, it works directly on the VEP's output

```
  - installing "SubsetVCF"
  - add "--plugin SubsetVCF" to your VEP command to use this plugin
  - OK
```

And others like REVEL need additional data to work properly

```
 - installing "REVEL"
 - This plugin requires data
 - See /scicore/home/heinzelv/lombardo/.vep/Plugins/REVEL.pm for details
 - OK
```

VEP Plugins: Manual install

Plugins can enhance VEP's capabilities and can add additional depth of information to the annotations **(with the price of more complexity)**.

If this command does not work

```
# if installing the official plugins
vep_install -a p # a for action
```

Please use this manual option

```
# clone repository
git clone https://github.com/Ensembl/VEP_plugins.git

# make a folder in the ~/.vep folder
mkdir -p ~/.vep/Plugins/ # Assuming your cache folder is in your $HOME

# Copy all the files with extension *.pm to the Plugins folder
cp VEP_plugins/*.pm ~/.vep/Plugins/
```

Core Concepts

Transcript Selection

VEP options for handling multiple transcripts:

- `--pick`: One consequence per variant
- `--pick_allele`: One per variant allele
- `--pick_allele_gene`: One per variant allele per gene
- `--per_gene`: One per gene
- `--flag_pick`: Flags selected while keeping others

Consequence Priority

Top 10 most severe consequences:

1. Transcript ablation
2. Splice acceptor/donor variant
3. Stop gained
4. Frameshift variant
5. Stop lost
6. Start lost
7. Transcript amplification
8. Inframe insertion/deletion
9. Missense variant
10. Protein altering variant

VEPs consequences

SIF (Sequence Ontology Impact Factors) Categories

HIGH Impact:

Likely to cause a severe effect on protein structure/function.

- Examples:
  - Frameshift variants
  - Nonsense mutations (e.g. stop codon gained)
  - Essential splice-site mutations (e.g. disruption of canonical splice sites).

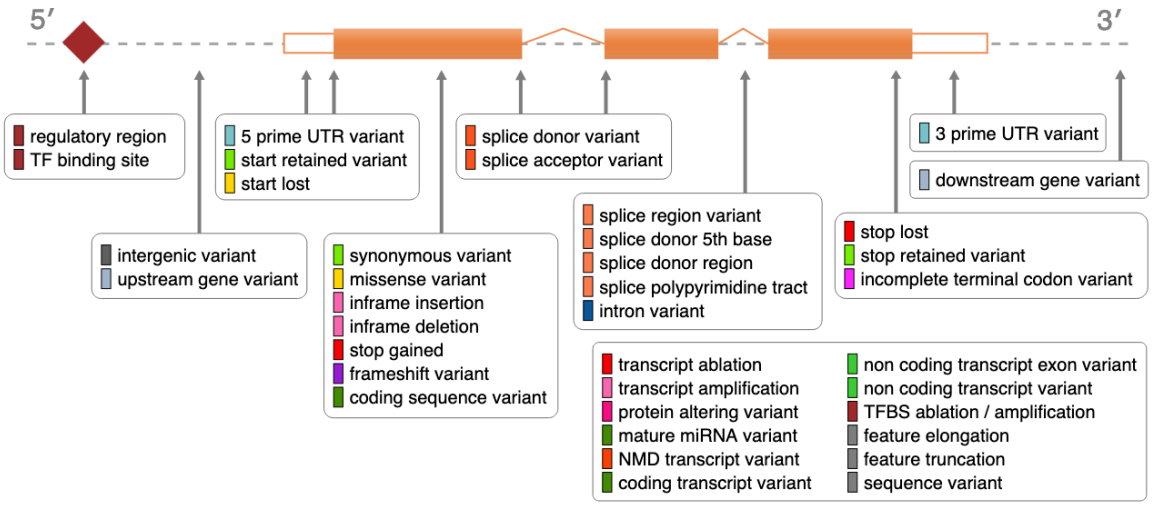

Figure 6: Cancer somatic mut

MODERATE Impact:

Non-disruptive changes that might affect protein structure or function.

- Examples:
  - Missense variants (amino acid substitutions).
  - In-frame insertions or deletions.

LOW Impact:

Variants that are less likely to have a significant effect on protein function.

- Examples:
  - Synonymous changes (no amino acid alteration, [Redundancy]).
  - Changes in non-coding regions (outside essential regulatory domains).

MODIFIER Impact:

Usually non-coding or intergenic variants with no expected impact on the protein but might influence gene regulation.

- Examples:
  - Intronic variants (non-coding).

– Variants in 5' or 3' UTR regions.

Interpretation of HIGH vs. MODERATE Impact

- HIGH: Typically results in loss of protein function or dominant-negative effects. These mutations are often more critical for cancer development and progression (e.g., TP53 inactivation).
- MODERATE: Produces proteins with altered but functional activity, often conferring selective growth advantages under specific contexts (e.g., KRAS or PIK3CA oncogenic activation).

LOW effects

- splice_donor_5th_base_variant A sequence variant that causes a change at the 5th base pair after the start of the intron in the orientation of the transcript SO:0001787 Splice donor 5th base variant LOW
- splice_region_variant A sequence variant in which a change has occurred within the region of the splice site, either within 1-3 bases of the exon or 3-8 bases of the intron SO:0001630 Splice region variant LOW
- splice_donor_region_variant A sequence variant that falls in the region between the 3rd and 6th base after splice junction (5' end of intron) SO:0002170 Splice donor region variant LOW
- splice_polypyrimidine_tract_variant A sequence variant that falls in the polypyrimidine tract at 3' end of intron between 17 and 3 bases from the end (acceptor -3 to acceptor -17) SO:0002169 Splice polypyrimidine tract variant LOW
- incomplete_terminal_codon_variant A sequence variant where at least one base of the final codon of an incompletely annotated transcript is changed SO:0001626 Incomplete terminal codon variant LOW
- start_retained_variant A sequence variant where at least one base in the start codon is changed, but the start remains SO:0002019 Start retained variant LOW
- stop_retained_variant A sequence variant where at least one base in the terminator codon is changed, but the terminator remains SO:0001567 Stop retained variant LOW
- synonymous_variant A sequence variant where there is no resulting change to the encoded amino acid SO:0001819 Synonymous variant LOW

PolyPhen-2 (another predictor included in VEP):

- Predicts impact of amino acid substitutions
- Analyzes both structure and evolutionary conservation
- Specific to human proteins
- Scores range from 0.0 (benign) to 1.0 (damaging)

Categories:

- **Probably Damaging** (score > 0.85)
  - High confidence prediction of affecting protein function
- **Possibly Damaging** (score 0.15-0.85)
  - Moderate prediction of damaging effects
- **Benign** (score < 0.15)
  - Likely to lack phenotypic effect

Key points: - Complements SIFT predictions - Used in conjunction with other predictors - Important for missense variant interpretation

Practical Exercises

Example 1: Basic Analysis

Input VCF:

```
##fileformat=VCFv4.2
#CHROM  POS     ID      REF  ALT   QUAL  FILTER  INFO
chr7       55242465  rs121913529  A    T     .     PASS    .
chr17      7577121   rs28934578    C    T     .     PASS    .
chr13      32936646  rs28897743    C    T     .     PASS    .
```

Results:

Then we get something more informative like:

- BRAF: p.Val600Glu (melanoma)
- TP53: p.Arg248Trp (multiple cancers)
- BRCA2: missense variant

VCF

Header:

- Metadata lines start with ##
- Column header starts with #
- Contains format version, reference genome info, tool info

- Required Fields:
  - CHROM - chromosome
  - POS - position
  - ID - variant identifier
  - REF - reference allele
  - ALT - alternate allele(s)
  - QUAL - quality score
  - FILTER - filter status
  - INFO - additional annotations

INFO field in a VCF file

This is coming from Mutect2

```
AS_FilterStatus=SITE;AS_SB_TABLE=70,82|1,2;DP=161;ECNT=1;GERMQ=93;MBQ=37,39;MFRL=267,318
```

Parameter	Description	Value Example
AS_FilterStatus	Site-level filter status	SITE
AS_SB_TABLE	Strand bias table	70,82|1,2 (ref fwd, ref rev|alt fwd, alt rev)
DP	Total depth	161 reads
ECNT	Number of events in this haplotype	1
GERMQ	Germline variant quality	93
MBQ	Median base quality	37,39 (ref,alt)
MFRL	Median fragment length	267,318 (ref,alt)
MMQ	Median mapping quality	60,60 (ref,alt)
MPOS	Median position in reads	12
NALOD	Negative log odds of artifact	1.93
NLOD	Normal likelihood for variant	24.63
POPAF	Variant population allele frequency	6
TLOD	Tumor LOD score	6.11

Somatic vs Germline Analysis with VEP

Key Differences

Somatic Analysis

- Focused on cancer-specific variants
- Uses `--check_existing` for known mutations
- Requires COSMIC database annotation
- Lower frequency thresholds

Germline Analysis

- Population frequencies critical (gnomAD, 1000G)
- Higher frequency filtering thresholds
- Disease-specific databases (ClinVar)

Filtering Strategies

- Somatic: Focus on cancer drivers, known hotspots
- Germline: Consider inheritance, population prevalence

Download data to COSMIC and ClinVar

For ClinVar

```
# Download latest ClinVar VCF
wget https://ftp.ncbi.nlm.nih.gov/pub/clinvar/vcf_GRCh38/clinvar.vcf.gz
wget https://ftp.ncbi.nlm.nih.gov/pub/clinvar/vcf_GRCh38/clinvar.vcf.gz.tbi
```

For COSMIC We cannot provide those files (for licenses issues), you need to register and download the files. `CosmicMutantExport.tsv.gz`

You can register here: ___ More in the practical part ___

Example Command for Somatic Analysis

```
vep -i somatic.vcf \
--cache \
--assembly GRCh38 \
--format vcf \
--symbol \
--check_existing \
--pick \
--output_file output.txt
```

instead of symbol one can select `--hgvs` or **both** `--hgvs \--symbol`

Exercise 2: Cancer Analysis

Adding more specific annotations:

```
vep -i input.vcf \
--cache \
--assembly GRCh38 \
--format vcf \
--symbol \
--pick \
--sift b \
--polyphen b \
--force_overwrite \
--output_file output.txt
```

- sift: SIFT is an algorithm for predicting whether a given change in a protein sequence will be deleterious to the function of that protein
- polyphen: PolyPhen-2 predicts the effect of an amino acid substitution on the structure and function of a protein using sequence homology **only for human**

Advanced Topics

VEP Plugins

Essential plugins for cancer analysis:

REVEL

```
--plugin REVEL,/path/to/revel/revel_all_chromosomes.tsv.gz
```

> REVEL paper: An Ensemble Method for Predicting the Pathogenicity of Rare Missense Variants

You can download here the data (~6.5GB): https://sites.google.com/site/revelgenomics/downloads or https://zenodo.org/records/7072866

```
unzip revel-v1.3_all_chromosomes.zip
cat revel_with_transcript_ids | tr "," "\t" > tabbed_revel.tsv
sed '1s/.*/#&/' tabbed_revel.tsv > new_tabbed_revel.tsv
bgzip new_tabbed_revel.tsv
```

Prepare for GRCh38

```
zcat new_tabbed_revel.tsv.gz | head -n1 > h
zgrep -h -v ^#chr new_tabbed_revel.tsv.gz | awk '$3 != "." ' | sort -k1,1 -k3,3n - | cat h -
tabix -f -s 1 -b 3 -e 3 new_tabbed_revel_grch38.tsv.gz
```

Usage:

```
--plugin REVEL,file=/path/to/revel/data.tsv.gz
```

AlphaMissense

> AlphaMissense's Paper: Accurate proteome-wide missense variant effect prediction with AlphaMissense

Download link

Prepare for GRCh38

```
wget https://storage.googleapis.com/dm_alphamissense/AlphaMissense_hg38.tsv.gz
tabix -s 1 -b 2 -e 2 -f -S 1 AlphaMissense_hg38.tsv.gz
```

Run it with VEP

```
--plugin AlphaMissense,file=/full/path/to/file.tsv.gz
```

dbNSFP

> dbNSFP v4 paper: a comprehensive database of transcript-specific functional predictions and annotations for human nonsynonymous and splice-site SNVs

A VEP plugin that retrieves data for missense variants from a tabix-indexed dbNSFP file.

```
--plugin dbNSFP,/path/to/dbNSFP.gz,SIFT_score,HDIV_score
```

Prepare the data

```
version=4.7c
wget ftp://dbnsfp:dbnsfp@dbnsfp.softgenetics.com/dbNSFP${version}.zip
unzip dbNSFP${version}.zip
zcat dbNSFP${version}_variant.chr1.gz | head -n1 > h
```

Prepare for GRCh38

```
zgrep -h -v ^#chr dbNSFP${version}_variant.chr* | sort -k1,1 -k2,2n - | cat h - | bgzip -c >
  tabix -s 1 -b 2 -e 2 dbNSFP${version}_grch38.gz
```

Run it with VEP

`--plugin dbNSFP,/path/to/dbNSFP.gz,LRT_score,GERP++_RS`

Filter Variants with VEP

You can use the tool that is included in the VEP's suite of tools. This tool generally works very well with data that have been VEP-annotated

Filter SIFT deleterious events '`filter_vep -i variant_effect_output.txt -filter "SIFT is deleterious" | grep -v "##" | head -n5` Can be used with pipes, for example (might save memory)

```
vep -i examples/homo_sapiens_GRCh38.vcf --cache --force_overwrite --sift b --canonical --symb
filter_vep --filter "CANONICAL is YES and SIFT is deleterious"
```

Notes

- There are many other plugins that can be used depending on the context and the specific biological question at hand. You can have a look here: Plugins for VEP
- You can run multiple plugins at the same time
- The more plugins the more computationally expensive it could become
- There is a nice `help` included in VEP that can be useful for consultation

Best Practices

Quality Control

- Filter low-quality variants before annotation
- Use matched normal samples when available
- Consider sequencing artifacts
- Document filtering criteria
- Use IGV for confirmation

Annotation Strategy

1. Use multiple prediction algorithms
2. Consider tissue-specific expression
3. Include population frequencies
4. Add clinical annotations
5. Follow standardized guidelines

Points to keep in mind

- Using not only the HIGH impact somatic mutations but also the MODERATE
- Remembering that the "driver" mutation is not necessarily fully representative, we simply do not know many of them
- CNVs (Copy number variants) calling is hard, especially for WES data (WES is only 2% of a genome)
- No tool is perfect, not even Google deepvariant or alphafold! This is especially true for CNV calling
  - If samples' tumor purity is too low, or the quality is too low one should be especially careful with the results
  - Experimental design is very important

Resources

Useful Links

- VEP Documentation
- gnomAD Browser
- COSMIC Database
- ClinVar
- OncoKB

Questions?

Feel free to drop a line in the chat or to contact us.

Contact Information

- VEP Support: https://www.ensembl.org/Help/Contact
- Documentation: http://www.ensembl.org/info/docs/tools/vep/
- GitHub: https://github.com/Ensembl/ensembl-vep
- Bioconductor: https://bioconductor.org/
- Email: flavio.lombardo@unibas.ch

Practical part!