VEST (Variant Effect Scoring Tool) Example workflow

Here is an example on how to run VEST. Additional examples are included in the examples directory in the readme file.

Files required to run VEST:

- The list of mutations to score from the sequencing study formatted in genomic or transcript coordinates

Mutation File formats:

Mutations can be specified as tab delimited lists of transcript and amino acid substitution with codon number ('transcript coordinates') or as genomic location and nucleotide substitution ('genomic coordinates'). In the later case, the –g flag should be used in VEST command lines to indicate that mutations should first be mapped to transcript coordinates. The expected format for each coordinate system is described here:

Transcript coordinates:

- (optional) mutation identifier
- transcript
- amino acid substitution (e.g. G12D or H1047R)

Example of transcript coordinates:

UID / Transcript / AA change

TR1	NM_001126116.1	D127Y
TR2	NM_001144919.1	R162Q
TR3	NM_000321.2	Q702K
TR4	NM_000245.2	A1108S
TR5	NM_004333.4	V600E
TR6	NM_001005862.1	G746V

Genomic Coordinates:

- (optional) mutation identifier
- chromosome (e.g. “chr1”)
- 0-based start position of the nucleotide
- 1-based end position of the nucleotide
- strand of the nucleotides being reported in the following column*
- reference nucleotide
- alternative nucleotide

*If the reference and alternative nucleotides match the forward strand of the reference genome this should be “+”, if they match the reverse strand it should be “-”

Example of genomic coordinates:

UID / Chr. / Start / Stop / Strand / Ref. base / Alt. base

TR1	chr17	7577505	7577506	-	G	T
TR2	chr10	123279679	123279680	-	G	A
TR3	chr13	49033966	49033967	+	C	A
TR4	chr7	116417504	116417505	+	G	T
TR5	chr7	140453135	140453136	-	T	A
TR6	chr17	37880997	37880998	+	G	T

Scoring mutations with VEST:

The RunVest python script is used to score a set of missense mutations with a trained VEST classifier. This script takes only two arguments: the classifier name and the mutation file. If the mutations are in genomic coordinates, the -g flag should be used.

Transcript coordinates:

./RunVest –c ClassifierName MutationFile

or

Genomic coordinates:

./RunVest –c ClassifierName MutationFile -g

Note: ClassifierName is optional and allows user to specify custom VEST classifiers. If the –c argument is left off, the VEST classifier distributed with the software package will be used by default.

Interpreting results:

The file ending with ".output" generated by RunVest is designed for simple parsing such that VEST scores, p-values and FDRs can easily be integrated into a larger spreadsheet with variant annotations. The input mutations can be matched to VEST scores using the mutation identifier field. VEST predictions can then be used to prioritize mutations for further analysis by sorting VEST scores from largest to smallest. This will rank the mutations by similarity to the disease mutation class of the VEST training set versus the neutral class mutations. VEST scores near 1 indicate a functional prediction.

The output file generated by RunVest can be sorted on VEST score at the command line using:

sort -n -k 3 filename.output > filename.output.sorted

Alternatively, it may be convenient to rename the output file with a '.txt' extension and load it with a spreadsheet program, such as Microsoft Excel. After sorting, the top scoring functional candidates will be at the top of the file. In general, the top scoring functional candidates include well-known disease mutations as well as a subset of mutations that have not previously been implicated in the disease under study.