BRIEFINGS IN BIOINFORMATICS. VOL 16. NO 2. 255–264 doi:10.1093/bib/bbu008
Advance Access published on 12 March 2014

Rich annotation of DNA sequencing variants by leveraging the Ensembl Variant Effect Predictor with plugins

Michael Yourshaw, S. Paige Taylor, Aliz R. Rao, Martín G. Martín and Stanley F. Nelson

Submitted: 14th November 2013; Received (in revised form): 21st January 2014

Abstract

High-throughput DNA sequencing has become a mainstay for the discovery of genomic variants that may cause disease or affect phenotype. A next-generation sequencing pipeline typically identifies thousands of variants in each sample. A particular challenge is the annotation of each variant in a way that is useful to downstream consumers of the data, such as clinical sequencing centers or researchers. These users may require that all data storage and analysis remain on secure local servers to protect patient confidentiality or intellectual property, may have unique and changing needs to draw on a variety of annotation data sets and may prefer not to rely on closed-source applications beyond their control. Here we describe scalable methods for using the plugin capability of the Ensembl Variant Effect Predictor to enrich its basic set of variant annotations with additional data on genes, function, conservation, expression, diseases, pathways and protein structure, and describe an extensible framework for easily adding additional custom data sets.

Keywords: *Ensembl Variant Effect Predictor; plugin; annotation; DNA sequencing; database*

INTRODUCTION

The recent development of technology to sequence the entire genome of an individual at moderate cost is revolutionizing clinical genetics and greatly accelerating the discovery of new genetic causes of disease [1, 2]. Next-generation sequencing (NGS) platforms now provide clinical laboratories with the ability to sequence nearly all of the thousands of genes known to be causal of human Mendelian diseases in a single process at a cost comparable with that of sequencing a single disease gene by conventional Sanger sequencing [3]. Similarly, researchers can use NGS for an unbiased examination of all genes and regulatory features to discover the relationship of unsuspected genes and pathways to diseases or traits with unknown causes [4]. In operation, a modern NGS platform typically reads the sequence of >100 million short DNA fragments extracted from an individual's

Corresponding author. Michael Yourshaw, Department of Pediatrics, David Geffen School of Medicine, University of California Los Angeles, Los Angeles, CA, USA. Tel.: +1-310-825-7920; Fax: +1-310-794-5446; E-mail: myourshaw@ucla.edu

Michael Yourshaw (PhD) is a postdoctoral scholar at the Department of Pediatrics, David Geffen School of Medicine, University of California, Los Angeles, CA. His research interests include next-generation sequencing and rare Mendelian disorders of the intestine and nervous system.

S. Paige Taylor is a PhD student at the Department of Human Genetics, David Geffen School of Medicine, University of California, Los Angeles, CA. Her research interests include dissecting the molecular basis of genetic disease, investigating the structure and function of primary cilia and understanding the regulation of developmental signaling pathways.

Aliz R. Rao is a PhD student at the Bioinformatics Interdepartmental Program at the University of California, Los Angeles, CA. Her research interests include improving variant interpretation and gene prioritization techniques and complex psychiatric diseases.

Martín G. Martín (MD, MPP) is a Professor in the Department of Pediatrics, Division of Gastroenterology, David Geffen School of Medicine, University of California, Los Angeles, CA. His research interests include monogenic forms of Pediatric intestinal failure and intestinal stem cell biology.

Stanley F. Nelson (MD) is a Professor in the Department of Human Genetics and the Department of Pathology and Laboratory Medicine, David Geffen School of Medicine, University of California, Los Angeles, CA, and co-director of the UCLA Clinical Genomics Center. His research interests include next-generation sequencing and rare Mendelian disorders including Duchenne muscular dystrophy.

blood, saliva or other tissue. These fragments may have been enriched during library preparation for protein-coding regions (the 'exome') or for targeted regions, such as those known to be involved in a class of diseases. Mature algorithms have been developed to align these short reads to a reference genome, assign read and mapping quality scores and genotype loci that vary from the reference [5]. The output of such a sequencing pipeline is a variant call format (VCF) file [6, 7] that succinctly and systematically describes the genomic position (POS), dbSNP ID, reference (REF) and alternate (ALT) alleles, genotype and other information related to each variant (Figure 1a). Where the entire exome is sequenced, a VCF file typically consists of >20 000 individual protein-coding variant records.

A basic VCF file does not contain most of the information that will be needed by a physician or researcher, such as the transcript and gene that contain the variant; the effect, if any, on protein encoding (synonymous, missense and nonsense) or structure; the likelihood that the variant is damaging; association with diseases or phenotypes; or tissue expression data. There are several applications that can add such annotations to a VCF file, each with strengths and weaknesses, and these have been reviewed elsewhere [5]. One characteristic of most of these tools is that they have little or no flexibility to include customized user-defined annotations. Furthermore, online tools, such as SeattleSeq [8], have the advantage of simplicity of use, but they may not be appropriate for confidential patient data or proprietary intellectual property.

Here we present an approach for developing a custom annotator that can be run on local servers, is not heavily dependent on an outside single researcher or small group for software development and maintenance and has a simple modular mechanism for adding new features. Thus, instead of a stand-alone software package, our goal is to share 'how-to' directions for using the plugin capability of the Ensembl Variant Effect Predictor (VEP) [9] to enrich its basic set of variant annotations with additional data from data sets such as Online Mendelian Inheritance in Man (OMIM), the Human Gene Mutation Database (HGMD pro), the Universal Protein Resource (UniProt), Kyoto Encyclopedia of Genes and Genomes (KEGG) Pathways, RefSeq, the MitoCarta Inventory of Mammalian Mitochondrial Genes, the Catalogue of Somatic Mutations in Cancer (COSMIC), Mouse Genome Informatics (MGI) and the Human Protein Atlas (HPA).

To satisfy the needs of our laboratory research projects and the initiation of the UCLA Clinical Genomics Center, we elected to use the Ensembl database and VEP as the basis of a custom annotator, which we call 'Variant Annotator eXtras' (VAX). Several factors were decisive in adopting this approach. Ensembl, a joint scientific project between the European Bioinformatics Institute and the Wellcome Trust Sanger Institute, provides access to genomic annotation for numerous species stored on a MySQL database that can be accessed programmatically via a Perl application programming interface (API). The database is supported by a large professional organization, is updated regularly and can be accessed remotely or by downloading a local copy [10]. The Ensembl database and VEP have a large and active user community and provide excellent and timely advice and support through the Ensembl developers list (dev@ensembl.org). The VEP is a mature open-source Perl script that can be run locally, connected either to the remote Ensembl database or a local copy thereof or with some limitations used with a local cache. Without any modifications, VEP produces many useful annotations, including genes affected by the variants, consequence of variants on the protein sequence, minor allele frequencies in the population and Sorting Intolerant From Tolerant (SIFT)/PolyPhen scores. VEP plugins are a powerful way to extend, filter and manipulate the output of the VEP, and form the foundation of our methods for integrating diverse data sets into our VAX annotation pipeline.

With the guidelines we present here, a research laboratory or clinical sequencing center, with access to a modest data processing infrastructure and having easily acquired basic Perl and SQL programming skills, can implement a custom annotation system similar to VAX. The following sections describe (i) installation of the data and programs from Ensembl that are required to run the VEP and plugin basics, (ii) methods for altering VEP output for downstream entry into a relational database, (iii) enriching basic annotations using Ensembl, (iv) examples of how to implement several useful annotations from non-Ensembl databases and (v) additional considerations for variant analysis. Computer code for the installer and the modules described herein is in the Supplementary File (available online at http://bib.oxfordjournals.org/) VAX_DIR.tar.gz, an index for which is in the README. The code is available under a GNU Public License on an 'as-is' basis; users

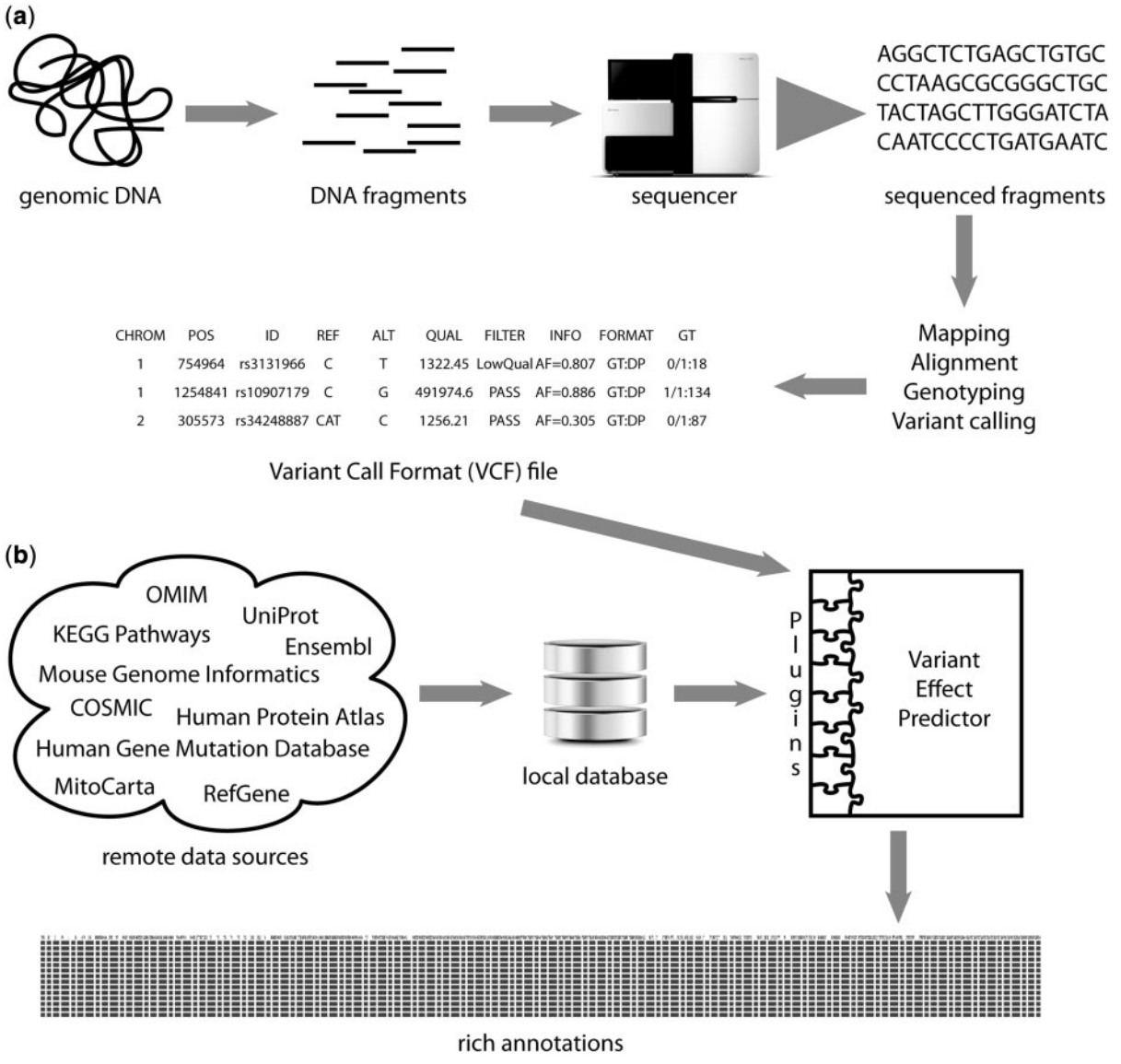

Figure 1: Overview of DNA sequencing and annotation. (**a**) DNA-sequencing pipeline. Fragmented genomic DNA is sequenced by a NGS and aligned to a reference genome. Each locus is genotyped and variants from the reference are output to a VCF file. (**b**) Rich variant annotation. Multiple data sets are stored on a local database server. Modular plugins integrated with the Ensembl VEP create an output file with rich annotations of each variant.

should expect to invest additional effort in adapting this code to their particular systems and needs. The current VAX code is available from https://github.com/myourshaw/vax.

BASE INSTALLATION AND PLUGINS

Ensembl data and VEP

VAX consists of a locally installed MySQL database system, which hosts the Ensembl database and custom data used by the annotator, local installations of the Ensembl Perl API and VEP and a library of custom VEP plugins (Figure 1b). Users must install the Ensembl API and VEP according to instructions on the Ensembl Web site [11, 12]. For improved performance, we recommend that users download the Ensembl databases from the Ensembl FTP site in the form of tab-delimited text files and import them into a local MySQL database, but it is possible to run small numbers of variants using all of our plugins by connecting to the Ensembl public MySQL server [13]. We include, with the attached files, an installation script (VAX_DIR/INSTALL)

that will create a MySQL database, download data and populate the database and install the plugins described herein. We suggest that users not use the VEP fasta files of the reference genome if they experience conflicts with the REF allele in their files. The INSTALL script requires a current version of BioPerl, rather than the version 1.2.3 recommended by Ensembl. Instructions for dealing with this are in the README.

VEP plugin interfaces

Adding functionality to VEP via plugins is straightforward and should be within the abilities of any researcher who has a working knowledge of the Perl programming language. ProteinSeq (Box 1) is a simple example plugin that, for each variant, adds an annotation of the amino acid sequence of the gene. The 'use base' line tells the plugin to inherit the properties of a base class defined by VEP, allowing the module to interact with VEP via well-defined methods. The 'new' method is called once by VEP to initialize the plugin. One-time code, such as establishing a database connection, would also be placed in the 'new' method. The 'version' method returns the version of VEP for which the plugin was designed, and the 'feature_types' method tells VEP only to call the plugin for variants that are within transcripts. The 'get_header_info' method defines the annotation. The 'run' method is where the plugin processes each given variation–allele–feature combination. In the example, $tva gets the transcript variation annotation object from VEP, which contains all information necessary to identify the variant's genomic context. The plugin uses the $tva object to access the related translation object and then returns the amino acid sequence of the translation to VEP, where the annotation will be output in the Extra column. There are a number of additional simple examples of plugins available through the Ensembl github Web site [14].

Database connection

For convenience and efficiency, we implemented a single plugin, vw.pm, to establish a connection to non-Ensembl MySQL databases, and a non–plugin module, VAX.pm, for commonly used functions such as get_unique, which removes duplicates from lists of annotations.

DATABASE-FRIENDLY OUTPUT

By default, VEP places many of its annotations as key–value pairs in the Extra column, for example, the amino acid sequence from the ProteinSeq plugin

Box 1. ProteinSeq plugin. This plugin illustrates the methods that a VEP plugin should implement (new, version, feature_types, get_header_info and run) and demonstrates a simple annotation of the complete amino acid sequence of the protein affected by a variant.

```
packageProteinSeq;

use base qw(Bio::EnsEMBL::Variation::Utils::BaseVepPlugin);

sub new {
my $class = shift;
my $self = $class->SUPER::new(@_);
return $self;
}

sub version { return '73'; }

subfeature_types { return ['Transcript']; }

subget_header_info {
return { ProteinSeq => "amino acid sequence of transcript's translated protein", };
}

sub run {
my ($self, $tva) = @_;
if ( defined $tva->transcript->translation ) {
      return { ProteinSeq => $tva->transcript->translation->seq() };
    }
return {};
}

1;
```

example and a SIFT score of likelihood of protein damage might be represented as 'ProteinSequence=MEAEESE...SLVRDS;SIFT=tolerated(0.34)'. For use in downstream analysis, it may be more convenient to separate data into one column for each annotation and even to create two columns for an annotation like SIFT: one for the verbal description and one for the numerical score. This approach works well for Excel spreadsheets and is almost essential for relational databases. The plugins we describe do not place data in the Extra or INFO columns to avoid creating columns of unwieldy length, instead, they put each item of data in a separate column consistent with standard database conventions. Users who wish, for consistency, to see this data in the Extra columns may add a code at the end of the plugin's run method to return appropriate keys and values, e.g. return {KEY => $value};.

ExtraCols plugin

VEP outputs many annotations, such as Human Genome Variation Society coding and protein sequence names and the SIFT and PolyPhen values, as key–value pairs in the Extra column. The ExtraCols plugin adds selected additional columns to the output file for each key, making the values more easily accessible to database queries. The command at the beginning of that plugin, 'use Bio::EnsEMBL::Variation::Utils::VEP qw(@OUTPUT_COLS);', gives the plugin access to VEP's list of columns that will appear in the output, and the get_header_info method demonstrates the technique for adding additional column headers. In the 'new' method, the plugin places data for each Extra annotation into its own column and also separates text from numbers for the SIFT, PolyPhen and similar scores.

Consequences plugin

By default, VEP produces a comma-separated list of all consequences of a variant in the Consequences column, e.g. 'splice_region_variant,synonymous_variant'. Also, by default the VEP produces multiple lines of data for each input variant, reflecting the variant consequences in the context of each transcript. VEP can identify a 'canonical' (longest) transcript and limit output to the canonical transcript [15]. However, the canonical transcript may not be the only significant transcript, and in the study of human disease, it may be desirable to detect transcripts, whether canonical or not, with damaging consequences. The Consequences plugin creates a column for the most severe consequence and another for a numerical ranking of the severity of the consequence. These columns support downstream analyses on a database system. Because VEP plugins process only one variant/transcript at a time, we find it more efficient to use an SQL database or other post-VEP scripts to identify and prioritize variant/transcript combinations of interest.

VCFCols plugin

VEP's output format does not preserve all of the columns originally present in the input VCF file, and it represents insertions and deletions in a way that is not directly comparable with the VCF standard use of POS, REF and ALT alleles. Some downstream applications require data in the original VCF form. The VCFCols plugin modifies the VEP output format to include all input VCF columns. The 'new' method scans the input VCF file to identify the columns and stores their names in $self->{_vcf_cols} for future use by the 'run' method. The get_header_info method adds output columns, and the 'run' method places data from the original input data line into these columns.

ADDITIONAL ENSEMBL ANNOTATIONS

The real power of plugins starts with the ability to add additional annotations from Ensembl's own rich data collection, as in the ProteinSeq plugin example, above. Two plugins, adapted from the NGS–Single nucleotide polymorphisms (SNP) collection of command-line scripts [16], and a plugin to get gene and variant phenotype data, illustrate this.

Protein plugin

The Protein plugin, derived from NGS–SNP, adds several useful annotations. Protein_Length is helpful for analysis when considering where in the protein a variant falls and the likelihood of protein mutation. Protein_Length_Decrease(%), Protein_Sequence_Lost, Protein_Length_Increase(%) and Protein_Sequence_Gained lend perspective to stop_gained and stop_lost variants. Reference_Splice_Site and Variant_Splice_Site clarify the effect of mutations in the essential splice site region. The plugin computes each of these from the consequence annotation and the amino acid sequence. The Overlapping_Protein_Domains annotation presents all the

domain features annotated in Ensembl's translation object that overlap the variant locus.

Alignment plugin

NGS–SNP's annotations for detailed comparisons with orthologous sequences are, to our knowledge, unique among existing variant annotation tools. The Alignment plugin adapts portions of NGS–SNP to function as a VEP plugin and illustrates the use of the Ensembl comparative genomics (Compara) database, which is not implemented in the basic VEP. The 'new' method accepts additional parameters used by the plugin and establishes connections to the compara database. This plugin calculates three values that are useful for evaluating the conservation of amino acid residues. Alignment_Score_Change is the alignment score for the variant amino acid versus the orthologous amino acids minus the alignment score for the reference amino acid versus the orthologous amino acids. C_blosum is a measure of the conservation of the reference amino acid with the aligned amino acids in orthologous sequences using the C_blosum formula given in [17]. Context_Conservation is the average percent identity obtained when the region of the reference protein containing the SNP-affected residue is aligned with the orthologous region from other species. Additionally, this plugin generates an alignment of amino acids in orthologous species, ordered by evolutionary distance from humans as calculated from the phylogenetic tree obtained from Ensembl [18]. These data are displayed in two compact columns: Amino_Acids_In_Orthologues lists the amino acids and Orthologue_Species lists the species from which sequences were obtained to generate the alignment as well as the three numerical measures.

Phenotypes plugin

The Phenotypes plugin creates columns for phenotypes associated with a gene or variant locus, cancer associations from COSMIC and the public HGMD data set, sourced from Ensembl (a plugin for the commercial HGMD pro data set is discussed below).

EXTERNAL DATABASES

It is possible to access some remote databases directly from within a VEP plugin, but, in our experience, this presents two difficulties: first, throughput may be slow and second, annotating multiple whole exome variant calls may place an undue burden on remote servers. Consequently, we elected to create local copies of external data sets and accepted the burden of performing regular (usually quarterly) updates. The attached INSTALL script largely automates the update process, limiting the human workload to a few hours per year, although inevitable changes to database schema, download procedures and the Ensembl API may need manual intervention. To access external data we use the MySQL ISAM engine, also used by Ensembl, for reasons of user familiarity, speed and cost, but any other database system with a Perl interface can easily connect with VEP plugins using code similar to that in the vw plugin. Using external databases within Ensembl plugins generally requires (i) obtaining data from an external source, (ii) preprocessing the data, (iii) creating and loading MySQL table(s), (iv) creating stored procedures as a secure and consistent interface between database and plugin and (v) developing a plugin to access the database and produce output. Steps 1 through 4 are automated by the INSTALL script.

A basic example of this process is the Mito plugin (Box 2), which produces a column named MT that contains '1' if the gene is annotated by MitoCarta [19] as being found in mitochondria, otherwise blank. The first step is to download the Human.MitoCarta.xls file from [20]. The second, preprocessing, step is to select from the SYM (gene symbol) column only those genes with a '1' in the MITOCARTA_LIST column (indicating strong support of mitochondrial localization) and remove any duplicates. Step 3 involves creating a MySQL table named 'mitocarta_gene' with a single column, 'mito_gene', which contains the selected mitochondrial gene symbols. For robust and secure access, we also create a stored procedure named 'get_mitocarta_gene' that takes a gene symbol as input and returns a list of one or zero matching symbols. Finally, the plugin creates an MT column with its get_header_info method and populates the column for each variant in its 'run' method by querying the database, using the database connection established by the generic vw plugin. Note the use of the $line_hash parameter passed from VEP to store data as a new column in the output line.

GeneIDs plugin

The GeneIDs plugin creates columns for the chromosomal strand containing the transcribed gene, the Ensembl permanent gene identifier, a gene

Box 2. Mito plugin. This plugin illustrates the use of data from an external database (the MitoCarta Inventory of Mammalian Mitochondrial Genes) that is stored on a local MySQL server. Consult the vw.pm file in the supplemental vax.code for details of the database connection.

```
package Mito;

use base qw(Bio::EnsEMBL::Variation::Utils::BaseVepPlugin);
use Bio::EnsEMBL::Variation::Utils::VEP qw(@OUTPUT_COLS);
usevw;

sub new {
my $class = shift;
my $self = $class->SUPER::new(@_);
return $self;
}

sub version { return '73'; }

subfeature_types { return ['Transcript']; }

subget_header_info {
my @new_output_cols = qw( MT );
@OUTPUT_COLS = (@OUTPUT_COLS, @new_output_cols);
return { MT => "annotated as in mitochondrion by MitoCarta", };
}

sub run {
my ($self, $tva, $line_hash) = @_;
my $config = $self->{config};
my $hgnc = $tva->transcript->{_gene_hgnc};
if (defined $hgnc){
        my $query = "CALL $vw::vw_database.get_mitocarta_gene('$hgnc')";
        my $qh = $vw::vw_conn->prepare($query);
        $qh->execute() or die "Unable to execute $query: $DBI::errstr\n";
        my @row = $qh->fetchrow_array();
        if( defined($row[0]) && $row[0] ne '' ) {
            $line_hash->{MT} = '1';
        }
        else {
            $line_hash->{MT} = '';
        }
    }
return {};
}

1;

# SQL stored procedure
#     CREATE DEFINER=`sa`@`%` PROCEDURE `get_mitocarta_gene`(hgncvarchar(15))
#     BEGIN
#     SELECT `mitocarta_gene`.`mito_gene`
#     FROM `vw`.`mitocarta_gene`
#     WHERE `mitocarta_gene`.`mito_gene` = hgnc;
#     END
```

description, a gene summary from RefSeq, the Entrez gene name, the UniProt KB_AC and ID and mitochondrial location. With two exceptions, these columns are populated from information in Ensembl. The RefSeq summary column contains brief summary paragraphs for >5000 well-understood genes, and it is an excellent starting point for an analyst when first confronted with an unfamiliar gene [21]. This plugin is implemented by downloading RefSeqGene and refseqgene*.genomic.gbff.gz from the National Center for Biotechnology Information ftp site [22] and converting to a database-friendly text file. The MT column is created as described above for the Mito plugin.

OMIM plugin

The OMIM plugin annotates gene associations with Mendelian disorders from OMIM [23]. To build the data set locally, the installer downloads the OMIM data files, converts them to database tables and imports genemap.txt into a MySQL table.

DiseasesPhenotypes plugin

The DiseasesPhenotypes plugin creates a convenience column intended to contain the union of all annotations of disease and phenotype associations from the HGMD, OMIM, Phenotypes and UniProt plugins. This plugin provides an easy way for a relational database to scan all of these annotations with one query. It is an example of how one plugin can create a column that will be populated by other plugins. Consult the code of the OMIM plugin for an example of how to interface with the DiseasesPhenotypes plugin.

HGMD plugin

The HGMD plugin obtains gene and locus disease associations from the commercial professional version of the HGMD® database (BIOBASE Biological Databases). This plugin requires a local installation of the database as documented in its distribution. Stored procedures for access to the data by the plugin are included in the plugin's internal documentation.

HPA plugin

The HPA plugin creates annotations of gene expression by tissue, cell type and subcellular location from the HPA [24]. The installer downloads the normal_tissue and subcellular_location tables and imports them into MySQL.

KEGG plugin

The KEGG plugin annotates gene participation in molecular pathways and interaction networks from the KEGG Pathway database [25]. Owing to usage restrictions on the database, the installer creates the gene_pathways table with a Perl script from the KEGG server.

MousePhenotypes plugin

Especially when a gene has no known associated phenotypes in humans, it is important to consider whether there is a phenotype in a model organism. The MGI database [26] has extensive annotations of phenotypes observed in mice when genes orthologous to human genes are mutated or knocked out. The MousePhenotypes plugin annotates variants with all known mouse phenotypes in equivalent human genes. The installer downloads the HMD_HumanPhenotype.rpt and VOC_MammalianPhenotype.rpt files, cleans up the format and loads the two tables into MySQL. Many other model organisms have similar human gene/phenotype data sets for which VEP plugins could be developed by similar methods.

UniProt plugin

The UniProt plugin creates columns for many of the extensive protein annotations in the UniProt database [27]. These include VARIANT, MUTAGEN, SITES, OTHER_OVERLAPPING_FEATURES, ALLERGEN, ALTERNATIVE_PRODUCTS, CATALYTIC_ACTIVITY, CAUTION, COFACTOR, DE, DEVELOPMENTAL_STAGE, DISEASE, DOMAIN, ENZYME_REGULATION, FUNCTION, GeneNames, INDUCTION, INTERACTION, KEGG, KEYWORDS, MIM_gene, MIM_phenotype, MISCELLANEOUS, PATHWAY, Pathway_Interaction, PE, POLYMORPHISM, PTM, Reactome, RecName, RefSeq_NM, RefSeq_NP, RNA_EDITING, SEQUENCE_CAUTION, SIMILARITY, SUBCELLULAR_LOCATION, SUBUNIT, TISSUE_SPECIFICITY, UCSC and WEB_RESOURCE. The plugin will annotate both proteins and specific protein features that overlap the variant. The installer downloads the UniProt data in its unique format, converts the data to database tables, loads them into MySQL and creates Ensembl transcript ID indexed tables. As explained in the README, the installer requires that the SWISS::Knife Perl module [28] be installed and linked in the PERL5LIB environment variable.

ADDITIONAL CONSIDERATIONS

VEP plugin–based annotations can work well for producing output that will be reviewed directly in an Excel spreadsheet. Even greater analytical power is available if annotated variants are used as input to downstream applications, possibly including relational databases. The ExtraCols, VCFCols and Consequences plugins enable output formatting that is more conducive to relational database analysis by ensuring that each column contains a single

discrete unit of data. Two other issues with the way the VCF input format provides for genotypes of multiple samples can complicate use of the data in some downstream applications. First, the VCF format requires that each genotyped sample's genotype and related data be stored in one column per sample. Second, because samples may have different ALT alleles at a given locus, the ALT column must contain a list of all observed alleles. Although recent versions of the VEP now have some provision for multiple alleles, we choose to preprocess VCF files before running the VEP to create two files: (i) a VCF file without sample genotype columns and with each ALT allele on a separate line (this file serves as input to VEP); and (ii) a file listing each sample's genotype on a separate line. After annotation by VEP, we relate sample genotypes and annotations in a database system. During preprocessing we also split the input into a number of smaller files to run VEP in parallel for faster throughput. None of these steps are essential to operation of a successful annotation pipeline based on VEP plugins.

CONCLUSION

A local installation of the Ensembl databases and Perl API provides a robust and flexible framework for annotating DNA sequencing variants from many different data sources using VEP plugin modules. We have outlined the design and usage of VEP plugins for a number of widely used databases. In addition, our plugins and installer can serve as a template for the design of new plugins that incorporate annotations from any external data set that is kept in a flat file or relational database, such as the Zebrafish Model Organism database [29] and the Rat Genome database [30]. We have used the VAX system for the discovery of the causes of rare Mendelian diseases and genes involved in neurological, gastroenterological and psychiatric disorders [4, 31, 32]. VAX is used routinely for Clinical Laboratory Improvement Amendments/College of American Pathologists-accredited whole exome sequencing by the UCLA Clinical Genomics Center, which has processed >1000 exomes to date [33]. An example of VAX output, including the use of the VAX plugins and several of the example plugins from Ensembl, is in the test subdirectory of the VAX_DIR.tar.gz file.

SUPPLEMENTARY DATA

Supplementary data are available online at http://bib.oxfordjournals.org/.

Key Points

- Richly annotated variants produced by next-generation sequencing are the foundation of modern clinical sequencing and gene discovery research.
- Ensembl Variant Effect Predictor (VEP) plugins provide a robust and flexible framework for annotating DNA sequencing variants.
- VEP plugins are Perl scripts that can use the extensive data in Ensembl, such as comparative genomics and variant annotations.
- Custom VEP plugins can associate variants with data from diverse external sources.
- An annotation pipeline incorporating VEP plugins is within the reach of small laboratories and clinical sequencing centers.

Acknowledgments

The authors wish to acknowledge Bret Harry for systems administration support and Valerie Arboleda, Vivian Chang, Ascia Eskin, Hane Lee and Kevin Squire for usage testing.

FUNDING

California Institute of Regenerative Medicine (CIRM), RT2-01985 to M.G.M.

References

1. Gonzaga-Jauregui C, Lupski JR, Gibbs RA. Human genome sequencing in health and disease. *Annu Rev Med* 2012;**63**:35–61.
2. Shendure J, Ji H. Next-generation DNA sequencing. *Nat Biotechnol* 2008;**26**:1135–45.
3. Bamshad MJ, Ng SB, Bigham AW, *et al*. Exome sequencing as a tool for Mendelian disease gene discovery. *Nat Rev Genet* 2011;**12**:745–55.
4. Wan J, Yourshaw M, Mamsa H, *et al*. Mutations in the RNA exosome component gene EXOSC3 cause pontocerebellar hypoplasia and spinal motor neuron degeneration. *Nat Genet* 2012;**44**:704–8.
5. Pabinger S, Dander A, Fischer M, *et al*. A survey of tools for variant analysis of next-generation genome sequencing data. *Brief Bioinform* 2014;**15**(2):256–78.
6. Genomes Project. *VCF (Variant Call Format) version 4.1.* http://www.1000genomes.org/wiki/Analysis/VariantCall Format/vcf-variant-call-format-version-41 (20 January 2014, date last accessed).
7. Danecek P, Auton A, Abecasis G, *et al*. The variant call format and VCFtools. *Bioinformatics* 2011;**27**:2156–8.
8. Ng SB, Turner EH, Robertson PD, *et al*. Targeted capture and massively parallel sequencing of 12 human exomes. *Nature* 2009;**461**:272–6.

9. McLaren W, Pritchard B, Rios D, *et al*. Deriving the consequences of genomic variants with the Ensembl API and SNP effect predictor. *Bioinformatics* 2010;**26**:2069–70.
10. Flicek P, Ahmed I, Amode MR, *et al*. Ensembl 2013. *Nucleic Acids Res* 2013;**41**:D48–55.
11. Ensembl. *Variant Effect Predictor: Download and Install*. http://www.ensembl.org/info/docs/tools/vep/script/vep_download.html (20 January 2014, date last accessed).
12. Ensembl. *API Installation*. http://www.ensembl.org/info/docs/api/api_installation.html (20 January 2014, date last accessed).
13. Ensembl. *Public MySQL Server*. http://www.ensembl.org/info/data/mysql.html (20 January 2014, date last accessed).
14. Ensembl. *Ensembl-variation/VEP.plugins*. https://github.com/ensembl-variation/VEP_plugins (20 January 2014, date last accessed).
15. Hubbard TJ, Aken BL, Ayling S, *et al*. Ensembl 2009. *Nucleic Acids Res* 2009;**37**:D690–7.
16. Grant JR, Arantes AS, Liao X, *et al*. In-depth annotation of SNPs arising from resequencing projects using NGS-SNP. *Bioinformatics* 2011;**27**:2300–1.
17. Kowarsch A, Fuchs A, Frishman D, *et al*. Correlated mutations: a hallmark of phenotypic amino acid substitutions. *PLoS Comput Biol* 2010;**6**:e1000923.
18. Ensembl. *Ensembl/UCSC Phylogenetic Tree*. http://tinyurl.com/ensembltree (20 January 2014, date last accessed).
19. Pagliarini DJ, Calvo SE, Chang B, *et al*. A mitochondrial protein compendium elucidates complex I disease biology. *Cell* 2008;**134**:112–23.
20. Broad Institute. *Human MitoCarta: 1013 Mitochondrial Genes*. http://www.broadinstitute.org/pubs/MitoCarta/human.mitocarta.html (20 January 2014, date last accessed).
21. Pruitt KD, Tatusova T, Brown GR, *et al*. NCBI Reference Sequences (RefSeq): current status, new features and genome annotation policy. *Nucleic Acids Res* 2012;**40**:D130–5.
22. NCBI. *RefSeqGene*. http://ftp.ncbi.nih.gov//refseq/H_sapiens/RefSeqGene/gene_RefSeqGene (20 January 2014, date last accessed).
23. Online Mendelian Inheritance in Man OMIM®. *Online Mendelian Inheritance in Man, OMIM®*. http://omim.org/ (4 November 2013, date last accessed).
24. Uhlen M, Oksvold P, Fagerberg L, *et al*. Towards a knowledge-based Human Protein Atlas. *Nat Biotechnol* 2010;**28**:1248–50.
25. Kanehisa M, Goto S, Kawashima S, *et al*. The KEGG databases at GenomeNet. *Nucleic Acids Res* 2002;**30**:42–6.
26. Eppig JT, Blake JA, Bult CJ, *et al*. The Mouse Genome Database (MGD): comprehensive resource for genetics and genomics of the laboratory mouse. *Nucleic Acids Res* 2012;**40**:D881–6.
27. Apweiler R, Martin MJ, O'Donovan C, *et al*. Update on activities at the Universal Protein Resource (UniProt) in 2013. *Nucleic Acids Res* 2013;**41**:D43–7.
28. Barrell D, Hermjakob H, Kersey P, *et al*. SWISS::Knife. http://sourceforge.net/projects/swissknife/ (20 January 2014, date last accessed).
29. Howe DG, Bradford YM, Conlin T, *et al*. ZFIN, the Zebrafish Model Organism Database: increased support for mutants and transgenics. *Nucleic Acids Res* 2013;**41**:D854–60.
30. Laulederkind SJ, Hayman GT, Wang SJ, *et al*. The Rat Genome Database 2013—data, tools and users. *Brief Bioinform* 2013;**14**:520–6.
31. Kerner B, Rao AR, Christensen B, *et al*. Rare genomic variants link bipolar disorder to CREB regulated intracellular signaling pathways. *Front Psychiatry* 2013;**4**:154.
32. Yourshaw M, Solorzano-Vargas RS, Pickett LA, *et al*. Exome sequencing finds a novel PCSK1 mutation in a child with generalized malabsorptive diarrhea and diabetes insipidus. *J Pediatr Gastroenterol Nutr* 2013;**57**:759–67.
33. Lee H, Nelson SF. Rethinking clinical practice: clinical implementation of exome sequencing. *Per Med* 2012;**9**:785–7.