*Database*, 2017, 1–11
doi: 10.1093/database/bax020
Database update

Database update

# Ensembl core software resources: storage and programmatic access for DNA sequence and genome annotation

**Magali Ruffier[1,*], Andreas Kähäri[1], Monika Komorowska[1], Stephen Keenan[1], Matthew Laird[1], Ian Longden[1], Glenn Proctor[1], Steve Searle[2], Daniel Staines[1], Kieron Taylor[1], Alessandro Vullo[1], Andrew Yates[1], Daniel Zerbino[1] and Paul Flicek[1,2,*]**

[1]European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Genome Campus, Hinxton, Cambridge CB10 1SD, UK and [2]Wellcome Trust Sanger Institute, Wellcome Genome Campus, Hinxton, Cambridge CB10 1SA, UK

*__Corresponding author:__ Tel: +44 (0)1223 494475; Fax: +44 (0)1223 494468; Email: mr6@ebi.ac.uk; Correspondence may also be addressed to Paul Flicek. Tel: +44 (0)1223 492581; Fax: +44 (0)1223 494468; Email: flicek@ebi.ac.uk

Present address: Andreas Kähäri, Institutionen för celloch molekylärbiologi, Uppsala University, Husargatan 3, Uppsala 752 37, Sweden.

Present address: Monika Komorowska, Eli Lilly and Company Limited, Lilly House, Priestley Road, Basingstoke, Hampshire RG24 9NL, UK.

Present address: Ian Longden, NanoporeTech, One Kendall Square, Building 200, Suite B2005, Cambridge, MA 02139, USA.

Present address: Glenn Proctor, Eagle Genomics, The Biodata Innovation Centre, Wellcome Genome Campus, Hinxton, Cambridge, CB10 1DR, UK.

Citation details: Ruffier,M., Kähäri,A., Komorowska,M., *et al.* Ensembl core software resources: storage and programmatic access for DNA sequence and genome annotation. *Database* (2017) Vol. 2017: article ID bax020; doi:10.1093/database/bax020.

Received 28 October 2016; Revised 7 February 2017; Accepted 20 February 2017

## Abstract

The Ensembl software resources are a stable infrastructure to store, access and manipulate genome assemblies and their functional annotations. The Ensembl 'Core' database and Application Programming Interface (API) was our first major piece of software infrastructure and remains at the centre of all of our genome resources. Since its initial design more than fifteen years ago, the number of publicly available genomic, transcriptomic and proteomic datasets has grown enormously, accelerated by continuous advances in DNA-sequencing technology. Initially intended to provide annotation for the reference human genome, we have extended our framework to support the genomes of all species as well as richer assembly models. Cross-referenced links to other informatics resources facilitate searching our database with a variety of popular identifiers such as UniProt and RefSeq. Our comprehensive and robust framework storing a large diversity of genome

annotations in one location serves as a platform for other groups to generate and maintain their own tailored annotation. We welcome reuse and contributions: our databases and APIs are publicly available, all of our source code is released with a permissive Apache v2.0 licence at http://github.com/Ensembl and we have an active developer mailing list (http://www.ensembl.org/info/about/contact/index.html).

**Database URL:** http://www.ensembl.org

## Introduction

The Ensembl database infrastructure was originally designed to support the storage and distribution of the reference assembly produced by the Human Genome Project (HGP) (1). Today, for any species or clade of interest, an Ensembl Core MySQL relational database can store the assembly structure, genomic sequence and genome annotations. Other Ensembl databases specialize in variation and phenotype data (2); whole genome alignments and other comparative genomics information (3); and epigenomic data and regulatory annotation (4).

Application Programming Interfaces (APIs) implemented in Perl provide primary access to these databases. The APIs offer simple functions to manage database connections, formulate queries and provide an object relational mapping (ORM) layer (i.e. they convert genomic annotations stored in a database into Perl objects). Key biological concepts such as genes, transcripts and exons are thus modelled as specialized objects, with links to their components and other related elements. In contrast, simpler features such as CpG islands can be stored as generic region objects located on the genome. Thus, the Ensembl Core database and API is the foundation for all Ensembl data resources (3–7) as well as our web browser, http://www.ensembl.org.

The number, complexity and diversity of available genome assemblies have been in constant growth since the early days of Ensembl. For example, our previous description of the Ensembl Core software libraries included a schema to represent genome assemblies resulting from the clone-by-clone based sequencing strategy used in the HGP (8), which was rendered intractable by whole genome shotgun based assembly methods. Starting with the human genome, sequencing projects focused on known haplotypes of the major histocompatibility complex (MHC) (9) or global population variation (10) captured genetic diversity, while essential curation has progressively identified and fixed errors in the reference assembly without updating the entire assembly version (11). Related variation discovery and targeted loci studies have been conducted in a number of species (12–14). At the same time the genomes of many species from across the phylogenetic tree have been sequenced and released, including tens of thousands of bacterial genomes (15).

The growing number of genome sequences has been accompanied by a substantial growth in genomics data resources (16). Established resources such as UniProt (17), RefSeq (18) and the HUGO Gene Nomenclature Committee (HGNC) (19) have increased in depth and value. Their identifiers and terminology have become a critical 'lingua franca' that ties together the field and makes the results of diverse studies quickly and effectively comparable.

To keep up with these developments, the Ensembl Core database and API infrastructure has been extended in many ways. It efficiently manages whole genome shotgun genome assemblies and many species within a single database. It represents modern assembly features, beyond consensus haploid sequence, including alternate sequences, pseudo-autosomal regions (PARs) and assembly corrections (patches), while maintaining a minimal data footprint. Methods to map stored gene identifiers between releases and to link them to external databases are more robust. There exists native API support for polycistronic transcripts and ontologies. We have also adopted alternative storage solutions where our MySQL database approach does not scale for particular data types such as RNA-seq read alignments (20).

We describe here our current methods for representing genome assemblies and annotation for all species. We detail the way we manage transitions of assembly coordinates and stable identifiers (i.e. Ensembl gene, transcript, exon and other ids) across releases. Improvements to the Core Perl API and our methods for cross-referencing identifiers from external databases will also be described.

## Materials and methods

### Representing genomes

The original Ensembl Core database schema, created in 1999, was designed for data from the draft releases of the HGP. As a consequence, the schema modelled the common units of that assembly including bacterial artificial chromosome clones, contigs and chromosomes (8). This model made a number of assumptions: chromosomes were

necessarily composed of contigs, which were necessarily composed of clone sequences. The design became obsolete in various ways: supercontigs could not be modelled; all DNA sequence had to be assigned to clones; and a database could only hold a single assembly version.

The schema was remodelled in 2004 for Ensembl release 20 into six tables as shown in Figure 1 and this part of it has remained stable since. An assembly is now composed of 'sequence regions'. Sequence regions are grouped by coordinate system (e.g. chromosome, contig, plasmid), an optional assembly version and whether they have DNA sequence associated with them. The 'assembly' table defines a hierarchy of sequence regions: any sequence region can be decomposed into component sequence regions much like an AGP file (https://www.ncbi.nlm.nih.gov/assembly/agp/AGP_Specification/) or a record from the INSDC CON division (21). The 'seqlevel' coordinate system is at the bottom of each assembly and refers to regions with DNA sequence (usually contigs). The 'toplevel' coordinate system represents all the regions that cannot be collected into larger regions. Typically, chromosomes are 'toplevel' sequences, as are supercontigs, scaffolds or contigs that have not been mapped to chromosomes. Genomic annotations are stored on 'toplevel' sequence regions for best query performance.

The reference genome has been enriched with a set of ancillary sequences, which we describe using our 'assembly exception' model. Assembly exceptions allow for the efficient storage of alternate loci as well as novel and patch sequences from the Genome Reference Consortium (GRC) (22) by only keeping the changes required to recreate them. Alternate loci are separate representations of regions found in the primary assembly such as the human MHC regions on chromosome 6. Patches either fix errors on the genome assembly or add novel sequence. For example, a patch on chromosome 9 of GRCh38 fixes a local misassembly and allows Ensembl to present the correct annotation for the ABO locus (ENSG00000175164) on a patch (KN196479.1) (Figure 2).

Although the underlying sequence data are stored without duplication, assembly exceptions are presented as whole chromosomes where the key region (i.e. alternate haplotype, sequence patch etc.) has been replaced by different sequence. Thus any feature coordinates on these patches are computed relative to the entire chromosome rather than to the modified region. Similarly, annotated features that are downstream of the exception may have modified coordinates when queried in the context of exception sequence.

For practical reasons, any annotation on the chromosome that is located across an assembly exception is duplicated and stored for both the primary chromosome and on the exception. In particular, multiple copies of the same gene are modelled as separate gene records, some of which may only be found on alternative assembly regions (Figure 2). The resulting multiple gene copies are called

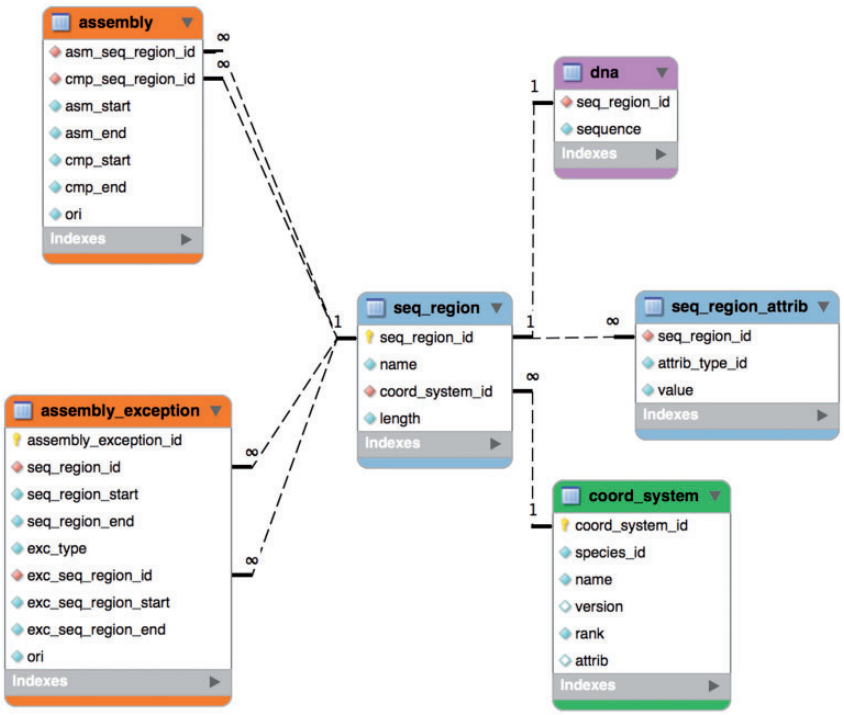


**Figure 1.** The core assembly schema.

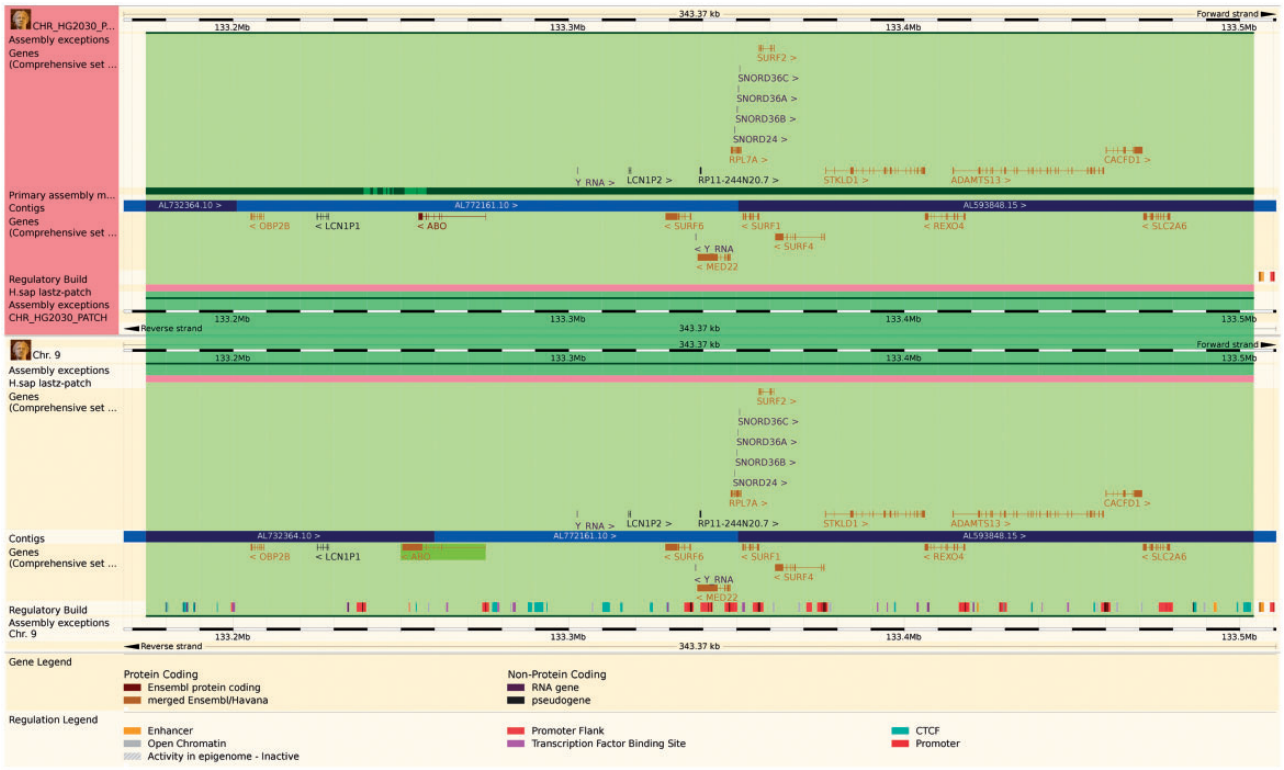


**Figure 2.** The Ensembl web browser can display the differences between a patch region and its equivalent in the primary assembly. Genes that are present in both regions are identified as alt_alleles. (http://e87.ensembl.org/Homo_sapiens/Location/Multi?db=core;g=ENSG00000175164;r=CHR_HG2030_PATCH:133174055-133504218;r1=9:133173980-133504143:1;s1=Homo_sapiens–9).

alternative alleles, or 'alt alleles', and collected together as a group of genes with a nominated reference copy. The alt alleles data structure also stores information about whether the duplicate annotation was due to sequencing error or individual variability and whether it was assigned automatically or manually.

Querying across an assembly exception requires searching separately the main assembly, upstream and downstream of the assembly exception location, and the assembly exception itself. When querying the flanking regions our API automatically handles these operations and discards all duplicated features.

We also use the assembly exception concept to store the PARs, which are the relatively short regions that behave as homologous autosomal regions within the otherwise highly differentiated mammalian sex chromosomes (23). For example, the human Y chromosome shares two regions (10001-2781479 and 56887903-57217415 on the GRCh38 assembly) with the X chromosome. One copy of the annotation and sequence is stored on X with equivalent regions on Y marked as assembly exceptions. When querying across these regions in Y, our API automatically projects any available genomic annotation on X to the corresponding coordinate space on the Y chromosome.

## Perl API

The Perl API is a powerful tool to store and retrieve genome annotation (http://www.ensembl.org/info/docs/Doxygen/core-api/hierarchy.html). The infrastructure relies on a clean separation between data representation and data access. All biological features including genes, transcripts and exons have a dedicated data object describing their attributes and logic. An adaptor class is responsible for querying the underlying databases providing a clean separation between business logic and database persistence and insulating API users from underlying schema changes. Data objects and adaptors are named after the entity they model or retrieve. For example, genes are modelled by the Gene object and are retrieved via the GeneAdaptor. In addition data objects are capable of retrieving additional records from the database via the adaptor framework allowing a number of attributes—such as all transcripts linked to a gene—to be loaded on-demand. Here we present some of the API's characteristics.

## Genomic features

All genomic annotations share some common attributes, such as start and end position on a sequence region, and are defined by the Feature class. More specialized features

are derived from this class and include regulatory features, variants, constrained elements and sequence repeats. The base Feature class offers key operations to retrieve co-located features of a genomic region, map features between assemblies, test object equality, detect feature overlap and project features onto assembly exceptions.

The Feature object hierarchy works in parallel to the adaptor hierarchy rooted by BaseFeatureAdaptor, which provides a framework for developing feature ORMs. The base class implements optimized SQL generation and automatic region expansion should a query span an assembly exception or when features are located on another coordinate system. If required, the coordinates of a retrieved feature are automatically recalculated providing locations relative to the queried region. All derived feature adaptors implement a number of stub methods that specify the tables to query, columns to retrieve and query optimization hints.

Features link back to their sequence region via objects of class Slice, which are portions of sequence regions with a coordinate system characterized by a start, end and orientation. The Slice class also provides a number of shortcut methods to retrieve sequence (repeat masked and reverse complemented) and co-located genomic features. Slice is therefore used as the main entry point to query for annotation by genomic position and holds a number of convenience methods for multiple data set retrieval.

## Efficient feature searching

A major obstacle when storing genomic features in a database is how to ensure the efficient querying of randomly accessed regions. As table sizes tend towards millions of entries, naive solutions scale poorly. Genome binning schemes such as those used by the UCSC Genome Browser (24) and the BAM file format (25) partly address these problems.

We have chosen an alternative indexing strategy: all feature tables are sorted by their sequence region identifier and start coordinate, and a B-tree index is generated over the same fields. Storing related data sequentially ensures that disk seeks are minimized during regional queries. Furthermore, we compute for each table and each coordinate system the maximum feature length and store this value in the 'meta_coord' table. For each genomic region query, a maximal search window for possible start positions is calculated based on the known maximum feature length and the region's start (Figure 3) and queried against the B-tree index. This query will return all data that overlaps the queried genomic range but could also return data that is located only 5′ of the queried start. Ensuring the end of a returned record is greater than or equal to the queried start

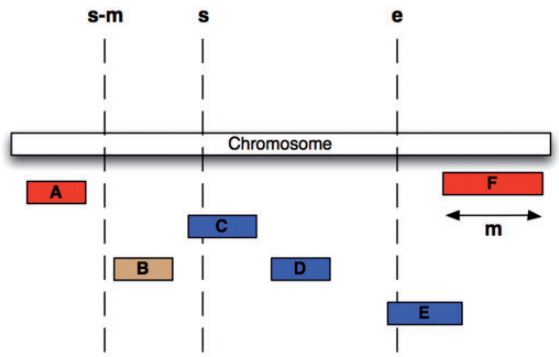


**Figure 3.** Efficient Searching of genomic features To find all features between coordinates s and e (i.e. C, D and E) in the situation were the maximum length of a feature for this coordinate system is m (i.e. the length of feature F), we extract all features whose start lies between s – m and e, then exclude B, since it ends before s.

further restricts the query. The range query performs an order of magnitude faster than querying with just a start and end and performs well in data volumes ranging from 1 to 1 000 000 bp.

## Ensembl database registry

Before retrieving data from the Perl API it is essential to locate all databases to be queried. Depending on the API call this could be a single or multiple databases. The Ensembl database registry solves this problem by providing two major functions. First, it is a global associative array that holds a reference to data adaptors keyed by species name, database type and data type. The registry also holds a list of species name aliases allowing the use of common terms such as 'human' to find a *Homo sapiens* adaptor. Second, the registry can automatically populate itself from a MySQL server (requiring a hostname and username be specified) and it searches for databases corresponding to a single release. Additional routines enable searching across multiple MySQL instances or from an external configuration file.

## Alternative data stores

In recent years, projects such as ENCODE (26), 1000 Genomes (10) and Illumina BodyMap 2.0 have produced a large number of genomic data sets. This situation has given rise to large indexed file formats including BigBed, BigWig (27) and BAM (25). We now use these data stores when existing database solutions fail to scale adequately. The data contained in these files is normally quite static between releases and not relational.

We have developed a metadata file storage system for locating these files called DataFiles. A data file entry describes the assembly version a file maps to, the analysis

method used to generate the data, a name, the file type and an optional versioning flag. The DataFiles API can construct a POSIX path from this information and provide an object capable of reading this data. For example, BAM files trigger creation of an object with bindings to HTSlib (http://www.htslib.org). DataFiles currently supports BAM, BigWig, BigBED and a specialized format BAMCOV, which combines a BAM file and pre-computed read coverage held in a BigWig file.

## Data management for non-vertebrate genomes

*Multi-species databases.* The Ensembl Core infrastructure was designed for vertebrate genome assemblies in general and human in particular and included an assumption that each species would be contained within a single database. As the number of genomes grew, putting each in its own database was putting significant pressure on our MySQL infrastructure and may have eventually exposed underlying file system limitations. To avoid these problems, we devised a method to store multiple genomes in a single Core database. For example, Ensembl Bacteria release 32 (August 2016) contains 41 610 species of bacteria and archaea held in only 173 databases.

Multi-species support involves associating separate coordinate systems to each species. Since every genomic annotation in an Ensembl database is mapped on a sequence region that is itself linked to a coordinate system, selecting the appropriate coordinate system will create species-specific queries. The Core API transparently handles the storage and querying of these multi-species databases by joining back to 'seq_region' and 'coord_system' tables when required. The API avoids these extra steps when dealing with a single-species database, but on a multi-species database (characterized by a name that includes the term 'collection') it runs the required queries to discover and register the species contained. This naming convention is not required if connecting directly to a multi-species database using its name. Controlled vocabulary tables are considered global to all species within a single database. Metadata linked to a species, such as its name, are held in the 'meta' table and are linked to its species identifier.

*Circular chromosomes.* Prokaryote species require native support for circular genomes. Specifically, on sequence regions marked as circular, a feature that passes through the origin of replication (ori) will have an end coordinate 'lower' than its start coordinate. We require that a feature can pass through the ori only once. The ori spanning queries are decomposed by the Core API into two queries: the first from the requested start coordinate to the sequence region's length and the second from position 1 to the requested end coordinate. Features appearing in results of both queries (i.e. those that pass through the ori) are returned once.

*Polycistronic transcripts.* Prokaryote and some eukaryote species also require native support for polycistronic genes. These are genes that are transcribed as a single polycistronic mRNA and undergo trans-splicing to result in multiple polypeptide products (28, 29). The Ensembl database schema was extended to represent polycistronic transcripts using three tables. The 'operon' and 'operon_transcript' tables locate the full range of polycistronic transcripts available and group them under a single regulatory element. Each operon_transcript record is linked to a set of polypeptide encoding genes via the 'operon_transcript_gene' table thereby annotating the available proteins. Every polypeptide encoded by a polycistronic transcript is held as a record in the gene and transcript tables. This allows the Ensembl Core API to translate the resulting transcript into a protein sequence without significant re-engineering. Polycistronic data is currently available for *Escherichia coli str.* 'K-12 substr. MG1655' (http://bacteria.ensembl.org/Escherichia_coli_str_k_12_substr_mg1655/Info/Index/).

## Projecting data between assembly versions and gene set updates

Periodically new assemblies are released for species that have previously been annotated in Ensembl. As these normally incorporate new sequence data and are generally of better quality than previous assembly versions, they tend to be quickly adopted for new genomic analysis efforts. However, because a new assembly will have a different coordinate system, directly comparing the results of analyses on two assembly versions is difficult or impossible. The Ensembl Core database and API store and manage assembly and identifier mapping data to help transition and compare results between assemblies.

*Mapping between assemblies.* We first generate a mapping from the old to the new assembly that takes advantage of shared contigs between the two assembly versions. The contig order, within each chromosome, is compared between versions. If contig accessions and versions are maintained between assembly versions, we look at the versioned INSDC accession; otherwise we generate a MD5 checksum of the underlying sequence. Neighboring contigs with conserved order and orientation are collapsed into a single mapping held at the chromosome level. Gaps in the mapping between chromosomes are filled in by a LASTZ

alignment (30). Finally, the resulting mappings are recorded in the assembly table.

Once a mapping is available between two assemblies, annotations can be transferred from one to another taking into account any insertions or deletions of DNA sequence. This is available from our Perl API via two subroutines: 'project' to report a target coordinate set and 'transform', which generates a new annotation object in the target coordinates. Both are available on any Ensembl Feature class including genes, transcripts, repeats and variants. These subroutines are used when annotating an updated assembly to transfer the existing annotation onto the new assembly for quality control (7). As such, our method attempts to minimize gaps between assemblies and help map gene annotation cleanly between assemblies.

In addition we create chain format files of our assembly version mappings to be used by liftover (31) or CrossMap (32). The latter is utilized for our online Assembly Mapper tool (33). Chain files are available from our FTP (ftp://ftp.ensembl.org/pub/current_assembly_chain) site and include additional mappings that use the UCSC naming scheme for regions (i.e. chr1 rather than 1).

*Mapping stable identifiers.* As new experimental sequence evidence becomes available or when a new genome assembly for a species is published, it is possible to generate updated and more accurate gene annotation. In these cases, linking new gene models to their older versions is particularly useful for reproducibility and other analyses.

The Ensembl gene annotation system, described in detail by Aken *et al.* (7), uses stable identifiers to uniquely identify each feature across releases. The stable ID mapping pipeline ensures the identity of these features is kept consistent across releases, whereas the version is incremented each time a feature is modified. For example, transcript ENST00000262097 (http://e85.ensembl.org/Homo_sapiens/Transcript/Summary?db=core;g=ENSG00000104763;r=8:18056416-18084370;t=ENST00000262097) in Ensembl release 85 has a longer UTR region than the same transcript in release 84 (http://e84.ensembl.org/Homo_sapiens/Transcript/Summary?db=core;g=ENSG00000104763;r=8:18056416-18084370;t=ENST00000262097), hence the version is different but the stable id is the same. As the Ensembl gene annotation is associated directly with an assembly, our challenge is to identify the gene and transcript models that are found on both assemblies and transfer our identifiers between those models. Other Ensembl resources such as GeneTrees and Ensembl Protein Families also use stable identifiers with procedures adapted to the specific data types (3). These pipelines are used to provide identifier continuity between releases. They are not however used to transfer annotations between assemblies, as all annotations are completely recomputed whenever a genome assembly is updated (7).

Our methodology is based on hierarchical identity, starting from exon overlaps: all exons from the source set are compared with all the exons from the target set. The comparison is based on coordinate overlap when both source and target are on the same assembly, and based on sequence alignment when going from one assembly to another. If a pair of exons share 90% or more of their bases they are considered the same and the exon stable id is transferred from source to target. Thus, when using coordinate overlap on the same assembly the new exon must share 90% of the bases to retain the stable id and when comparing between assemblies we require 90% identity in the alignment. The exon stable ID version is incremented if either the coordinates or the sequence is not a perfect match.

Transcript identifiers are then mapped by comparing sets of exons (as defined by the source and target annotation) and looking for sets with 25% or greater intersection. To reduce the chances of incorrect transfer of identifiers, matches are penalized when the compared transcript functions do not match each other. A simple change in function will incur a 10% penalty increasing to 20% if that change causes a significant alteration in assigned function. For example, changing a transcript's function from short nuclear RNA to short cytoplasmic RNA will only incur a 10% penalty since both of these functions are linked to small non-coding RNAs. If a transcript changes from coding to non-coding we apply a 20% penalty since this represents a significant change in assigned function. A transcript version will increment when there is a change in the underlying cDNA sequence or exon-splicing pattern. The identity of a transcript is thus defined by the list of its exon coordinates and its underlying sequence.

Protein identifiers are mapped via the underlying transcript mapping and versions are incremented when protein sequence changes.

Genes identifiers are mapped based on the shared transcripts between source and target data sets. Any stable identifier or version change in the underlying transcripts will cause an increment in the gene's stable identifier version.

In some cases, a source feature can map with comparable scores to two or more features. These ambiguous cases are resolved by comparing the biotypes, the parent and children features or the internal ids. If it is not possible to clearly identify the best candidate, all existing

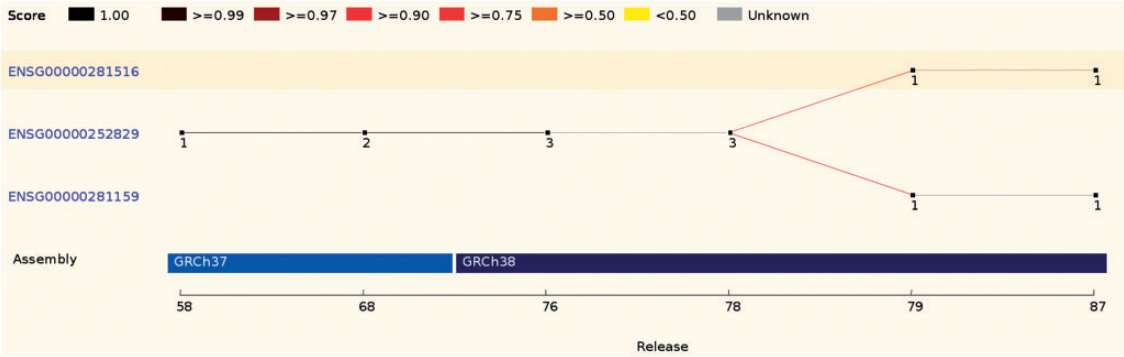


**Figure 4.** This ID History Map for the SCARN4 gene (http://e87.ensembl.org/Homo_sapiens/Gene/Idhistory?g=ENSG00000281516) aligns Ensembl release numbers, genomic assembly versions, and version numbers of that gene across multiple Ensembl IDs. The different updates in the version ID are represented as a chain of small nodes, connected by lines. The colour of the line reflects how well consecutive versions match, for recent releases. If a score was not calculated (typically in older releases), the line is grey.

**Table 1.** A selection of resources mapped by the Ensembl cross reference system indicating the methods used to map, the Ensembl feature mapped to, any additional resources brought in by this association and whether the resource is used to name genes

| Resource | Mapping method | Linked to | Transitive resources | Used for naming |
|---|---|---|---|---|
| HGNC | Direct | Genes | | Yes |
| UniProt Knowledgebase (UniProtKB) | Direct, alignment | Proteins | PDBe | Yes |
| RefSeq mRNA | Location overlap, alignment | Transcripts | EntrezGene | Yes |
| RefSeq Proteins | Alignment | Proteins | | No |
| UniProt Archive (UniParc) | Checksum | Proteins | | No |
| RNACentral | Checksum | Non-coding transcripts | | No |
| MGI | Direct | Genes | | Yes |

IDs are discarded and new stable IDs are assigned (Figure 4).

## External references

Linking Ensembl entities such as genes, transcripts and proteins to external resources such as UniProt, RefSeq, RNACentral (34) and HGNC increases the value of our annotation by enabling cross resource querying (http://www.ensembl.org/Help/View?id=166).

We employ four methods to perform this linkage: direct (mappings provided by a third party), location overlap (annotation overlapping the same genomic location), checksum matches (MD5 digests of sequences) and alignment executed by exonerate (35). Alignment based linkages can be performed between any resource where sequence is attached to an identifier. After our sequence sets have been aligned against an external set, results are limited to the best hit or a range of hits such as the top five. Hits are filtered by their percentage identity and all alignments are subject to a minimum threshold of identity. If we manage to align multiple sequences to the same degree of identity then all are linked. A selection of the resources linked and the mapping method is shown in Table 1. For most resources, we allow a certain amount of mismatching to account for the differences in annotation strategies. For those cases, we provide an alignment score (Figure 5).

Our approach selects the best matches according to a priority rating. Direct links are assumed to have the highest priority followed by location, checksum and then alignment based links. Transitive links are also imported including links from UniProt to PDBe (36). Once all links are assembled we use these to select a name for each gene. If a species has a nomenclature committee such as HGNC or MGI (37) these links will take the highest priority. Otherwise we revert to using Rfam (38), mirbase (39), UniProt or EntrezGene (40) names and, if there is no other option, a clone name. Links are then stored with gene, transcript and protein records. If a link was generated by alignment we retain the percentage identities of the mapping and a CIGAR string representation of the alignment. Alternatively if the link was from an ontology, we record the three letter evidence code assigned to the link as described by the Gene Ontology Consortium such as IEA (Inferred from Electronic Annotation) for unreviewed annotation assigned by a computational method (41).

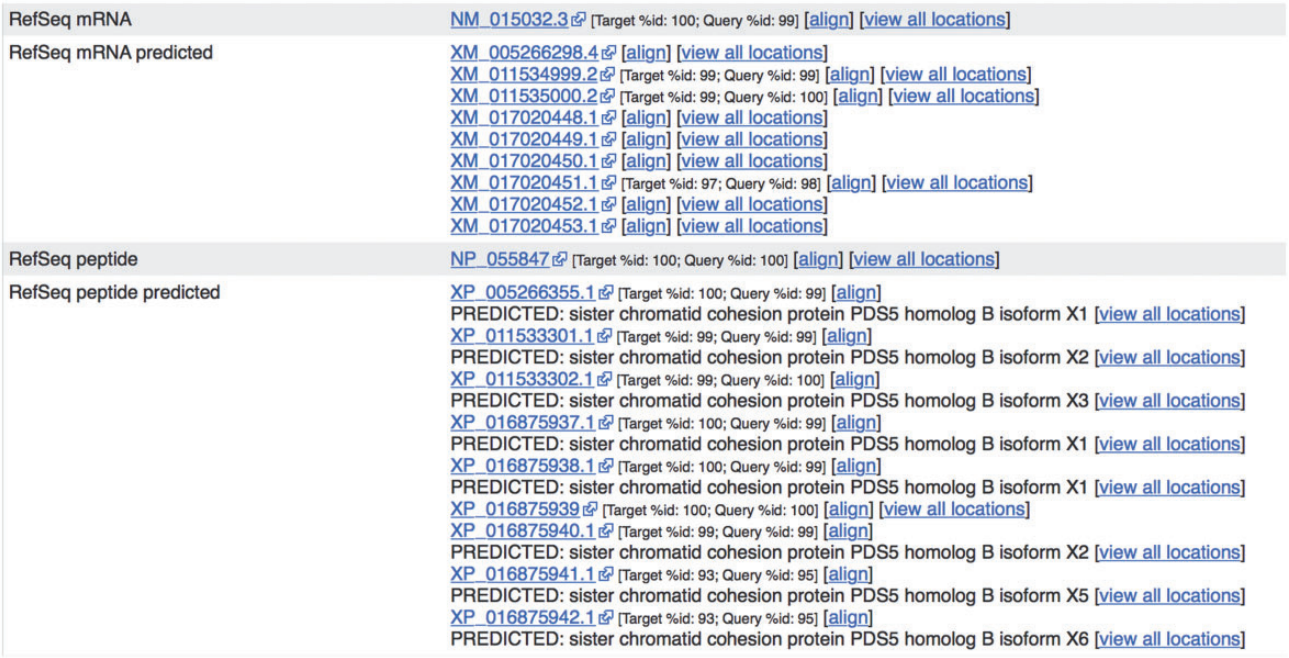


**Figure 5.** Sequences from external sources are aligned against Ensembl features. For ENST00000315596.14 (http://e87.ensembl.org/Homo_sapiens/Transcript/Similarity?db=core;g=ENSG00000083642;r=13:1-50000000;t=ENST00000315596), a number of predicted RefSeq peptide sequences have been aligned with small mismatches. The curated RefSeq peptide NP_055847 aligns perfectly but the corresponding mRNA sequence, NM_015032.3, does not, indicating that there is a difference in the UTR sequence of this transcript.

## Discussion

Over the past several years, we have engineered substantial enhancements to our API without significantly changing the way existing developers interact with it via existing public interfaces. Updates such as multi-species and polycistronic transcript support necessitated a number of schema and code changes, but created minimal API changes. For example, circular genome support required API changes only, while polycistronic transcript support required changes to the database schema but did not alter existing API methods. None of these updates required existing code to be modified, except to use a new feature. Indeed, Ensembl's gene orthology and paralogy annotation pipeline (42) was run on multi-species core databases with no modifications. To implement these changes with such a minimal level of impact requires the level of abstraction described earlier.

Schema changes are made only when necessary and incompatible changes are applied when we feel the current schema falls short of its intended goals. Where possible, migration plans are offered between schemas for a number of releases, giving developers time to migrate. Such was the solution offered when stable identifier tables merged with their parent tables in Ensembl release 65 (December 2011). Where a method is considered no longer applicable or supplanted by a superior method the original is deprecated and developers are pointed to a replacement. An extensive set of unit tests (4873 in Ensembl release 87) ensures that code development does not cause a break in our expected API behaviour. Due to the API's ubiquitous usage within our data production cycles, performance regressions are generally quickly identified and corrected.

The API continues to support novel methods of information delivery across multiple species as highlighted by our REST Service (43). By building on the Perl API, the REST API has a solid platform for development and a stable robust model of data transfer. It is compatible with any Ensembl infrastructure based resource including Ensembl Genomes (44), which has released its own installation of this service. We see the exposure of Ensembl data to alternative programming languages as an important development.

The flexibility of the Ensembl Core infrastructure is most clearly displayed by the many large-scale projects that reuse it, including Ensembl Genomes, Gramene (45), WormBase and WormBase ParaSite (46), VectorBase (47), PomBase (48) and AvianBase (49). The versatility of the Ensembl Core software infrastructure, including the Perl and REST APIs, is further demonstrated by the third party tools that incorporate and extend it (50–52) as well as companion software for creating Ensembl instances (53). These vibrant and active research communities regularly bring in new demands and requirements that, together with the challenges of new data types described above, ensure the development of our Core infrastructure is as active now as it has ever been.

## Conclusions

The Ensembl core API and database is a highly extensible, performant, adaptable and consistent interface to genomic data. This has allowed other projects to create their own resources using the Ensembl infrastructure. The Perl APIs continue to be the primary mechanism of data access within the project, supporting the Ensembl Website and our newer data distribution mechanisms such as the Ensembl REST API. As more species are sequenced and the amount of species-specific data increases, the foundation of the Ensembl core infrastructure will ensure successful scaling to meet the future needs of the genomics community.

## Availability

All Ensembl software is freely available under an Apache 2.0 license and can be downloaded from our website or github repository (http://github.com/Ensembl). Ensembl is updated approximately five times year as a specific numbered release. Starting in 2016, Ensembl releases will be maintained as accessible archive web sites for at least 5 years the date of original release and data will be maintained for at least 10 years.

## Acknowledgements

We acknowledge the work and contributions to the core database infrastructure of Arne Stabenau, Ewan Birney and other early Ensembl developers. We thank all users of the Ensembl infrastructure especially those who have taken the time to contact us with comments, bug reports and suggestions.

## Funding

Ensembl receives majority funding from the Wellcome Trust (WT108749/Z/15/Z) with additional funding for specific project components from the Biotechnology and Biological Sciences Research Council (BB/L024225/1, BB/I025506/1, BB/I025360/2, BB/M011615/1, BB/M01844X/1); Open Targets; the Wellcome Trust (WT200990/Z/16/Z, WT201535/Z/16/Z); the National Institutes of Health (U41HG007234, U41HG007823-S1) and the European Molecular Biology Laboratory. The research leading to these results has received funding from the European Union's Sixth Framework Programme under the thematic area 'Life sciences, genomics and biotechnology for health', contract number LSHG-CT-2003-503265 (BioSapiens). The research leading to these results has received funding from the European Union's Seventh Framework Capacities Specific Programme under grant agreement no. 284209 (BioMedBridges). Funding for open access charge: the Wellcome Trust (WT108749/Z/15/Z).

*Conflict of interest*. Paul Flicek is a member of the Scientific Advisory Board for Omicia, Inc.

## References

1. International Human Genome Sequencing Consortium. (2001) Initial sequencing and analysis of the human genome. *Nature*, 409, 860–921.
2. Rios,D., McLaren,W.M., Chen,Y. *et al.* (2010) A database and API for variation, dense genotyping and resequencing data. *BMC Bioinformatics*, 11, 238.
3. Herrero,J., Muffato,M., Beal,K. *et al.* (2016) Ensembl comparative genomics resources. *Database (Oxford)*, 2016, bav096.
4. Zerbino,D.R., Johnson,N., Juetteman,T. *et al.* (2016) Ensembl regulation resources. *Database (Oxford)*, 2016, bav119.
5. Chen,Y., Cunningham,F., Rios,D. *et al.* (2010) Ensembl Variation Resources. *BMC Genomics*, 11, 293.
6. Yates,A., Akanni,W., Amode,M.R. *et al.* (2016) Ensembl 2016. *Nucleic Acids Res.*, 44, D710–D716.
7. Aken,B.L., Ayling,S., Barrell,D. *et al.* (2016) The Ensembl gene annotation system. *Database (Oxford)*, 2016, baw093.
8. Stabenau,A., McVicker,G., Melsopp,C. *et al.* (2004) The Ensembl core software libraries. *Genome Res.*, 14, 929–933.
9. Alper,C.A., Larsen,C.E., Dubey,D.P. *et al.* (2006) The haplotype structure of the human major histocompatibility complex. *Hum. Immunol.*, 67, 73–84.
10. The 1000 Genomes Project Consortium (2015) A global reference for human genetic variation. *Nature*, 526, 68–74.
11. Church,D.M., Schneider,V.A., Steinberg,K.M. *et al.* (2015) Extending reference assembly models. *Genome Biol.*, 16, 13.
12. Gao,J., Liu,K., Liu,H. *et al.* (2010) A complete DNA sequence map of the ovine major histocompatibility complex. *BMC Genomics*, 11, 466.
13. Keane,T.M., Goodstadt,L., Danecek,P. *et al.* (2011) Mouse genomic variation and its effect on phenotypes and gene regulation. *Nature*, 477, 289–294.
14. Atanur,S.S., Diaz,A.G., Maratou,K. *et al.* (2013) Genome sequencing reveals loci under artificial selection that underlie disease phenotypes in the laboratory rat. *Cell*, 154, 691–703.
15. Loman,N.J., Constantinidou,C., Chan,J.Z. *et al.* (2012) High-throughput bacterial genome sequencing: an embarrassment of choice, a world of opportunity. *Nat. Rev. Microbiol.*, 10, 599–606.
16. Rigden,D.J., Fernández-Suárez,X.M., and Galperin,M.Y. (2016) The 2016 database issue of Nucleic Acids Research and an updated molecular biology database collection. *Nucleic Acids Res.*, 44, D1–D6.
17. UniProt Consortium (2015) UniProt: a hub for protein information. *Nucleic Acids Res.*, 43, D204–D212.
18. O'Leary,N.A., Wright,M.W., Brister,J.R. *et al.* (2016) Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res.*, 44, D733–D745.
19. Gray,K.A., Yates,B., Seal,R.L. *et al.* (2015) Genenames.org: the HGNC resources in 2015. *Nucleic Acids Res.*, 43, D1079–D1085.
20. Collins,J.E., White,S., Searle,S.M.J. *et al.* (2012) Incorporating RNA-seq data into the zebrafish Ensembl genebuild. *Genome Res.*, 22, 2067–2078.
21. Cochrane,G., Karsch-Mizrachi,I., Takagi,T. *et al.* (2016) The International Nucleotide Sequence Database Collaboration. *Nucleic Acids Res.*, 44, D48–D50.

22. Church,D.M., Schneider,V.A., Graves,T. *et al.* (2011) Modernizing reference genome assemblies. *PLoS Biol.*, 9, e1001091.
23. Helena Mangs,A., and Morris,B.J. (2007) The Human Pseudoautosomal Region (PAR): Origin, Function and Future. *Curr. Genomics*, 8, 129–136.
24. Speir,M.L., Zweig,A.S., Rosenbloom,K.R. *et al.* (2016) The UCSC Genome Browser database: 2016 update. *Nucleic Acids Res.*, 44, D717–7D725.
25. Li,H., Handsaker,B., Wysoker,A. *et al.* (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25, 2078–2079.
26. ENCODE Project Consortium (2012) An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489, 57–74.
27. Kent,W.J., Zweig,A.S., Barber,G. *et al.* (2010) BigWig and BigBed: enabling browsing of large distributed datasets. *Bioinformatics*, 26, 2204–2207.
28. Jacob,F., Perrin,D., Sanchez,C. *et al.* (1960) Operon: a group of genes with the expression coordinated by an operator. *C R Hebd Seances Acad. Sci.*, 250, 1727–1729.
29. Blumenthal,T., and Gleason,K.S. (2003) Caenorhabditis elegans operons: form and function. *Nat. Rev. Genet.*, 4, 112–120.
30. Harris,R.S. (2007) Improved pairwise alignment of genomic DNA. PhD Thesis, The Pennsylvania State University.
31. Hinrichs,A.S., Karolchik,D., Baertsch,R. *et al.* (2006) The UCSC Genome Browser Database: update 2006. *Nucleic Acids Res.*, 34, D590–D598.
32. Zhao,H., Sun,Z., Wang,J. *et al.* (2014) CrossMap: a versatile tool for coordinate conversion between genome assemblies. *Bioinformatics*, 30, 1006–1007.
33. Cunningham,F., Amode,M.R., Barrell,D. *et al.* (2015) Ensembl 2015. *Nucleic Acids Res.*, 43, D662–D669.
34. Bateman,A., Agrawal,S., Birney,E. *et al.* (2011) RNAcentral: a vision for an international database of RNA sequences. *RNA*, 17, 1941–1946.
35. Slater,G.S.C. and Birney,E. (2005) Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics*, 6, 31.
36. Velankar,S., van Ginkel,G., Alhroub,Y. *et al.* (2016) PDBe: improved accessibility of macromolecular structure data from PDB and EMDB. *Nucleic Acids Res.*, 44, D385–D395.
37. Eppig,J.T., Blake,J.A., Bult,C.J. *et al.* (2015) The Mouse Genome Database (MGD): facilitating mouse as a model for human biology and disease. *Nucleic Acids Res.*, 43, D726–D736.
38. Nawrocki,E.P., Burge,S.W., Bateman,A. *et al.* (2015) Rfam 12.0: updates to the RNA families database. *Nucleic Acids Res.*, 43, D130–D137.
39. Kozomara,A. and Griffiths-Jones,S. (2014) miRBase: annotating high confidence microRNAs using deep sequencing data. *Nucleic Acids Res.*, 42, D68–D73.
40. Maglott,D., Ostell,J., Pruitt,K.D. *et al.* (2011) Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Res*, 39, D52–D57.
41. Gene Ontology Consortium. (2015) Gene Ontology Consortium: going forward. *Nucleic Acids Res.*, 43, D1049–D1056.
42. Vilella,A.J., Severin,J., Ureta-Vidal,A. *et al.* (2009) EnsemblCompara GeneTrees: complete, duplication-aware phylogenetic trees in vertebrates. *Genome Res.*, 19, 327–335.
43. Yates,A., Beal,K., Keenan,S. *et al.* (2015) The Ensembl REST API: Ensembl Data for Any Language. *Bioinformatics*, 31, 143–145.
44. Kersey,P.J., Allen,J.E., Armean,I. *et al.* (2016) Ensembl Genomes 2016: more genomes, more complexity. *Nucleic Acids Res.*, 44, D574–D580.
45. Monaco,M.K., Stein,J., Naithani,S. *et al.* (2014) Gramene 2013: comparative plant genomics resources. *Nucleic Acids Res.*, 42, D1193–D1199.
46. Howe,K.L., Bolt,B.J., Cain,S. *et al.* (2016) WormBase 2016: expanding to enable helminth genomic research. *Nucleic Acids Res.*, 44, D774–D780.
47. Giraldo-Calderón,G.I., Emrich,S.J., MacCallum,R.M. *et al.* (2015) VectorBase: an updated bioinformatics resource for invertebrate vectors and other organisms related with human diseases. *Nucleic Acids Res.*, 43, D707–D713.
48. McDowall,M.D., Harris,M.A., Lock,A. *et al.* (2015) PomBase 2015: updates to the fission yeast database. *Nucleic Acids Res.*, 43, D656–D661.
49. Eöry,L., Gilbert,M.T.P., Li,C. *et al.* (2015) Avianbase: a community resource for bird genomics. *Genome Biol.*, 16, 21.
50. Gallone,G., Simpson,T.I., Armstrong,J.D. *et al.* (2011) Bio::Homology::InterologWalk–a Perl module to build putative protein-protein interaction networks through interolog mapping. *BMC Bioinformatics*, 12, 289.
51. Fallmann,J., Sedlyarov,V., Tanzer,A. *et al.* (2016) AREsite2: an enhanced database for the comprehensive investigation of AU/GU/U-rich elements. *Nucleic Acids Res.*, 44, D90–D95.
52. Veidenberg,A., Medlar,A., and Löytynoja,A. (2016) Wasabi: An Integrated Platform for Evolutionary Sequence Analysis and Data Visualization. *Mol. Biol. Evol.*, 33, 1126–1130.
53. Challis,R.J., Kumar,S., Lewis,S. *et al.* (2016) EasyMirror and EasyImport: Simplifying the setup of a custom Ensembl database and webserver for any species. *PeerJ Preprints*, 4, e2401v1.