The Ensembl Database Schema

European Bioinformatics Institute
Requirements for the schema

- Store data for human genome
- … and all the other genomes we have
- … and all the genomes we might get
- Flexible to add more data
- Easy to adapt to new genome
- Responds fast enough for web site display and pipelined genebuild
System Context

Other Scripts & Applications

Apollo
Java API (EnsJ)
Ensembl DBs
Perl API
www
Pipeline
MartShell
MartView
Mart DB
Sequence regions

- Everything which represents a length of nucleotide sequence is a sequence region.
  - chromosome, BAC-clone, supercontig, scaffold, contig …

- Sequence regions of the same type belong to the same coordinate system.
  - “1”, “2”, and “3” are sequence regions with coordinate system “chromosome”

- Sequence regions have names and lengths.
Sequence regions

- **seq_region**
  - seq_region_id: int
  - name: varchar
  - coord_system_id: int
  - length: int

- **coord_system**
  - coord_system_id: int
  - name: varchar
  - version: varchar
  - attrib: "default_version", "sequence_level"
  - rank: int

- **dna**
  - seq_region_id: int
  - sequence: mediumtext

The diagram illustrates the relationships and attributes of sequence regions, showing how they are interconnected and structured within the Ensembl database.
Example sequence region

- Chromosome, 1, 200MB
- Clone, AL123123.4, 132KB
- NT_contig, NT_1245675, 17MB
- Contig, AC332232.1.1.123223, 123223
Coordinate system

- The `coord_system` describes the type of the sequence region
  - Name ("chromosome", "contig", ...)
  - Version (e.g. NCBI35, ZFISH3)
  - Internal id (`coord_system_id`)
  - Attrib – (default, `sequence_level`)
  - rank (1..n)

- If you have 2 coordinate systems with the same name, choose a "default" one. They need to have different versions (NCBI34, NCBI35).
- The lower the rank, the bigger the sequence region. Choose 1 for your biggest regions (chromosomes).
- Only one coordinate system is allowed to contain sequence regions with actual sequence attached. Flag it with `Attrib = sequence_level`. 
Coordinate system

- "contig"
  - Contiguous sequence.
  - "N"'s should be rare and of short length.
  - Can serve as your basic sequence holder
- "clone"
  - Should have a real BAC or PAC or maybe YAC behind it.
  - Might not be contiguous
- "supercontig"
  - Assembled from smaller contiguous sequences.
  - May have small gaps (eg between read pairs)
- "chromosome"
  - Use it only for real chromosomes.
  - or for alternative sequences of reference chromosomes.
- "chunk"
  - Artificial coordinate system to hold sequence regions for technical reasons.
  - Create, when none of the other coordinate systems can hold your sequence (eg. You only have full length chromosomes as coordinate system but they are too long to store)
  - or when you have 2 real sequence containing coordinate systems.
Assemblies

- An assembly defines how sequence regions in one coordinate system are made up of sequence regions from another coordinate system.
- For example human chromosomes are assembled from a “tiling path” of BAC clones.
- Assembly information stored in Ensembl makes it possible to obtain features or sequence from arbitrary sequence regions.
Assemblies

- A row in the assembly table references an assembled and component sequence region.
- How a piece of the assembled sequence region is made from a piece of a component region is defined by a pair of coordinates and an orientation.
- Gaps are represented by the absence of assembly information.
The assembly table

assembly
- asm_seq_region_id (int)
- cmp_seq_region_id (int)
- asm_start (int)
- asm_end (int)
- cmp_start (int)
- cmp_end (int)
- ori (int)

seq_region
- seq_region_id (int)
- name (varchar)
- coord_system_id (int)
- length (int)

coord_system
- coord_system_id (int)
- name (varchar)
- version (varchar)
- attrib (varchar)
- rank (int)

dna
- seq_region_id (int)
- sequence (mediumtext)

The diagram illustrates the relationships and attributes of the assembly table in the Ensembl database.
Sequence region attributes

- Arbitrary attributes may be associated with a sequence region via the seq_region_attrib table.
  - sanger ids for certain clones.
  - htg phases for clones.
## The seq_region_attrib table

<table>
<thead>
<tr>
<th>assembly</th>
<th>dna</th>
<th>coord_system</th>
<th>dnac</th>
<th>attrib_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq_region_id</td>
<td>seq_region_id</td>
<td>coord_system_id</td>
<td>seq_region_id</td>
<td>attrib_type_id</td>
</tr>
<tr>
<td>name</td>
<td>name</td>
<td>version</td>
<td>sequence</td>
<td>code</td>
</tr>
<tr>
<td>length</td>
<td>length</td>
<td>version</td>
<td>mediumtext</td>
<td>name</td>
</tr>
<tr>
<td>seq_region_id</td>
<td>seq_region_id</td>
<td>version</td>
<td>n_line</td>
<td>description</td>
</tr>
<tr>
<td>asm_seq_region_id</td>
<td>sequence</td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cmp_seq_region_id</td>
<td>mediumtext</td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>asm_start</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>asm_end</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cmp_start</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cmp_end</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ori</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>seq_region_id</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>attrib_type_id</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>value</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>seq_region_id</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>attrib_type_id</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>value</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>seq_region_id</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>attrib_type_id</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>value</td>
<td></td>
<td>version</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The diagram illustrates the relationships and data structures for these tables, showing how they interconnect and the various fields they contain.
Features

• Features are annotation information placed on the genome.
• A feature is stored as a position on a sequence region.
A standard feature

### simple_feature
- simple_feature_id: int
- seq_region_id: int
- seq_region_start: int
- seq_region_end: int
- seq_region_strand: tinyint
- display_label: varchar
- analysis_id: int
- score: double

### any feature
- usually has a any_feature_id: int
- contains a sequence position with or without strand on a sequence region
- usually contains a string to display: varchar
- usually links to the analysis responsible for calculating it: int
- contains any number of other attributes: ..
- ..

### analysis
- analysis_id: int
- created: datetime
- logic_name: varchar
- db: varchar
- db_version: varchar
- db_file: varchar
- program: varchar
- program_version: varchar
- program_file: varchar
- parameters: varchar
- module: varchar
- module_version: varchar
- gff_source: varchar
- gff_feature: varchar
### other Features

<table>
<thead>
<tr>
<th>Table</th>
<th>Columns</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>dna_align_feature</td>
<td>dna_align_feature_id int</td>
<td>Sequence position</td>
</tr>
<tr>
<td></td>
<td>hit_start int</td>
<td></td>
</tr>
<tr>
<td></td>
<td>hit_end int</td>
<td></td>
</tr>
<tr>
<td></td>
<td>hit_strand tinyint</td>
<td></td>
</tr>
<tr>
<td></td>
<td>hit_name varchar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>analysis_id int</td>
<td></td>
</tr>
<tr>
<td></td>
<td>score double</td>
<td></td>
</tr>
<tr>
<td></td>
<td>evalue double</td>
<td></td>
</tr>
<tr>
<td></td>
<td>perc_ident float</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cigar_line text</td>
<td></td>
</tr>
<tr>
<td>protein_align_feature</td>
<td>protein_align_feature_id int</td>
<td>Sequence position</td>
</tr>
<tr>
<td></td>
<td>hit_start int</td>
<td></td>
</tr>
<tr>
<td></td>
<td>hit_end int</td>
<td></td>
</tr>
<tr>
<td></td>
<td>hit_name varchar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>analysis_id int</td>
<td></td>
</tr>
<tr>
<td></td>
<td>score double</td>
<td></td>
</tr>
<tr>
<td></td>
<td>evalue double</td>
<td></td>
</tr>
<tr>
<td></td>
<td>perc_ident float</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cigar_line text</td>
<td></td>
</tr>
<tr>
<td>repeat_feature</td>
<td>repeat_feature_id int</td>
<td>Sequence position</td>
</tr>
<tr>
<td></td>
<td>repeat_start int</td>
<td></td>
</tr>
<tr>
<td></td>
<td>repeat_end int</td>
<td></td>
</tr>
<tr>
<td></td>
<td>repeat_name varchar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>repeat_class varchar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>consensus text</td>
<td></td>
</tr>
<tr>
<td>repeat_consensus</td>
<td>repeat_consensus_id int</td>
<td></td>
</tr>
<tr>
<td></td>
<td>repeat_name varchar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>repeat_class varchar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>consensus text</td>
<td></td>
</tr>
</tbody>
</table>

The diagram illustrates the relationships between these tables, with arrows indicating the direction of the associations.
Genes are features

sequence == seq_region

exons

have stable ids
ENSE0001  ENSE0002  ENSE0003  ENSE0004

alternative spliced transcripts

.. with different translations

stable ids
ENST0001
ENST0002
ENSP0003
ENSP0004
External references

• Ensembl objects can reference objects in other databases.
  – eg a SWISSPROT identifier, GO identifier, Refseq, HUGO, …

• External references are used for display ids in Genes and Transcripts. These links are provided directly in Gene and Transcript.
Misc features

- are features with user definable attributes
- it can belong to a set to provide a trackname for the feature.
- a misc feature can be in more than one set.
Archive tables

- For some species there is a record of old predictions available.
  - human, mouse
- You can get
  - old peptide sequences
  - how an older gene prediction was made up from transcripts/translations.
  - how genes and transcripts were merged and split from older prediction to newer prediction.
Archive tables

- A mapping session describes the event when a set of ids is mapped from an older database to a newer database
  - Version number are a relatively new addition, so you need the mapping session to uniquely specify a Gene.
- A stable id event for a gene states that some part of the old gene is to be found in the new gene.
  - Same for Transcript.
- The gene archive records the gene structure of the older gene, when the gene has changed during a mapping session.
- The peptide archive records the peptide sequence of the old version of the peptide, when the peptide changes.
## Archive tables

### gene_archive
- **gene_stable_id**: varchar
- **gene_version**: smallint
- **transcript_stable_id**: varchar
- **transcript_version**: smallint
- **translation_stable_id**: varchar
- **translation_version**: smallint
- **mapping_session_id**: int

### stable_id_event
- **old_stable_id**: varchar
- **old_version**: smallint
- **new_stable_id**: varchar
- **new_version**: smallint
- **mapping_session_id**: int
- **type**: “Translation”, “Gene”, “Transcript”

### mapping_session
- **mapping_session_id**: int
- **old_db_name**: varchar
- **new_db_name**: varchar
- **created**: timestamp

### peptide_archive
- **translation_stable_id**: varchar
- **translation_version**: smallint
- **peptide_seq**: mediumtext
Markers and marker features

```sql
-- Table: marker_map_location
marker_id | int
map_id    | int
chromosome_name | varchar
marker_synonym_id | int
position      | varchar
lod_score     | double

-- Table: map
map_id | int
map_name | varchar

-- Table: marker
marker_id | int
display_marker_synonym_id | int
left_primer | varchar
right_primer | varchar
min_primer_dist | int
max_primer_dist | int
priority | int
type | "est", "microsatellite"

-- Table: marker_synonym
marker_synonym_id | int
marker_id | int
source | varchar
name | varchar

-- Table: marker_feature
marker_feature_id | int
marker_id | int
seq_region_id | int
seq_region_start | int
seq_region_end | int
analysis_id | int
map_weight | int
```
QTLs
Meta information

- Meta table contains general key-value pairs
  - eg. species name
  - taxonomy id
- which coordinates can be mapped and how
- future additions likely
- Meta_coord says which feature is stored in which coordinate system
  - more than 1 entry possible
  - no feature retrieval without it.
Protein features

- Features can be added to the peptide sequence
- hit_id is usually Pfam, Prosite, prints identifier.
- interpro table links these to interpro ids.
- xrefs have further information for them.
Density feature

- Density features assign numeric values to regions.
  - GC content
  - gene count
  - repeat coverage

- The blocksize and value_type enable interpolation by API
Karyotype bands

- Karyotype table defines the banding pattern of the chromosomes and how to draw the ideogram.
- A single band is just like any other feature in the database.
- Band naming convention depends on species and resolution.
- Stain could be (“acen”, “gvar”, “gpos25”, “gpos50”, “gpos75”, “gpos100”, “gneg” ...)

<table>
<thead>
<tr>
<th>karyotype</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>karyotype_id</td>
<td>int</td>
</tr>
<tr>
<td>seq_region_id</td>
<td>int</td>
</tr>
<tr>
<td>seq_region_start</td>
<td>int</td>
</tr>
<tr>
<td>seq_region_end</td>
<td>int</td>
</tr>
<tr>
<td>band</td>
<td>varchar</td>
</tr>
<tr>
<td>stain</td>
<td>varchar</td>
</tr>
</tbody>
</table>
Supporting Evidence

- Exons can be linked to features.
- These are alignment features that were used as evidence when the exon was created.
  - supporting evidence

<table>
<thead>
<tr>
<th>supporting_feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>exon_id</td>
</tr>
<tr>
<td>feature_type</td>
</tr>
<tr>
<td>feature_id</td>
</tr>
</tbody>
</table>
New tables

<table>
<thead>
<tr>
<th>transcript_attrib</th>
</tr>
</thead>
<tbody>
<tr>
<td>transcript_id</td>
</tr>
<tr>
<td>attrib_type_id</td>
</tr>
<tr>
<td>value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>assembly_exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>assembly_exception_id</td>
</tr>
<tr>
<td>seq_region_id</td>
</tr>
<tr>
<td>seq_region_start</td>
</tr>
<tr>
<td>seq_region_end</td>
</tr>
<tr>
<td>exc_type</td>
</tr>
<tr>
<td>exc_seq_region_id</td>
</tr>
<tr>
<td>exc_seq_region_start</td>
</tr>
<tr>
<td>exc_seq_region_end</td>
</tr>
<tr>
<td>ori</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>translation_attrib</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation_id</td>
</tr>
<tr>
<td>attrib_type_id</td>
</tr>
<tr>
<td>value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>alt_allele</th>
</tr>
</thead>
<tbody>
<tr>
<td>alt_allele_id</td>
</tr>
<tr>
<td>gene_id</td>
</tr>
</tbody>
</table>
Ensembl Core Software Team:

- Arne Stabenau
- Glenn Proctor
- Craig Melsopp
- Ian Longden

The Rest of the Ensembl Team.