

Database Schema

As described “Data Model”, locus data can come from a variety of sources. Besides the user's own data, one of the main sources of data will be pre-existing databases. Hocus Locus will contain it's own database array to serve two purposes:

1. To provide a local, fast lookup of common data sets. This allows the user ease of retrieval, without having to depend on 3rd party sources.
2. To provide specially structured and accessed database tables of additional annotation. This allows the user rapid recovery of the additional data which is normally slow and resource-intensive to generate.

The Hocus Locus database array is conceptually modeled after the UCSC database system. The data is structured in a hierarchical fashion, based on species and assembly (version of the genome sequence). For a particular species and assembly, there will be a number of data sets available. Much of the actual data itself is taken directly from UCSC, matching table schema, indexing, and content. Other 3rd party data sources can be leveraged as well. This allows for ease of portability and maintenance, and allows for a local copy of this data to be present. However the Hocus Locus database array contains a number unique attributes which add to the functionality of the system.

See Fig 1

The Hocus Locus database system includes:

- Specialized meta-data tables which describe *what* information is available and *how* it is structured.
- The ability to use these meta-data tables to add new data sets to the system on the fly, and have them become immediately available.
- Uniquely structured tables of additional annotation, allowing for rapid retrieval of large repositories of information with minimal overhead.

The meta-data layer consists of a main database which acts as a central point of access, and contains information describing the remainder of the array. Each species and assembly combination is housed in it's own database, and specific tables in the “main” database list what combinations are available. Other meta-data describes how to access those data sets, as well as global table structure descriptions for each unique set of content taken from UCSC.

As mentioned previously, each species/assembly database contains some number of data sets gathered from 3rd party sources such as UCSC or others. When describing this data, the **genomic location** attribute (chromosomal coordinates) is the focus of the Hocus Locus system. However there are many other attributes of importance – the most common being the sequence – that may be required as part of the analysis. The Hocus Locus database array provides a means by which this information can quickly and easily accompany the loci in a data set. Currently the two additional annotation sets provided are nucleotide sequence, and phylogenetic conservation. In each case, an attribute of each nucleotide must be maintained: a sequence 'letter' (A,T,C,G,etc), or a conservation score. Each table is structured in a similar manner: The attributes of each nucleotide are grouped together into equal length short segments, and each segment is given it's corresponding chromosomal position. In this case only the the chromosome and first nucleotide (start position) are tracked. An index is also created based on the chromosomal coordinates, thus giving a unique index. In this way, data that was previously very “horizontal” - for instance an entire chromosome sequence – is transformed into easily indexable

“vertical” data. This allows us to draw upon the power of the database engine to perform extremely fast retrieval of large amounts of information.

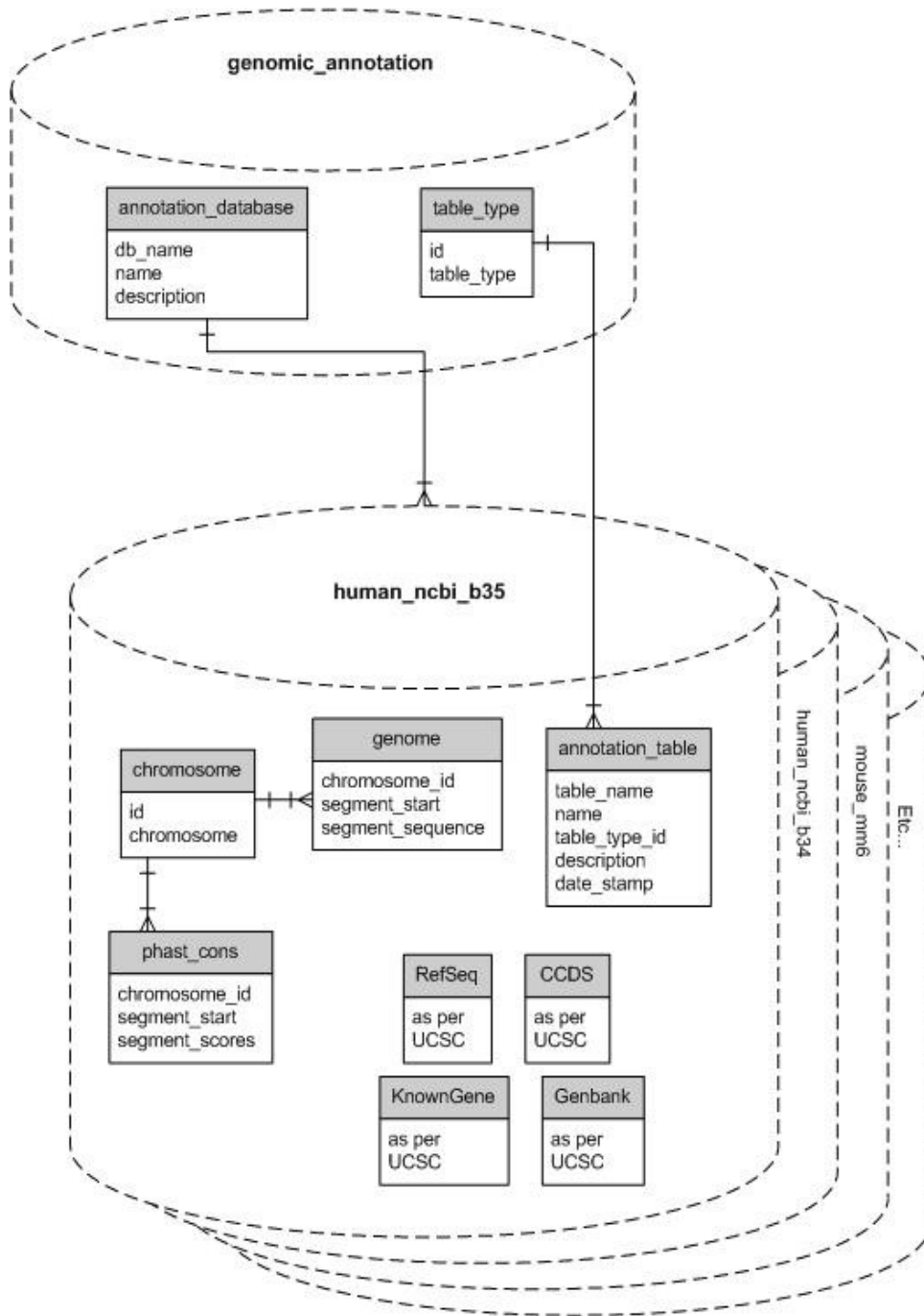
Certainly many other retrieval-speed problems still exist, such as disk access speeds, data caching, network traffic, etc. However the Hocus Locus storage schema allows us to all but eliminate the bottleneck of *seek time*, while allowing all the benefits of storing your raw data in a relational database.

See Fig 2

Potentially Novel Items:

- 1) Database table structure and accompanying functionality for rapid retrieval of annotation data (as described above and in Figure 2).

Figure 1



The HocusLocus database schema. A “genomic_annotation” database acts as a central point of access and contains meta-data describing the remainder of the array. This database is used to discover what species and assembly combinations are available, how to access those tables, as well as global table structure descriptions for each unique set of content. There then exists a separate database to house each species/assembly combination, and whatever corresponding data exists.

Figure 2

A.

chromosome	position	sequence
2	256	CGCGATCGTATAGTGCACGACTGTAGTCGAGCTAGGCTAT
2	511	ACGATGTGCAGCATGCTAGCTGAGCGAGCGTAGCTAGCT
2	766	ACACGTAGCTAGGCCGCGATTATATGCAGCTGACTGTAGC
2	1021	CGATGCGAGCTCGCCATGTAGCGACTGATTTGCAAACGTG
2	1276	CGCGATCGTATAGTGCACGACTGTAGTCGAGCTAGGCTAT
2	1531	ACGATGTGCAGCATGCTAGCTGAGCGAGCGTAGCTAGCT
2	1786	ACACGTAGCTAGGCCGCGATTATATGCAGCTGACTGTAGC
2	2041	CGATGCGAGCTCGCCATGTAGCGACTGATTTGCAAACGTG

B.

ACGATGTGCAGCATGCTAGCTGAGCGAGCGTAGCTAGCTACACGTAGCTAGGCCGCGATTATATGCAGCTGACTGTAGCCGATGCGAGCTCGCCATGTAGCGACTGATTTGCAAACGT

C.

Final Sequence

CTGAGCGAGCGTAGCTAGCTACACGTAGCTAGGCCGCGATTATATGCAGCTGACTGTAGCCGATGCGAGCTCGCCATGTAG

Example data retrieval from a genomic sequence table. First all records from the table are fetched which contain all or part of the requested sequence (A). Then the sequence segments are concatenated into a single string (B). Finally the excess sequence is removed from the beginning and end, resulting in the final requested sequence.

By adding these small amounts of pre and post- processing, the seek time of the sequence retrieval process becomes almost negligible, while still allowing all the benefits of storing the raw data in an RDBMS.