Databases at UCSC

It just *looks* like 200,000 columns.

The Databases

- Genome databases - one for each assembly of each organism: hg16, mm4, sacCer1, etc.
- hgFixed - mostly microarray data.
- swissProt - Relationalized swissProt database.
- go - Gene ontology terms and term/gene associations.
- Protein databases - Shared across organisms. Each genome database associated with a particular protein database.
- hgCentral - home to dbDb and user settings info. One database shared by all web servers.

Genome Databases

- Track data
- Parsed out GenBank data
- Data associated with knownGenes
- Proteome Browser data.
- trackDb - a table about tracks

Track Table Data

- Most tracks are independent of each other.
- Most tracks are in one of several formats:
  - genePred - stored gene structures
  - alignment formats (psl, chain, net, axt, maf)
  - bed, a flexible format used for simpler stuff.
    - Initial field of a bed are defined, later fields can be anything
- Older and larger tracks may be split across chromosomes.
- In addition to primary table, tracks may use other tables - typically joining via the ‘name’ or ‘qName’ field of the primary table.

GenBank mRNA Data

- Most of the information in a GenBank flat file record ends up in the genome database.
- The mrna table contains an entry for every mRNA, EST, and RefSeq.
- The mrna table itself just contains the GenBank accession, and id's that link into other tables.
  - Select mrna.acc, tissue.name from mrna,tissue where mrna.tissue = tissue.id

Known Genes Data

- KnownGene, and to a lesser extent RefGene link to a *lot* of other tables.
- The knownToXxx tables are used as the basis of many Family Browser columns. kgXref has much of the same data in one place.
- knownCanonical/knownIsoforms group together splicing varients.
- Various 'BlastTab' tables link known genes to homologs in other species.
- sangerGene (worm), bdgpGene (fly), sgdGene (yeast) play similar role to knownGene in model organisms.

Proteome Browser Data

- Known genes linked to swissProt via accession at kgXref.spId.
  - (SwissProt displayId is not as stable.)
- Pathway info in keggPathway, bioCycPathway, cgapBiocPathway
- GO links via swissProt accession in go.goaPart.dbObjectSymbol

TrackDb

- Every genome database has a trackDb table.
- trackDb contains a row for each track. Fields include:
  - tableName - primary table
  - short & long labels - seen in user interface
  - type - track type
  - visibility - default hide/dense/pack/full state
- Build from src/hg/makeDb/trackDb .ra files
- README in that directory describes format.
- Each developer has a trackDb_user table that controls hgwdev-user.cse.ucsc.edu.

hgFixed - expression data

- Each set of expression data is associated with two types of tables:
  - A table ending with Exps that has information about all the mRNA samples (tissues etc)
  - A table not ending in Exps that has the level of mRNA observed for each Gene.
- In some cases there may be separate tables with log-2 based ratios as well as absolute expression values.
- In some cases there may be separate tables with median values for replicated experiments.

swissProt vs. SwissProt

- SwissProt is a beautiful database, but it is represented at Geneva as a bunch of ‘managed’ files, and externally in a flat-file format.
- swissProt is an efficient relationalized version. Best to link into this with the accession, but can also use ‘displayId’.
- See spdb.h for C library modules to access.
- Contains a wealth of protein info, and also some good functional info in nicely structured comments. Good xrefs to other databases.
- Programmers at SwissProt have unofficially double-checked the relationalization.

GO Database

- This is imported directly form geneontology.org.
- Use goaPart table to find which GO terms are associated with a SwissProt accession
- Highly relational. Use term and term_definition to find meaning of terms.

Protein Databases

- Used in proteome browser, in building known genes, and some in hgGene/hgNear.
- proteins040115 is latest. Switched from month/day/year format to year/month/day format part way through. Earlier ones lack year. (Suggest change to ‘protYYMMDD’ to avoid confusion in the future and have ‘prot’ sym-linked to latest.)
- Mostly contains stuff you can get through swissProt. Formerly was bits of SwissProt Fan needed when using ugly bioperl relational swissProt

hgCentral

- has dbDb - a table with a row for each genome database. This includes organism name, DNA location, etc.
- sessionDb - user ‘cart’ setting for current session
- userDb - cart settings saved between sessions
- gdbPdb - relates genome and protein databases.

Database Documentation

- find src/hg -name *.as -print
- http://genome.ucsc.edu/goldenPath/gbdDescriptions.html
- src/hg/makeDb/make*.doc
- src/hg/makeDb/schema/all.joiner
- src/hg/makeDb/schema/joiner.doc
- src/hg/makeDb/*/*.c

.as Files - table and field docs

```
table cpgIsland
"Describes the CpG Islands"
   (
   string chrom;        "Human chromosome or FPC contig"
   uint   chromStart;   "Start position in chromosome"
   uint   chromEnd;     "End position in chromosome"
   string name;         "CpG Island"
   uint   length;       "Island Length"
   uint   cpgNum;       "Number of CpGs in island"
   uint   gcNum;        "Number of C and G in island"
   float  perCpg;       "Percentage of island that is CpG"
   float  perGc;        "Percentage of island that is C or G"
   )
```

autoSql generates code from these. They also help document.

Other Docs

- gbdDescriptions.html - updated by Donna. Merges together all .as files for whole projects, and has some overview text.
- Description button in table browser will fetch relevant .as file most of the time.
- makeHg16.doc and other database build docs - describes how database was built.
- all.joiner file - describes how tables are linked together.

all.joiner - basic example

```
identifier softberryGeneName
"Link together Fshgene++ gene structure, peptide, and homolog"
    $gbd.softberryGene.name
    $gbd.softberryPep.name
    $gbd.softberryHom.name
```

- The central concept is an identifier that appears in fields in multiple table, sometimes even multiple databases.
- $gbd is a variable that contains a comma-separated list of databases.
- An identifier record ends with a blank line.

Databases by organism

```
# Define databases used for various organisms
set hg hg13,hg15,hg16
set mm mm3,mm4
set rn rn2,rn3
set fr fr1
set ce ce1
set cb cb1
set dm dm1
set dp dp1
set sc sc1
set sacCer sacCer1
set panTro panTro1
set galGal galGal2
```

```
# Define all organism/assembly specific databases.
set gbd $hg,$mm,$rn,$fr,$ce,$cb,$dm,$dp,$sc,$sacCer,$panTro,$galGa

# Only consider one of members of gbd at a time.
exclusiveSet $gbd

# Define other databases that we check
set otherDb swissProt,go,hgFixed
                                   …

# Set up list of databases we ignore and those we check.  Program
# will complain about other databases.
databasesChecked $gbd,$otherDb
databasesIgnored mysql,lost+found,$proteinDb,$zooDb,hgcentraltest,hg
```

```
# Define databases that support known genes
set kgDb $hg,$mm,$rn

# Define databases that support family browser
set familyDb $hg,$mm,$ce,$sacCer,$dm

# Magic for tables split between chromosomes
set split splitPrefix=chr%_
```

```
# Stuff to link together alignment chains and nets

identifier chainSelf
"Link together self chain info"
    $gbd.chainSelf.id $split
    $gbd.chainSelfLink.chainId $split
    $gbd.netSelf.chainId exclude=0

identifier chain[${chainDest}]Id
"Link together chain info"
    $gbd.chain[].id $split
    $gbd.chain[]Link.chainId $split
    $gbd.net[].chainId
    $gbd.allChain[].id
    $gbd.netRxBest[].chainId exclude=0
    $gbd.net[]NonGap.chainId exclude=0
    $gbd.netSynteny[].chainId exclude=0
```

```
# Genbank/trEMBL Accessions and meaningful subsets thereof
identifier genbankAccession external=genbank
"Generic Genbank Accession.  More specific Genbank accessions follow
    $gbd.seq.acc

identifier stsAccession external=genbank typeOf=genbankAccession
"Genbank accession of a Sequence Tag Site (STS) sequence."
    $gbd.stsInfo2.genbank dupeOk

identifier bacEndAccession typeOf=genbankAccession
"Genbank accession of a BAC end read."
    $gbd.all_bacends.qName dupeOk
    $gbd.bacEndPairs.lfNames comma
    $hg.fishClones.beNames comma minCheck=0.70
```

The typeOf line allows joins between parent and child, but not between siblings.

```
identifier hugoName external=HUGO fuzzy
"International Human Gene Identifier"
    $hg.refLink.name
    $hg.atlasOncoGene.locusSymbol
    $hg.kgAlias.alias
    $hg.kgXref.geneSymbol
    $hg.refFlat.geneName
    $hg.jaxOrtholog.humanSymbol
    hg13,hg15.geneBands.name
```

“Biological” names for human genes are so messy, no validation is done (note ‘fuzzy’ keyword).

```
identifier ensemblTranscriptId external=Ensembl dependency
"Ensembl Transcript ID"
    $gbd.ensGene.name chopAfter=.
    $gbd.superfamily.name
    $gbd.ensGeneXref.transcript_name chopAfter=. minCheck=0.20
    mm3,hg13.ensemblXref.transcript_name chopAfter=. minCheck=0.2
    mm3.ensemblXref2.transcript_name chopAfter=. minCheck=0.20
    $gbd.ensGtp.transcript chopAfter=. minCheck=0.98
    $gbd.ensPep.name chopAfter=. minCheck=0.98
    $gbd.ensTranscript.transcript_name chopAfter=. minCheck=0.20
    $kgDb.knownToEnsembl.value chopAfter=.
    $gbd.sfDescription.name chopAfter=.
    mm3.superfamily.name chopAfter=.
```

Ensembl isn’t ‘fuzzy’ but requires relaxed ‘minCheck’

```
# Dependencies not already captured in identifiers

dependency $mm.affyGnfU74ADistance $mm.knownToU74  hgFixed.g
dependency $mm.affyGnfU74BDistance $mm.knownToU74  hgFixed.g
dependency $mm.affyGnfU74CDistance $mm.knownToU74  hgFixed.g
dependency $hg.gnfU95Distance $hg.knownToU95 hgFixed.gnfHuman
dependency $ce.kimExpDistance hgFixed.kimWormLifeMedianRatio
dependency $dm.arbExpDistance $dm.bdgpToCanonical hgFixed.arbFly
dependency $sacCer.choExpDistance hgFixed.yeastChoCellCycle
```

```
# Ignored tables - no linkage here that we check at least.

tablesIgnored go
    instance_data
    source_audit

tablesIgnored $gbd
    ancientRepeat
    axtInfo
    chromInfo
    cpgIsland
    …
    trackDb%
    chr%_mrna
```

joinerCheck squawks about any table (or database) not mentioned

joinerCheck - the tool you’ll love to hate!

```
joinerCheck - Parse and check joiner file
usage:
   joinerCheck file.joiner
options:
   -parseOnly just parse joiner file, don't check database.
   -fieldListOut=file - List all fields in all databases to file.
   -fieldListIn=file - Get list of fields from file rather than mysql.
   -identifier=name - Just validate given identifier.
   -database=name - Just validate given database.
   -keys - Validate (foreign) keys.  Takes about an hour.
   -noTableCoverage - No check that all tables are mentioned in joiner file
   -noDbCoverage - No check that all databases are mentioned in joiner file
   -noTimes - Check update times of tables are after tables they depend on
```