

Abstract

This HOWTO provides several sample SQL queries to view basic genome data from a Chado database. Some of these are drawn from the [GMODTools](#) configuration file `GMODTools/conf/bulkfiles/chadofeatsql.xml` Example output of some of these is shown in the tables at

http://insects.eugenes.org/genome/Drosophila_melanogaster/current/tables/

PostgreSQL queries

The simplest way to test contents of a Chado database is with the `psql` command line program that is part of the [PostgreSQL](#) software. The following sample SQL code can be used this way from a Unix or MacOSX command line.

```
dgbook% psql -l
          List of databases
  Name          | Owner      | Encoding
-----+-----+-----
 dev_chado_01b | gilbertd  | SQL_ASCII

dgbook% psql dev_chado_01b
 dev_chado_01b=# select organism_id,count(*) from feature group by organism_id;
 organism_id | count
-----+-----
           7 |      8
           6 |     10
           3 |     10
          10 |    1605
```

organism_summary

This lists organisms and number of features per organism.

```
SELECT o.organism_id,o.abbreviation,o.genus,o.species,o.common_name,
       count(f.feature_id) as n_features, o.comment
FROM organism o LEFT JOIN feature f USING (organism_id)
GROUP by o.organism_id,o.abbreviation,o.genus,o.species,o.common_name,o.comment
ORDER BY o.genus,o.species
;
```

sample result

Organism_id	Abbreviation	Genus	Species	Common_name	N_features	Comment
105	\N	Abrostola	asclepiadis	\N	1	\N
..						
1	Dmel	Drosophila	melanogaster	fruit fly	725035	\N
214	Dpse	Drosophila	pseudoobscura	\N	137045	\N

feature_summary

This lists number of features and sequences by species and type.

```
SELECT
  f.type_id,
  t.name as Feature_type,
  count(f.feature_id) as N_features,
  sum(length(f.residues)) as N_residues,
  sum(f.seqlen) as Tot_len,
  ROUND( AVG(f.seqlen), 0 ) as Ave_len,
  MIN(f.seqlen) as Min_len,
  MAX(f.seqlen) as Max_len,
  (select genus || '_' || species from organism where organism_id = f.organism_id) as Species
FROM feature f, cvterm t
WHERE f.type_id = t.cvterm_id
GROUP BY f.organism_id, f.type_id, t.name
ORDER BY species, feature_type
;
```

Sample result

Type_id	Feature_type	N_features	N_residues	Tot_len	Ave_len	Min_len	Max_len
256	EST	308722	143832868	\N	\N	\N	\N
562	cDNA	13464	26962334	\N	\N	\N	\N
210	chromosome_arm	14	246701062	246701062	18977005	1237870	27905053
257	exon	64498	\N	31208553	484	3	27725
219	gene	14828	\N	70649778	4899	16	171463
720	insertion_site	622	\N	28	0	0	1
368	mRNA	19389	44104476	44104476	2275	132	69571
450	miRNA	66	1498	1498	23	20	29
426	ncRNA	116	115179	115179	993	19	14084
733	point_mutation	1444	\N	43	1	1	1
499	polyA_site	122	\N	\N	\N	\N	\N
1179	protein	22219	12111034	10921981	563	25	23015

chromosome_summary

This is an extension of feature summary, to identify source features. These are the chromosomes, scaffolds, contigs or other features on which other (genes, etc.) are located. It is essential for Chado software to know about these source features. If you are working with someone else's Chado database, this query will tell you explicitly which features contain others.

```
SELECT
  f.type_id,
  t.name as Feature_type,
  count(f.feature_id) as N_features,
  sum(length(f.residues)) as N_residues,
  sum(f.seqlen) as Tot_len,
  sum(CASE WHEN fl.srcfeature_id = f.feature_id THEN 1 ELSE 0 END) as N_issource,
  sum(CASE WHEN fl.feature_id = f.feature_id THEN 1 ELSE 0 END) as N_istarget,
  (select genus || '_' || species from organism where organism_id = f.organism_id) as Species
FROM cvterm t, feature f
left join featureloc fl on (fl.srcfeature_id = f.feature_id or fl.feature_id = f.feature_id)
```

Sample_Chado_SQL

```
WHERE f.type_id = t.cvterm_id
GROUP BY f.organism_id, f.type_id, t.name
ORDER BY species, feature_type
```

See the **n_issource** column for source features, the **n_istarget** lists the target features contained in sources.

type_id	feature_type	n_features	n_residues	tot_len	n_issource	n_istarget
66	centromere	16		1883	0	16
64	chromosome	146644	132193461745	132193461745	146644	0
124	gene	6609		8889442	0	6609
133	intron	392		122291	0	392
156	ncRNA	483		76213	0	483

Note the **n_issource** count is the number of features contained in the chromosomes, not number of chromosomes (see above feature_summary for that).

analysis_summary

This lists analyses and number of features per analysis.

```
SELECT
  an.analysis_id,
  CASE WHEN (an.sourcename IS NULL OR an.sourcename = 'dummy') THEN 'match:' || an.program
        ELSE 'match:' || an.program || ':' || an.sourcename
  END AS Analysis_type,
  count(f.feature_id) as N_features,
  ROUND( (AVG(af.rawscore)::numeric), 2 ) as Ave_score,
  ROUND( (AVG(af.significance)::numeric), 2 ) as Ave_sig,
  (select genus || '_' || species from organism where organism_id = f.organism_id) as Species
FROM feature f, analysisfeature af, analysis an
WHERE an.analysis_id = af.analysis_id and af.feature_id = f.feature_id
GROUP BY f.organism_id, an.analysis_id, Analysis_type
ORDER BY species, Analysis_type
;
```

Sample result

Analysis_id	Analysis_type	N_features	Ave_score	Ave_sig	Species	
68	match:aubrey_cytolocator:cytology		5770	\N	Computational_result	
70	match:augustus 60764	\N	\N		Computational_result	
53	match:blastx_masked:aa_SPTR.insect		53629	200.60	\N	Computational_result
44	match:repeatmasker	23486	3922.55	\N		Computational_result
78	match:tblastn:Dmel r3.1	12179	\N	\N		Computational_result

sequence_ontology

Specifying the sequence ontology section of the cv and cvterm tables is a small but essential bit of a Chado database that software needs for configuration. Not everyone uses the same name, though *Sequence Ontology Feature Annotation* is recommended.

```
select cv_id,name from cv where cv_id in (select cv_id from cvterm where name = 'exon');
cv_id      name
-----
8  Sequence Ontology Feature Annotation
```

property_summary

This lists properties and number of features per analysis.

```
SELECT
  fp.type_id,
  t.name as Property_type,
  count(fp.featureprop_id) as N_properties,
  count(distinct f.feature_id) as N_features,
  count(distinct fp.value) as N_values,
  (select genus || '_' || species from organism where organism_id = f.organism_id) as Species
FROM feature f, featureprop fp, cvterm t
WHERE fp.type_id = t.cvterm_id and fp.feature_id = f.feature_id
GROUP BY f.organism_id, fp.type_id, t.name
ORDER BY Species, Property_type
;
```

Sample result

Type_id	Property_type	N_properties	N_features	N_values	Species	
57127	citation	5467	4073	1141	Drosophila_melanogaster	
59945	comment	13005	9699	11264	Drosophila_melanogaster	
7	description	312199	312196	312171	Drosophila_melanogaster	
59954	dicistronic	109	109	1	Drosophila_melanogaster	
59959	mutant_in_strain	14	14	5	Drosophila_melanogaster	
59951	non_canonical_splice_site		488	488	2	Drosophila_melanogaster
59953	problem	6350	6350	12	Drosophila_melanogaster	

gene_page

This is a sample gene page view to list most attributes for a gene feature. **NOTE:** this kind of multi-table join view can be very slow to execute on a large genome database.

Usage: `dev_chado_01c=# select v.* from v_genepage2 v join feature as f using (feature_id) where f.name = 'PAU1';`

```
CREATE OR REPLACE VIEW v_genepage2
  (feature_id, field, value)
AS
  SELECT feature_id AS feature_id, 'Name' as field, name as value FROM feature
UNION ALL
  SELECT feature_id AS feature_id, 'uniquename' as field, uniquename as value FROM feature
UNION ALL
  SELECT feature_id AS feature_id, 'seqlen' as field, cast(seqlen as text) as value FROM feature
UNION ALL
  SELECT f.feature_id AS feature_id, 'type' as field, c.name as value
  FROM feature f, cvterm c WHERE f.type_id = c.cvterm_id
UNION ALL
  SELECT f.feature_id AS feature_id, 'organism' as field, o.abbreviation as value
```

Sample_Chado_SQL

```
FROM feature f, organism o WHERE f.organism_id = o.organism_id

UNION ALL
SELECT fs.feature_id AS feature_id,
CASE WHEN fs.is_current IS FALSE THEN 'Synonym_2nd' ELSE 'Synonym' END AS field,
s.name as value
FROM feature_synonym fs, synonym s
WHERE fs.synonym_id = s.synonym_id

UNION ALL
SELECT f.feature_id AS feature_id, 'Dbxref' as field, gd.name||':'||gx.accession as value
FROM feature f, db gd, dbxref gx
WHERE f.dbxref_id = gx.dbxref_id and gx.db_id = gd.db_id

UNION ALL
SELECT fs.feature_id AS feature_id,
CASE WHEN fs.is_current IS FALSE THEN 'Dbxref obsolete' ELSE 'Dbxref 2' END AS field,
(d.name || ':' || s.accession)::text AS value
FROM feature_dbxref fs, dbxref s, db d
WHERE fs.dbxref_id = s.dbxref_id and s.db_id = d.db_id

UNION ALL
SELECT fc.feature_id AS feature_id, c.name AS field,
substr(cv.name,1,40) || ';' || dx.accession AS value
FROM feature_cvterm fc, cvterm cv, cv c, dbxref dx
WHERE fc.cvterm_id = cv.cvterm_id and cv.cv_id = c.cv_id
and cv.dbxref_id = dx.dbxref_id

UNION ALL
SELECT fp.feature_id AS feature_id, cv.name AS field, fp.value AS value
FROM featureprop fp, cvterm cv
WHERE fp.type_id = cv.cvterm_id

UNION ALL
SELECT fl.feature_id AS feature_id, 'location' as field,
chr.uniquename ||':'|| cast( fl.fmin+1 as text) ||'..'|| cast( fl.fmax as text)
|| CASE
WHEN fl.strand IS NULL THEN ' '
WHEN fl.strand < 0 THEN ' [-]'
ELSE ' [+]'
END AS value
FROM featureloc fl, feature chr
WHERE fl.srcfeature_id = chr.feature_id

UNION ALL
SELECT af.feature_id AS feature_id,
'an:' ||
CASE
WHEN a.name IS NOT NULL THEN a.name
WHEN a.sourcename IS NOT NULL THEN (a.program || '.' || a.sourcename)::text
ELSE a.program
END AS field,
CASE
WHEN af.rawscore IS NOT NULL THEN cast(af.rawscore as text)
WHEN af.normscore IS NOT NULL THEN cast(af.normscore as text)
WHEN af.significance IS NOT NULL THEN cast(af.significance as text)
ELSE cast(af.identity as text)
END AS value
FROM analysisfeature af, analysis a
```

```
WHERE af.analysis_id = a.analysis_id
;
```

simple gene_page output

```
dev_chado_01c=# select v.* from v_genepage2 v join feature as f using (feature_id) where f.name
feature_id | field | value
-----+-----+-----
          23 | Name | PAU1
          23 | uniquename | PAU1
          23 | seqlen |
          23 | type | gene
          23 | organism | S.cerevisiae
          23 | Synonym | PAU1
          23 | Dbxref | GeneID:853232
          23 | Dbxref 2 | GFF_source:GenBank
          23 | Dbxref 2 | GeneID:853232
          23 | gene | PAU1
          23 | locus_tag | YJL223C
          23 | location | NC_001142:8776..9138 [-]
```

The above data was loaded from Yeast GenBank Genome (i.e. not very complex)

longer gene_page output

See this [Sample Chado gene report](#) for a well studied gene from FlyBase chado release 5.

More Information

Please send questions to the GMOD developers list:

gmod-devel@lists.sourceforge.net

Authors

- [Dongilbert](#) 16:05, 16 April 2007 (EDT)