==========

ENSEMBL GENEBUILD WORKSHOP
14-16 September 2008
==========

Welcome to our Ensembl Developers' Workshop 2008!

In this worksheet, we will go through some aspects of the genebuild. We will be working with a small part of the mouse (NCBIM37) genome throughout this workshop. We will assume that you have some basic knowledge of Linux and MySQL. Please don't hesitate to ask us if you have any questions.

Enjoy,
Bronwen Aken and Jan Vogel

Overview

1. What we've already done for you
2. Why do we use a pipeline/database system?
3. What is an Ensembl database?
4. Regions used in this exercise – Mouse genome
5. Exercise One: RepeatMask a sequence manually
6. Exercise two: RepeatMask sequences automatically and write results into database

6.1 Check out Ensembl code and set PERL5LIB
6.2 Create a reference database (schema 50)
6.3 Download sequence and load into seq_region table
6.4 Loading assembly mappings into assembly-table
6.5 Populate some additional tables and set toplevel attributes
6.6 Setup rawcomputes for repeat-masking
6.7 Test_RunnableDB
6.8 Add rules
6.9 Make input_ids to run the analysis on CONTIG
6.10Run RepeatMasker

7.

1. What we've already done for you

We have set up a working environment for you that is very similar to the type of environment we work in on a day-to-day basis. We have down loaded the Ensembl API from http://www.ensembl.org/info/using/api/api_installation.html.
eg.

cd /home/training/ensembl-src/
cvs -d :pserver:cvsuser@cvs.sanger.ac.uk:/cvsroot/ensembl checkout –r branch-ensembl-50 ensembl
cvs -d :pserver:cvsuser@cvs.sanger.ac.uk:/cvsroot/ensembl checkout –r HEAD ensembl-analysis

Two directories in /home/training/ensembl-src/ were not downloaded because they contain data that is not publicly available:
ensembl-config and ensembl-personal
The ensembl-config directory contains config for running analyses in a pipeline. The ensembl-personal directory is for your own documentation and scripts.

Some data relevant to this workshop can be found in:
/home/training/ensembl-data/genebuild/assembly and
/home/training/ensembl-data/genebuild/input_ids

We have also created an output directory:
/home/training/ensembl-data/genebuild/output
Output from pipeline jobs will be written here, as specified in: /home/training/ensembl-src/ensembl-config/mouse/NCBIM37/Bio/EnsEMBL/Pipeline/Config/BatchQueue.pm

Raw data can be found here:
/home/training/ensembl-data/genebuild/

2. Why do we use a pipeline/database system?

We think of the genebuild process as a 'pipeline' where we put raw data in (eg. proteins, cDNAs and ESTs) and we get gene models out. Running analyses across a large genome uses up a lot of computer time and so we run analyses on short sequences in parallel. The code used to run these batches of jobs is the 'pipeline code' (ensembl-pipeline). The code that handles all of our analyses is the 'analysis code' (ensembl-analysis).

The mouse genome contains seq of ~20.000 contigs (~ 3,400,000,000 bp total) - 3.4 gig bases
- hard to run 20.000 jobs by hand and to monitor which ones fail
- hard to deal with flat files
- hard to extract sequence or repeats partly / small regions

Solution:

- load sequence in db and store results in db
- control of jobs to run is also handled by database
- use mysql

- job-control system use by Ensembl is LSF, but other systems can also be used
- jobs run on big compute farm

3. What is an Ensembl database?

An Ensembl database is a MySQL database with a specific schema. The overall schema has been stable for a long time, although there are small changes to the schema every release. The database has tables to hold DNA, alignments of raw data, gene models, repeats, and cross-references to external databases. In addition, the database also keeps track of each job that has been run.

4. Regions used in this exercise - Mouse genome

chr "2" 3326467 3329254 2kb
chr "3" 94740817 94748115 7kb
chr "6" 129365239 129372376 7kb
chr "10" 81053780 81057330 3kb
chr "11" 69782598 69788762 6kb
chr "13" 21570962 21576605 5kb

5. Exercise One: RepeatMask a sequence manually

The genebuild process:
Unmasked DNA sequence is loaded into a database. The first analyses we run on this sequence will mask out repetitive regions eg. RepeatMasker and Dust. We mask/filter out the DNA before aligning proteins, cDNAs and ESTs because these alignments. Alignment of proteins, cDNAs and ESTs is done on masked sequence because the alignments are compute-intensive and it wastes compute time to align them to unmasked DNA. We do not want to predict alignments in repeat regions.

Location of RepeatMasker source code:
perl /lustre/work1/ensembl/jhv/bin/RepeatMasker_3_1_8/RepeatMasker

RepeatMasker options

RepeatMasker uses different libraries for different species eg. primate library, mammalian library. RepeatMasker also has a number of options which can be configured to allow for quicker masking, or more sensitive masking (this takes longer). Example of options:

-species <SPECIES> : choose species
-qq : run quicker
-nolow : Does not mask low_complexity DNA or simple repeats

-lib : if you'd like to use a custom library
-nolow -species <SPECIES> (used for ensembl builds)

Set up your directories

mkdir -p /home/training/ensembl-data/genebuild/output/repeatmask_output

todo: backup species-specific libraries and make sure analysis runs from cmdline

depending on if libraries are installed for repeatMasker or not, use species-specific
library with -lib option. mouse repeats can be downloaded from sanger ftp site

RepeatMask sequence :NCBIM37:11:69703858:69950060:1.fa

Paste the following command in your terminal:

```
perl /lustre/work1/ensembl/jhv/bin/RepeatMasker_3_1_8/RepeatMasker
-species mouse -qq \
-dir /home/training/ensembl-data/genebuild/output/repeatmask_output \
/home/training/ensembl-
data/genebuild/output/repeatmask_output/chromosome:NCBIM37:11:69703858:6995006
0:1.fa
```

RepeatMask sequence: test_sequence_to_repeatmask.fa

It takes RepeatMasker about 30 seconds to mask the test-sequence of length approx. 10kb.

Paste the following command in your terminal:

```
perl /lustre/work1/ensembl/jhv/bin/RepeatMasker_3_1_8/RepeatMasker \
-species mouse -qq \
-dir /home/training/ensembl-data/genebuild/output/repeatmask_output \
/home/training/ensembl-
data/genebuild/output/repeatmask_output/test_sequence_to_repeatmask.fa
```

Have a look at your RepeatMasker output

The output from your RepeatMasker run (above) has been written to:
/home/training/ensembl-data/genebuild/output/repeatmask_output

```
ls -1 /home/training/ensembl-data/genebuild/output/repeatmask_output
test_sequence_to_repeatmask.fa.log - log
test_sequence_to_repeatmask.fa.masked - hardmasked fa-file
test_sequence_to_repeatmask.fa.tbl - summary of identifed rpt-types (%perc)
```

test_sequence_to_repeatmask.fa.out - list of identifed rpt and positions
test_sequence_to_repeatmask.fa.cat - different format of out-file

Check your answer

Jan's results are stored in:
/home/training/ensembl-genebuild-data/2_seq_data_loading/test_seqs/repeatmask_output_results

Look at the repeats identified in files

Summary in: /home/training/ensembl-genebuild-data/2_seq_data_loading/test_seqs/clones_finished_mini_testseq.fa

less /home/training/ensembl-genebuild-data/2_seq_data_loading/test_seqs/clones_finished_mini_testseq.fa.masked

SINEs: 84 11167 bp 30.21 %
Alu/B1 38 4235 bp 11.46 %
B2-B4 45 6851 bp 18.54 %
IDs 1 81 bp 0.22 %
MIRs 0 0 bp 0.00 %
etc.

SINEs LINEs LTR elements DNA elements

The hardmasked file: REPEAT is replaced by NNNNNNNN

6. Exercise two: RepeatMask sequences automatically and write results into database

There are a number of steps we need to do before we can run the RepeatMask analysis. These can be time-consuming, but careful set-up is important.

6.1 Check out Ensembl code and set PERL5LIB

Documentation here: http://www.ensembl.org/info/software/api_installation.html

Section 6.1 has already been done for you, so you can skip to section 6.2 after setting your PERL5LIB.

Option (i):

less /home/training/ensembl-src/ensembl-personal/usr/set_perl_path.sh
source /home/training/ensembl-src/ensembl-personal/usr/set_perl_path.sh
(use with /bin/tcsh)

or Option(ii):

```
setenv HOME /home/training/ensembl-src/
setenv PERL5LIB ${HOME}/bioperl-live
setenv PERL5LIB ${PERL5LIB}:${HOME}/ensembl/modules
setenv PERL5LIB ${PERL5LIB}:${HOME}/ensembl-analysis/modules
setenv PERL5LIB ${PERL5LIB}:${HOME}/ensembl-pipeline/modules
```

or Option (iii) if you use bash:

```
PERL5LIB=${PERL5LIB}:${HOME}/bioperl-live
PERL5LIB=${PERL5LIB}:${HOME}/ensembl/modules
PERL5LIB=${PERL5LIB}:${HOME}/ensembl-pipeline/modules
PERL5LIB=${PERL5LIB}:${HOME}/ensembl-analysis/modules
export PERL5LIB
```

Now check:

```
echo $PERL5LIB
```

6.2 Create a reference database (schema 50)

* Note address of mysql-server and port

Create database with name ‘usrname_mouse37’:

```
mysql -uens-training -pworkshop -hmyhost -Pmyport –e "create database
usrname_mouse37_ref ;"
```

Load core tables:

```
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref <
$HOME/ensembl/sql/table.sql
```

Load pipeline tables:

```
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref <
$HOME/ensembl-pipeline/sql/table.sql
```

See which tables are loaded:

```
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref –e 'show
tables '
```

Look at the gene table:

```
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref –e 'desc
gene '
```

Check that the seq_region table is empty:
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref –e 'select * from seq_region'

6.3 Download sequence and load into seq_region table

Make a mini chr_contig.agp - file
cd /home/training/ensembl-genebuild-data/2_seq_data_loading
vi /home/training/ensembl-genebuild-data/2_seq_data_loading/contigs_javier_needed.txt

```
AL589742
AL596185
AC153919
AC138620
AC087062
AL732620
```

grep -f contigs_javier_needed.txt
/lustre/work1/ensembl/sd3/mouse/NCBIM37/raw_sequence/chr_contig.agp >\
/home/training/ensembl-genebuild-data/2_seq_data_loading/mini_chr_contig.agp

setenv SCRIPTS /home/training/ensembl-src/ensembl-personal/usr/scripts/

Load chromosome level. This script writes to seq_region and coord_system tables.

```
perl $SCRIPTS/load_chromosome_seq_regions.pl \
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref -dbpass ensembl \
-coord_system_name chromosome -coord_system_version NCBIM37 -rank 1 -default_version \
-agp_file /home/training/ensembl-genebuild-data/2_seq_data_loading/mini_chr_contig.agp
```

Look at the regions entered into seq_region:
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref –e 'select * from seq_region'

Make mini file to load supercontig level:

```
grep -f /home/training/ensembl-genebuild-data/2_seq_data_loading/contigs_javier_needed.txt supercontig_contig.agp > \
/home/training/ensembl-genebuild-data/2_seq_data_loading/mini_supercontig_contig.agp
cat /home/training/ensembl-genebuild-data/2_seq_data_loading/mini_supercontig_contig.agp
```

Load supercontig level. Script writes to seq_region and coord_system table

```
perl $SCRIPTS/load_superctg_seq_regions.pl \
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref -
dbpass workshop \
-coord_system_name supercontig \
-default_version -rank 2 \
-coord_system_version NCBIM37 \
-agp_file /home/training/ensembl-genebuild-
data/2_seq_data_loading/mini_supercontig_contig.agp
```

Look at the regions entered into seq_region:

```
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref -e'select
* from seq_region'

mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref -e'select
* from coord_system ' ;
```

Make mini file to load contig level:

```
grep -f /home/training/ensembl-genebuild-
data/2_seq_data_loading/contigs_javier_needed.txt \
/lustre/scratch1/ensembl/sd3/mouse37/clones_finished.1.fa

perl ~/scripts/extractSequence_outof_fasta.pl -fasta $SCR/clones_finished.1.fa -file \
/home/training/ensembl-genebuild-data/2_seq_data_loading/contigs_javier_needed.txt\
-outfile /home/training/ensembl-genebuild-
data/2_seq_data_loading/clones_finished_mini.fa

perl ~/cvs_checkout/ensembl-personal/jhv/builds/buildscripts/fingerprint_db.pl\
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref -
dbpass workshop
```

Load contig level. Script writes to coord_system, dna and seq_region table

```
perl $SCRIPTS/load_mouse_seq_regions.pl \
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref -
dbpass workshop \
-coord_system_name contig \
-default_version \
-rank 3 \
-sequence_level \
-coord_system_version NCBIM37 \
-fasta_file /home/training/ensembl-genebuild-
data/2_seq_data_loading/clones_finished_mini.fa
```

Look at the regions entered into seq_region:

```
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref -e'select * from seq_region'

mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref -e'select * from coord_system ' ;

mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref -e'select count(*) from dna' ;
```

Extract sequence out of database – just to test that it works

Make mini file to load clone level
Load clone2contig mapping into coord_system and seq_region table

```
grep -f /home/training/ensembl-genebuild-data/2_seq_data_loading/contigs_javier_needed.txt /lustre/work1/ensembl/sd3/mouse/NCBIM37/raw_sequence/clone_contig.agp >\
/home/training/ensembl-genebuild-data/2_seq_data_loading/mini_clone_contig.agp
```

```
cp /lustre/work1/ensembl/sd3/mouse/NCBIM37/raw_sequence/clone_contig.agp \
$HOME/projects/project_workshop
```

Load clone level.

```
perl $HOME/ensembl-pipeline/scripts/load_seq_region.pl \
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref -dbpass workshop \
-coord_system_name clone \
-default_version \
-coord_system_version NCBIM37 \
-rank 4 \
-agp_file /home/training/ensembl-genebuild-data/2_seq_data_loading/mini_clone_contig.agp
```

NOTE : corrected mini_clone_contig.agp - changed length of AC153919.8 from 80852 to 264561

Query seq_region table:

```
select count(*) from seq_region s1, seq_region s2 where s1.name = s2.name and s1.length != s2.length and s1.coord_system_id = 3 and s2.coord_system_id = 4 limit 1 ;
```

Delete version numbers from coord_system table for contig and clone

```
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref
select * from coord_system ;
update coord_system set version=NULL where name ='clone' ;
```

update coord_system set version=NULL where name ='contig' ;
select * from coord_system ;

```
+-----------------+------------+---------+------+--------------------
----------+
| coord_system_id | name | version | rank | attrib |
+-----------------+------------+---------+------+--------------------
----------+
| 1 | chromosome | NCBIM37 | 1 | default_version |
| 2 | supercontig | NCBIM37 | 2 | default_version |
| 3 | contig | NULL | 3 | default_version,sequence_level |
| 4 | clone | NCBIM37 | 4 | default_version |
+-----------------+------------+---------+------+--------------------
----------+
```

snapshot : usrname_mouse37_ref_snap_one
mysqldump -h genebuild1 -u ensro usrname_mouse37_ref > \
/home/training/ensembl-genebuild-data/db_snap_shots/usrname_mouse37_ref_snap_one

6.4 Loading assembly mappings into assembly-table

setenv SCRIPTS ~/perl_code/ensembl-personal/jhv/builds/mmusculus/scripts/
setenv DATA /lustre/work1/ensembl/sd3/mouse/NCBIM37
setenv SCR /lustre/scratch1/ensembl/sd3/mouse37

no mini prep - load_file already made
chromosome ---> contig
perl ~/cvs_checkout/ensembl-personal/jhv/builds/buildscripts/fingerprint_db.pl\
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref -
dbpass workshop
writes to asssembly- and meta table

perl $HOME/ensembl-pipeline/scripts/load_agp.pl \
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref -
dbpass workshop \
-assembled_name chromosome \
-component_name contig \
-agp_file /home/training/ensembl-genebuild-
data/2_seq_data_loading/mini_chr_contig.agp
warnings MSG: You are already using AL589742.21 in another place in your assembly
are you sure you want to
#
supercontig ---> contig
#
mini prep - load_file already made
grep -f /home/training/ensembl-genebuild-
data/2_seq_data_loading/contigs_javier_needed.txt\

/lustre/work1/ensembl/sd3/mouse/NCBIM37/raw_sequence/supercontig_contig.agp >\
/home/training/ensembl-genebuild-data/2_seq_data_loading/mini_supercontig_contig.agp
load NT* -> AC (supercontig-to-contig) mapping
cat /home/training/ensembl-genebuild-
data/2_seq_data_loading/mini_supercontig_contig.agp
NT_039202 208471 436586 2 F AL732620.14 2001 230116 +
NT_039240 44047803 44249301 335 F AC087062.25 22953 224451 +
NT_039353 81573483 81742068 594 F AC138620.4 41261 209846 +
NT_039500 23370008 23450859 212 F AC153919.8 1 80852 -
NT_096135 35208736 35454938 263 F AL596185.12 2001 248203 +
NT_039578 10375984 10489377 86 F AL589742.21 2001 115394 +
NT_039578 10489378 10499548 87 F AL589742.21 115471 125641 +

Load supercontig to contig mapping. Script writes to asssembly- and meta table
perl $HOME/ensembl-pipeline/scripts/load_agp.pl \
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref -dbpass workshop \
-assembled_name supercontig \
-component_name contig \
-agp_file /home/training/ensembl-genebuild-
data/2_seq_data_loading/mini_supercontig_contig.agp

mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref -e'select * from assembly '

asm_seq_region_id	cmp_seq_region_id	asm_start	asm_end	cmp_start	cmp_end	ori
1	14	129260521	129429106	41261	209846	1
2	17	69703858	69950060	2001	248203	1
3	13	94665450	94866948	22953	224451	1
4	16	21549325	21662718	2001	115394	1
4	16	21662719	21672889	115471	125641	1
5	15	81038621	81119472	1	80852	-1
6	18	3208471	3436586	2001	230116	1
7	15	23370008	23450859	1	80852	-1
8	14	81573483	81742068	41261	209846	1
9	13	44047803	44249301	22953	224451	1
10	16	10375984	10489377	2001	115394	1
10	16	10489378	10499548	115471	125641	1
11	17	35208736	35454938	2001	248203	1
12	18	208471	436586	2001	230116	1

Load clone to contig mapping. Script writes to assembly and meta-table
eg. mapping-entry in meta-table : assembly.mapping clone#contig

```
perl $HOME/ensembl-pipeline/scripts/load_agp.pl \
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref -
dbpass workshop \
-assembled_name clone \
-component_name contig \
-agp_file /home/training/ensembl-genebuild-
data/2_seq_data_loading/mini_clone_contig.agp
```

snapshot : usrname_mouse37_ref_snap_two

```
mysqldump -h genebuild1 -u ensro usrname_mouse37_ref > \
/home/training/ensembl-genebuild-data/db_snap_shots/usrname_mouse37_ref_snap_two
# i do not add the direct mapping between chromosome and NT-contig as i think it's not
needed ....
```

6.5 Populate some additional tables and set toplevel attributes

The tables we'll need to populate are:
external_db
attrib_types
unmapped_object_types

```
echo $HOME
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref
```

```
truncate external_db;
load data local infile $HOME/ensembl/misc-scripts/external_db/external_dbs.txt' into
table external_db;
delete from external_db where external_db_id = 0;
```

```
truncate attrib_type;
load data local infile $HOME/ensembl/misc-scripts/attribute_types/attrib_type.txt' into
table attrib_type;
select * from attrib_type where attrib_type_id = 6;
```

```
truncate unmapped_reason;
 load data local infile $HOME/ensembl/misc-
scripts/unmapped_reason/unmapped_reason.txt' into table unmapped_reason ;
```

Set toplevel attributes

```
perl $HOME/ensembl-pipeline/scripts/set_toplevel.pl \
-dbhost genebuild1 -dbport 3306 -dbuser ens-training -dbpass workshop -dbname
usrname_mouse37_ref
```

Check loaded sequence by dumping it out

```
mkdir -p /home/training/ensembl-data/genebuild/output/unmasked_seq

perl $HOME/ensembl-analysis/scripts/sequence_dump.pl\
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref -
dbpass workshop \
-coord_system_name chromosome \
-output_dir /home/training/ensembl-data/genebuild/output/unmasked_seq
```

The dumped sequence, in /home/training/ensembl-data/genebuild/output/unmasked_seq, should contain lots of NN and tail contains sequence.

```
tail /home/training/ensembl-data/genebuild/output/unmasked_seq/*
```

You could also diff your sequence against Jan's:

6.6 Setup rawcomputes for repeat-masking

```
# using libraries provided by www.girinst.org

mkdir -p /home/training/ensembl-data/genebuild/output/repeatmask_output
cd /home/training/ensembl-data/genebuild/output

#
# either get the test seq out of the created ensembl database or use
# clones_finished_mini_testseq.fa
#
# seq written to : chromosome:NCBIM37:11:69703858:69803858:1.fa
#

# /home/training/ensembl-genebuild-
data/2_seq_data_loading/test_seqs/clones_finished_mini_testseq.fa
```

We choose a random piece of the genome and fetch its sequence:

```
perl /home/training/ensembl-genebuild-data/scripts/get_sequence.pl \
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref\
-dbpass workshop \
-coord_system_name chromosome:NCBIM37:11:69703858:69803858:1 \
-onefile -output_dir /home/training/ensembl-data/genebuild/output
```

We will be running the following three analyses:

- RepeatMask
- TRF
- tRNAscan

All analyses will be run on the contig level.

Set up an analysis for RepeatMasker - analysis, rules and the role of input_id_types

Let's first look at the schema:

mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref -e'desc analysis ' ;

```
+-----------------+---------------------+------+-----+----------------
----+----------------+
| Field | Type | Null | Key | Default | Extra |
+-----------------+---------------------+------+-----+----------------
----+----------------+
| analysis_id | smallint(5) unsigned | NO | PRI | NULL | auto_increment
|
| created | datetime | NO | | 0000-00-00 00:00:00 | |
* | logic_name | varchar(40) | NO | UNI | | |
| db | varchar(120) | YES | | NULL | |
| db_version | varchar(40) | YES | | NULL | |
| db_file | varchar(120) | YES | | NULL | |
* | program | varchar(80) | YES | | NULL | |
| program_version | varchar(40) | YES | | NULL | |
* | program_file | varchar(80) | YES | | NULL | |
* | parameters | varchar(255) | YES | | NULL | |
* | module | varchar(80) | YES | | NULL | |
| module_version | varchar(40) | YES | | NULL | |
| gff_source | varchar(40) | YES | | NULL | |
| gff_feature | varchar(40) | YES | | NULL | |
+-----------------+---------------------+------+-----+----------------
----+----------------+
```

mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref -e'desc input_id_analysis ; '

```
+---------------+----------------------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+---------------+----------------------+------+-----+---------+-------+
| input_id | varchar(100) | NO | PRI | | |
| input_id_type | varchar(40) | NO | MUL | | |
| analysis_id | smallint(10) unsigned | NO | PRI | | |
| created | datetime | NO | | | |
| runhost | varchar(20) | NO | | | |
| db_version | varchar(40) | NO | | | |
| result | smallint(10) unsigned | NO | | | |
+---------------+----------------------+------+-----+---------+-------+
```

Add analysis using script

less $HOME/ensembl-config/mouse/NCBIM37/pipe_conf/repeatmask_ana.conf
Check that the program_file and other information are correct.

Your file should look something like this:

```
[RepeatMask]
db=repbase
db_version□0129
db_file=repbase
program=RepeatMasker
program_version=3.1.8
program_file=/path/to/repmasker/RepeatMasker
parameters=-nolow -species mouse -s
module=RepeatMasker
gff_source=RepeatMasker
gff_feature=repeat
input_id_type=CONTIG
```

snapshot : usrname_mouse37_ref_snap_wo_rules
mysqldump -h genebuild1 -u ensro usrname_mouse37_ref > \
/home/training/ensembl-genebuild-data/db_snap_shots/usrname_mouse37_ref_wo_rules

And now run the script:

perl $HOME/ensembl-pipeline/scripts/analysis_setup.pl \
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref \
-dbpass workshop -read \
-file $HOME/ensembl-config/mouse/NCBIM37/pipe_conf/repeatmask_ana.conf

Gotcha! The above command does not work because we need to add the pipeline-scripts dir to our path !
=> need to add $HOME/ensembl-pipeline/scripts/ to the PERL5LIB

setenv PERL5LIB ${PERL5LIB}:${HOME}/ensembl-pipeline/scripts

And now try running analysis_setup.pl again ....

Check that the analysis has been added correctly:
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref -e'select * from analysis ;'

```
INSTALL
INSTALL REPEATMASKER
INSTALL
INSTALL cd /home/training/ensembl-genebuild-
data/bin/repeatmasker_318
INSTALL perl ./configure
INSTALL
INSTALL perl_path : /vol/software/linux-i386/farm/bin/perl
INSTALL repeatmasker_dir :
/nfs/acari/jhv/cshl_workshop/bin/repeatmasker_318/
INSTALL search_engine : crossmatch
```

```
INSTALL crossmatch_path :/nfs/acari/jhv/cshl_workshop/bin/
INSTALL
INSTALL
```

6.7 test_RunnableDB

Before running any job on the farm, it's a good idea to first run a job in test_RunnableDB.

```
perl $HOME/ensembl-analysis/scripts/test_RunnableDB\
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref \
-dbpass workshop -analysis RepeatMask -input_id
chromosome:NCBIM37:3:94665450:94676948:1\
-verbose
```

The command fails if we have not set up the relevant config files. We have already set some of the config files for you, but it's worth having a look at them anyway.

These files have been copied from $HOME/ensembl-analysis/modules/Bio/EnsEMBL/Analysis/Config:

```
cd $HOME/ensembl-config/mouse/NCBIM37/Bio/EnsEMBL/Analysis/Config
ls
General.pm
```

These files have been copied from $HOME/ensembl-pipeline/modules/Bio/EnsEMBL/Pipeline/Config:

```
cd $HOME/ensembl-config/mouse/NCBIM37/Bio/EnsEMBL/Pipeline/Config
ls
General.pm    BatchQueue.pm
```

For BatchQueue, check the output_dir (and create if necessary), queue,

Make sure that the config directory ($HOME/ensembl-config/mouse/NCBIM37) is added to your PERL5LIB:

```
echo $PERL5LIB
```

6.8 Add rules

```
perl $HOME/ensembl-pipeline/scripts/analysis_setup.pl \
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbname usrname_mouse37_ref \
-dbpass workshop -read \
-file $HOME/ensembl-config/mouse/NCBIM37/pipe_conf/submit_contig_ana.conf
```

- If we would like to run the analysis RepeatMasker on the Contig level, then our RepeatMask analysis needs to have input_id_type of CONTIG. Check this by querying the input_id_type_analysis table.
- All analyses (eg. RepeatMasker, Genscan, Exonerate) need a 'dummy analysis' having the same input_id_type as the analysis we'd like to run.
- Two analyses can share the same dummy analysis if they both have the same input_id_type. Eg. If both RepeatMasker and Pmatch are to be run on the contig level, then they can both use the SubmitContig dummy analysis. However, if RepeatMasker is to be run on contig level and Pmatch is to be run on the chromosome level, then they will each need their own dummy analysis – SubmitContig and SubmitChromosome respectively.
- Every analysis requires at least one rule that defines the conditions under which the analysis is allowed to run. RepeatMasker is allowed to run if SubmitContig has been run. Other analyses (eg. alignment of proteins) might require both the dummy analysis and another analysis (eg. RepeatMasker) to have been successful.
- Dummy analyses do not require rules
- Each analysis usually has its own 'accumulator', although this is not required. An accumulator for Pmatch would be called Pmatch_wait. It is only allowed to run after ALL Pmatch jobs have run successfully. Pmatch-dependent analyses will have Pmatch_wait as a condition. Thus, the accumulator prevents any jobs from Pmatch-dependent analyses from running until all Pmatch jobs are done.

Add a rule for RepeatMasker to your database, saying that SubmitContig must have finished before RepeatMasker can run:

```
perl $HOME/ensembl-pipeline/scripts/RuleHandler.pl\
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbpass workshop \
-dbname usrname_mouse37_ref \
-insert -goal RepeatMask -condition SubmitContig
```

Show rules:

```
perl $HOME/ensembl-pipeline/scripts/RuleHandler.pl\
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbpass workshop -dbname
usrname_mouse37_ref \
-rules
```

We can also use files to set up rules, eg:

```
[RepeatMask]
condition=SubmitContig
```

The script used is similar to the analysis_setup script that we used earlier.

```
perl $HOME/ensembl-pipeline/scripts/rule_setup.pl\
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbpass workshop \
```

-read \
-file /home/training/ensembl-genebuild-data/analyses/repeatmask_rules.conf

Rules are written to the rules_conditions and the rule_goal table :
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref -e'desc rule_conditions ; '

```
+----------------+----------------------+------+-----+---------+-------
+
| Field | Type | Null | Key | Default | Extra |
+----------------+----------------------+------+-----+---------+-------
+
| rule_id | smallint(10) unsigned | NO | MUL | | |
| rule_condition | varchar(40) | YES | | NULL | |
+----------------+----------------------+------+-----+---------+-------
+
```

mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref -e'desc rule_goal'

```
+---------+----------------------+------+-----+---------+-------------
--+
| Field | Type | Null | Key | Default | Extra |
+---------+----------------------+------+-----+---------+-------------
--+
| rule_id | smallint(10) unsigned | NO | PRI | NULL | auto_increment |
| goal | varchar(40) | YES | | NULL | |
+---------+----------------------+------+-----+---------+-------------
--+
```

A useful query to list rules and goal :
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref
select rc.*, rg.*, logic_name as analysis_to_run from rule_conditions rc, rule_goal rg , analysis a where rc.rule_id = rg.rule_id and goal = analysis_id ;
exit ;

```
+---------+----------------+---------+------+-----------------+
| rule_id | rule_condition | rule_id | goal | analysis_to_run |
+---------+----------------+---------+------+-----------------+
| 1 | SubmitContig | 1 | 1 | RepeatMask |
+---------+----------------+---------+------+-----------------+
```

snapshot
mysqldump -h genebuild1 -u ensro usrname_mouse37_ref > \
/home/training/ensembl-genebuild-data/db_snap_shots/usrname_mouse37_ref_rules

6.9 Make input_ids to run the analysis on CONTIGS :

select * from input_id_analysis ;
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref
select ia,a.logic_name

```
from input_id_analysis ia.*, analysis a
where ia.analysis_id = a.analysis_id ;
```

Run script:

```
perl $HOME/ensembl-pipeline/scripts/make_input_ids \
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbpass workshop \
-dbname usrname_mouse37_ref \
-logic_name SubmitContig -slice -coord_system contig
```

```
# snapshot
mysqldump -h genebuild1 -u ensro usrname_mouse37_ref > \
/home/training/ensembl-genebuild-data/db_snap_shots/usrname_mouse37_ref_input_ids
```

6.10 Run RepeatMasker

And finally…!

```
bsub -o local_pipeline_run.out \
perl $HOME/ensembl-pipeline/scripts/rulemanager.pl \
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbpass workshop -dbname
usrname_mouse37_ref \
-analysis RepeatMask
```

bjobs

```
JOBID USER STAT QUEUE FROM_HOST EXEC_HOST JOB_NAME
SUBMIT_TIME
888830 jhv RUN normal bc-9-1-01 bc-12-1-03 *epeatMask Oct 24 14:24
888831 jhv RUN normal bc-9-1-01 bc-12-4-09 *epeatMask Oct 24 14:24
888832 jhv RUN normal bc-9-1-01 bc-12-2-04 *epeatMask Oct 24 14:24
888833 jhv RUN normal bc-9-1-01 bc-12-2-04 *epeatMask Oct 24 14:24
888834 jhv RUN normal bc-9-1-01 bc-12-3-06 *epeatMask Oct 24 14:24
888835 jhv RUN normal bc-9-1-01 bc-12-3-01 *epeatMask Oct 24 14:24
```

takes too long to run in local

Running this script creates an entry in meta_table to lock the pipeline:

```
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref
select * from meta where meta_key ='pipeline.lock'
+---------+-----------------+------------------------------------------
------------+
| meta_id | meta_key | meta_value |
+---------+-----------------+------------------------------------------
------------+
| 14 | pipeline.lock | jhv@bc-9-1-
01.internal.sanger.ac.uk:20677:1193228390 |
+---------+-----------------+------------------------------------------
```

```
------------+
```

Rulemanager also creates a status for each job running:

```
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref -e'select
input_id from input_id_analysis \
where input_id_type ="CONTIG" limit 1 '

# only run on 5 jobs on very small sequence slices ( 10.000 k )
# generate input_ids for that with insert_sql.statement - they need to be of
# type 'CONTIG' - than delete repeat featuers which they have computed and
# source new table ( repeat_feature / repeat_repeat_consensus )
#
# after RepeatMask analysis finished take snapshot, than create input_ids for 5 small jobs
in file .....
# SubmitContig has to have analysis_id 2 !!!!
# file $HOME//cshl_workshop/4_input_ids_sql/submit_contig_small_input_ids.sql
insert into input_id_analysis(input_id, input_id_type, analysis_id) values
('chromosome:NCBIM37:3:94661000:94662000:1','CONTIG',2);
insert into input_id_analysis(input_id, input_id_type, analysis_id) values
('chromosome:NCBIM37:3:94662000:94663000:1','CONTIG',2);
insert into input_id_analysis(input_id, input_id_type, analysis_id) values
('chromosome:NCBIM37:3:94663000:94664000:1','CONTIG',2);
insert into input_id_analysis(input_id, input_id_type, analysis_id) values
('chromosome:NCBIM37:3:94664000:94665000:1','CONTIG',2);
insert into input_id_analysis(input_id, input_id_type, analysis_id) values
('chromosome:NCBIM37:3:94665000:94666000:1','CONTIG',2);
insert into input_id_analysis(input_id, input_id_type, analysis_id) values
('chromosome:NCBIM37:3:94666000:94667000:1','CONTIG',2);
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref -e'delete
from input_id_analysis where analysis_id_id = 2 '
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref <
/home/training/ensembl-genebuild-
data/4_input_ids_sql/submit_contig_small_input_ids.sql
# run RepeatMask analysis on fake, small input_ids
perl $PS/rulemanager.pl \
-dbhost genebuild1 -dbuser ens-training -dbport 3306 -dbpass workshop \
-analysis RepeatMask
```

Look at output file and at repeat_feature / repeat_consensus analysis

```
# delete these features and load the proper ones for the whole region
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref <
/home/training/ensembl-genebuild-data/4_input_ids_sql/repeat_feature_full_dump.sql
mysql -uens-training -pworkshop -hmyhost -Pmyport -D usrname_mouse37_ref <
/home/training/ensembl-genebuild-data/4_input_ids_sql/repeat_consensus_full_dump.sql
```

this will run in paralell :
NAME : input_id_type: module: rule_condition :

SubmitContig ---> [CONTIG] Dummy ---
RepeatMask --> [CONTIG] RepeatMask SubmitContig
SubmitSlice --> [SLICE] Dummy ---
Genscan --> [SLICE] Genscan SubmitSlice
#
ACCUMULATORS wait until analysis is finished .....
#
NAME : input_id_type: module: rule_condition :

SubmitContig --> [CONTIG] Dummy ---
RepeatMask --> [CONTIG] RepeatMask SubmitContig
WAIT_RepeatM --> [ACCUMULATOR] Accumulator RepeatMask *new*
SubmitSlice --> [SLICE] Dummy ---
Genscan --> [SLICE] Genscan SubmitSlice AND RepeatMask_WAIT *new*
another example:

Analysis, RunnableDB, input size, section
Exonerate PipelineExonerate CHUNK_FILE A
EST_GeneBuilder EST_GeneBuilder 1MSLICE B
Here is a description of the dependancies you may should see in this system
SubmitCDNA SubmitSlice
CDNACHUNK 1MSLICE
Dummy Dummy
| |
| |
| |
| |
cdna_exonerate |
CDNACHUNK |
Exonerte2Genes |
| |
| |
| /
cdna_exonerate_wait /
ACCUMULATOR /
Accumulator /
| /
| /
EST_GeneBuilder/
1MSLICE
EST_GeneBuilder
None of these settings are specific to the anaysis but are their for the

pipeline functionailty or just descriptive purposes
Analysis to set up :

[SubmitCDNA]
module=Dummy
typeÍNACHUNK
[cdna_exonerate]
module=Exonerate2Genes
typeÍNACHUNK
[cdna_exonerate_wait]
module¬cumulator
type¬CUMULATOR
[Submit1MSlice]
module=Dummy
type=1MSLICE
[EST_GeneBuilder]
module=EST_GeneBuilder
type=1MSLICE
#
rules :
- NO rule for submit-analysis (SubmitCDNA, Submit1MSlice)
#
cdna_exonerate_wait = condition to 'start' is that all cdna_exonerate have finished
if this analysis 'runs' an entry in input_id_analysis will be placed
which indicates that this analysis 'ran'
EST_GeneBuilder = condition to start : - entry in input_id_analysis of analysis
'cdna_exonerate_wait' [ACCUMULATOR]
- entry in input_id_analysis of analysis 'Submit1MSlice' [1MSLICE]
[cdna_exonerate]
condition=SubmitCDNA
[cdna_exonerate_wait]
conditionÍna_exonerate
[EST_GeneBuilder]
condition=Submit1MSlice
condition=exonerate_wait
# who wants can set up analysis for TRF :
- run on contigs
- program is called 'trf'
- module is called 'TRF'
who wants can set up analysis for tRNAscan :
- run on contigs
- porgram-file to use is 'tRNAscan-SE'
- module is tRNAscan_SE
Dust
- program-file is tcdust
- module : Dust

- run on chromosomes
CpG
- program-file cpg
- module : CPG
- run on 100k slices
set up rules for the analysis above
#
repeats now stored in DB - hooray !
#
check how much sequence has been masked :
#
perl $PE/scripts/repeat_coverage.pl \
-repeattypes RepeatMask -path NCBIM37\
-dbhost genebuild1 -dbuser ens-training -dbport 3306 \
-dbname usrname_mouse37_ref -dbpass workshop
dump some softmasked sequence :
Align protein-seqs to DNA : exonerate
=
Genscan
=
great full db out of region
RUN stuff by hand :
cd cshl_programs_test
test_run for exonerate :

./bin/exonerate-0.9.0 --model protein2genome \
--query ./tests/protein.fa --target
./tests/chromosome:NCBIM37:3:94665450:94866948:1.fa
run-output : test_output_exonerate.txt
test_run for genscan :

./bin/genscan ./bin/genscan_matrices/HumanIso.smat
./tests/chromosome:NCBIM37:3:94665450:94866948:1.fa
run-output : test_output_genscan.txt
test_run for tRNAscan-SE :

./bin/tRNAscan-SE ./tests/chromosome:NCBIM37:3:94665450:94866948:1.fa
run-output : test_output_tRNAscan-SE
test_run for trf :

./bin/trf ./tests/chromosome:NCBIM37:3:94665450:94866948:1.fa 2 5 7 80 10 40 500 -d
-h
run-output : test_output_tRNAscan-SE
test_run for cross_match (silly one)

./bin/cross_match ./tests/chromosome:NCBIM37:3:94665450:94866948:1.fa

./tests/protein.fa
run-output : test_output_tRNAscan-SE
cDNA's to align :

NM_001039115
AK034247.1
AK037097.1
NM_207572