Release coordination documentation

Preparation

Preparing this document itself

This document is usually inherited from the previous release cycle and tends to have all the check-boxes ticked. The fastest way to untick them all to start afresh is to go to "Edit...", then switch over to the XML view (a button on the top right with <> on it), and then perform a mass-replace of <ac:task-status>complete</ac:task-status> by <ac:task-status>incomplete</ac:task-status> .

Declaration of Intentions

☑ Once the release coordinator has sent out the email for the declaration of intentions, set up a web page with intentions in the Confluence wiki system to allow easy tracking of the progress.Release plans

☑ Ask compara team members of their intentions
Compara has one extra day to declare their intentions because of the need to know what the genebuilders and associated teams (eg wormbase, ensembl genomes) will declare

☑ Discuss with the team which declarations are worth being put on the front page (3 at most) "*Headline position on the homepage*"

☑ Submit the declaration of intentions on the http://admin.ensembl.org/index.html website and fill in the correct category ("API/Schema changes", "New alignments", etc)

Environment variables

☑ Define $ENSEMBL_CVS_ROOT_DIR
This is necessary to run the Hive and is used by many scripts/files in this document. Make sure this is defined in your terminal

☑ Define $ENSADMIN_PSW
The password for the mysql 'ensadmin' user also needed for many scripts

- ☑ Define $COMPARA_REG variable to simplify connecting to databases via registry
  export COMPARA_REG="-reg_conf $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl -reg_type compara -reg_alias"

Update your checkouts

Ensure you have up-to-date git checkouts of at least the following repositories, pointing at master branch:

- ☑ ensembl-compara
- ☑ ensembl
- ☑ ensembl-hive
- ☑ ensembl-analysis
- ☑ ensj-healthcheck

NCBI taxonomy data

The production team updates the ncbi_taxonomy database on livemirror just before the handover to us (please check that this has been done).

We then need to update the tables on our master DB. The current (rel.75) master database is sf5_ensembl_compara_master on compara1

- ☑ Update the ncbi_taxa_node and ncbi_taxa_name in the master database

  Click here for details

  The ncbi_taxonomy database is located in mysql://ens-livemirror:3306/ncbi_taxonomy

  mysqldump

```
time db_cmd.pl $COMPARA_REG ncbi_taxonomy -executable mysqldump --prepend
--extended-insert --prepend --compress --prepend --delayed-insert
ncbi_taxa_node ncbi_taxa_name | db_cmd.pl $COMPARA_REG compara_master
```

  Times

  rel.64: 45 sec
  rel.65: 47 sec
  rel.66: 47 sec
  rel.67: 30 sec
  rel.68: 30 sec
  rel.69: 35 sec
  rel.70 32 sec
  rel.71 34 sec
  rel.72 36 sec
  rel.74 36 sec
  rel.75 38 sec
  rel.76 44 sec
  rel.78 45 sec

  rel.81 45 sec

- ☑ Check new extant taxon names

  Each new species must have a 'ensembl alias name' tag in the $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/taxonomy/ensembl_aliases.sql file. It should match the "web_name" used by the production team in the "species" table of their "ensembl_production" database on staging1. This may have already been added by Ensembl Production.

  check species/taxa from ensembl_production

```
db_cmd.pl $COMPARA_REG ensembl_production -sql 'select production_name, web_name,
taxon from species'
```

- □ Update the ancestral taxon names

  Each new extant species is anchored to the species tree at a certain taxon. This taxon must be described with two fields in the ncbi_taxa_name table:
   - 'ensembl alias name': a "simple English" description of the taxon.

  The script in scripts/taxonomy/place_species.pl helps in placing the new species in the current compara species tree and discover the new extant species:

  check species/taxa from ensembl_production

  ```
  $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/taxonomy/place_species.pl
  -master_url mysql://ensro@compara1/sf5_ensembl_compara_master -taxon_ids
  9940,9361,7994,7918
  ```

   - 'ensembl timetree mya': the age of the taxon. It can be obtained from the TimeTree database (http://www.timetree.org)

  For any new species, update the file $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/taxonomy/ensembl_aliases.sql to add the two new tags

- ☑ Load ensembl_aliases.sql onto the master database
  Click here for details

  The script will report any discrepancies that need to be resolved ie any nodes which have been deleted from the ncbi_taxonomy database but still have entries in the ensembl_aliases.sql file. Check if these have an entry in the species_set_tag table. If not, it is probably safe to delete them. Check with other compara team members.

  load ensembl_aliases

  ```
  db_cmd.pl $COMPARA_REG compara_master <
  $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/taxonomy/ensembl_aliases.sql
  ```

Compara servers

Check out the current space on the compara servers and delete the last but one release. Leave the previous release for healthchecks. Check with the other compara team members before deleting.

- □ Check space on the admin website

- ☑ Ask compara team members to tidy up any unwanted databases (run the command below for all compara servers) and inform them of the intention to delete the last but one release

  how much space do databases take?

  ```
  perl $ENSEMBL_CVS_ROOT_DIR/ensembl/misc-scripts/db/db-space.pl -host compara4
  -port 3306 -user ensadmin -password $ENSADMIN_PSW
  ```

- ☑ Delete ???_ensembl_compara_xx
- ☑ Delete ???_ensembl_compara_ancestral_xx

Configuration file

- □ Update production_reg_conf.pl and check back into git:

Click for details

Update the registry configuration file $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl that will be used throughout the release process.

Make sure to have edited the release numbers, added external core databases and fixed name prefixes.

The convention right now (since rel.66) is to have the release database in compara3.

Update schema version

□ Update the schema_version in the master database

```
update meta set meta_value = XX where meta_key = 'schema_version';
```

Update table.sql and create patch files

Update the $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/table.sql file and create any patch files.

□ Create a patch file for the schema_version

□ Check if any other patch files need creating by looking at the Declaration of Intentions and checking with other compara team members

□ Update the schema_version in table.sql

□ Delete the previous patch INSERT statements from table.sql

□ Add an INSERT statement for the new schema_version in table.sql and for any other new patches

Check the patch files

The schema defined in the current table.sql must be obtainable by patching the previous database. There is a shell script to do the comparison

Click here for details

Run the script $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/production/schema_diff.sh (you may have to press "y" to validate the patches to apply to the old schema) and check the output. The script relies on several things:

- The API version declared by the Core API (in Bio::EnsEMBL::ApiVersion) has been updated
- The meta keys in the live database are correct (they should !)

The only allowed difference is that peptide_align_feature_XX tables are only found in the previous database, not the new one. The script writes 3 files to the current directory: old_schema.sql, patched_old_schema.sql, and new_schema.sql. Feel free to compare them with sdiff (automatically done by the script):

```
sdiff -w 200 -bs patched_old_schema.sql new_schema.sql
```

git commit table.sql and any patch files

Master database

ⓘ After Handover

Run CheckTaxon healthcheck

☑ Run the CheckTaxon healthcheck early to find any discrepancies between the ncbi_taxon_name table and the core databases (information about how to set up the healthchecks can be found here)

Run healthcheck

```
# make sure you are using the right version of JAVA:
export JAVA_HOME=/software/jdk1.6.0_14

# if you need to recompile (submit to the farm, because you need more memory than is
available on the head) :
bsub -I ant clean jar

# run the healthchecks (submit to the farm, because you need more memory than is
available on the head) :
time bsub -I ./run-configurable-testrunner.sh -h compara1 -d
sf5_ensembl_compara_master --host2 ens-staging2 -t
org.ensembl.healthcheck.testcase.compara.CheckTaxon
```

We can use the compara master database as the source, before the creation of the release database.

(see xxxx on how to set run the healthchecks)

Add new genome_db to compara master database

The current master database (rel.75) is called sf5_ensembl_compara_master on compara1. You have to create new genome_dbs and dnafrags when there is a new assembly or a new species. Any new genome_dbs, dnafrags and method_link_species_set_ids need to be added before production starts.

☑ Add genome_db

Click here for details

This may have already been done if the dna guys started early, please check.

Add genome_db

```
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/update_genome.pl
--reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
--compara compara_master --species "gadus_morhua" --collection ensembl
```

> ⚠ Do not use the --collection option if you are adding a species that is scheduled for a later release. There will be a way of adding the species to the collection during the next release.

Add the new genome_db_id to the confluence page Release plans. This script may take a while if the species you are adding is new. You can check the progress by counting dnaFrag entries in the master database:

```
select count(*) from dnafrag;
```

```
NB: The updated update_genome.pl should do the following automatically. Please check that it does,
and remove the italicised text if ok:
```

If it is a new genebuild and the assembly hasn't changed, you can just edit the entry in master (genome_db table) introducing the new genebuild from meta.genebuild.start_date in the core database.

```
update genome_db set genebuild ="2011-07-FlyBase" where genome_db_id = 105;

Also, check that the first/last_release columns in the genome_db table are correct. The
first_release value should correspond to the current release version (eg. 80) if the genome_db is
new or has an updated assembly. If it is an updated assembly, check that the last_release column
from the previous assembly is set to the previous release date (eg. 79).
```

☑ Add in extra non-reference patches.

Click here for details

This is currently done when a new patch for either human or mouse is released. This may have already been done, please ask.

Details about the patches can be found here ftp://ftp.ncbi.nlm.nih.gov/genbank/genomes/Eukaryotes/vertebrates_mammals/Homo_sapiens/

eg for patch 11: ftp://ftp.ncbi.nlm.nih.gov/genbank/genomes/Eukaryotes/vertebrates_mammals/Homo_sapiens/GRCh37.p11/README

It is first necessary to find if any patches have been deleted or updated since these need to be deleted from the master before the new and replacement patches are added. This is done by running the find_assembly_patches.pl script on the new and previous release of the core database

Find assembly patches

```
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/find_assembly_patches.pl -new_core
"mysql://ensro@ens-staging1:3306/homo_sapiens_core_81_38?group=core&species=homo_sapiens" -prev_core
"mysql://ensro@ens-livemirror:3306/homo_sapiens_core_80_38?group=core&species=homo_sapiens"
```

The output looks like this (format: name, seq_region_id, date)

Sample ouptut

```
NEW patches
  HG29_PATCH 1001061122 2013-02-18 14:16:55
  HG1592_PATCH 1001061114 2013-02-18 14:16:55
  HG385_PATCH 1001061116 2013-02-18 14:16:55
  HSCHR6_2_CTG5 1001061112 2013-02-18 14:16:55
  HG1079_PATCH 1001061118 2013-02-18 14:16:55
CHANGED patches
  HG1436_HG1432_PATCH new=1001061124 2013-02-18 14:16:55
prev=1000859885 2012-10-08 16:48:36
  HG1292_PATCH new=1001061108 2013-02-18 14:16:55       prev=1000759258
2012-04-27 12:12:07
  HSCHR22_1_CTG1 new=1001061120 2013-02-18 14:16:55     prev=1000057052
2010-09-07 14:22:38
  HG1287_PATCH new=1001061110 2013-02-18 14:16:55       prev=1000859831
2012-10-08 16:48:36
DELETED patches
Patches to delete:
("HG1436_HG1432_PATCH","HG1292_PATCH","HSCHR22_1_CTG1","HG1287_PATCH")
```

In this case, there are 5 NEW patches, 4 CHANGED patches and no DELETED patches. Any CHANGED or DELETED patches must be deleted from the master before importing the new patch set.

To add extra non-reference patches to an existing assembly, you need the -force option to just add those dnafrags which aren't already in the database.

Add patches

```
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/update_genome.pl
--reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
--compara compara_master --species human --force
```

The steps for running the pairwise alignment pipeline for new patches can be found here:

`$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/docs/pipelines/READMEs/pair_aligner_patches.txt`

☑ Add in the new LRGs

Click here to expand...

LRGs are needed by the Family pipeline, and have to be updated every release. This is done by running the update_genome.pl script on human with the --force option

```
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/update_genome.pl
--reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
--compara compara_master --species human --force
```

Note that the new LRGs may have already been loaded by the previous step (*add in the human patches*) as the same update_genome.pl command is run

To check if everything loaded OK, compare the output of the following queries:

```
db_cmd.pl --reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
--reg_type core --reg_alias human -sql 'select count(*) from seq_region join
coord_system cs using(coord_system_id) where cs.name="lrg"'

db_cmd.pl --reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
--reg_alias compara_master -sql 'select count(*) from dnafrag where
coord_system_name="lrg"'
```

they should be the same.

Add method_link_species_set entries to compara master database

These are usually added by the people that need them, please check.

The release coordinator (or any team member) should create a new method_link_species_set in the master database before starting a new pipeline in order to get a unique method_link_species_set_id. Ideally they can be created before starting to build the new database although new method_link_species_sets can be added later on.

□ Add dna method_link_species_set entries

Pairwise method_link_species_set

```
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/create_mlss.pl
--method_link_type LASTZ_NET --genome_db_id 90,142 --source "ensembl" --reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
--compara compara_master
```

□ Add synteny method_link_species_set entries

Synteny method_link_species_set

```
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/create_mlss.pl
--method_link_type SYNTENY --genome_db_id 90,142 --source "ensembl" --reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
-compara compara_master
```

□ Add species tree method_link_species_set entry

Synteny method_link_species_set

```
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/create_mlss.pl
--method_link_type SPECIES_TREE --f --collection "ensembl" --name "species tree"
--source "ensembl" --reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
-compara compara_master
```

□ Add homology method_link_species_set_entries

Click here for details

Choose a temp. directory where the output will be generated:

Choose temp directory

```
export MLSS_DIR="/tmp/mlss_creation"
mkdir $MLSS_DIR
```

Run the loading script several times:

--pw stands for all pairwised genome_db_ids in the list provided

--sg stands for keep genome_db_id in the list alone (singleton)

Protein method_link_species_set

```
# orthologues
echo -e "201\n" | perl
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/create_mlss.pl --f \
--reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
\
--pw --collection ensembl 1>$MLSS_DIR/create_mlss.ENSEMBL_ORTHOLOGUES.201.out
2>$MLSS_DIR/create_mlss.ENSEMBL_ORTHOLOGUES.201.err

# paralogues wth
echo -e "202\n" | perl
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/create_mlss.pl --f \
--reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
\
--sg --collection ensembl
1>$MLSS_DIR/create_mlss.ENSEMBL_PARALOGUES.wth.202.out
2>$MLSS_DIR/create_mlss.ENSEMBL_PARALOGUES.wth.202.err

# proteintrees
echo -e "401\n" | perl
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/create_mlss.pl --f \
--reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
\
--name "protein trees" --collection ensembl
1>$MLSS_DIR/create_mlss.PROTEIN_TREES.401.out
2>$MLSS_DIR/create_mlss.PROTEIN_TREES.401.err

# nctrees
echo -e "402\n" | perl
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/create_mlss.pl --f \
--reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
\
--name "nc trees" --collection ensembl
1>$MLSS_DIR/create_mlss.NC_TREES.402.out
2>$MLSS_DIR/create_mlss.NC_TREES.402.err

# families
echo -e "301\n" | perl
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/create_mlss.pl --f \
--reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
\
--name "families" --collection ensembl 1>$MLSS_DIR/create_mlss.FAMILY.301.out
2>$MLSS_DIR/create_mlss.FAMILY.301.err
```

If output/error files are ok, remove them all:

Remove files

```
rm -rf $MLSS_DIR
```

Otherwise make yourself a nice cup of tea and then *PANIC*

Add new species to phylogenetic tree

□ Add new species to phylogenetic tree

Click here for details

The easiest way to use this is to use the phylowidget.
Paste in the content of $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/species_tree.eukaryotes.topology.nw, select Arrow and select where you want the new species to go (use ncbi taxonomy or wikipedia etc) eg Gadus morhua.
Then select in the menu "Tree Edit > Add > Sister". Click on the empty node and edit name (add new name). Leave the branch length as it is.
The tree should appear in the Toolbox: save it into
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/species_tree_blength.nh
git commit

Final checks to compara master database

□ Check if any new species have been postponed
If a new species is postponed for this release, check that no entries (genome_db, dnafrags, etc) were added to the master database. If they were, you can simply switch the assembly_default value in the genome_db table.
[1] for species making it / used in the pipeline
[0] for species not making it / or old assemblies

□ Check that all the new species have been added to the 'ensembl' collection:

Click here for details...

```
select name from species_set_tag join species_set using(species_set_id) join
genome_db using(genome_db_id) where value like 'collection-ensembl' order by
genome_db_id;
```

□ Drop method_link_species_set entries for alignments which did not make it.

Click here for details

Check with other members of compara. Remove redundant entries.

mlss

```
SELECT ml.* FROM method_link ml LEFT JOIN method_link_species_set mlss ON
ml.method_link_id=mlss.method_link_id WHERE mlss.method_link_id IS NULL;
```

Production

Now we can run all our production pipelines

End of production window

Create Release Database

Create the new database for the new release and add it to your registry configuration file. Use the $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/sql/table.sql file to create the tables and populate the database with the relevant primary data and genomic alignments that can be reused from the previous release. This can be done with the $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/populate_new_database.pl script. It requires the master database, the previous released database and the fresh new database with the tables already created. The script will copy relevant data from the master and the old database into the new one.

- ☑ Create new database
  - Click here for details

Create database

```
db_cmd.pl $COMPARA_REG compara_curr -sql "CREATE DATABASE"
db_cmd.pl $COMPARA_REG compara_curr <
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/sql/table.sql
```

- ☑ Populate the new database
  - Click here for details

Before you start copying, make a dry run of the populate_new_database.pl with -intentions flag to review the list of mlss_ids to be copied:

populate_new_database intentions

```
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/populate_new_datab
ase.pl --reg-conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_con
f.pl \
--master compara_master --old compara_prev --new compara_curr
--intentions > populate_new_database.intentions
```

This takes about a minute and produces a long list.

If you believe some of the entries should NOT be copied, you should manually add 'skip_mlss' and 'skip_ss' entries into master database meta table.

NB There are cases where the mlss does not change but the underlying data does eg the "patch-to-ref" alignment (H.sap-H.sap lastz-patch and M.mus-M.mus lastz-patch). These have a mlss_id of 556 (H.sap) and 624 (M.mus) and are currently set in the skip_mlss. If there are no new patches, this needs to be removed to allow the existing data to be copied. If there are new patches, please ensure the 'skip_mlss' is set in the meta table. However, the entry in the method_link_species_set table will not be copied and will need to be added manually.

Start the copying:

populate_new_database

```
time
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/populate_new_datab
ase.pl \
    --reg-conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_con
f.pl --master compara_master --old compara_prev --new compara_curr >
populate_new_database.out
```

Click here for run times

took 3 hours for rel.pre57 (copied from rel.56)
took 3 hours for rel.57 (copied from rel.pre57)
took 2:15 hours for rel.58 (copied from rel.57)
took 2:09 hours for rel.59 (copied from rel.58)
took 3 hours for rel. 60 (copied from rel.59)
rel.64: 2.6h
rel.65: 2.5h
rel.66: 4.8h
rel.67: 2.1h (launched from compara3)
rel.68: 1h40m (run on compara3)
rel.69: 2.5h
rel.70: ~3.5h (compara1 was slow)
rel.71: 4.1h (compara3)
rel.72: 5.1h (compara3)
rel.73: 5.5h (compara2)
rel.74: 2h:3' (compara3)
rel.75: 5.5h (compara5)
rel.77: 9.7h (compara5)
rel.78: 6.0h (compara4)
rel.79:
rel.80:
rel.81: 6h (compara5)

If new method_link_species_sets are added in the master after this, you use this script again to copy the new relevant data. In such case, you will have to:

- skip the old_database in order to avoid trying to copy the dna-dna alignments and syntenies again

- empty ncbi_taxa_name before running

populate_new_database from master only

```
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/populate_new_database.pl \
    --reg-conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl --master compara_master --new compara_curr
```

☑ Delete any pairwise alignments on non-reference patches that have been DELETED or UPDATED.

Click here for details

Find the output of find_assembly_patches.pl script that you ran previously (usually for Human and Mouse) and combine their "Dnafrags to delete" into one common list:

delete patches

```
DNAFRAGS_2_DELETE="(14025314,14025313)"

db_cmd.pl $COMPARA_REG compara_curr -sql "SELECT count(*) FROM genomic_align
ga1, genomic_align ga2, genomic_align_block gab WHERE
ga1.genomic_align_block_id = ga2.genomic_align_block_id AND
ga1.genomic_align_id != ga2.genomic_align_id AND  ga1.genomic_align_block_id =
gab.genomic_align_block_id AND ga1.dnafrag_id in $DNAFRAGS_2_DELETE"

db_cmd.pl $COMPARA_REG compara_curr -sql "DELETE ga1, ga2, gab FROM
genomic_align ga1, genomic_align ga2, genomic_align_block gab WHERE
ga1.genomic_align_block_id = ga2.genomic_align_block_id AND
ga1.genomic_align_id != ga2.genomic_align_id AND  ga1.genomic_align_block_id =
gab.genomic_align_block_id AND ga1.dnafrag_id in $DNAFRAGS_2_DELETE"
```

☑ Run healthchecks on the release database

Click here for details

Run the healthchecks to make sure the the release database is consistent after the initial population of data.

Click here for how to setup and run the healthchecks

Run the compara_external_foreign_keys healthcheck

healthcheck

```
cd $ENSEMBL_CVS_ROOT_DIR/ensj-healthcheck

# make sure you are using the right version of JAVA:
export JAVA_HOME=/software/jdk1.6.0_14

# if you need to recompile (submit to the farm, because you need more memory
than is available on the head) :
bsub -I ant clean jar

# some tests need more memory than the farm3's default:
time bsub -q yesterday -M8000  -R"select[mem>8000]  rusage[mem=8000]" -I
./run-configurable-testrunner.sh -h compara5 -d lg4_ensembl_compara_81 --host2
ens-staging2 -g ComparaShared
```

Click here for run times

rel.81: 13 minutes, 2 expected complaints (CheckSpeciesSetSizeByMethod may complain about Human-on-Human lastz-new and ForeignKeyMasterTables may complain about empty MethodLink entries (to be populated or deleted later in the merging process))

and correct any newly detected problems

Merge DNA data

NOTE: All the runs of copy_data.pl (except the last one) should have the flag "-re_enable 0" to avoid constantly recomputing the indices

☑ Pairwise alignments: LASTZ_NET, BLASTZ_NET, TRANSLATED_BLAT_NET and the special case of LASTZ_PATCH.

NOTE : For merging pw alignments involving haplotypes, go to the next point

Click here for details

These data are usually in separate production databases. You can copy them using the $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/copy_data.pl script. This script requires write access to the production database if the dnafrag_ids need fixing. Use the flag -re_enable 0 on all calls apart from the last one to avoid recomputing the indices.

Example:

copy_data

```
# for each source URL: first plug in the --from_url and add --dry_run to check
that the script has found the right MLSS:
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/copy_data.pl --reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
--to_reg_name compara_curr --method_link_type LASTZ_NET --re_enable 0
--from_url mysql://ensadmin:${ENSADMIN_PSW}@compara4/sf5_ggal_falb_lastz_73
--dry_run

    # if happy, remove the --dry_run flag and run it again, preferably on the
farm:
bsub -q yesterday -R "select[mem>5000] rusage[mem=5000]" -M5000 -I time
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/copy_data.pl --reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
--to_reg_name compara_curr --method_link_type LASTZ_NET --re_enable 0
--from_url mysql://ensadmin:${ENSADMIN_PSW}@compara4/sf5_ggal_falb_lastz_73

    # sometimes you will also need to copy Human_ref_vs_Human_patches (note
the method_link_type is LASTZ_PATCH !)
bsub -q yesterday -R "select[mem>5000] rusage[mem=5000]" -M5000  -I time
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/copy_data.pl --reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
--to_reg_name compara_curr --method_link_type LASTZ_PATCH --re_enable 0
--from_url
mysql://ensadmin:${ENSADMIN_PSW}@compara3/sf5_compara_human_lastz_patch_and_ha
plotype_73
```

☑ Pairwise alignments: non-reference patches for the high coverage LASTZ_NET alignments. This is ito be used when merging pw alignments involving haplotypes.

Click here for details

This step is now very similar to the previous.

Do not forget --merge and --patch_merge options.

Also, if it's the last one you might want to switch keys back on

copy_data --merge --patch_merge

```
# first plug in the --from_url and add --dry_run to check that the script has
found the right MLSS:
bsub -q yesterday -R "select[mem>5000] rusage[mem=5000]" -M5000  -I time
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/copy_data.pl --reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
--to_reg_name compara_curr --method_link_type LASTZ_NET --method_link_type
BLASTZ_NET --method_link_type TRANSLATED_BLAT_NET --re_enable 1 --merge
--patch_merge --from_url  mysql://ensro@compara1/mm14_lastz_human --dry_run

    # if happy, remove the --dry_run flag and run it again, preferably on the
farm:
bsub -q yesterday -R "select[mem>5000] rusage[mem=5000]" -M5000  -I time
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/copy_data.pl --reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
--to_reg_name compara_curr --method_link_type LASTZ_NET --method_link_type
BLASTZ_NET --method_link_type TRANSLATED_BLAT_NET --re_enable 1 --merge
--patch_merge --from_url  mysql://ensro@compara1/mm14_lastz_human
```

☑ Multiple alignments: PECAN, EPO, EPO_LOW_COVERAGE, GERP_CONSTRAINED_ELEMENT, GERP_CONSERVATION_SCORE

⌄ Click here for details

These data are usually in separate production databases. You can copy them using the $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/copy_data.pl script. This script requires write access to the production database if the dnafrag_ids need fixing or the data must be copied in binary mode (this is required for conservation scores).

Some alignments produce conservation scores and constrained elements (check the Release plans) and these need to be copied separately.

eg

copy_data multiple alignment

```
bsub -q yesterday -R "select[mem>5000] rusage[mem=5000]" -M5000 \
    -I time
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/copy_data.pl \
    --reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
--to_reg_name compara_curr \
    --method_link_type EPO --method_link_type EPO_LOW_COVERAGE
--method_link_type PECAN \
    --method_link_type GERP_CONSTRAINED_ELEMENT --method_link_type
GERP_CONSERVATION_SCORE \
    --from_url
mysql://ensadmin:${ENSADMIN_PSW}@compara2/sf5_epo_low_8way_fish_71 -re_enable
0
```

EPO alignments produce ancestral sequences and a separate core database which must also be copied. See below.

⌄ Click here for run times

rel 71.
2m: kb3_hsap_ggal_lastz_71 mlss_id=632
1m kb3_mmus_ggal_lastz_71 mlss_id=633
1m kb3_ggal_mgap_lastz_71 mlss_id=634
1m kb3_ggal_xtro_tblat_71 mlss_id=638

```
1m  kb3_hsap_ggal_tblat_71  mlss_id=637
1m  sf5_olat_gmor_lastz_71  mlss_id=625
3m  kb3_pecan_20way_71      mlss_id=630
4m  kb3_pecan_20way_71      mlss_id=631
35m  kb3_pecan_20way_71      mlss_id=50045
3m  sf5_compara_epo_6way_71 mlss_id=548
1m  sf5_olat_onil_lastz_71  mlss_id=626
1m  sf5_olat_xmac_lastz_71  mlss_id=627
1m  sf5_epo_low_8way_fish_71 mlss_id=628
2m  sf5_epo_low_8way_fish_71 mlss_id=629
9m  sf5_epo_low_8way_fish_71 mlss_id=50044
1m  kb3_ggal_drer_tblat_71   mlss_id=639
1m  kb3_ggal_csav_tblat_71   mlss_id=640
1m  sf5_ggal_acar_lastz_71   mlss_id=636
91m  sf5_ggal_tgut_lastz_7    mlss_id=635 (re-enable 1)
93m  sf5_compara_epo_3way_birds_71 mlss_id=641 (re-enable 1)
14m  sf5_compara_epo_3way_birds_71 mlss_id=642 (re-enable 1)
16m  sf5_compara_epo_3way_birds_71 mlss_id=50046 (re-enable 1)
```

□ Check the keys have been re-enabled

Click here for details

Use mysqlshow to highlight if the table still has disabled keys. The text "disabled" will be shown in the Comment column if the key is disabled. An empty Comment column indicates the keys are enabled.

mysqlshow interprets any underscores in the last argument as a wildcard so to get round this, we need to use % as the last argument.

mysqlshow

```
db_cmd.pl $COMPARA_REG compara_curr --executable mysqlshow --append --keys
--append genomic_align_block %
db_cmd.pl $COMPARA_REG compara_curr --executable mysqlshow --append --keys
--append genomic_align %
db_cmd.pl $COMPARA_REG compara_curr --executable mysqlshow --append --keys
--append genomic_align_tree %
db_cmd.pl $COMPARA_REG compara_curr --executable mysqlshow --append --keys
--append conservation_score %
db_cmd.pl $COMPARA_REG compara_curr --executable mysqlshow --append --keys
--append constrained_element %
```

If there are still tables with keys disabled run the following on them:

```
db_cmd.pl $COMPARA_REG compara_curr -sql "ALTER TABLE <table_name> ENABLE
KEYS";
```

☑ Syntenies

Click here for details

First make sure the entries in $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl file point at the latest (staging) versions of the core databases.

Load the synteny data by running the $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/synteny/LoadSytenyData.pl script. This requires a synteny file. The location of this should be on the Release plans.

The "-ref" and "-nonref" species should be taken from the name of the method_link_species_set that corresponds to the pairwise alignments containing both species

Example

load synteny data

```
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/synteny/LoadSyntenyData.pl
--reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
\
    --dbname compara_curr -ref "Homo sapiens" -nonref "Callithrix jacchus"
-mlss_id 10052 \

/lustre/scratch101/ensembl/kb3/scratch/hive/release_64/kb3_hsap_cjac_synteny_6
4/synteny/all.100000.100000.BuildSynteny
```

☑ Build a new ancestral sequence core database

Click here for details

Putting together the database of ancestral sequence is now done using a dedicated Hive-Core mini-pipeline.

Check you have the most recent core checkout ie the correct schema and patch files are added to the meta table.

Go to $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/modules/Bio/EnsEMBL/Compara/PipeConfig and open the PipeConfig file AncestralMerge_conf.pm .

Make sure you have edited/checked the following:

1) current release number

2) names and locations of current and previous ancestral core databases

3) the table of ancestral sequence sources in the second analysis (some entries might point to the previous release ancestral database, some will be new)

For (3), you can run the following query on your release database and on the previous database: (NB: method_link_id=13 is equivalent to method_link_type = "EPO")

EPO query

```
SELECT * FROM method_link_species_set WHERE method_link_id = 13;
```

The new mlss_id should be attached to their production database:
'641' => 'mysql://ensadmin:$ENSADMIN_PSW@compara3/sf5_3birds_ancestral_sequences_core_71'
The mlss_id that are reused should be linked to the previous database
'505' => $self->o('prev_ancestral_db'),

The current (as or rel.75) list of ancestral alignments are:

5 teleost fish
6 primates
4 sauropsids ("birds")
15 eutherian mammals

Save the changes, exit the editor and run init_pipeline.pl with this file:

init_pipeline

```
init_pipeline.pl AncestralMerge_conf.pm -host compara5
```

Then run both -sync and -loop variations of the beekeeper.pl command suggested by init_pipeline.pl . This pipeline will merge the separate ancestral core sources into ensembl_ancestral_{rel_number}.

You may want to check the msg table for errors and have a look at the result of the merger:

Which Ancestral sequences do we have?

```
SELECT left(name,12) na, count(*), min(seq_region_id), max(seq_region_id),
max(seq_region_id)-min(seq_region_id)+1 FROM seq_region GROUP BY na;
```

If everything is ok, measure the time:

how much time did running of the pipeline take?

```
call time_analysis('%')
```

Click here for run times

rel.67: 20min
rel.71: 20min
rel.75: 21min

Then drop hive-specific tables:

drop hive tables

```
CALL drop_hive_tables;
```

Make sure all tables are myISAM.

```
 SHOW TABLE STATUS where engine != 'MyISAM';
```

or, if no new multiple alignments were run, copy it over from the previous release

Click here for details

Create a new database for ancestral sequences:

```
 db_cmd.pl $COMPARA_REG ancestral_curr -reg_type core -sql 'CREATE DATABASE'
```

Copy over the data from the previous release:

```
 time db_cmd.pl $COMPARA_REG ancestral_prev -reg_type core -executable
mysqldump | db_cmd.pl $COMPARA_REG ancestral_curr -reg_type core
# took 45 minutes in rel.81
```

Patch the database to the current release by applying the relevant patches from $ENSEMBL_CVS_ROOT_DIR/ensembl/sql or use a schema patcher script.

□ Update all the stats (because they rely on the gene-set which may have changed)

Merge GeneTrees+Families+NCTrees+PatchProjectionsAsHomologies

☑ Check that Protein-trees have been run and handed-over ("*Compara hands over Homologies*" date)

☑ Check that ncRNA-trees have been run

☑ Check that Families have been run

☑ Check that LRGs were included in the Families

☑ Once Production has updated all the xrefs (incl. gene names and descriptions), the pipeline that merges the gene side (*ImportAltAlleGroupsAsHomologies_conf*) can be run.
NB: make sure that all the source databases' URLs are correctly registered in the master_db's MLSS table (if the pipelines have moved in between, make sure to edit the relevant MLSS.url columns).

Click here for times

rel.81 : 1h03m

☑ Run the Hive pipeline (*MergeDBsIntoRelease_conf*) to merge tables from all the four products into the release database

Click here for details

Go to $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/modules/Bio/EnsEMBL/Compara/PipeConfig and open the PipeConfig file MergeDBsIntoRelease_conf.pm

It has a 'urls' hash where you will have to change the names of the databases and possibly their locations:

master_db - is the main compara master
prev_rel_db - should point to the previous release database
curr_rel_db - should point to the current release database being merged into (not the Hive pipeline database, but purely Compara schema product)

protein_db - should point to the current GeneTrees pipeline database
family_db - should point to the current Families pipeline database
ncrna_db - should point to the current ncRNAtrees pipeline database
projection_db - should point to the current PatchProjectionsAsHomologies pipeline database.

Also choose the server to run the merging pipeline on (you don't need a lot of resources or memory, as it is purely Hive book-keeping) and set the 'host' default_option.

Save the changes, exit the editor, init and run the merging pipeline with this file:

running the merging pipeline

```
init_pipeline.pl MergeDBsIntoRelease_conf.pm
beekeeper.pl ... -sync
beekeeper.pl ... -loop
```

This pipeline will merge all the protein-side products into the release database.

Click here for times

rel.73 was the first experimental run, code had to be fixed, servers had to be reconfigured, so merging took one whole working day.

In the merging database run: call time_analysis('%');

rel.75 : 5 hours
rel.76 : 5.6 hours

□ Drop the homology_projections database

Click here for details

After you are happy about the result of protein side merging you can drop the "YourName_homology_projections_ThisRelease" database.

□ git commit the changes to the PipeConfig files that you have made.

Final database checks

☑ Remove redundant method_link entries

⌄ Click here for details

In most cases they can be removed, but check with other members of Compara. Remove redundant method_link entries

method_link entries

```
SELECT ml.* FROM method_link ml LEFT JOIN method_link_species_set mlss
USING(method_link_id) WHERE mlss.method_link_id IS NULL;
DELETE ml FROM method_link ml LEFT JOIN method_link_species_set mlss
USING(method_link_id) WHERE mlss.method_link_id IS NULL;
```

☑ Check that all the schema patches have been declared and applied.

⌄ Click here for details

If unsure, recheck the current schema against the previous schema. See Check the patch files for details

☑ Load the species-trees (needed for the Species-tree view)

```
# Create a MLSS ID (FIXME: this must be done on the master database and the
release database, but they must use the same mlss_id)
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/create_mlss.pl
--method_link_type SPECIES_TREE --f --collection "ensembl" --name "species tree"
--source "ensembl" --reg_conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
-compara compara_master
# The Ensembl-curated tree (replace 60002 with the mlss_id !)
$ time standaloneJob.pl Bio::EnsEMBL::Compara::RunnableDB::MakeSpeciesTree
-reg_conf
/nfs/users/nfs_l/lg4/work/ensembl-compara/scripts/pipeline/production_reg_conf.pl
-compara_db compara_curr -label "Ensembl" -species_tree_input_file
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/species_tree.ensembl.topology.nw -mlss_id 60002
# The NCBI tree (automatically populated) (replace 60002 with the mlss_id !)
$ time standaloneJob.pl Bio::EnsEMBL::Compara::RunnableDB::MakeSpeciesTree
-reg_conf
/nfs/users/nfs_l/lg4/work/ensembl-compara/scripts/pipeline/production_reg_conf.pl
-compara_db compara_curr -label "NCBI Taxonomy" -mlss_id 60002
```

```
# Filter the tree by clade

db_cmd.pl $COMPARA_REG compara_curr -sql "INSERT INTO method_link_species_set_tag
VALUES
(60002,'filter:Amniota','Amniotes'),(60002,'filter:Mammalia','Mammals'),(60002,'filter:Neopterygii','Fish'),(60002,'filter:Sauria','Sauropsids')"
```

Run the healthchecks

☑ Update the code

⌄ Click here for details

The healthchecks are written in java and need to be recompiled after a git pull.

compile healthchecks

```
cd $ENSEMBL_CVS_ROOT_DIR/ensj-healthcheck
export JAVA_HOME=/software/jdk1.6.0_14
git pull
bsub -I ant clean jar
```

We don't need to configure a database.properties any more. Everything is done from the command line

☑ Run the healthchecks for ancestral database

⌄ Click here for details

```
time bsub -M8000  -R"select[mem>8000]  rusage[mem=8000]" -I
./run-configurable-testrunner.sh -h compara5 -d sf5_ensembl_ancestral_77 -g
ComparaAncestral
```

It should take less than a minute (if the tables are analyzed / optimized) and usually complains about 1 thing that you can ignore:

```
org.ensembl.healthcheck.testcase.generic.AssemblySeqregion [Team
responsible: GENEBUILD]
  mm14_ensembl_ancestral_80: 0 rows found in assembly table
```

☑ Update the max_alignment_length IF NECESSARY.

⌄ Click here for details

Check that the max_alignment_lengths have been computed.

update max_alignment_length

```
time bsub -I ./run-configurable-testrunner.sh -h compara5 -d
sf5_ensembl_compara_77 -t
org.ensembl.healthcheck.testcase.compara.MLSSTagMaxAlign
```

If not (the healthcheck is failing), you can repair it by adding the --repair flag:

update max_alignment_length

```
time bsub -I ./run-configurable-testrunner.sh -h compara5 -d
sf5_ensembl_compara_77 -t
org.ensembl.healthcheck.testcase.compara.MLSSTagMaxAlign --repair 1 --user
ensadmin --password $ENSADMIN_PSW
```

☑ Update the alignment mlss_id of the conservation score IF NECESSARY

⌄ Click here for details

update conservation score mlss_id

```
time bsub -I ./run-configurable-testrunner.sh -h compara5 -d
sf5_ensembl_compara_77 -t
org.ensembl.healthcheck.testcase.compara.MLSSTagGERPMSA
```

If the healthcheck is failing, you can repair it by adding the --repair flag:

update conservation score mlss_id

```
time bsub -I ./run-configurable-testrunner.sh -h compara5 -d
sf5_ensembl_compara_77 -t
org.ensembl.healthcheck.testcase.compara.MLSSTagGERPMSA --repair 1 --user
ensadmin --password $ENSADMIN_PSW
```

☑ Run the ComparaAll group of healthchecks on the release database

Click here for details

The 'stdbuf -o0' is a trick to prevent the pipe from buffering the output, since in addition to storing it we also want to examine the output visually.

compara_external_foreign_keys

```
time bsub -q yesterday -M8000  -R"select[mem>8000]  rusage[mem=8000]" -I
stdbuf -o0 ./run-configurable-testrunner.sh -h compara5 -d
lg4_ensembl_compara_81 --host2 ens-staging2 -g ComparaAll | tee
healthchecks_after_merge.txt
```

Click here for previous results

rel.81 --

Test web server

☐ Ask ensembl-production to point the test web server to the compara release database
Upon confirmation from the release coordinator ask other members of Compara to check their data on:
http://staging.ensembl.org/

Final handover of databases

Analyze / Optimize one last time

☑ This is required for the CopyDbOverServer script to work properly.

Click here to expand...

1. Run ANALYZE_TABLE on compara and ancestral databases

   Click here for details

analyze table

```
time db_cmd.pl $COMPARA_REG compara_curr   -executable mysqlanalyze --append --verbose
time db_cmd.pl $COMPARA_REG ancestral_curr -reg_type core -executable mysqlanalyze --append --verbose
```

2. Run OPTIMIZE_TABLE on compara and ancestral databases
   Click here for details

optimize table

```
time db_cmd.pl $COMPARA_REG compara_curr   -executable mysqloptimize --append --verbose
time db_cmd.pl $COMPARA_REG ancestral_curr -reg_type core -executable mysqloptimize --append --verbose
```

Both operations will take between a few seconds and a few hours, depending on the state of the tables / indexes

Copy databases to staging servers

□ Logon to ens-staging1 and import the databases
Click here for details

The import script MUST be run on the DESTINATION machine with the *mysqlens* user. NB: ask for the password for *mysqlens* well in advance - there may be no-one around you at the right moment!
You should also check whether there is enough space (~270Gb) on the disk before starting the copy.

The script assumes that your environment is set ($ENSEMBL_CVS_ROOT_DIR and $ENSADMIN_PSW)

```
ssh ens-staging1
df -h /mysql
su -m -c $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/production/import_compara_release_databases.pl mysqlens
```

It should take 1h30-2h to run

□ Logon to ens-staging2 and import the databases (same procedure)

Final bits

☑ Send an email to ensembl-production to announce the handover the databases and cc ensembl-compara@

☑ Update the Declaration of Intentions on the admin website to indicate what has been handed over and what didn't make it and has been postponed

□ Dump the master database and place the copy in a safe place
Click here to expand...

It should take a couple of minutes at most to run:

dump master database

```
db_cmd.pl $COMPARA_REG compara_master --prepend -t --executable mysqldump | gzip - > /warehouse/ensembl01/compara/master_dumps/ensembl_compara_master_${RELEASE_VERSION}.mysql.gz
```

Post-handover

Update documentation and diagrams

ⓘ It is now time to update the static files. This should be done before we branch the code

☑ Update the pipeline diagrams for all the pipelines that have been run this release

Click here for details

Go to the docs directory

pipeline diagrams

```
cd $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/docs/pipelines/diagrams
generate_graph.pl $COMPARA_REG compara_ptrees -output ProteinTrees.png
generate_graph.pl $COMPARA_REG compara_nctrees -output ncRNAtrees.png
generate_graph.pl $COMPARA_REG compara_families -output Families.png
generate_graph.pl -url mysql://ensro@compara3/kb3_pecan_20way_71 -output MercatorPecan.png
generate_graph.pl -url mysql://ensro@compara4/sf5_epo_35way_68 -output EpoLowCoverage.png
generate_graph.pl -url mysql://ensro@compara4/sf5_compara_epo_13way_69 -output epo_pt3.png
```

Commit any changed diagrams to git and push.

☑ Update the schema documentation and diagrams

Click here for details

generate new schema documentation

```
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-production/scripts/sql2html.pl -i $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/sql/table.sql -o $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/docs/schema/compara_schema.html -d Compara -host compara5 -user ensro -dbname lg4_ensembl_compara_75 -sort_headers 0 -sort_tables 0 -intro $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/docs/schema/intro.html
```

Open the output file $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/docs/schema/compara_schema.html in your browser and check that no example errors are reported.

If everything looks fine, copy this file to public-plugins and commit&push both (the compara one and the webcode one) :

update schema documentation for web

```
cp $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/docs/schema/compara_schema.html
$ENSEMBL_CVS_ROOT_DIR/public-plugins/docs/htdocs/info/docs/api/compara/
```

If necessary, update schema diagrams by loading the $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/sql/table.sql schema file into MySQL Workbench, rearrange/colour the nodes and export into PNG.

The schema diagrams will have to be copied both to $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/docs/schema/diagrams and to public-plugins and committed&pushed in both repositories:

update schema diagrams for web

```
cp -r $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/docs/schema/diagrams/*.png
$ENSEMBL_CVS_ROOT_DIR/public-plugins/docs/htdocs/info/docs/api/compara/diagrams/
```

☑ Update the tutorial documentation

Click here for details

Update the tutorial documentation compara_tutorial.html in this directory:

> $ENSEMBL_CVS_ROOT_DIR/ensembl-webcode/htdocs/info/docs/api/compara/

Be careful that the embedded Perl snippets must use HTML-escaped characters (e.g. &lt; and &gt;) and be wrapped in a <pre class="code sh_perl">

Open the URL /info/docs/api/compara/compara_tutorial.html from a sandbox / test website and export it as a PDF in

> $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/docs/ComparaTutorial.pdf

To make the pdf look nicer, you can issue a few JavaScript commands to remove the Ensembl headers. See Creating PDF version of VEP docs for more details

☑ Update main ensembl species tree if there are any new species

See $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/docs/species_tree/README

The end result should go here (check how it should be moved) : http://www.ensembl.org/info/about/species_tree.pdf

☐ Check examples work in ensembl-compara/scripts/examples/

☑ git commit and push any modified files or added tutorial examples

☐ Update the declared intention with removed / deprecated methods

Click for details

We need to generate the list of methods exported by the objects / adaptors on the master and release/{$n-1} branches, and compare (diff) them.

Check deprecated / removed methods

```
# on master/ensembl-compara/modules/Bio/EnsEMBL/Compara/
grep "^sub " *pm | sort > ~/MASTER
# on the previous release/ensembl-compara/modules/Bio/EnsEMBL/Compara/
grep "^sub " *pm | sort > ~/RELEASE75
sdiff -w 200 -bs ~/RELEASE75 ~/MASTER | less

# Let's do the same for the adaptors
# on master/ensembl-compara/modules/Bio/EnsEMBL/Compara/
grep "^sub " DBSQL/*pm | sort > ~/MASTER
# on the previous release/ensembl-compara/modules/Bio/EnsEMBL/Compara/
grep "^sub " DBSQL/*pm | sort > ~/RELEASE75
sdiff -w 200 -bs ~/RELEASE75 ~/MASTER | less
```

In both cases, make sure the methods are really removed, and not moved to a base / sub class, etc

Branch the code

- ☑ Check with the rest of Compara that it is ok to branch the code as it is, then create the 'release/THIS_RELEASE_NUMBER' branch in git and push it to the server.

Data dumps

These should only be done once ensembl-production has given the go-ahead for this. This is to avoid overloading the databases whilst biomart is being run.

Most dumps (except homology) are currently generated in /lustre/scratchXXX areas and then have to be manually assembled into the /nfs/ensembl/ensembl/ftp_ensembl/release-XX/ tree.
Look at the previous release tree to get the idea. The first level of directories normally defines the file type, and the second level is the team name (except fasta/ where species are mixed).

Ensembl-compara is responsible for the following dumps:

bed/ensembl-compara (MSA)
emf/ensembl-compara (MSA and homologies)
maf/ensembl-compara (multiple_alignments and pairwise_alignments)
xml/ensembl-compara (homologies)
fasta/ancestral_alleles (the only one without ensembl-compara in the path)

Make sure they are all either generated by running new dumps, or if some MSAs did not run in this release - sym-linked from the previous release's /warehouse/ens_ftp_arch_03/release-Xx/ tree.

DO NOT LINK OR COPY FROM PREVIOUS RELEASE's /lustre/scratch areas, as they may no longer point to real data!!!
If you do not have enough permissions for reading from /warehouse/ens_ftp_arch_03/release-Xx/ talk to the web-team and ask for it to be fixed.

- ☑ DNA data dumps
  Generally the person who ran the pipeline will also do the data dumps. The instructions are in $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/docs/pipelines/READMEs/multi_align.dumps.txt

- ☑ Gene tree dumps
  Generally the person who ran the pipeline will also do the data dumps.

  Click here for details

  Go to $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/modules/Bio/EnsEMBL/Compara/PipeConfig and open the PipeConfig file

DumpTrees_conf.pm

Check that you are happy about all parameters. In usual cases, they can all be set from the command line and the config file does not need editing.

Make sure you have the XML::Writer module in your PERL5LIB (there is a copy in ~mm14/src/perl/orthoxml/)

Run init_pipeline.pl with this file:

init_pipelline

```
init_pipeline.pl DumpTrees_conf.pm -host compara5 -member_type protein
```

> rel.64: testing sqlite mode failed: too many occurrences of "database locked". We should stick to mysql.

Then run both -sync and -loop variations of the beekeeper.pl command suggested by init_pipeline.pl .

This pipeline takes about 3 hours and will produce protein_tree dumps in the directory pointed at by 'target_dir' parameter. ('/lustre/scratch110/ensembl/'.$self->o('ENV', 'USER').'/'.$self->o('pipeline_name'))

It also automatically copies the file to /nfs/ensembl/ensembl/ftp_ensembl/release-XX/, so just check that the files are there with correct sizes, and you're done.

Create the ncRNA pipeline from the same config file:

init_pipeline

```
init_pipeline.pl DumpTrees_conf.pm -host compara5 -member_type ncrna
```

Then run both -sync and -loop variations of the beekeeper.pl command suggested by init_pipeline.pl .

This pipeline will produce ncrna_tree dumps in the directory pointed at by 'target_dir' parameter and copy them to /nfs/ensembl/ensembl/ftp_ensembl/release-XX/. This is much faster: less than an hour

Commit the DumpTrees_conf.pm file into git if you'd like to keep the changes.

☑ Copy the tree content dump for Uniprot

Click here for details

The file 'target_dir'/ensembl.GeneTree_content.{release}.txt.gz needs to be copied to the EBI ftp server, and then MD5 checksum computed and stored next to it:

init_pipeline

```
scp
/lustre/scratch110/ensembl/lg4/protein_81_dumps/ensembl.GeneTree_content.e81.t
xt.gz \
login.ebi.ac.uk:/nfs/ftp/pub/databases/ensembl/ensembl_compara/gene_trees_for_
uniprot/ssh login.ebi.ac.ukcd
/nfs/ftp/pub/databases/ensembl/ensembl_compara/gene_trees_for_uniprot/md5sum
ensembl.GeneTree_content.e81.txt.gz >
ensembl.GeneTree_content.e81.txt.gz.MD5SUM
```

☑ Ancestral alleles (for the Variation team)

Click here to expand...

```
perl
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/ancestral_sequences/get_ancestra
l_sequence.pl --conf
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/production_reg_conf.pl
--compara_url mysql://ensro@compara5/sf5_epo_8primates_77 --species
homo_sapiens
dirname=homo_sapiens*
cd $dirname
perl
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/ancestral_sequences/get_stats.pl
> summary.txt
cd ..
tar -jcf ${dirname}.tar.bz2 $dirname
md5sum ${dirname}.tar.bz2 > ${dirname}.tar.bz2.MD5SUM
```

☑ Let the production team know that the dumps are ready in their common /nfs/ensembl/ensembl/ftp_ensembl/release-XX/ location.

Final things

□ Create a word document and a pdf dump of this document

Click here for details

In the top-right menu of this Confluence page, choose "Tools -> Export to PDF" and "Tools -> Export to Word".

Put these files into $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/docs/

git commit and push

□ Web-site testing on staging and finally on the test server when it is ready. Ask the compara team to test as much as possible to catch problems as early as possible.
http://staging.ensembl.org/
http://test.ensembl.org/