Release coordination documentation

Preparation

Preparing this document itself

- [x] 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

Intentions for the upcoming releases are discussed at the Compara group meetings, and then approved at the Ensembl Operations meetings. They can be consulted online at (https://www.ebi.ac.uk/seqdb/confluence/display/EnsCom/Intentions+for+release+XX)

- [x]

- [x] At the beginning the release cycle, the release coordinator sets up a web page with intentions in the Confluence wiki system to allow easy tracking of the progress. Release plans > Click on the three dots at the tip next to the "Create" button , and select the *Intentions for Release* template.
  The JIRA board for the release can be found by following the link to the correct release number on https://www.ebi.ac.uk/panda/jira/projects/ENSCOMPARASW?selectedItem=com.atlassian.jira.jira-projects-plugin:release-page
  Generate the image of the dependency graph with this command:

```
/usr/bin/dot -Tpng < $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts
/pipeline/compara_ensembl.dot > $ENSEMBL_CVS_ROOT_DIR/ensembl-compara
/scripts/pipeline/compara_ensembl.png
```

  Do not commit the .png file. If the dependencies have changed, update and commit th .dot file, and upload a new version of the image to the Intentions page

Update your checkouts

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

- [x] ensembl-compara
- [x] ensembl
- [x] ensembl-hive
- [x] ensembl-analysis
- [x] ensj-healthcheck
- [x] ensembl-taxonomy
- [x] public-plugins
- [x] ensembl-test

Environment variables

- [x] Define `$ENSEMBL_CVS_ROOT_DIR`
  This is necessary to run the Hive and is used by many scripts/files in this document. Make sure this is defined in your terminal
- [x] Define `$ENSADMIN_PSW`
  The password for the mysql 'ensadmin' user also needed for many scripts
- [x] 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_ebi_conf.pl -reg_type compara -
reg_alias"
```

- [x] Define `$CURR_ENSEMBL_RELEASE` variable to simplify naming of databases

```
export CURR_ENSEMBL_RELEASE=`perl -mBio::EnsEMBL::ApiVersion -e
'print Bio::EnsEMBL::ApiVersion::software_version()."\n"'`

        # make sure that you got the right value (your ensembl
checkout has to be up-to-date)
echo $CURR_ENSEMBL_RELEASE

        # it is also very handy to have the previous release number
in a variable:
export PREV_ENSEMBL_RELEASE=`expr $CURR_ENSEMBL_RELEASE - 1`
```

- [x] source the RH7 env

```
source /nfs/software/ensembl/latest/envs/basic.sh
```

- [x] Make sure you are using the production queue (as everyone in the team should !)

```
export LSB_DEFAULTQUEUE=production-rh7
```

- [x] Backup the master database

```
mysql-ens-compara-prod-1 mysqldump ensembl_compara_master > /nfs
/production/panda/ensembl/warehouse/compara/master_db_dumps
/ensembl_compara_master.$(date '+%Y%m%d').pre${CURR_ENSEMBL_RELEASE}.
sql
```

Preparing JIRA tickets

As of release 89, JIRA ticket creation for the release process is performed automatically using a script.

- [x] The script assumes that the relco is running it, but you can use the `-relco` option to override the user name.

Automatically create JIRA tickets

```
cd $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/jira_tickets
perl create_compara_release_JIRA_tickets.pl
```

The script will print the paths of the configuration files it's read, and the release number it's targeting (taken from the Core API), and then ask for your JIRA password

> Only run this script once the list of tickets has been updated based on last release's run ! (look for a ticket named "*Update the Jira_recurrent_tickets.txt*")

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.

- [x] Check space on http://www.ebi.ac.uk/~muffato/disk/compara.html
  You can click on "Show per-database disk usage" to see the list of databases and their size

  Click here for the guidelines for the deletion of databases...

  - Research databases should be fully backed up
  - Release databases: keep only the last one (for HCs)
  - Production databases: keep the last production run of each pipeline on each species-set, e.g. keep the primate EPO **and** the mammal EPO **and** etc
    Before dropping any of the databases, back up the eHive tables

```
standaloneJob.pl Bio::EnsEMBL::Hive::RunnableDB::
DatabaseDumper -exclude_list 1 -db_conn $URL -debug 9 -
output_file path/to/file.sql.gz
```

Configuration file

- [x] Update `production_reg_ebi_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_ebi_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 is to keep the merged release database on mysql-ens-compara-prod-1.
  - DB connection details of production databases (families, nctrees, etc.) can be removed from the file until merge.

Schema preparation

> All the schema changes must be ready by the handover of the core databases. See SOP for API / schema changes for the procedure about changing the schema

Update table.sql and create patch files

Here we need to prepare table.sql for the new schema and create the relevant patch files. The general procedure is defined in SOP for API / schema changes.

The other Compara members may have already created patches, so check with them what's there. At the minimum, there should be the schema version increase.

Click here for details

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

- [x] Create a patch file for the schema_version
- [x] Update the schema_version in table.sql and delete the other patch INSERT statements
- [x] Check if any other patch files need creating by looking at the Declaration of Intentions and checking with other compara team members
- [x] Add an INSERT statement for the new schema_version in table.sql and for any other new patches
- [x] Check that all the INSERT statements in table.sql are correct (version numbers and letters) and match the patches

Check the patch files

- [x] 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, and automatically runs sdiff. If you need to look at the diff later on, run this:

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

  git commit table.sql and any patch files

Patch the reused databases to the latest schema

- [x] Use the following script to detect the what schema the database is on and to apply all the required patches to bring it to the latest schema (double-check the database names !)

  ```
  $ENSEMBL_CVS_ROOT_DIR/ensembl/misc-scripts/schema_patcher.pl $(mysql-ens-compara-prod-1-ensadmin details script) --database=ensembl_compara_${PREV_ENSEMBL_RELEASE}
  $ENSEMBL_CVS_ROOT_DIR/ensembl/misc-scripts/schema_patcher.pl $(mysql-ens-compara-prod-1-ensadmin details script) --database=ensembl_ancestral_${PREV_ENSEMBL_RELEASE}
  ```

Patch the test databases

- [x] Be sure you are on a fresh master and follow the instructions from Patching test databases
  Check that all the databases have been patched with

  ```
  grep "schema_version." modules/t/test-genome-DBs/*/*/meta.txt
  ```

  Rerun the entire test-suite to check that it still passes (fix it otherwise !)

  ```
  prove -rv modules/t/
  ```

  Commit and push all the changes under modules/t/test-genome-DBs/

Branch the code

- [x] 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.

```
cd $ENSEMBL_CVS_ROOT_DIR/ensembl-compara
git pull master
git branch release/$CURR_ENSEMBL_RELEASE
git checkout release/$CURR_ENSEMBL_RELEASE
git push origin release/$CURR_ENSEMBL_RELEASE
```

Specialize the branch

- [x] The name of the branch ("master") is hardcoded in a number of places. Because the release branch is meant to be coupled to the other release branches, replace "master" with "release/<release number>" in these places
    - README.md
    - .travis.yml . For all the repos that are eventually branched (ensembl-XXX, except ensembl-hive, and the external dependencies) we want to switch over to release/XX, but since they may not yet be branched, we also need to add a fallback to master, e.g.

```
# replace 'git clone --branch master --depth 1 https://github.com
/Ensembl/ensembl-rest.git' with:
git clone --branch release/89 --depth 1 https://github.com
/Ensembl/ensembl-rest.git || git clone --branch master --depth 1
https://github.com/Ensembl/ensembl-rest.git
```

    Take a look at .travis.yml on the release/89 branch for an example

    However on master, we want to stick to the master branch of all dependencies, so here are the commands to update the files on the release/XX branch only without affecting master

```
cd $ENSEMBL_CVS_ROOT_DIR/ensembl-compara
git checkout release/$CURR_ENSEMBL_RELEASE
(edit and commit the files as explained above)
git push origin release/$CURR_ENSEMBL_RELEASE
git checkout master
git merge -s ours release/$CURR_ENSEMBL_RELEASE
git diff origin/master..master  # should be empty
git push origin master
```

    Pay attention to the merge command. The "-s ours" option tells git to make a merge commit that doesn't change any files (git diff will confirm that). The merge is still important so that bugfixes put on the release branch can be naturally merged with a standard git merge.

- [x] Patch the previous production and ancestral databases using the above script
- [x] Patch the master database (run the commands contained in the patch files individually, given that some patches may fail due to the master having an incomplete set of tables)

> From this point, all the production should happen on the release branch, with bugfixes pushed there. We shouldn't worry about master until the end of production

- [x] Ask Matthieu or a git paladin (#git channel on Slack) to *protect* the new release branch. https://github.com/Ensembl/ensembl-compara/settings/branches
  Only tick "Protect this branch", none of the other tickboxes
- [x] Ask Matthieu or a git paladin (#git channel on Slack) to add a daily run of the branch on Travis CI https://travis-ci.org/Ensembl/ensembl-compara/settings (choose "Always Run").
  Disable the daily runs on branches that are not live any more (usually release n-2)

Master database

> The Master database can be updated at any time but must be ready before the main pipelines start. For more information about its role, consult Master database.

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.90) master database is ensembl_compara_master on mysql-ens-compara-prod-1

- [x] 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://ensro@mysql-ensembl-mirror:4240/ncbi_taxonomy

  mysqldump

  ```
  time db_cmd.pl $COMPARA_REG ncbi_taxonomy -reg_type taxonomy -
  executable mysqldump --prepend --extended-insert --prepend --
  compress ncbi_taxa_node ncbi_taxa_name | sed 's/ENGINE=MyISAM
  /ENGINE=InnoDB/g' | db_cmd.pl $COMPARA_REG compara_master
  ```

  It usually takes between 30 and 60 seconds.

  Then check that all the GenomeDBs still have a valid taxon_id

  ```
  db_cmd.pl $COMPARA_REG compara_master -sql 'SELECT genome_db.*
  FROM genome_db LEFT JOIN ncbi_taxa_node USING (taxon_id) WHERE
  genome_db.taxon_id IS NOT NULL AND ncbi_taxa_node.taxon_id IS NULL'
  ```

  Only genome_db_id 76 (tarsius_syrichta/tarSyr1_OLD, the old tarsier) should be out of sync, which is fine

- [x] 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.

  load ensembl_aliases

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

- [x] Run the CheckTaxon healthcheck

  Click here to expand...

  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

```
#cd to your local healthcheck git repo :
cd ensj-healthcheck/

# make sure you are using the right version of JAVA:
export JAVA_HOME=/nfs/software/ensembl/latest/linuxbrew/opt/jdk@8

# 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 $(mysql-ens-compara-
prod-1 details script) -d ensembl_compara_master --release
$CURR_ENSEMBL_RELEASE -t org.ensembl.healthcheck.testcase.compara.
CheckTaxon
```

Taxonomy mismatches

Sometimes, there can be mismatches in the taxonomy between master and the core database. This check lives with compara for historical reasons, but in fact, it is more relevant to the production and genebuild teams. Report such occurrences to production.

Example mismatch error

```
ensembl_compara_master: classification:: Drosophilini
Drosophilinae Drosophilidae Ephydroidea Acalyptratae
Schizophora Cyclorrhapha Eremoneura Muscomorpha Brachycera
Diptera Holometabola Neoptera Pterygota Dicondylia Insecta
Hexapoda Pancrustacea Mandibulata Arthropoda Panarthropoda
Ecdysozoa Protostomia Bilateria Eumetazoa Metazoa
Opisthokonta Eukaryota is not in
drosophila_melanogaster_core_92_6
drosophila_melanogaster_core_92_6: classification::
Drosophila Drosophiliti Drosophilina Drosophilini
Drosophilinae Drosophilidae Ephydroidea Acalyptratae
Schizophora Cyclorrhapha Eremoneura Muscomorpha Brachycera
Diptera Endopterygota Neoptera Pterygota Dicondylia Insecta
Hexapoda Pancrustacea Mandibulata Arthropoda Panarthropoda
Ecdysozoa Protostomia Bilateria Eumetazoa Metazoa
Opisthokonta Eukaryota is not in ensembl_compara_master
```

Add new entries to compara master database

The current master database (e90) is called ensembl_compara_master on mysql-ens-compara-prod-1. 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.

- [x] Add the new species / assemblies (i.e. genome_dbs)

Click here to expand...

Run this to generate a file with the new / updated assemblies

```
mysql-ens-sta-1 ensembl_production_$CURR_ENSEMBL_RELEASE -Ne
"SELECT production_name FROM species JOIN db AS db_curr USING
(species_id) LEFT JOIN db AS db_prev ON db_prev.species_id =
species.species_id AND db_prev.db_release=db_curr.db_release-1 AND
db_prev.db_type=db_curr.db_type WHERE db_curr.
db_release=$CURR_ENSEMBL_RELEASE AND db_curr.db_type='core' AND
(db_prev.db_id IS NULL OR db_prev.db_assembly!=db_curr.
db_assembly) ORDER BY production_name" > $ENSEMBL_CVS_ROOT_DIR
/ensembl-compara/updated_species_names.$CURR_ENSEMBL_RELEASE
```

The load all the genomes at once

```
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline
/update_genome.pl --reg_conf $ENSEMBL_CVS_ROOT_DIR/ensembl-compara
/scripts/pipeline/production_reg_ebi_conf.pl --compara
compara_master --release --file $ENSEMBL_CVS_ROOT_DIR/ensembl-
compara/updated_species_names.$CURR_ENSEMBL_RELEASE
```

in case you ever have to update a single genome

```
 perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline
/update_genome.pl --reg_conf $ENSEMBL_CVS_ROOT_DIR/ensembl-compara
/scripts/pipeline/production_reg_ebi_conf.pl --compara
compara_master --species Medaka --force --release
```

Add a summary of the new genome_dbs to the confluence page Release plans. The script may take a while if the species you are adding is new and has a lot of scaffolds. You can check the progress by counting *dnafrag* entries in the master database:

```
SELECT COUNT(*) FROM dnafrag;
```

- [x] Update the collection

Click here to expand...

Run this to generate a file with all the current assemblies

```
db_cmd.pl $COMPARA_REG compara_master -sql 'SELECT name FROM
genome_db WHERE first_release IS NOT NULL AND last_release IS
NULL' -- -BN > $ENSEMBL_CVS_ROOT_DIR/ensembl-compara
/all_species_names.$CURR_ENSEMBL_RELEASE
```

And then update the collection

```
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline
/edit_collection.pl --reg_conf $ENSEMBL_CVS_ROOT_DIR/ensembl-
compara/scripts/pipeline/production_reg_ebi_conf.pl --compara
compara_master --collection ensembl --file $ENSEMBL_CVS_ROOT_DIR
/ensembl-compara/all_species_names.$CURR_ENSEMBL_RELEASE --update
--release
```

- [x] Add in extra non-reference patches.

Click here for details

This is currently done when a new patch for human, mouse or zebrafish 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/ e.g. 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 alignments on these need to be deleted from the Compara database. 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 -compara "mysql://ensro@mysql-ens-
compara-prod-1:4485/ensembl_compara_master" -new_core
"mysql://ensro@mysql-ens-sta-1:4519/homo_sapiens_core_88_38?
group=core&species=homo_sapiens" -prev_core "mysql://ensro@mysql-
ensembl-mirror:4240/homo_sapiens_core_87_38?
group=core&species=homo_sapiens"
```

Sample ouptut

```
NEW patches
  CHR_HG2088_PATCH 2006369791 2017-01-19 11:45:17
  CHR_HSCHR19KIR_HG2394_CTG3_1 2006369821 2017-01-19 11:45:17
  CHR_HSCHR19KIR_0019-4656-A_CTG3_1 2006369811 2017-01-19 11:45:17
  CHR_HSCHR19KIR_7191059-1_CTG3_1 2006369829 2017-01-19 11:45:17
  CHR_HSCHR19KIR_7191059-2_CTG3_1 2006369835 2017-01-19 11:45:17
  CHR_HSCHR19KIR_0019-4656-B_CTG3_1 2006369831 2017-01-19 11:45:17
  CHR_HSCHRX_3_CTG7 2006369841 2017-01-19 11:45:17
  CHR_HG1708_PATCH 2006369799 2017-01-19 11:45:17
  CHR_HG2236_PATCH 2006369783 2017-01-19 11:45:17
  CHR_HSCHR19KIR_0010-5217-AB_CTG3_1 2006369827 2017-01-19 11:45:17
  CHR_HSCHR17_11_CTG4 2006369809 2017-01-19 11:45:17
  CHR_HSCHR19KIR_CA01-TB01_CTG3_1 2006369819 2017-01-19 11:45:17
  CHR_HSCHR19KIR_HG2393_CTG3_1 2006369839 2017-01-19 11:45:17
  CHR_HSCHR1_6_CTG3 2006369781 2017-01-19 11:45:17
  CHR_HSCHR19KIR_CA04_CTG3_1 2006369833 2017-01-19 11:45:17
  CHR_HG926_PATCH 2006369801 2017-01-19 11:45:17
  CHR_HSCHR19KIR_CA01-TA01_1_CTG3_1 2006369813 2017-01-19 11:45:17
  CHR_HSCHR19KIR_CA01-TB04_CTG3_1 2006369817 2017-01-19 11:45:17
```

```
  CHR_HSCHR4_12_CTG12 2006369785 2017-01-19 11:45:17
  CHR_HG2068_PATCH 2006369795 2017-01-19 11:45:17
  CHR_HSCHR19KIR_502960008-1_CTG3_1 2006369825 2017-01-19 11:45:17
  CHR_HG2046_PATCH 2006369805 2017-01-19 11:45:17
  CHR_HG2285_HG106_HG2252_PATCH 2006369803 2017-01-19 11:45:17
  CHR_HG2266_PATCH 2006369793 2017-01-19 11:45:17
  CHR_HG2067_PATCH 2006369797 2017-01-19 11:45:17
  CHR_HSCHR19KIR_CA01-TA01_2_CTG3_1 2006369815 2017-01-19 11:45:17
  CHR_HSCHR19KIR_HG2396_CTG3_1 2006369837 2017-01-19 11:45:17
  CHR_HG30_PATCH 2006369789 2017-01-19 11:45:17
  CHR_HSCHR19KIR_502960008-2_CTG3_1 2006369823 2017-01-19 11:45:17
  CHR_HSCHR5_8_CTG1 2006369787 2017-01-19 11:45:17
  CHR_HSCHR17_3_CTG1 2006369807 2017-01-19 11:45:17
CHANGED patches
DELETED patches

DnaFrags to delete:
  names:
  dnafrag_ids:
Input for create_patch_pairaligner_conf.pl:
--patches chromosome:CHR_HG1708_PATCH,chromosome:
CHR_HSCHR19KIR_CA01-TA01_1_CTG3_1,chromosome:CHR_HSCHR4_12_CTG12,
chromosome:CHR_HSCHR19KIR_HG2396_CTG3_1,chromosome:
CHR_HSCHR17_3_CTG1,chromosome:CHR_HSCHR5_8_CTG1,chromosome:
CHR_HSCHR19KIR_7191059-1_CTG3_1,chromosome:CHR_HSCHR19KIR_0019-
4656-B_CTG3_1,chromosome:CHR_HSCHR19KIR_CA04_CTG3_1,chromosome:
CHR_HG926_PATCH,chromosome:CHR_HSCHR19KIR_CA01-TB04_CTG3_1,
chromosome:CHR_HG2046_PATCH,chromosome:
CHR_HG2285_HG106_HG2252_PATCH,chromosome:CHR_HG2067_PATCH,
chromosome:CHR_HSCHR19KIR_HG2394_CTG3_1,chromosome:
CHR_HSCHR19KIR_0019-4656-A_CTG3_1,chromosome:
CHR_HSCHR19KIR_7191059-2_CTG3_1,chromosome:CHR_HSCHRX_3_CTG7,
chromosome:CHR_HG2236_PATCH,chromosome:CHR_HSCHR17_11_CTG4,
chromosome:CHR_HSCHR1_6_CTG3,chromosome:CHR_HSCHR19KIR_502960008-
1_CTG3_1,chromosome:CHR_HG2266_PATCH,chromosome:
CHR_HSCHR19KIR_502960008-2_CTG3_1,chromosome:CHR_HG2088_PATCH,
chromosome:CHR_HSCHR19KIR_0010-5217-AB_CTG3_1,chromosome:
CHR_HSCHR19KIR_HG2393_CTG3_1,chromosome:CHR_HSCHR19KIR_CA01-
TB01_CTG3_1,chromosome:CHR_HG2068_PATCH,chromosome:
CHR_HSCHR19KIR_CA01-TA01_2_CTG3_1,chromosome:CHR_HG30_PATCH
```

Copy the output to the Intentions for Release page as it will be needed to clean-up the alignments

In this case, there are 14 NEW patches, 1 CHANGED patch and 1 DELETED patch. They can be imported to / deleted from the master database by running update_genome.pl with the --force option

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_ebi_conf.pl --compara
compara_master --species human --force
```

Running LASTZ for patches

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.rst

- [x] 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_ebi_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_ebi_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_ebi_conf.pl --reg_alias compara_master -
sql 'select count(*) from dnafrag where coord_system_name="lrg"'
```

they should be the same.

- [x] Load the divergence times from TimeTree

Click here to expand...

This will do searches on the TimeTree website to find divergence times of all the clades that are represented in the master database

```
standaloneJob.pl Bio::EnsEMBL::Compara::RunnableDB::SpeciesTree::
LoadTimeTree -reg_conf $ENSEMBL_CVS_ROOT_DIR/ensembl-compara
/scripts/pipeline/production_reg_ebi_conf.pl -compara_db
compara_master -debug 1
```

Additionally you could check the results by running the following query on the master database (and compare with the previous release database if needed):

```
SELECT name_class , COUNT(*) FROM ncbi_taxa_name GROUP BY
name_class;
rel. 93: | ensembl timetree mya |       79 |
rel. 94: | ensembl timetree mya |       99 |
```

If TimeTree is down, copy the data from the last release:

```
db_cmd.pl $COMPARA_REG compara_prev -executable mysqldump --
prepend --insert-ignore --prepend --no-create-info --prepend
--where="name_class = 'ensembl timetree mya'" ncbi_taxa_name
| db_cmd.pl $COMPARA_REG compara_master
```

Add method_link_species_set entries to compara master database

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.

- [x] Specific mlss_ids can be created with `create_mlss.pl` (see Master database), but we have another script that will bulk-create all the mlss_ids we need

```
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline
/create_all_mlss.pl --reg_conf $ENSEMBL_CVS_ROOT_DIR/ensembl-compara
/scripts/pipeline/production_reg_ebi_conf.pl --compara compara_master
-xml $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline
/compara_ensembl.xml --release --verbose
```

The script will report all the MethodLinkSpeciesSets that it needs to create, with a summary of what it has created

- [x] Reset the URL of reused mlss_ids
In case the same mlss_id can be reused, the pipeline will probably complain that there is already a URL attached to it. You need to reset these URLs

```
UPDATE method_link_species_set SET url = "" WHERE
method_link_species_set_id IN (<LIST_OF_mlss_ids>);
```

Usually, this happens when there are no new assemblies, in which case you need to give the mlss_id of the Family, ncRNA-tree and protein-tree pipelines

Final edits to compara master database

This runs once all the species have been added / updated.

- [x] Compare the staging servers to the master database

Click here to expand...

This script will list the genomes of the staging servers, and compare them to the master database.

```
# Run once in dry-run mode to see the changes
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline
/update_master_db.pl --reg_conf $ENSEMBL_CVS_ROOT_DIR/ensembl-
compara/scripts/pipeline/production_reg_ebi_conf.pl --compara
compara_master --dry_run
```

** note: The above might throw a warning about not finding any species name in the ancestral db, this is expected and should not break the test!!

Some things (like the genebuild date is different) can be directly changed by the script (use the --nodry-run option instead). If there is a new assembly / species on the staging servers that is not yet in the master database, run update_genome.pl (see above) to add it, and re-run update_master_db.pl to make sure things are solved.

- [ ] Healthcheck the master database

Click here to expand...

See JAVA Healthchecks and use the ComparaMaster group

```
#cd to your local repo of healthcheck
cd ensj-healthcheck/
$ENSEMBL_CVS_ROOT_DIR/ensj-healthcheck/run-configurable-testrunner.
sh $(mysql-ens-compara-prod-1 details script) -d
ensembl_compara_master --release $CURR_ENSEMBL_RELEASE -g
ComparaMaster
```

possible errors:

"""

org.ensembl.healthcheck.testcase.compara.CheckTaxon [Team responsible: COMPARA]
ensembl_compara_master: name::Saccharomyces cerevisiae is not in saccharomyces_cerevisiae_core_94_4 saccharomyces_cerevisiae_core_94_4: name::Saccharomyces cerevisiae S288C is not in ensembl_compara_master ensembl_compara_master: classification:: Saccharomycetaceae Saccharomycetales Saccharomycetes Saccharomycotina saccharomyceta Ascomycota Dikarya Fungi Opisthokonta Eukaryota is not in saccharomyces_cerevisiae_core_94_4 saccharomyces_cerevisiae_core_94_4: classification:: Saccharomyces cerevisiae Saccharomycetaceae Saccharomycetales Saccharomycetes Saccharomycotina saccharomyceta Ascomycota Dikarya Fungi Opisthokonta Eukaryota is not in ensembl_compara_master

"""

There is nothing that compara can do about this error. Report it to production/core. The name in core needs to be updated to what is on the NCBI taxon tables

- [ ] Backup the master database (one more time)

```
mysql-ens-compara-prod-1 mysqldump ensembl_compara_master > /nfs
/production/panda/ensembl/warehouse/compara/master_db_dumps
/ensembl_compara_master.$(date '+%Y%m%d').
during${CURR_ENSEMBL_RELEASE}.sql
```

Production

Pre-production tasks

Core databases

- [x] Some of our pipelines intensively use the Core databases. To avoid overloading the Ensembl-wide staging server (mysql-ens-sta-1), copy all the core databases to mysql-ens-vertannot-staging with this script:

```
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline
/copy_all_core_databases.pl
```

But first try to clean up the server by removing the databases of the previous release !
The script will list:

- the databases already present on mysql-ens-vertannot-staging with the same *database name*. Check with the genebuilders that the assembly and geneset they contain is the same as on mysql-ens-sta-1
- the databases already present on mysql-ens-vertannot-staging with the same *species name*. Check with the genebuilders what they contain and whether they can be dropped, as the Registry will be confused if it finds two databases for the same species

If there are any, you will have to add the --force option to ignore the warnings and do the copy.

- [x] if you are interested in updating all the core db in your target server use the --update option
In reality, the script doesn't copy the databases itself, but *submits* copy jobs on the Ensembl Production self-service API. Check on http://ens-prod-1.ebi.ac.uk:8000/#!/copy_list that the jobs have completed successfully.

Linuxbrew paths

- [x] As the linuxbrew installation can sometimes go awry, run this test to check that all the paths are valid

```
prove $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/modules/t
/housekeeping_checkAllLinuxbrewPaths.t
```

If you find anything wrong, report that to the group so that it can be fixed before starting the pipelines

Main site

List the pipelines that have to be run on the *Intentions for Release* page and schedule the work with the rest of the team.

LastZs

- [ ] Any programmed LastZ can already be started at this point. It is important to start them ASAP in order to save time.

Members

- [x] As of release 90, members are preloaded before any gene-homology pipelines are run. Use the following pipeline to load the members:

```
init_pipeline.pl Bio::EnsEMBL::Compara::PipeConfig::EBI::Ensembl::
LoadMembers_conf --collection ensembl
```

> Remember to pass the `member_db` parameter to the other production pipelines at initialisation (ProteinTrees, Families, etc), or update the parameter in the PipeConfig files!!

Genome dumps

- [x] As of release 94, genome DNA sequences are dumped to FASTA file before any multiple-alignment pipelines are run. Use the following pipeline to start the dumps

```
init_pipeline.pl Bio::EnsEMBL::Compara::PipeConfig::EBI::
DumpGenomes_conf --collection ensembl
```

The dumps will be put under `/hps/nobackup2/production/ensembl/compara_ensembl/genome_dumps/`, which all of multiple-alignment pipelines are configured to use.

Species-tree

- [x] We now have a species-tree pipeline that generates the species-tree used by all the pipelines. As soon as the **unmasked** genome dumps are done, use the following command to generate the new species tree:

```
init_pipeline.pl Bio::EnsEMBL::Compara::PipeConfig::
CreateSpeciesTree_conf -host mysql-ens-compara-prod-X -port XXXX -
output_file $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline
/species_tree.ensembl.branch_len.nw
```

Note that, by default, the pipeline runs on all members of the '*ensembl*' collection.

Check that the tree has been rooted correctly. If it hasn't, we can do it in FigTree:

1. open the file
2. select *saccharomyces_cerevisiae* and click the 'Reroot' button (

).
3. go to File > Export Trees. Select Newick from the dropdown menu and check the box marked 'Save as currently displayed'

Once finished, copy the output file to `$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/species_tree.ensembl.branch_len.nw` , commit it and push.

GRCh37 site

Ensembl maintains a special archive site for the previous human assembly (GRCh37): http://grch37.ensembl.org . Whenever we release an important new method / feature, we need to consider doing it for GRCh37 too, see GRCh37 website for more information.

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.

NOTE

Remember to run `mysqlanalyze` and `mysqloptimize` regularly throughout the merging on the release db.

It can be a bit expensive, like re-enabling the keys, so not something to do at every step. But from time to time it can help

- [x] 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
  ```

- [x] 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_database.pl --reg-conf $ENSEMBL_CVS_ROOT_DIR/ensembl-
  compara/scripts/pipeline/production_reg_ebi_conf.pl \
  --master compara_master --old compara_prev --new compara_curr --
  intentions > populate_new_database.intentions
  ```

  This normally takes less than a minute and produces a long list.

  If you believe some of the MLSS or SS entries should NOT be copied, connect to the master database and change the last_release of the unwanted entries to the previous release number. Conversely, if you want to prepare some new MLSS or SS entries to be copied (e.g. the newly run pipelines), change the first_release of the wanted entries to the current release number.

  NB: OLD INSTRUCTIONS FOR PRE-first_release/last_release API: *There are cases where the mlss does not change but the underlying data does, e.g. 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.*

The removal of old data shouldn't be necessary unless the skip_mlss entries are not up to date. However if you need to remove old data from the release db follow the instructions below

Removal of old data from Release DB

Start the copying:

populate_new_database

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

Click here for run times

rel. 94: 679m18.736s

rel. 93: 343m46.507s

rel. 91: 240m55.310s

rel. 88: 640m18.007s

took 351m23.029s (~5.85 hours) for rel.87

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)

rel.82: 5.5h (compara5)

rel.83: 4.8h (compara5)

rel.85: 7.5h (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
```

- [x] 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"
```

- [x] 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=/nfs/software/ensembl/latest/linuxbrew/opt/jdk@8

# 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 production-rh7 -M8000  -R"select[mem>8000]  rusage
[mem=8000]" -I ./run-configurable-testrunner.sh $(mysql-ens-
compara-prod-1 details script) -d ensembl_compara_88 --release
$CURR_ENSEMBL_RELEASE -g ComparaShared
```

At this point its OK to have some unused method_links since they will be removed later. But should NOT be removed now since they may still be used by the merging.

Click here for run times

rel. 94: 17m46.121s

rel.88 1:15m

rel.83: 13 minutes, 2 expected complaints (CheckSpeciesSetSizeByMethod may complain about Human-on-Human lastz-new and ForeignKeyMasterTables will complain about empty MethodLink entries (this will be deleted later in the merging process))

rel.85: 20mins

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 recomputing the indices in the end of each run.

Running "-re_enable 1" will add *at least 2 hours* (rel.82) to the merging time (but it is necessary in the final product) so make sure you only do it once.

- [x] Pairwise alignments: LASTZ_NET (and, formerly, BLASTZ_NET or TRANSLATED_BLAT_NET)

  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.

  Also, check first_release of these databases. In case it hasn't been set, you need to do it **now** on **both** the production database and the master database,

  Example:

  copy_data

```
    # for each source URL [there may be several sources, remember
to add all of them]: 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://ensro@mysql-ens-compara-prod-3:4523
/mateus_lastz_cat_human_93 --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://ensro@compara4/sf5_ggal_falb_lastz_73
```

- [x] The curious case of LASTZ_PATCH alignments. There is **always** something to copy, even if there are no new patches

  Click here for details

  You will also have to copy Human_ref_vs_Human_patches and Mouse_ref_vs_Mouse_patches LASTZ_PATCH alignments, but mind the source:

If there were new patches, you'll import them in a way similar to other LASTZ:

copy_data

```
#First run the following pipeline to import the alignments between
patches / haplotypes and primary regions.

init_pipeline.pl Bio::EnsEMBL::Compara::PipeConfig::
ImportPatchAlignmentsToRef_conf -host comparaX

#then
# 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_ebi_conf.pl --to_reg_name
compara_curr --method_link_type LASTZ_PATCH --re_enable 0 --
from_url <the_url_of_the_pipeline_db_from_above>
```

If there were no new patches, you will still have to copy them from compara_prev, since LASTZ_PATCH alignments are automatically skipped by populate_new_database.pl script. You simply have to refer to the previous database as the source:

copy_data

```
    # note the method_link_type is LASTZ_PATCH !
bsub -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_ebi_conf.pl --to_reg_name compara_curr --
method_link_type LASTZ_PATCH --re_enable 1 --from_reg_name
compara_prev
```

- [x] 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 the --merge option.

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 -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_ebi_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 0 --merge --
from_url  mysql://ensro@mysql-ens-compara-prod-1.ebi.ac.uk:4485
/carlac_lastz_human_patches_88 --dry_run

    # if happy, remove the --dry_run flag and run it again,
preferably on the farm:
bsub -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_ebi_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 --
from_url  mysql://ensro@mysql-ens-compara-prod-1.ebi.ac.uk:4485
/carlac_lastz_human_patches_88
```

- [x] 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_ebi_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://ensro@compara2/sf5_epo_low_8way_fish_71 -
re_enable 0
```

- [x] 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  -- --
keys  genomic_align_block %
db_cmd.pl $COMPARA_REG compara_curr --executable mysqlshow -- --
keys genomic_align %
db_cmd.pl $COMPARA_REG compara_curr --executable mysqlshow  -- --
keys  genomic_align_tree %
db_cmd.pl $COMPARA_REG compara_curr --executable mysqlshow  -- --
keys  conservation_score %
db_cmd.pl $COMPARA_REG compara_curr --executable mysqlshow  -- --
keys  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";
```

- [x] Syntenies

Click here for details

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

Inititate Synteny pipeline on the merged database

```
init_pipeline.pl Bio::EnsEMBL::Compara::PipeConfig::EBI::
Synteny_conf -pipeline_name synteny_93 -alignment_db
mysql://ensro@mysql-ens-compara-prod-1:4485/ensembl_compara_93
```

Before running the code... Ensure that you check the synteny coverage and if it is less than 1%, It must be deleted from mlss, mlss_tag, dnafrag_region and synteny_region tables. **This should be automated in the synteny pipeline by release 84.

Example

load synteny data

```
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/copy_data.
pl --reg_conf $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts
/pipeline/production_reg_ebi_conf.pl --to_reg_name compara_curr --
method_link_type SYNTENY --from_url mysql://ensro@compara1
/cc21_synteny_83 --dry_run
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline/copy_data.
pl --reg_conf $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts
/pipeline/production_reg_ebi_conf.pl --to_reg_name compara_curr --
method_link_type SYNTENY --from_url mysql://ensro@compara1
/cc21_synteny_83
```

- [x] 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://ensro@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 mysql-ens-compara-
prod-1
```

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.94: 30min

rel.93: 59min

rel.87: 29.5min

rel.86: 58min

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';
```

Healthcheck Ancestral Core

A non-java healthcheck is available to ensure that the dnafrags in the release database are in sync with the seq_regions in the new core. Run as follows:

```
perl $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/production
/verify_ancestral_dnafrags.pl -compara mysql://ensro@mysql-
ens-compara-prod-1:4485/ensembl_compara_92 -ancestral
mysql://ensro@mysql-ens-compara-prod-1:4485
/ensembl_ancestral_92
```

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
# rel.88: 54m44.503s
# rel.85: 40mins
# took 45 minutes in rel.81
# took 42 minutes in rel.82
# took 38 minutes in rel.83
# took 38 minutes in rel.84
# took 54 minutes in rel.89
```

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.

```
$ENSEMBL_CVS_ROOT_DIR/ensembl/misc-scripts/schema_patcher.pl --
host=compara5 --user=ensadmin --pass=${ENSADMIN_PSW} --
database=lg4_ensembl_ancestral_${CURR_ENSEMBL_RELEASE}
```

If patches were applied, make sure you have both analyzed and optimized the tables:

```
db_cmd.pl $COMPARA_REG ancestral_curr -reg_type core -executable
mysqlcheck -- --analyze --verbose
db_cmd.pl $COMPARA_REG ancestral_curr -reg_type core -executable
mysqlcheck -- --optimize --verbose
```

Merge the homology pipelines

- [x] Check that all the pipelines have been run and have completed. Usually, this means protein-trees & ncRNA-trees (for the default species and for the mouse-strains), incl. the WGA Orthology QC, and families
- [x] Run the *ImportAltAlleGroupsAsHomologies_conf* pipeline
- [x] Run the *EBI/Ensembl/MergeDBsIntoRelease_conf* pipeline to merge tables from all the products into the release database

Click here for details

Go to $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/modules/Bio/EnsEMBL/Compara/PipeConfig/EBI/Ensembl/ 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 ProteinTrees pipeline database
family_db - should point to the current Families pipeline database
ncrna_db - should point to the current ncRNAtrees pipeline database
mouse_prot_db - should point to the current mouse-strains ProteinTrees pipeline database
mouse_ncrna_db - should point to the current mouse-strains ncRNAtrees pipeline database
projection_db - should point to the latest run of the ImportAltAlleGroupsAsHomologies_conf pipeline
members_db - should point to the database that contains all members

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 Bio::EnsEMBL::Compara::PipeConfig::EBI::Ensembl::
MergeDBsIntoRelease_conf.pm
beekeeper.pl ... -sync
runWorker.pl ... -job_id 1
# If the first job passes
beekeeper.pl ... -loop
```

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

**** NOTE: you may have to do some cleaning of the mouse strain db e.g. deleting some duplicated mlss ids.

Merge family members

Members are exclusively copied from the members db, but extra members need to be merged from the families pipeline database. This includes gene_members, seq_members and sequences with a dbID >= 900000001

- [x] Load the species-trees (needed for the Species-tree view)

```
$ init_pipeline.pl Bio::EnsEMBL::Compara::PipeConfig::
LoadSpeciesTrees_conf -compara_alias_name compara_curr -host mysql-
ens-compara-prod-1 -port 4485
# Then run beekeeper as suggested by init_pipeline.pl
```

Note: the last analysis of this pipeline failed in rel.82 (all 4 jobs of this analysis) trying to insert duplicated entries into MLSS_tag table, but all the data was there, so I just carried on. same in rel.85

- [x] Run the `Bio::EnsEMBL::Compara::PipeConfig::Example::EnsemblPostHomologyMerge_conf` pipeline:

Click here to expand...

This pipeline combines a few pipelines we have to run at this stage: GeneMemberHomologyStats_conf and HighConfidenceOrthologs_conf. Each of these can be disabled with a flag, and they're also all available as a standalone pipelines.

- GeneMemberHomologyStats_conf. It populates the gene_member_hom_stats table (absence/presence of gene-tree, number of orthologues, etc) which is used by Web to grey out the menu items. This part comes pre-seeded with two jobs (one for the *default* collection and one for the *murinae* collection).
- HighConfidenceOrthologs_conf. This pipeline marks some orthologues as high-confidence based on their % identity, GOC and WGA scores. Because GOC is not computed on ncRNAs, the pipeline has two stream of jobs: one for protein-coding genes and one for ncRNAs.  This needs the WGA scores to be in !

init_pipeline.pl Bio::EnsEMBL::Compara::PipeConfig::Example::EnsemblPostHomologyMerge_conf -compara_db mysql://ensadmin:${ENSADMIN_PSW}@mysql-ens-compara-prod-1:4485/ensembl_compara_88

- [x] Drop the three databases used for merging.

Click here for details

```
        # After you are happy about the result of protein side
merging you can drop the
"YourName_homology_projections_ThisRelease" database.
$ db_cmd.pl -url mysql://ensadmin:${ENSADMIN_PSW}@compara5
/lg4_homology_projections_${CURR_ENSEMBL_RELEASE} -sql 'drop
database'

        # same for the LoadSpeciesTrees database:
$ db_cmd.pl -url mysql://ensadmin:${ENSADMIN_PSW}@compara5
/lg4_load_species_trees_${CURR_ENSEMBL_RELEASE} -sql 'drop
database'

        # same for the MergeDBsIntoRelease database:
$ db_cmd.pl -url mysql://ensadmin:${ENSADMIN_PSW}@compara5
/lg4_pipeline_dbmerge_${CURR_ENSEMBL_RELEASE} -sql 'drop database'
```

- [x] git commit the changes to the PipeConfig files that you have made.

Final database checks

- [x] 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

```
-- prepend with:     db_cmd.pl -reg_conf $ENSEMBL_CVS_ROOT_DIR
/ensembl-compara/scripts/pipeline/production_reg_ebi_conf.pl -
reg_alias compara_curr -sql

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;

note**** we deleted 18 mlss_ids in rel.83, rel.86 & rel.88, rel.
89; 19 in rel. 90; 19 in rel 93 and 94;
```

- [x] 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

Run the healthchecks

- [x] 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=/nfs/software/ensembl/latest/linuxbrew/opt/jdk@8
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

- [x] 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 $(mysql-ens-compara-prod-1 details
script) -d mateus_ensembl_ancestral_88 --release
$CURR_ENSEMBL_RELEASE -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
```

If healthcheck indicates that tables need to be analysed, follow instructions here: Analyze / Optimize the databases

- [x] 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 $(mysql-ens-compara-
prod-1 details script) -d ensembl_compara_88 --release
$CURR_ENSEMBL_RELEASE -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 --release $CURR_ENSEMBL_RELEASE -t org.
ensembl.healthcheck.testcase.compara.MLSSTagMaxAlign --repair 1 --
user ensadmin --password $ENSADMIN_PSW
```

- [x] 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 $(mysql-ens-compara-
prod-1 details script) -d ensembl_compara_88 --release
$CURR_ENSEMBL_RELEASE -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 --release $CURR_ENSEMBL_RELEASE -t org.
ensembl.healthcheck.testcase.compara.MLSSTagGERPMSA --repair 1 --
user ensadmin --password $ENSADMIN_PSW
```

- [x]

Run the ComparaAll group of healthchecks on the release database. NOTE: If pressed for time, this test can divided into 2 separate tests that can be run simultaneously to save some time. You just have substitute "ComparaAll" in the code block below to "ComparaHomology" in one run and "ComparaGenomic" in the other run

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.

```
time bsub -M8000  -R"select[mem>8000]  rusage[mem=8000]" -I stdbuf
-o0 ./run-configurable-testrunner.sh $(mysql-ens-compara-prod-1
details script) -d ensembl_compara_88 --release
$CURR_ENSEMBL_RELEASE -g ComparaAll | tee healthchecks_after_merge.
txt
```

- [x] Run the ControlledComparaTables 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 -M8000  -R"select[mem>8000]  rusage[mem=8000]" -I stdbuf
-o0 ./run-configurable-testrunner.sh $(mysql-ens-compara-prod-1
details script) -d ensembl_compara_89 --release 89 --
compara_master.database ensembl_compara_master -g
ControlledComparaTables | tee
healthchecks_controlled_tables_after_merge.txt
```

- [x] Run the ComparaSanity group of healthchecks on the release database

Click here for details

"Sanity" tests are tested that are expected to fail and have to be manually approved

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 -M8000  -R"select[mem>8000]  rusage[mem=8000]" -I stdbuf
-o0 ./run-configurable-testrunner.sh $(mysql-ens-compara-prod-1
details script) -d ensembl_compara_89 --release 89 --
compara_master.database ensembl_compara_master -g ComparaSanity |
tee healthchecks_sanity_after_merge.txt
```

Typically, there are three failures:

- CheckConservationScoreSanity: Most of the multiple-genome WGAs will have a few blocks with no conservation scores. This is fine as long as it's only a few of them.
- CheckSyntenySanity: For distant species, some chromosomes may not have any synteny blocks. Again, there shouldn't be more than a few.
- CheckTableSizes: This compares the size of each table to the previous versions of the database and complains if there is a significant change (either way), if the tables have exactly the same size, or if some tables have appeared / disappeared. As the relco, you need to check that all the differences correspond to schema or dataset changes

Test web server

- [x] Ask web's relco to point the test web server to the compara release database
- [x] Follow instructions here to test staging data

Final handover of databases [edit]

Handover

- [x] Let the production team know that our database is ready, and where. From this point onwards, everybody **hands-off** the database. Nothing is allowed to change in it
- [ ] Generate a new Age of Base file

  Click here for more information...

  If the mammals EPO alignment has been rerun, we will need to generate a new Age of Base file for human only. This can be done using the following pipeline:

  ```
  init_pipeline.pl Bio::EnsEMBL::Compara::PipeConfig::EBI::
  BaseAge_conf -pipeline_name human_base_age_${release_number}
  ```

  The pipeline will generate a number of .bed files and one bigBed (`.bb`) file. Copy the `.bb` to the following location, adhering to the naming standard `Hsap_ages_${epo_mlss_id}_${release_number}.bb` :

  `/nfs/production/panda/ensembl/production/ensemblftp/data_files/homo_sapiens/GRCh38/compara/`

  Let the web-team know if you have copied a new file or if they should consider the file from the previous release. Usually, they prefer to get this information as a Jira ticket.

  > **NB**
  >
  > The newest variation database must be handed over before running this pipeline

- [x] 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

Final bits

- [x] 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

  ```
  become -- compara_ensembl
  db_cmd.pl $COMPARA_REG compara_master --executable mysqldump |
  gzip - > /nfs/production/panda/ensembl/warehouse/compara
  /master_db_dumps/ensembl_compara_master_${CURR_ENSEMBL_RELEASE}.
  mysql.gz
  ```

Post-handover

Update documentation and diagrams [edit]

- [x] 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/production/diagrams

generate_graph.pl $COMPARA_REG compara_ptrees -output ProteinTrees.
png
**At the current time, due to the size of the protein tree db, we
have to use the older version of "dot" which is not in the
linuxbrew environment. for this to work 1) export PATH=/usr/bin
/:$PATH 2) export PATH=/nfs/software/ensembl/RHEL7-JUL2017-core2
/plenv/shims:/nfs/software/ensembl/RHEL7-JUL2017-core2/plenv/bin
/:$PATH

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@mysql-ens-compara-prod-1:4485
/mateus_dump_release_93 -out Dumps.png
generate_graph.pl --url mysql://ensro@mysql-ens-compara-prod-2:4522
/mateus_amniotes_mercator_pecan_93 -output MercatorPecan.png
generate_graph.pl --url mysql://ensro@mysql-ens-compara-prod-2.ebi.
ac.uk:4522/muffato_mammals_epo_low_coverage_93 -output
EpoLowCoverage.png
generate_graph.pl -url mysql://ensro@mysql-ens-compara-prod-2.ebi.
ac.uk:4522/waakanni_mammals_epo_93 -output epo_pt3.png
```

  Commit any changed diagrams to git and push.

- [x] 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/public-plugins/docs/htdocs/info/docs/api
/compara/compara_schema.html -d Compara $(mysql-ens-compara-prod-1
```

```
details script) -dbname ensembl_compara_$CURR_ENSEMBL_RELEASE -
sort_headers 0 -sort_tables 0 -intro $ENSEMBL_CVS_ROOT_DIR/ensembl-
compara/docs/schema_intro.html -out_diagram_dir diagrams
```

Open the output file `$ENSEMBL_CVS_ROOT_DIR/public-plugins/docs/htdocs/info/docs/api/compara/compara_schema.html` in your browser and check that no example errors are reported.

If everything looks fine, commit&push public-plugins.

- [x] Update the API 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. < and >) 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

- [x] Update the tutorial about Compara resources

  Click here to expand...

  Do the same with `$ENSEMBL_CVS_ROOT_DIR/ensembl-webcode/htdocs/info/website/tutorials/compara.html` .
  This pages can be viewed online at http://staging.ensembl.org/info/website/tutorials/compara.html

- [x] Check examples work in ensembl-compara/scripts/examples/
- [ ] 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/production`

- [ ] git commit and push any modified files or added tutorial examples

Test the sites

Once the branch is in, the web and core teams will set up the test sites. Each one has to be tested idealy following the same protocol as before the handover

- [ ] Main ensembl site http://test.ensembl.org
- [ ] GRCh37 site: http://test.ensembl.org
- [ ] Main REST server: http://test.rest.ensembl.org
  Open each Compara endpoint and check that there is an output and that it is similar to the live site
  We now have a new script to automatically test compara rest endpoints

  ```
  perl ensembl-compara/scripts/production/Verify_Compara_REST_Endpoints.
  pl http://e89.rest.ensembl.org
  ```

- [ ] GRCh37 REST server: http://test.grch37.rest.ensembl.org/

```
perl ensembl-compara/scripts/production/Verify_Compara_REST_Endpoints.
pl http://test.grch37.rest.ensembl.org
```

Data dumps

Most dumps (except homology) are currently generated in `/hps/nobackup/production/ensembl/${USER}/dumps_${release_number}`, areas and then have to be assembled into the `/nfs/production/panda/ensembl/production/ensemblftp/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)
- `tsv/ensembl-compara` (homologies)
- `xml/ensembl-compara` (homologies)
- `fasta/ancestral_alleles` (the only one without ensembl-compara in the path)
- `compara/`

> All dumps have to be present in the release-XX tree, either newly generated by running pipelines, or copied over from the previous release (e.g. if some MSAs did not run in this release).
> Compare to the previous release and check that we have more / bigger files.

A new pipeline was implemented to perform all the dumps:

Run this pipeline:

- [x] DumpAllForRelease for e93

- [x] Let the production team know that the dumps are ready in their common location.

This is now obsolete

All the copying has to be done with the `compara_ensembl` virtual user:

```
become -- compara_ensembl
```

- [ ] BED files

The `bed/ensembl-compara/` directory should contain dumps of constrained elements calculated on (a) EPO low coverage alignments and (b) Pecan alignments. If there have been new runs of these pipelines during the release, the data must be re-dumped. Otherwise, they can be copied from the last release.

> The constrained elements mlss_id should be passed to this pipeline!

The files can be generated using the following pipeline:

```
init_pipeline.pl Bio::EnsEMBL::Compara::PipeConfig::
DumpConstrainedElements_conf --compara_url $COMPARA_DB --
registry $ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts
/pipeline/production_reg_ebi_conf.pl --mlss_id $CE_MLSS_ID
```

- [ ] EMF & MAF (Alignments)

Ensure that the environment variable `$CURR_ENSEMBL_RELEASE` is set before running this pipeline! If it is not set, the pipeline will dump **everything**, rather than just freshly-run pipelines (freshly-run = MLSSes where `first_release >= $CURR_ENSEMBL_RELEASE`)

Data can be dumped in these formats for both multiple and pairwise alignments using the following pipeline:

```
# MSA pipeline settings
init_pipeline.pl Bio::EnsEMBL::Compara::PipeConfig::
DumpMultiAlign_conf --compara_db $RELEASE_DB --registry
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline
/production_reg_ebi_conf.pl --format emf+maf --
method_link_types EPO:PECAN:EPO_LOW_COVERAGE

# pairwise pipeline settings
init_pipeline.pl Bio::EnsEMBL::Compara::PipeConfig::
DumpMultiAlign_conf --compara_db $RELEASE_DB --registry
$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/scripts/pipeline
/production_reg_ebi_conf.pl --format maf --
method_link_types LASTZ_NET --make_tar_archive 1
```

- Formats:
  - For multiple alignments, the format should be "emf+maf"
  - For pairwise alignments, the format should be "maf" only
- `--method_link_types` can be replaced with `--mlss_id` in cases where only select alignments should be dumped (`--method_link_types` will result in *all* matching alignments being dumped)
- `--make_tar_archive` should be set to true for pairwise alignments

Further information can be found at `$ENSEMBL_CVS_ROOT_DIR/ensembl-compara/docs/production/README s/multi_align.dumps.rst`

- [ ] EMF, TSV & XML (Homologies)

Check all parameters in `Bio::EnsEMBL::Compara::PipeConfig::DumpTrees_conf`. This config file defines parameters for each set of trees, but we will use a wrapper pipeline, `DumpAllTrees_conf`, to dump everything (inheriting params from DumpTrees_conf).

init_pipelline

```
init_pipeline.pl DumpAllTrees_conf.pm -
dump_per_species_tsv 1 -host mysql-ens-compara-prod-1.ebi.
ac.uk -port 4485
```

The pipeline will produce tree dumps in the defined `target_dir`

Copy tree content dump for Uniprot

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

```
cp /nfs/production/panda/ensembl/compara/${USER}
/dumps_XX/ensembl.GeneTree_content.e81.txt.gz /nfs
/ftp/pub/databases/ensembl/ensembl_compara
/gene_trees_for_uniprot/
cd /nfs/ftp/pub/databases/ensembl/ensembl_compara
/gene_trees_for_uniprot
md5sum ensembl.GeneTree_content.
e<CURR_RELEASE_NUMBER>.txt.gz > ensembl.
GeneTree_content.e<CURR_RELEASE_NUMBER>.txt.gz.MD5SUM
```

compara/

Species Trees

Species trees should be dumped and placed in `release-XX/compara/species_trees/`. All species trees in a database can be dumped using a single pipeline run:

```
init_pipeline.pl Bio::EnsEMBL::Compara::PipeConfig::
DumpSpeciesTrees_conf --compara_url mysql://ensro@mysql-
ens-compara-prod-1:4485/ensembl_compara_XX
```

Alternatively, individual trees can be dumped using the `ensembl-compara/scripts/example/species_getSpeciesTree.pl` script:

```
perl species_getSpeciesTree.pl -url mysql://ensro@mysql-
ens-compara-prod-1.ebi.ac.uk:4485/ensembl_compara_XX -
mlss_id 60004 -label 'NCBI Taxonomy'
```

TreeFam HMMs

A dump of all HMM profiles used in TreeFam should be created and moved to `release-XX/compara/multi_division_hmm_lib.tar.gz`. The file is generated once when the library is created (outside of the production cycle), and then copied over release to release.

fasta/ancestral_alleles

Ancestral alleles are computed from EPO data and can be copied from the previous release if no new alignments have been run

Ancestral sequences for the members of the primate EPO can be dumped using a mini-pipeline:

```
init_pipeline.pl Bio::EnsEMBL::Compara::PipeConfig::
DumpAncestralAlleles_conf -host mysql-ens-compara-prod-X -
port XXXX
```

These are required by the variation team - update them when dumps have completed.

- [ ] Backup the master database (one last time)

```
mysql-ens-compara-prod-1 mysqldump ensembl_compara_master > /nfs
/production/panda/ensembl/warehouse/compara/master_db_dumps
/ensembl_compara_master.$(date '+%Y%m%d').post${CURR_ENSEMBL_RELEASE}.
sql
```