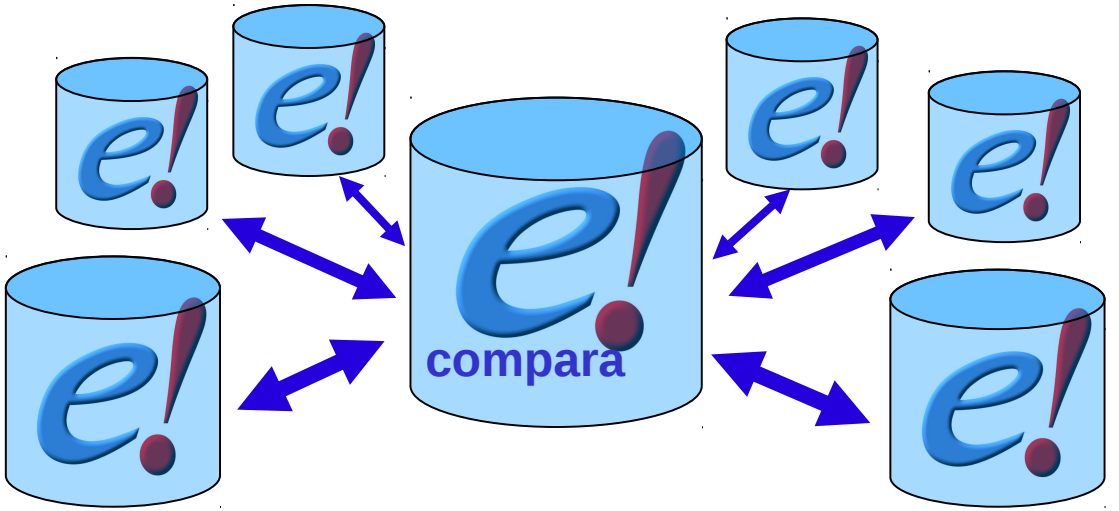
Ensembl Compara Perl API



EMBL-EBI

wellcome trust sanger institute

What is Ensembl Compara?

A single database which contains precalculated comparative genomics data and which is linked to all the Ensembl Species databases.

Access via perl API and mysql

A production system for generating that database

(not in this presentation)

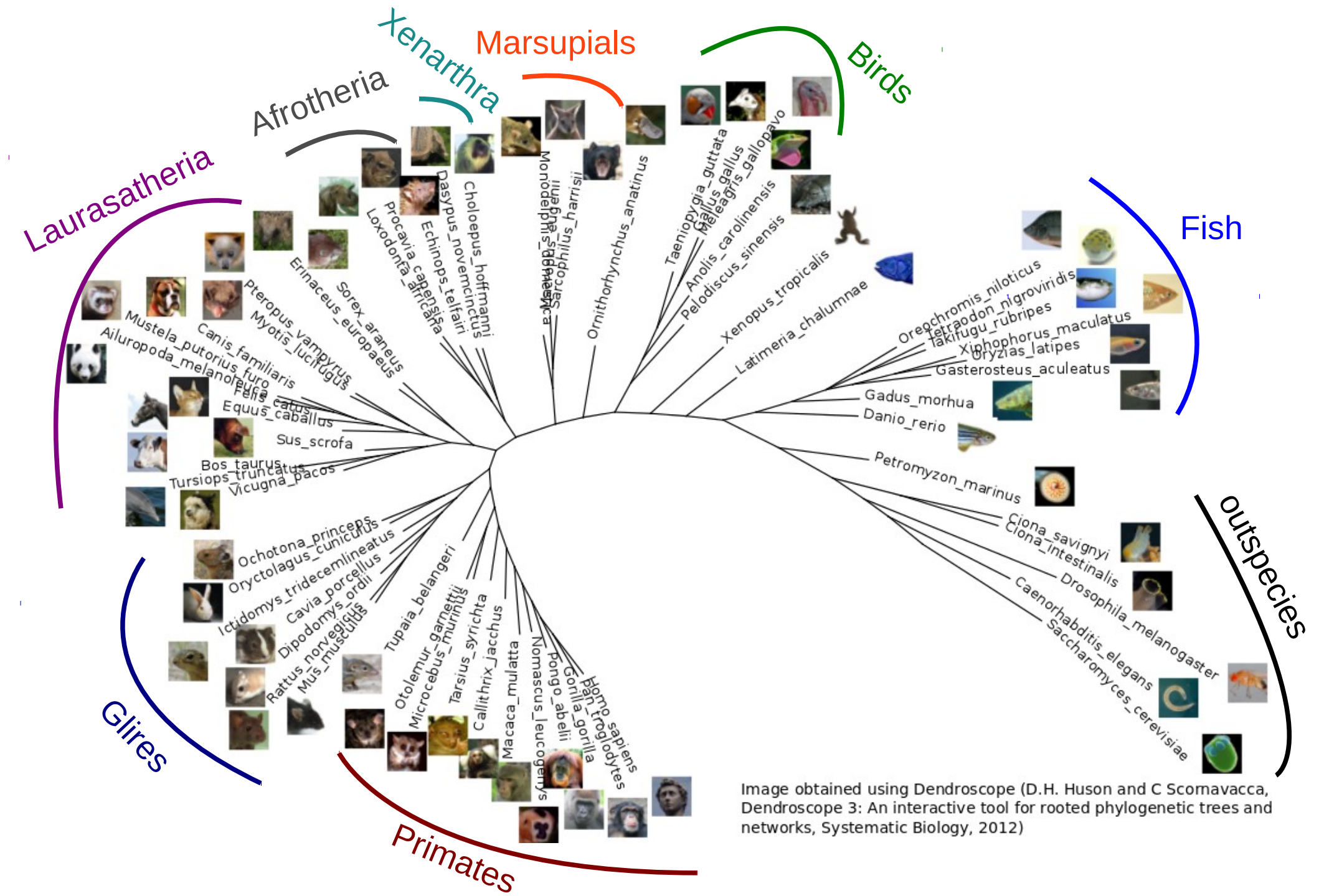
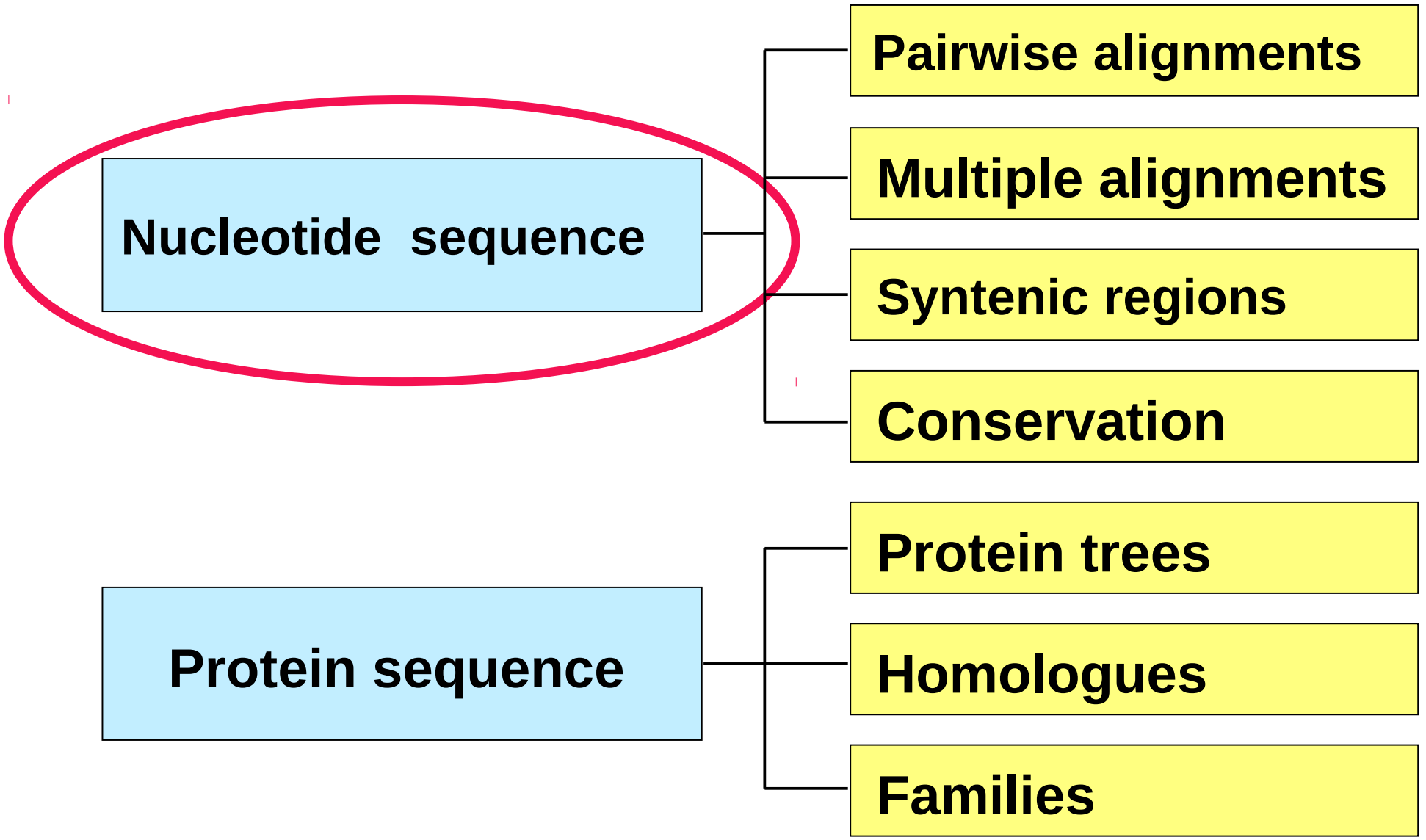


Image obtained using Dendroscope (D.H. Huson and C Scornavacca, Dendroscope 3: An interactive tool for rooted phylogenetic trees and networks, Systematic Biology, 2012)

Sequence types and outputs



Nucleotide sequence analyses

Pairwise Alignments

- BLASTZ-net
- LASTZ-net
- t-BLAT-net

Syntenic regions

- Only for species with chromosomal mappings

Multiple alignments

- Mercator-Pecan
- Enredo-Pecan-Ortheus

Conservation

- GERP Cons. Scores
- GERP Constr. Elements

Nucleotide sequence analyses

Pairwise Alignments

- BLASTZ-net
- LASTZ-net
- t-BLAT-net

BLASTz-net / LASTz-net

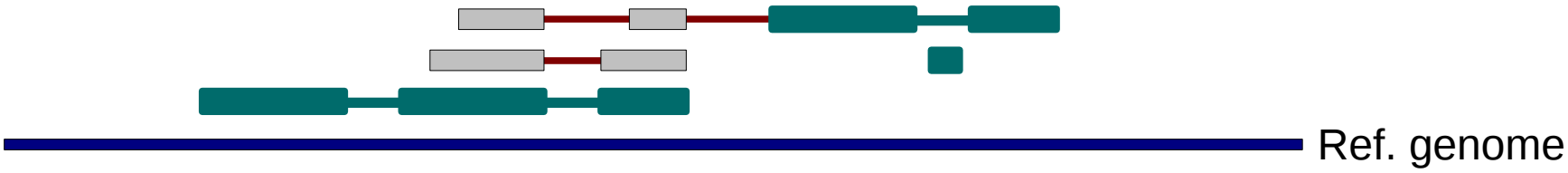
- Closely related species
- LASTz is a replacement for BLASTz

T-BLAT-net

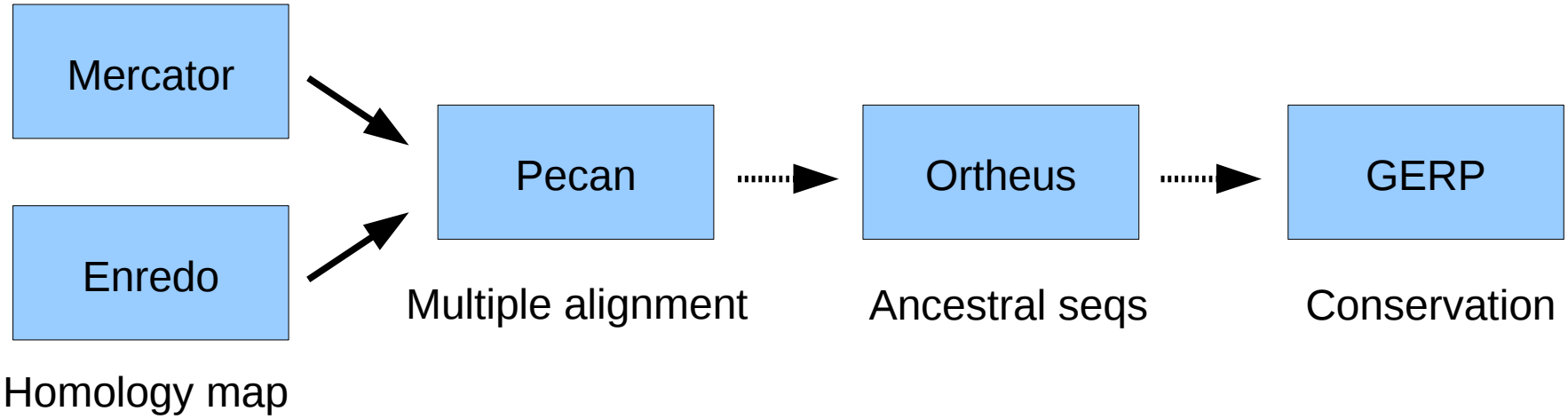
- Distantly related species
- Coding + highly conserved

Chaining/Netting

- BlastZ-raw → BlastZ-chain → BlastZ-net



Nucleotide sequence analyses



Multiple alignments

- Mercator-Pecan
- Enredo-Pecan-Ortheus

Conservation

- GERP Cons. Scores
- GERP Constr. Elements

Compara database is coupled to Ensembl core databases

Compara stores relationships between the genomes by loading references or ‘handles’ to external data.

Since there is minimal primary data inside Compara, to gain full access to the data these external links must be re-established

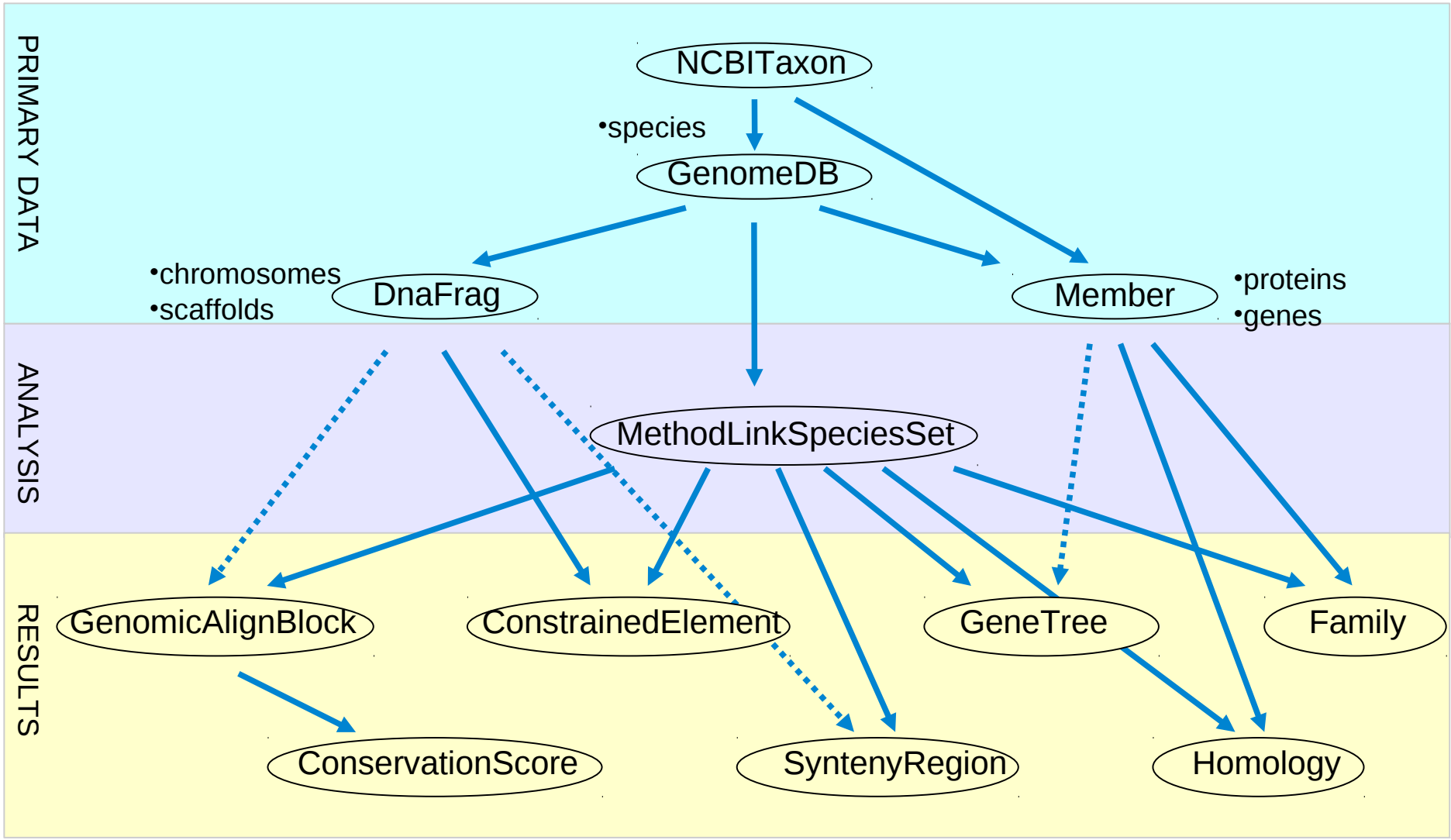
Example: compara_70 must be linked with the Ensembl core_70 databases

Proper REGISTRY configuration is critical (auto-loading is OK)

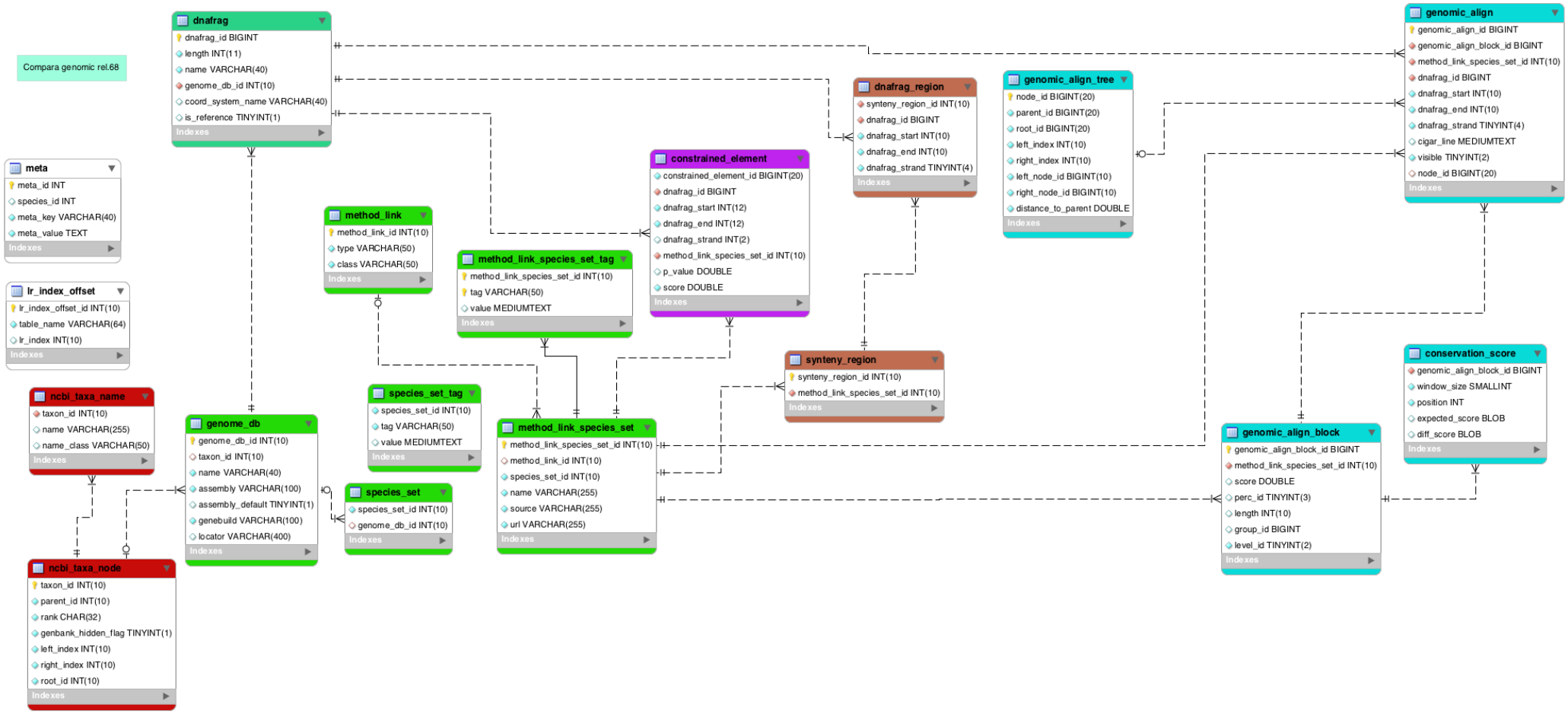
The Compara Perl API

- Written in Object-Oriented Perl
- Used to retrieve data from and store data into ensembl-compara database (only the production pipeline generates the alignments, trees, etc)
- Links species together for Ensembl website
- Generalized to extend to non-ensembl genomic data (Uniprot)
- Follows same ‘Data Object’, ‘Object Adaptor’ and 'DBAdaptor' design as the other Ensembl APIs

Compara object model overview



Database schema (genomic part)



Primary data

Conventions in these slides:

Methods returning values()
Methods returning objects()
Object Adaptor fetching method()

- **NCBITaxon**: list of all species
  - taxon_id(), classification(), binomial()
- **GenomeDB:** relates to a particular Ensembl core DB
  - name(), assembly(), genebuild(), **taxon()**
  - fetch_by_name_assembly(), fetch_by_registry_name(), fetch_by_Slice(), fetch_all()
- **DnaFrag:** relates to all "top level" SeqRegions
  - name(), length(), **genome_db()**, **slice**(), coord_system_name()
  - fetch_by_Slice(), fetch_by_GenomeDB_and_name()
- **Member**: list all Ensembl genes + SwissProt + SPTrEMBL
  - source_name(), stable_id(), **genome_db(), taxon()**, sequence(), **get_all_peptide_Members(), get_longest_peptide_Member(), gene_member()**
  - fetch_by_source_stable_id()
  - possible sources are: ENSEMBLGENE, ENSEMBLPEP, ENSEMBLTRANS, Uniprot/SPTREMBL, Uniprot/SWISSPROT

GenomeDB example code

```
use strict;
use Bio::EnsEMBL::Registry;
my $reg = "Bio::EnsEMBL::Registry";

$reg->load_registry_from_db(
    -host=>"ensembldb.ensembl.org",
    -user => "anonymous");

my $genome_db_adaptor = $reg->get_adaptor(
    "Multi", "compara", "GenomeDB");

my $genome_db = $genome_db_adaptor->
    fetch_by_registry_name("human");

print "Name: ", $genome_db->name, "\n";
print "Assembly: ", $genome_db->assembly, "\n";
print "GeneBuild: ", $genome_db->genebuild, "\n";
```

DnaFrag example code

```
use strict;
use Bio::EnsEMBL::Registry;
my $reg = "Bio::EnsEMBL::Registry";

$reg->load_registry_from_db(
    -host=>"ensembldb.ensembl.org",
    -user => "anonymous");

my $genome_db_adaptor = $reg->get_adaptor(
    "Multi", "compara", "GenomeDB");

my $genome_db = $genome_db_adaptor->
    fetch_by_registry_name("human");

my $dnafrag_adaptor = $reg->get_adaptor(
    "Multi", "compara", "DnaFrag");

my $dnafrag = $dnafrag_adaptor->
    fetch_by_GenomeDB_and_name($genome_db, "13");


print "Name: ", $dnafrag->name, "\n";
print "Length: ", $dnafrag->length, "\n";
print "CoordSystem: ", $dnafrag->coord_system_name,
    "\n";
```

API documentation & Help

- perldoc – Viewer for inline API documentation.
  - `shell> perldoc Bio::EnsEMBL::Compara::GenomeDB`
  - `shell> perldoc Bio::EnsEMBL::Compara::DBSQL::MemberAdaptor`
  - online at: http://www.ensembl.org/info/software/Pdoc/
- Tutorial document:
  - cvs: ensembl-compara/docs/ComparaTutorial.pdf
- ensembl-dev mailing list:
  - <dev@ensembl.org>

Exercises – GenomeDB & DnaFrag

- A GenomeDB is used to link the Compara database to each of the Core species databases
  - Print the name, assembly version and genebuild version for all the GenomeDBs in the compara DB
- A DnaFrag represents a top-level SeqRegion in the Compara database
  - Print all the DnaFrags for chimp

Analysis

Conventions in these slides:

Methods returning values()
Methods returning objects()
Object Adaptor fetching method()

- **Method**: type of analysis
  - type(), class(), toString()
  - Possible values for the type are: BLASTZ_NET, LASTZ_NET, TRANSLATED_BLAT_NET, PECAN, EPO, EPO_LOW_COVERAGE, SYNTENY, FAMILY, ENSEMBL_ORTHOLOGUES, ENSEMBL_PARALOGUES, PROTEIN_TREES, etc.
  - The class is used to tell the web code about the type of data that one expects for this method (pairwise alignment, conservation, gene trees...)

- **SpeciesSet**: group of species/genomes involved in an analysis
  - **genome_dbs()**

- **MethodLinkSpeciesSet**: one particular analysis set
  - **method**(), **species_set_obj**(), name(), source(), url()
  - fetch_all(), fetch_by_method_link_type_species_set_name(), fetch_by_method_link_type_registry_aliases, fetch_all_by_method_link_type, etc.

MethodLinkSpeciesSet example code

```
use strict;
use Bio::EnsEMBL::Registry;
my $reg = "Bio::EnsEMBL::Registry";

$reg->load_registry_from_db(
    -host=>"ensembldb.ensembl.org",
    -user => "anonymous");

my $mlssa = $reg->get_adaptor("Multi", "compara",
    "MethodLinkSpeciesSet");

my $mlss = $mlssa->
    fetch_by_method_link_type_registry_aliases(
      "LASTZ_NET", ["human", "mouse"]);

print $mlss->name, "\n";

print "type: ", $mlss->method->type, "\n";

my $species_set = $mlss->species_set_obj();

foreach my $this_genome_db (@{$species_set->genome_dbs}) {
  print $this_genome_db->name(), "\n";
}
```

Exercises – MethodLinkSpeciesSet

- The MethodLinkSpeciesSet is a central component in the Compara database, it stores information connecting the various analyses (method) with a set of species (species_set_obj)
  - Print the total number of MethodLinkSpeciesSet entries stored in the database
  - Print a unique list of method_link_types and a count of their number in the database
  - Print a list of the species and their internal Ids (dbIDs) for the 12 eutherian mammal EPO alignments

GenomicAlignBlock

Conventions in these slides:

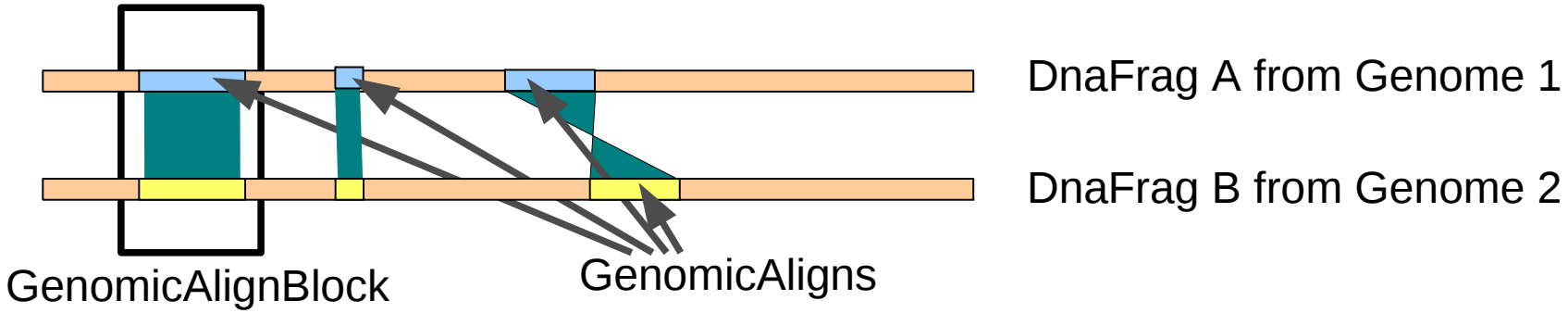
Methods returning values()
Methods returning objects()
Object Adaptor fetching method()

GenomicAlignBlock

- represents a genomic alignment
- contains 1 GenomicAlign per sequence
- **method_link_species_set()**, score(), length(), perc_id(), **get_all_GenomicAligns(), get_SimpleAlign()**
- fetch_all_by_MethodLinkSpeciesSet_Slice()

GenomicAlign

- **dnafrag(), genome_db()**, **get_Slice()**, dnafrag_start, dnafrag_end(), dnafrag_strand(), aligned_sequence()



Alignments are stored in the genomic_align and genomic_align_block tables

For example:

```
gorilla_gorilla/MT/935-953    gacat-ttaactaaaac-ccc
macaca_mulatta/MT/1469-1488   aacatcttaactaaacg-ccc
pan_troglodytes/MT/934-953    gatac-ttaacttaaaccccc
pongo_pygmaeus/MT/940-958     actac-ctaactaaaac-ccc
homo_sapiens/MT/1516-1534     gacat-ttaactaaaac-ccc
                                 *    ***** **    ***
```

```
GACATTTAACTAAAACCCC       5MD11MD3M
AACATCTTAACTAAACGCCC      17MD3M
GATACTTAACTTAAACCCCC      5MD15M
ACTACCTAACTAAAACCCC       5MD11MD3M
GACATTTAACTAAAACCCC       5MD11MD3M
```

5 genomic_align entries
1 genomic_align_block

Sequences from core

cigar lines

Multiple sequence alignments

Mercator-Pecan

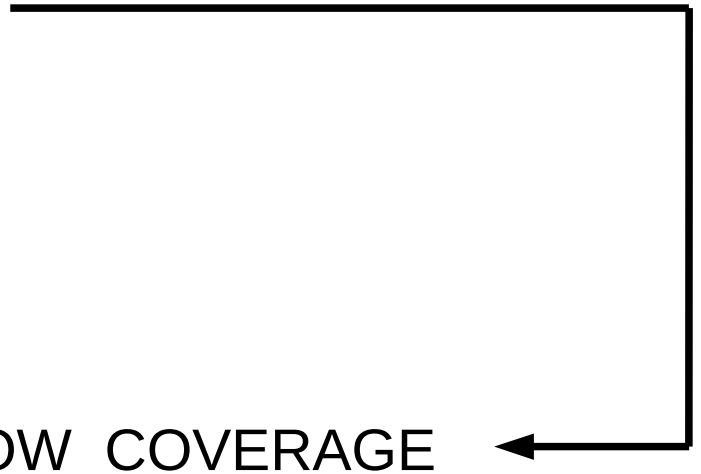
- 19-way amniota vertebrates Pecan

EPO

- 6 primates EPO
- 12 eutherian mammals EPO
- 3 neognath birds EPO
- 5 teleost fish EPO

EPO-2X (EPO_LOW_COVERAGE)

- 35 eutherian mammals EPO_LOW_COVERAGE



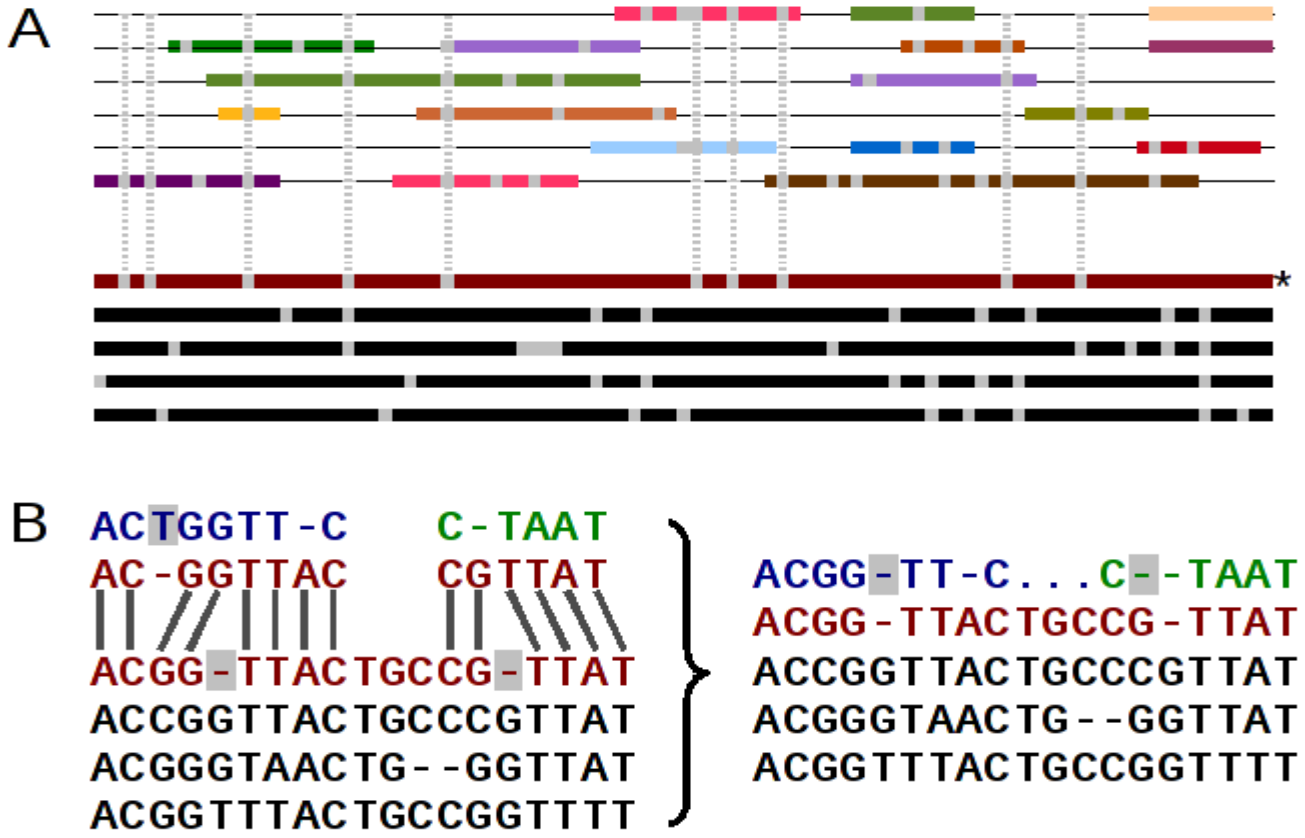
Adding low-coverage (2X) genomes

Low coverage genomes cannot be fully assembled

Resulting assembly is too scattered to be used with Enredo

Run EPO on high-coverage genomes only

Map 2X genomes using pairwise alignments

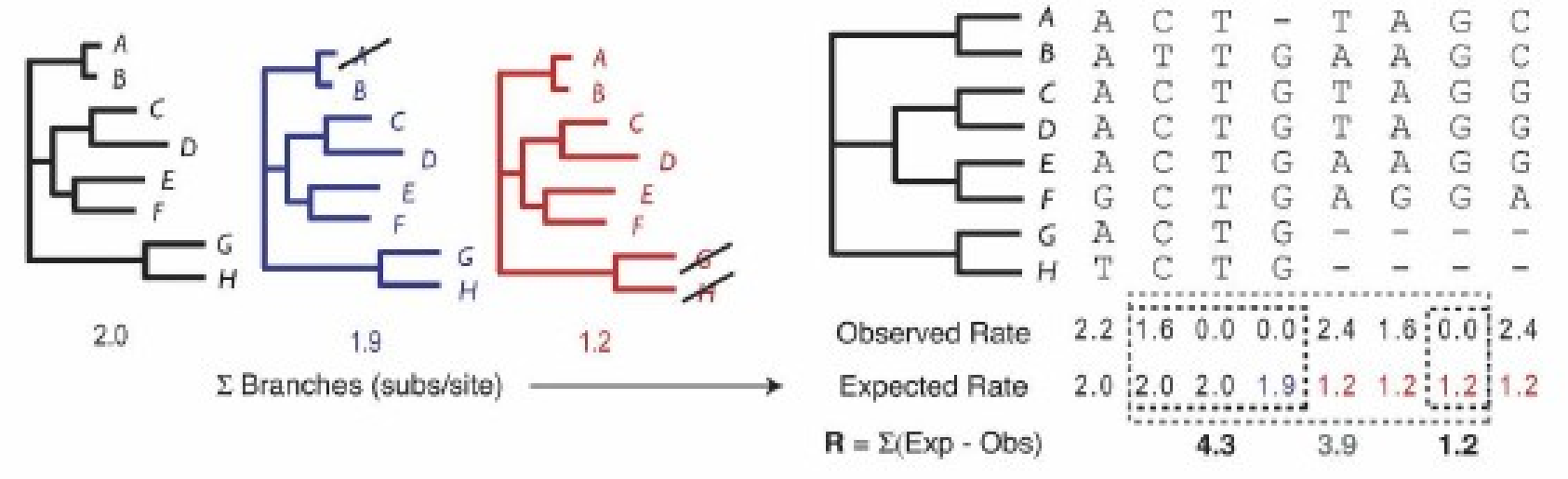


Exercises – GenomicAlignBlock

- A GenomicAlignBlock represents an alignment between two or more pieces of DNA. Every piece of DNA is represented by a GenomicAlign
  - Print the LASTZ_NET alignments for pig chromosome 15 with cow (using pig coordinates 89151597 and 89157190)
  - Change the above example so that it prints the alignments for the 12 eutherian mammals EPO

Gerp Constrained Elements

Stretches of the alignment with a high conservation



Cooper et al. Genome Research, 2005

Constrained elements and coding exons

74% of coding exons are associated with constr. elem.

22% of constr. elem. are associated with coding exons

Constrained elements

Conventions in these slides:

Methods returning values()
Methods returning objects()
Object Adaptor fetching method()

- **ConstrainedElement**: a constrained element
  - slice, start, end, strand, seq_region_start, seq_region_end
  - **get_SimpleAlign**(): this method gets the alignment from the corresponding GenomicAlignBlock and returns a Bio::SimpleAlign object.
  - fetch_all_by_MethodLinkSpeciesSet_Slice, fetch_all_by_MethodLinkSpeciesSet_DnaFrag

Exercises – ConstrainedElements

- A Constrained Element represents regions in the multiple alignment which appear to be under evolutionary constraint.
  - Print the constrained element alignments from the previous pig locus (use the Constrained elements generated from the EPO_LOW_COVERGAGE mammals alignments)

Synteny

Conventions in these slides:

Methods returning values()
Methods returning objects()
Object Adaptor fetching method()

Based on BlastZ-net alignments

- group syntenic alignments closer than 200 kb
- link syntenic groups closer than 3Mb
- minimum length of the syntenic block: 100 kb

SyntenyRegion

- **method_link_species_set()**, **get_all_DnaFragRegions()**
- fetch_all_by_MethodLinkSpeciesSet_Slice(), fetch_all_by_MethodLinkSpeciesSet_DnaFrag()

DnaFragRegion

- **slice(), dnafrag()**, dnafrag_start(), dnafrag_end(), dnafrag_strand()

Synteny example code

```
[...]
my $synteny_region_adaptor = $reg->get_adaptor(
        "Multi", "compara", "SyntenyRegion");

my $synteny_regions = $synteny_region_adaptor->
        fetch_all_by_MethodLinkSpeciesSet_Slice(
        $human_mouse_synteny_method_link_species_set,
        $human_slice);

foreach my $this_synteny_region (@$synteny_regions) {

  my $these_dnafrag_regions =
      $this_synteny_region->get_all_DnaFragRegions();

  foreach my $this_dnafrag_region
      (@$these_dnafrag_regions) {

    print $this_dnafrag_region->dnafrag->
        genome_db->name, ": ",
        $this_dnafrag_region->slice->name, "\n";
  }
  print "\n";
}
```

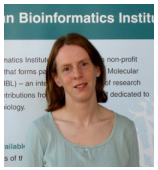
Exercises – Synteny

- Syntenies represent large collinear regions. Although syntenies are inferred from pairwise alignments, details about the alignments are not provided within the synteny
  - Print the pig-cow syntenic map for the pig chr. 15

Acknowledgements

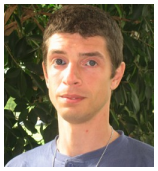
Compara Team

Javier



Kathryn

Stephen



Leo

Matthieu

Miguel

D48–D55 Nucleic Acids Research, 2013, Vol. 41, Database issue *Published online 30 November 2012*
doi:10.1093/nar/gks1236

Ensembl 2013

Paul Flicek[1,2,*], Ikhlak Ahmed[1], M. Ridwan Amode[2], Daniel Barrell[2], Kathryn Beal[1], Simon Brent[2], Denise Carvalho-Silva[1], Peter Clapham[2], Guy Coates[2], Susan Fairley[2], Stephen Fitzgerald[1], Laurent Gil[1], Carlos García-Girón[2], Leo Gordon[1], Thibaut Hourlier[2], Sarah Hunt[1], Thomas Juettemann[1], Andreas K. Kähäri[2], Stephen Keenan[1], Monika Komorowska[1], Eugene Kulesha[1], Ian Longden[1], Thomas Maurel[1], William M. McLaren[1], Matthieu Muffato[1], Rishi Nag[2], Bert Overduin[1], Miguel Pignatelli[1], Bethan Pritchard[2], Emily Pritchard[1], Harpreet Singh Riat[2], Graham R. S. Ritchie[1], Magali Ruffier[1], Michael Schuster[1], Daniel Sheppard[2], Daniel Sobral[1], Kieron Taylor[1], Anja Thormann[1], Stephen Trevanion[2], Simon White[2], Steven P. Wilder[1], Bronwen L. Aken[2], Ewan Birney[1], Fiona Cunningham[1], Ian Dunham[1], Jennifer Harrow[2], Javier Herrero[1], Tim J. P. Hubbard[2], Nathan Johnson[1], Rhoda Kinsella[1], Anne Parker[2], Giulietta Spudich[1], Andy Yates[1], Amonida Zadissa[2] and Stephen M. J. Searle[2]

[1]European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton Cambridge CB10 1SD, UK and [2]Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SA, UK

wellcome trust

EMBL

National Human Genome Research Institute

BBSRC bioscience for the future

European Commission
Framework Programme 7

SEVENTH FRAMEWORK PROGRAMME

GEN2PHEN

BLUEPRINT epigenome

Quantomics
From Sequence to Consequence :
Tools for the Exploitation of Livestock Genomes