

Ensembl Browser Workshop



Aleena Mushtaq Stolworthy

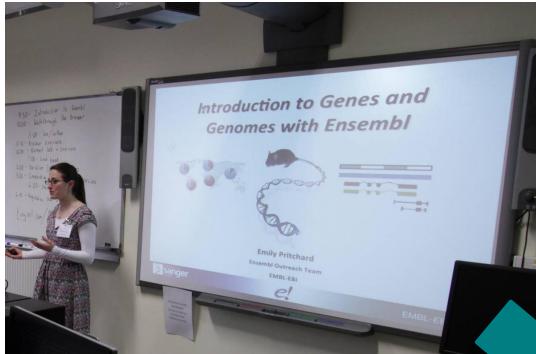
Ensembl Outreach

EMBL-EBI

Course agenda

- 10:00 - 10:45 Introduction to Ensembl**
- 10:45 - 11:30 Region in detail**
- 11:30 - 11:45 *Coffee break***
- 11:45 - 13:00 Genes and transcripts in Ensembl**
- 13:00 - 14:00 *Lunch break***
- 14:00 - 14:45 Exporting Ensembl data with BioMart**
- 14:45 - 15:00 *Coffee break***
- 15:00 - 16:00 Overview of the Ensembl REST API and GET queries**
- 16:00 - 17:00 Scripting around REST API calls**

Structure



Presentation:
What the data/tool is
How we produce/process the data

Demo:
Getting the data
Using the tool



Follow along if you
want to

Exercises:
Trying things out for yourself (alone/pairs?)
Going beyond the demo
Not a test!

Pick and choose which ones best suit your use-case



Course material

<http://training.ensembl.org/events/>

- Presentations
- Demos
- Exercises (with text and video answers)

Objectives

- What is Ensembl?
- What type of data can you get in Ensembl?
- How to navigate the Ensembl browser website.
- Where to go for help and documentation.

Exploring the Ensembl genome browser

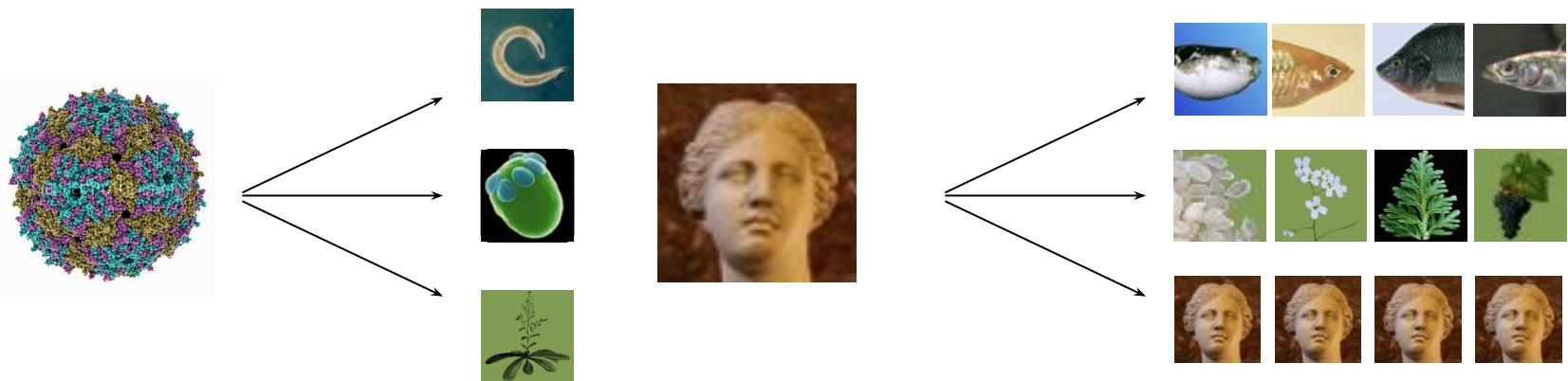


Introduction

Why do we need genome browsers?

1977: 1st genome to be sequenced (5 kb)

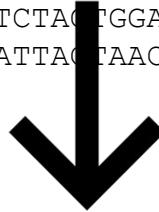
2004: finished human sequence (3 Gb)



CGGCCTTGGCTCCGCTTCAGCTCAAGACTTAACCTCCCTCCCAGCTGTCCCAGATGACGCCATCTGAAATTCTTGAA
ACACGATCACTTAACGGAATATTGCTGTTGGGGAAAGTGTAACTACAGCTGCTGGGCACGCTGTATTGCCTACTTAAGC
CCCTGGTAATTGCTGTATTCCGAAGACATGCTGATGGAAATTACAGGCGCGTGGTCTCTAACTGGAGCCCTCTGTCCCC
ACTAGCCACCGCGTCACTGGTTAGCGTGATTGAAACTAAATCGTATGAAAATCCTCTTAGTCGCACTAGCCACGTTCG
AGTGCTTAATGTGGTAGTGGCACCGGTTGGACAGCACAGCTGAAAATGTTCCCATCCTCACAGTAAGCTGTTACCCTTC
CAGGAGATGGGACTGAATTAGAACAAATTTCAGCGCTTGAGTTACCTCAGTCACATAATAAGGAATGCAT
CCCTGTGTAAGTGCATTTGGTCTCTGTTGCAGACTTACCAAGCATTGGAGGAATATCGTAGGTAAAAATGCCTA
TTGGATCCAAGAGAGGCCAACATTGGAAATTAAAGACACGCTGCAACAAAGCAGGTATTGACAAATTATATAAAC
TTTATAAATTACACCGAGAAAGTGTCTAAAAAATGCTGCTAAAACCCAGTACGTACAGTGTGCTTAGAACCATAA
ACTGTTCTTATGTGTGTTAAATCCAGTTAACACATAATCATCGTTGCAGGTTAACCATGATAAAATAGAACGTCT
AGTGGATAAAGAGGAAACTGGCCCTTGACTAGCAGTAGGAACAATTACTAACAAATCAGAACGATTAATGTTACTTTATGG
CAGAAGTTGCCAACTTTGGTTCACTTACTCCCTATACTCTTAAAATGATCTAGGACCCCCGGAGTGTGCTTGTATG
TAGCTTACCATATTAGAAATTAAACTAAGAACATTAAAGGCTGGCGTGGCTCACGCCGTAAATCCCAGCAGTGG
GCCGAGGTGGCGGATCACTTGAGGCCAGAAGTTGAGACCAGCCTGGCCAACATGGTAAACCCATCTACTAAAAAT
ACAAAAAAATGTGCTGCGTGTGGTGCCTGTAATCCCAGCTACACGGGAGGTGGAGGCAGGAGAACGCTGAACCC
TGGAGGCAGAGGTTGCAGTGAGCCAAGATCATGCCACTGCACTCTAGCCTGGGCCACATAGCATGACTCTGTCTAAAACAA
ACAAACAAACAAAAACTAAGAACATTAAAGTTAATTACTTAAATGAAAGCTAACCCATTGCATATTATCACACAT
TCTTAGGAAAAATAACTTTGAAAACAAGTGAGTGAATAGTTTACATTTCAGTTCTCTTTAATGTCTGGCTAAAT
AGAGATAGCTGGATTCACTTATCTGTGCTAATCTGTTATTGGTAGAAGTATGTGAAAAAAATTAACCTCACGTTGAAA
AAAGGAATATTAAATAGTTCACTTGTACTTTGGTATTTCCTGTACTTGCATAGATTTCAAAGATCTAATAGAT
ATACCATAGGTCTTCCATGCGAACATCATGCAGTGATTATTGGAGATAGTGGTGTCTGAATTATAACAAAGTTCC
AAATATTGATAAAATTGCATTAAACTATTAAATCTCATTCAATTACCAACATGGATGTCAGAAAAGTCTTTAAGAT
TGGGTAGAAATGAGCCACTGGAAATTCTAATTTCATTGAAAGTTCACATTGACAAACAAACTGTTCTTGC
AGCAACAAGATCACTCATTGATTGTGAGAAAATGTCTACCAAATTATTAAAGTTGAAATAACTTGTCA
AAGTAAAAATGACTTTCAATTGAAAAAAATTGCTGTTCACTACAGCTCAACATGAGTGCTTTCTAGGCAGTATTGTACT
TCAGTATGCAGAAGTGCTTATGTATGCTCCTATTGTCAGAGATTATTAAAGAAGTGCTAAAGCATTGAGCTCGAAA
TTAATTTCATTGCTCATTAGGACATTCTACATTAAACTGGCATTATTACTATTATTAAACAAGGACACTCAGTG
GTAAGGAATATAATGGCTACTAGTATTAGTTGGTGCCTGCAACTCATGCAAATGTGCCAGCAGTTACCCAGCAT
CATCTTGCACTGTTGATACAAATGTCAACATCATGAAAAAGGGTTGAAAAAAAGGAATATTAAATAGTTCTAGTTACTTT

What is Ensembl?

```
AGTGCTTAATGTGGCTAGTGGCACCGGTTGGACAGCACAGCTGTAA  
AATGTTCCCATTCTCACAGTAAGCTGTTACCGTTCCAGGAGATGGGA  
CTGAATTAGAACAAACAAATTTCAGCGCTCTGAGTTTACCT  
CAGTCACATAATAAGGAATGCATCCCTGTGTAAGTGCATTTGGTCT  
TCTGTTTGCAGACTTACCAAGCATTGGAGGAATATCGTAGGT  
AAAAATGCCTATTGGATCCAAGAGAGGCCAACATTGGAAATT  
TTAAGACACGCTGCAACAAAGCAGGTATTGACAATTATATAACT  
TTATAAATTACACCGAGAAAGTGTCTAAAAATGCTGCTAAAA  
ACCCAGTACGTACAGTGTGCTTAGAACCATAAACTGTTCTTATG  
TGTGTATAAATCCAGTTAACACATAATCATCGTTGCAGGTTAAC  
ACATGATAAATATAGAACGTCTACTGGATAAAGAGGAAACTGGCCCC  
TTGACTAGCAGTAGGAACAAATTAACTAACAAATCAGAACGATTAATGT
```



Ensembl annotates and maps genomic features from genome sequences

What is Ensembl?

```
AGTGCTTAATGTGGCTAGTGGCACCGGTTGGACAGCACAGCTGTAA  
AATGTTCCCATCCTCACAGTAAGCTGTTACCGTTCCAGGAGATGGGA  
CTGAATTAGAACAAACAAATTTCAGCGCTCTGAGTTTACCT  
CAGTCACATAATAAGGAATGCATCCCTGTGTAAGTGCATTTGGTCT  
TCTGTTTGCAGACTTACCAAGCATTGGAGGAATATCGTAGGT  
AAAAATGCCTATTGGATCCAAGAGAGGCCAACATTGGAAATT  
TTAAGACACGCTGCAACAAAGCAGGTATTGACAATTATATAACT  
TTATAAATTACACCGAGAAAGTGTCTAAAAATGCTGCTAAAA  
ACCCAGTACGTACAGTGTGCTTAGAACCATAAACTGTTCTTATG  
TGTGTATAAATCCAGTTAACACATAATCATCGTTGCAGGTTAAC  
ACATGATAAATATAGAACGTCTACTGGATAAAGAGGAAACTGGCCCC  
TTGACTAGCAGTAGGAACAAATTACAAACAAATCAGAACGATTAATGT
```



Ensembl is an ‘added value resource’ bringing together information from a wide range of other databases in a single site

What is Ensembl?

```
AGTGCTTAATGTGGCTAGTGGCACCGGTTGGACAGCACAGCTGTAA  
AATGTTCCCATTCTCACAGTAAGCTGTTACCGTTCCAGGAGATGGGA  
CTGAATTAGAACAAACAAATTTCAGCGCTCTGAGTTTACCT  
CAGTCACATAATAAGGAATGCATCCCTGTGTAAGTGCATTTGGTCT  
TCTGTTTGAGACTTACCAAGCATTGGAGGAATATCGTAGGT  
AAAAATGCCTATTGGATCCAAGAGAGGCCAACATTTGAAATT  
TTAAGACACGCTGCAACAAAGCAGGTATTGACAATTATATAACT  
TTATAAATTACACCGAGAAAGTGTCTAAAAATGCTGCTAAAA  
ACCCAGTACGTACAGTGTGCTTAGAACCATAAACTGTTCTTATG  
TGTGTATAAATCCAGTTAACACATAATCATCGTTGCAGGTTAAC  
ACATGATAAATATAGAACGTCTACTGGATAAAGAGGAAACTGGCCCC  
TTGACTAGCAGTAGGAACAAATTACAAACAAATCAGAACGATTAATGT
```



Genome Data Viewer

[www.ncbi.nlm.nih.gov/
genome/gdv/](http://www.ncbi.nlm.nih.gov/genome/gdv/)



www.ensembl.org



www.genome.ucsc.edu

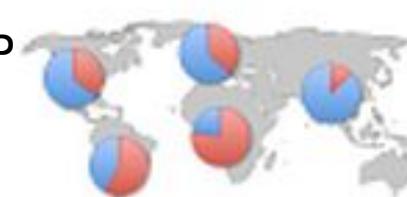
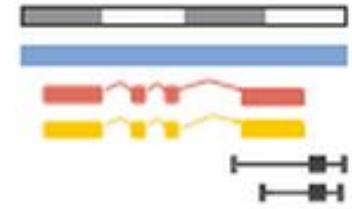
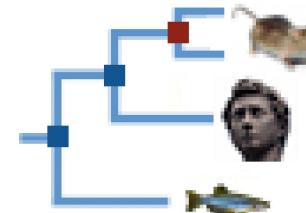


www.ensemblgenomes.org

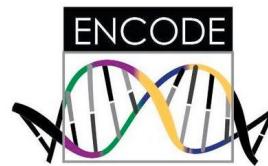
g

Ensembl features

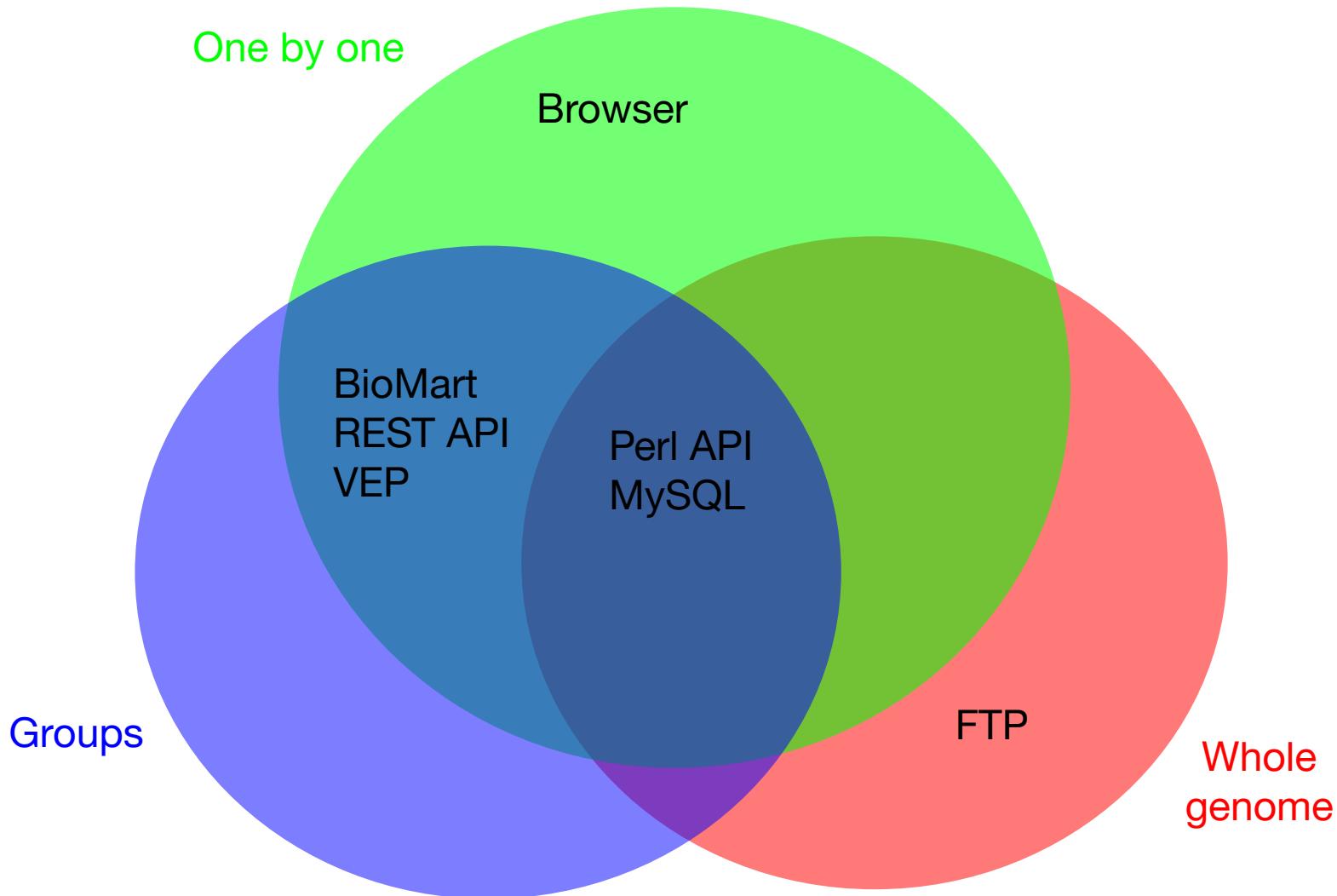
- Genomes and gene builds for >300 species
- Variation data
- Compara (alignments, gene trees, homologues)
- Regulatory build
- BioMart (data export)
- Tools for data processing, e.g. VEP
- Display your own data
- Programmatic access via APIs
- Completely Open Source (FTP, GitHub)



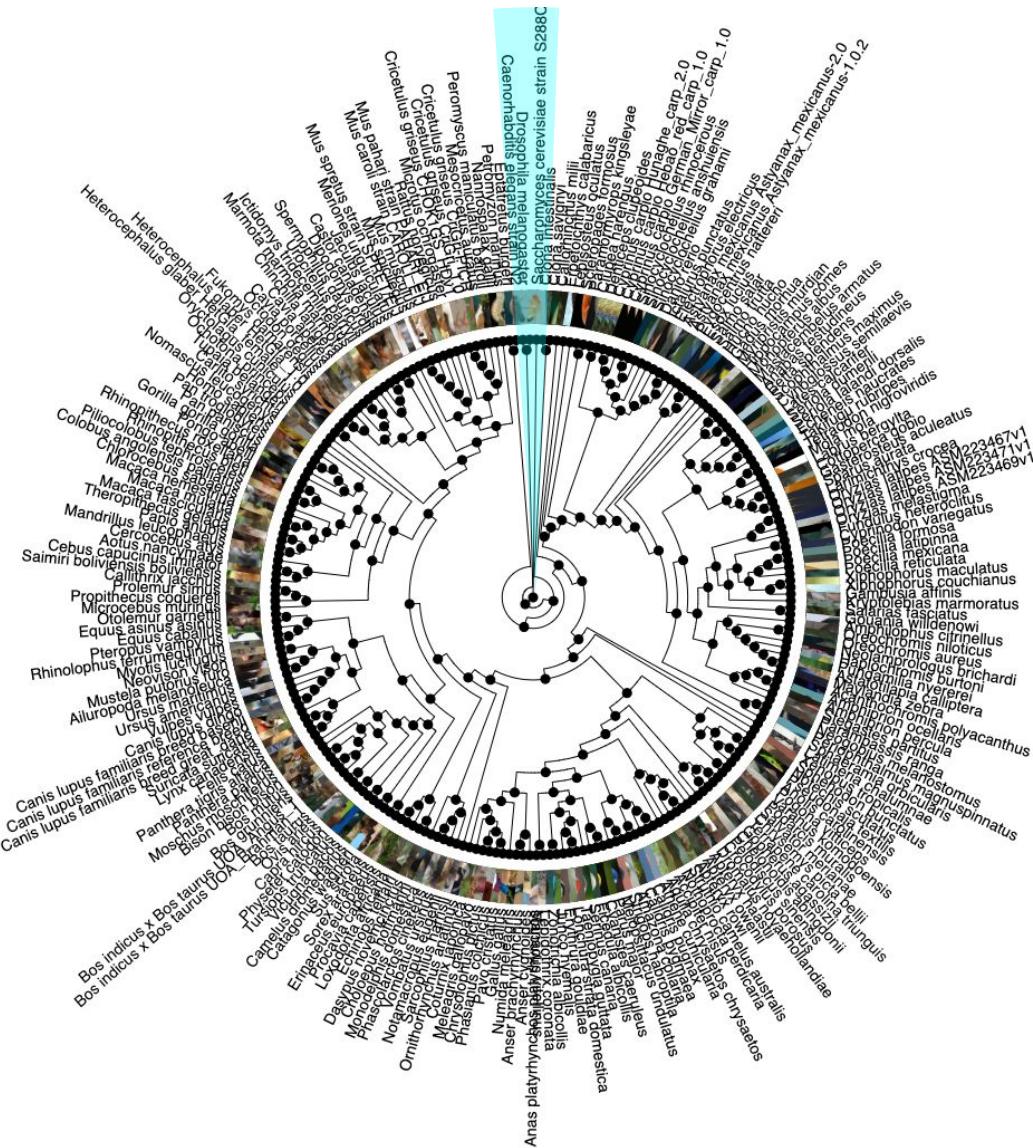
Ve!P



Access scales



Vertebrate species in Ensembl



Non-vertebrates on Ensembl genomes

[EnsemblBacteria](#) ▾ BLAST | More ▾ [Search Ensembl Bacteria species...](#)

Help & Documentation Species List

Find a Species

Ensembl Bacteria Species

Bacillus collection

78 genomes

<i>Bacillus amyloliqufaciens</i> European Nucleotide Archive	<i>Bacillus anthracis A0248</i> European Nucleotide Archive	<i>Bacillus anthracis Ames</i> European Nucleotide Archive
<i>Bacillus anthracis Ames ancestor</i> European Nucleotide Archive	<i>Bacillus anthracis CDC 684</i> European Nucleotide Archive	<i>Bacillus anthracis Sterne</i> European Nucleotide Archive
<i>Bacillus cereus 03BB102</i> European Nucleotide Archive	<i>Bacillus cereus 172560W</i> European Nucleotide Archive	<i>Bacillus cereus 95/8201</i> European Nucleotide Archive
<i>Bacillus cereus AH1271</i> European Nucleotide Archive	<i>Bacillus cereus AH1272</i> European Nucleotide Archive	<i>Bacillus cereus AH1273</i> European Nucleotide Archive
<i>Bacillus cereus AH187</i> European Nucleotide Archive	<i>Bacillus cereus AH603</i> European Nucleotide Archive	<i>Bacillus cereus ATCC 10876</i> European Nucleotide Archive
<i>Bacillus cereus ATCC 10987</i> European Nucleotide Archive	<i>Bacillus cereus ATCC 14579</i> European Nucleotide Archive	<i>Bacillus cereus ATCC 4342</i> European Nucleotide Archive
<i>Bacillus cereus B4264</i> European Nucleotide Archive	<i>Bacillus cereus BORD-Bc04</i> European Nucleotide Archive	<i>Bacillus cereus BORD-ST196</i> European Nucleotide Archive
<i>Bacillus cereus BORD-ST24</i> European Nucleotide Archive	<i>Bacillus cereus F68158</i> European Nucleotide Archive	<i>Bacillus cereus MM3</i>

Bacteria

[EnsemblProtists](#) ▾ BLAST | More ▾ [Search Ensembl Protists species...](#)

Help & Documentation Species List

Find a Species

Ensembl Protists Species

Alveolata

<i>Plasmodium berghei</i> GenoDB Plasmodium berghei ANKA	<i>Plasmodium knowlesi</i> Wellcome Trust Sanger Institute Plasmodium knowlesi	<i>Toxoplasma gondii</i> ToxoDB Toxoplasma gondii
<i>Plasmodium chabaudi</i> GenoDB Plasmodium chabaudi	<i>Plasmodium vivax</i> The Institute for Genomic Research Plasmodium vivax	
<i>Plasmodium falciparum</i> GenoDB Plasmodium falciparum 3D7	<i>Tetrahymena thermophila</i> The Institute for Genomic Research Tetrahymena thermophila SB210	

Amoebozoa

<i>Dictyostelium discoideum</i> DictyBase Dictyostelium discoideum	<i>Entamoeba histolytica</i> AmoebaDB Entamoeba histolytica HM-1:IMSS	
---	--	--

Stramenopiles

<i>Albugo laibachii</i> The Sainsbury Laboratory Albugo laibachii Nc14	<i>Phytophthora infestans</i> BROAD Phytophthora infestans	<i>Pythium ultimum</i> Pythium Genome Database Pythium ultimum
---	---	---

Protists

[EnsemblFungi](#) ▾ BLAST | More ▾ [Search Ensembl Fungi species...](#)

Help & Documentation Species List

Find a Species

Ensembl Fungi Species

Coprinidae

<i>Mycosphaerella graminicola</i> JGI Mycosphaerella graminicola IPO323
--

Euotiales

<i>Aspergillus clavatus</i> CADRE Aspergillus clavatus	<i>Aspergillus fumigatus</i> CADRE Aspergillus fumigatus A1163	<i>Aspergillus oryzae</i> CADRE Aspergillus oryzae
<i>Aspergillus flavus</i> CADRE Aspergillus flavus	<i>Aspergillus nidulans</i> CADRE Aspergillus nidulans FGSC A4	<i>Aspergillus terreus</i> CADRE Aspergillus terreus
<i>Aspergillus fumigatus</i> CADRE Aspergillus fumigatus A1393	<i>Aspergillus niger</i> CADRE Aspergillus niger CBS 513.88	<i>Neosartorya fischeri</i> CADRE Neosartorya fischeri

Hypocreales

<i>Fusarium oxysporum</i> Broad Institute Fusarium oxysporum 4287	<i>Gibberella zeae</i> Broad Institute Gibberella zeae PH-1	<i>Trichoderma virens</i> JGI Trichoderma virens C8
--	--	--

Fungi

[EnsemblMetazoa](#) ▾ BLAST | More ▾ [Search Ensembl Metazoa species...](#)

Help & Documentation Species List

Find a Species

Ensembl Metazoa Species

Diptera

Metazoa

[EnsemblPlants](#) ▾ BLAST | More ▾ [Search Ensembl Plants species...](#)

Help & Documentation Species List

Find a Species

Ensembl Plants Species

Liliopsida

eudicotyledons

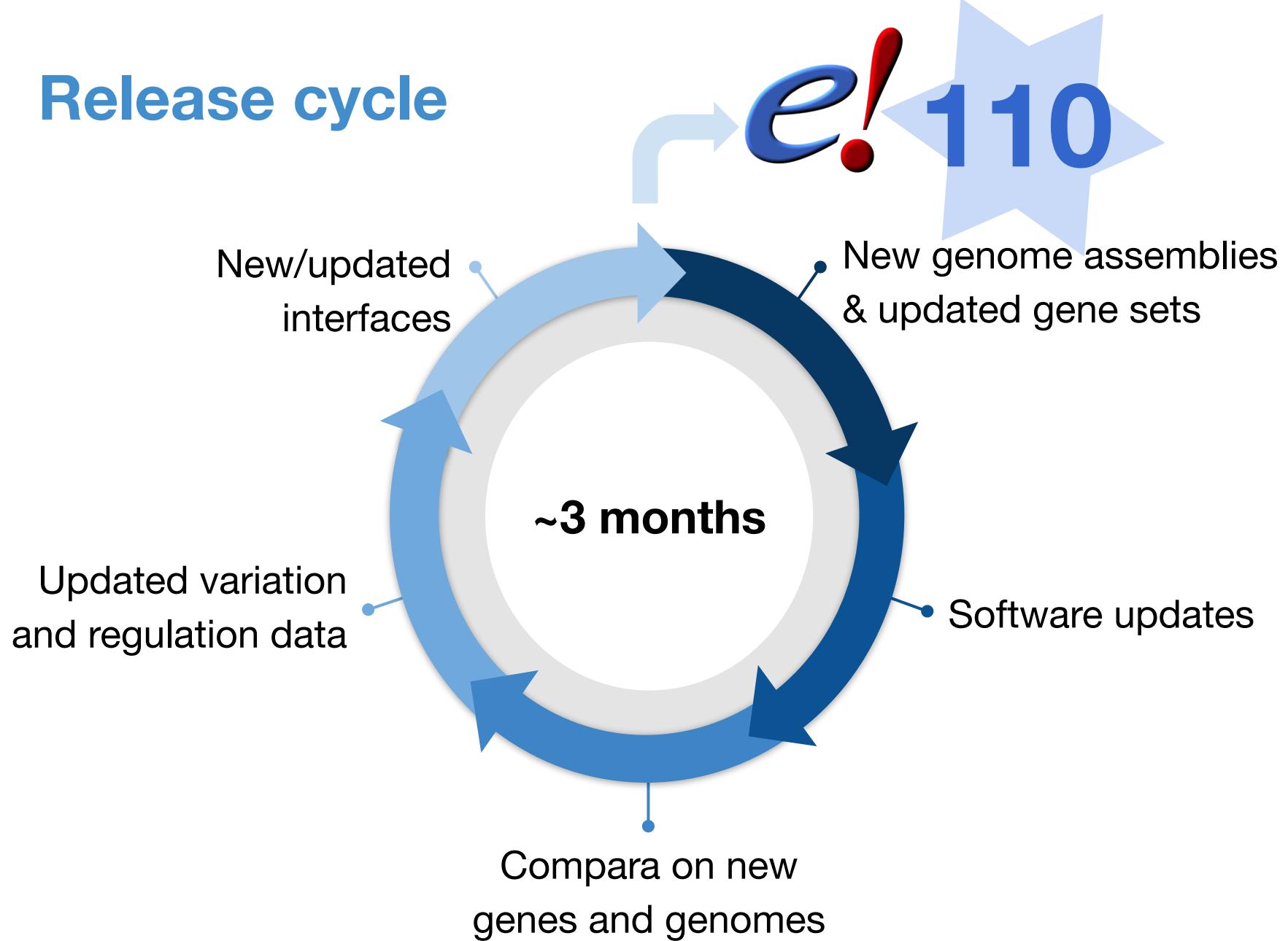
Plants



Ensembl and Ensembl Genomes

	Ensembl	Ensembl Genomes
Released	2000	2009
Species	Vertebrates (fly, worm and yeast as outgroups)	Non-vertebrates (protists, plants, fungi, metazoa, bacteria)
Annotation	by Ensembl	in collaboration with the scientific communities
URL	www.ensembl.org	www.ensemblgenomes.org

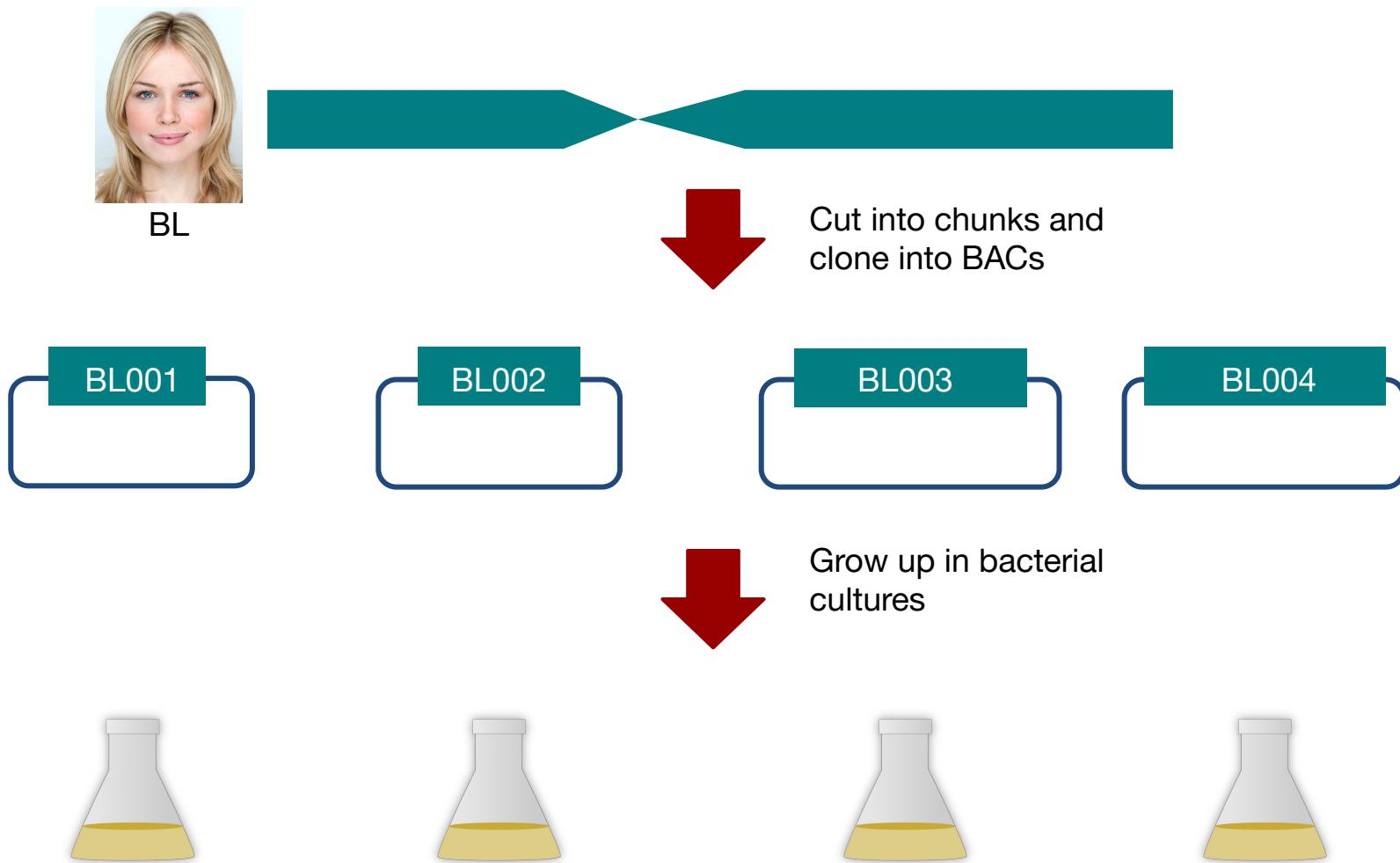
Release cycle



Ensembl Rapid Release

- Released every two weeks ✓
- Genome with gene annotation only ✓
- BLAST ✓
- Homology predictions ✓
- No BioMart ✗
- No variation ✗

Cloning into BACs



Making a contig

Sequence reads

CGGCCTTGCGCTTCAGCTCAAGA

CAGCTGTCCCAGATGAC ACTTAACCTCCCTCCCCAGCTGTCC

GGGCTCCGCCTTCAGCTC

AAC TTCCCTCCCAGCT TCCCCAGCTGTCCCCAGATGACGCCATC

CGGCCTTGGGCTCC

CAGATGACGCC TCCGCCTTCAGCTCAAGACTTAACCTTC

Match up overlaps

CGGCCTTGCTCCGCCTCAGCTCAAGA AACTTCCCTCCCAGCT CAGATGACGCC
TCCGCCTTCAGCTCAAGACTTAACCTC TCCCAGCTGTCCCAGATGACGCCATC
GGGCTCCGCCTTCAGCTC ACTTAACTTCCCTCCCAGCTGTCC
CGGCCTTGCTCC CAGCTGTCCCAGATGAC

Contig

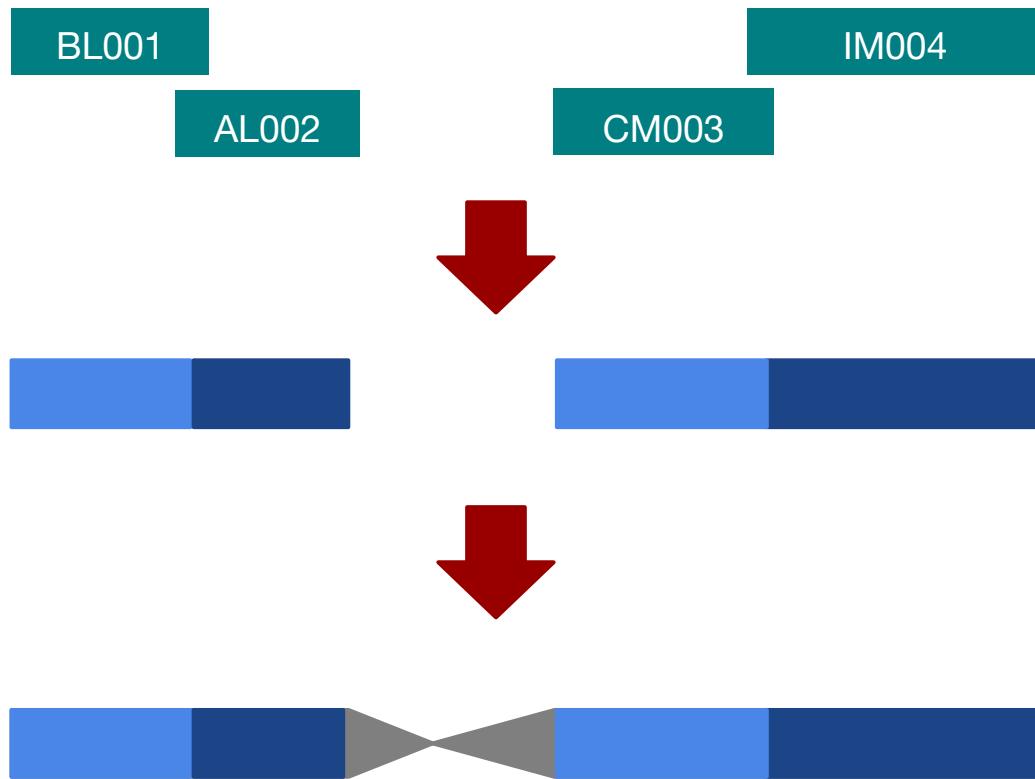
CGGCCTTTGGGCTCCGCTTCAGCTAAAGACTTAACTTCCTCAGCTGTCCCCAGATGACGCCATG

Contigs to scaffolds

BACs from different individuals assembled together with overlaps
Tilepath

Overlaps trimmed to give **contigs**. A run of contigs with no gaps is a **scaffold**.

Genetic maps are used to assemble scaffolds into a chromosome

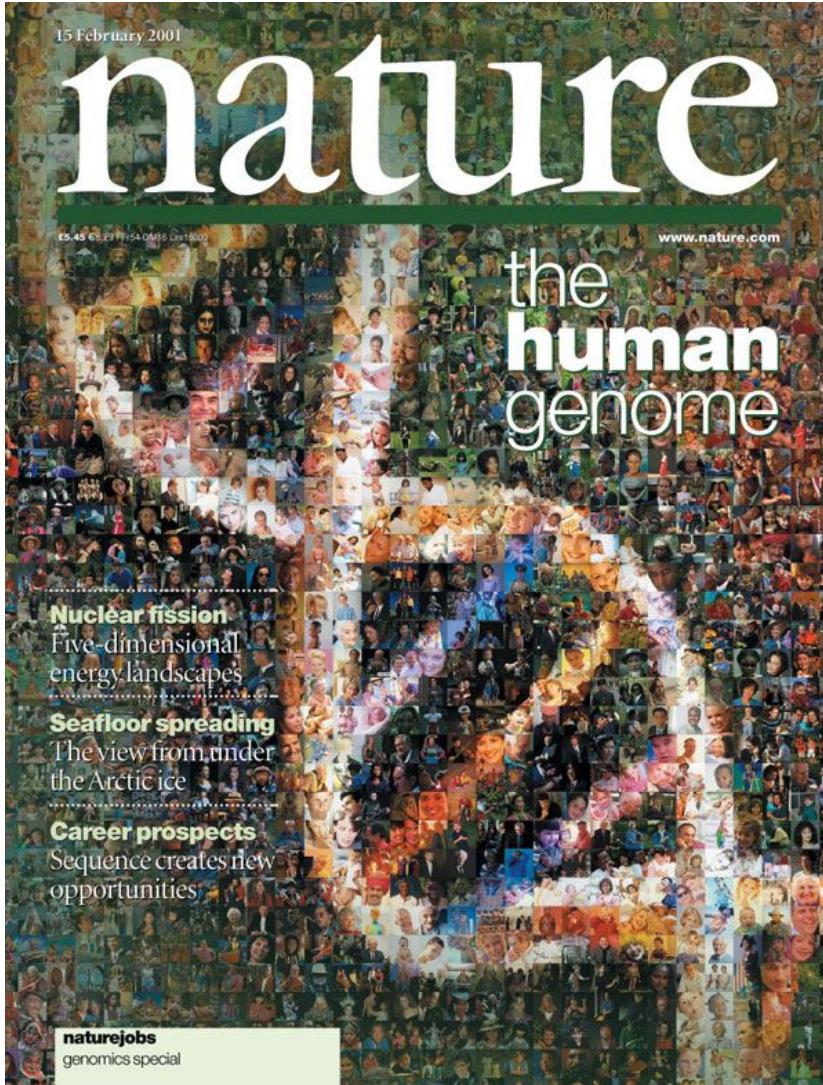


Tilepath

Contig/
scaffold

Chromosome

Genome assemblies



BL



AL

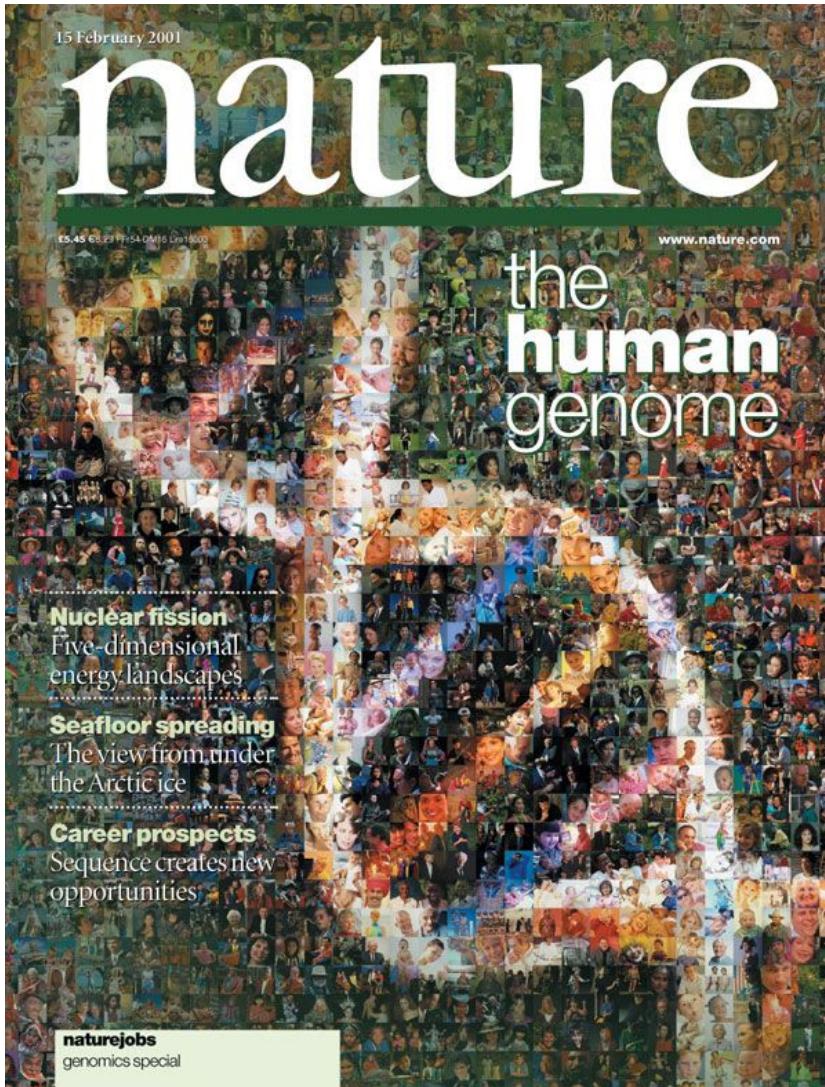


CM



IM

Genome contigs



BL



AL



CM



IM

BL001

AL002

CM003

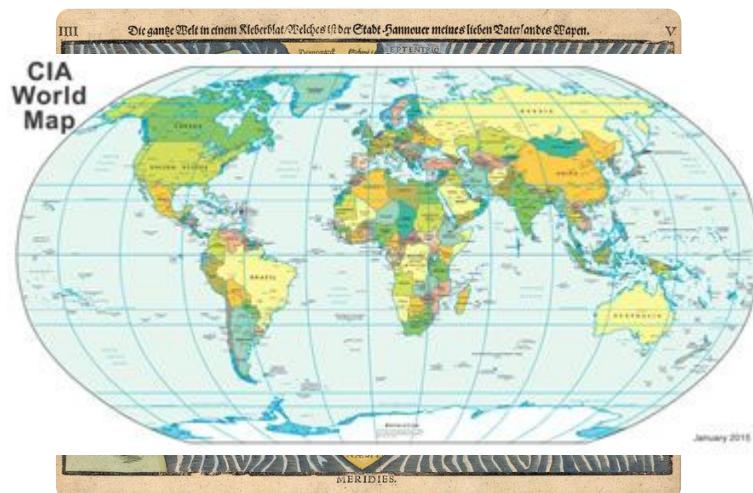
IM004

Genome assemblies

Genome
“DNA within a cell”



Genome assembly
Representation of a genome
Contains errors and gaps
Coordinate system



Human genome assemblies

- GRCh38 (aka hg38)
• Many rare/private alleles replaced.
• www.ensembl.org
• Most up-to-date and supported
- GRCh37 (aka hg19)
• Some large gaps
• grch37.ensembl.org
• Limited data and software updates
• Still the preferred genome of the clinical community
- NCBI36 (aka hg18)
• Many gaps
• ncbi36.ensembl.org
• No longer updated



Hands on

We're going to look at the Ensembl homepage and how to find information about the species and genome assemblies in Ensembl.

Hands on

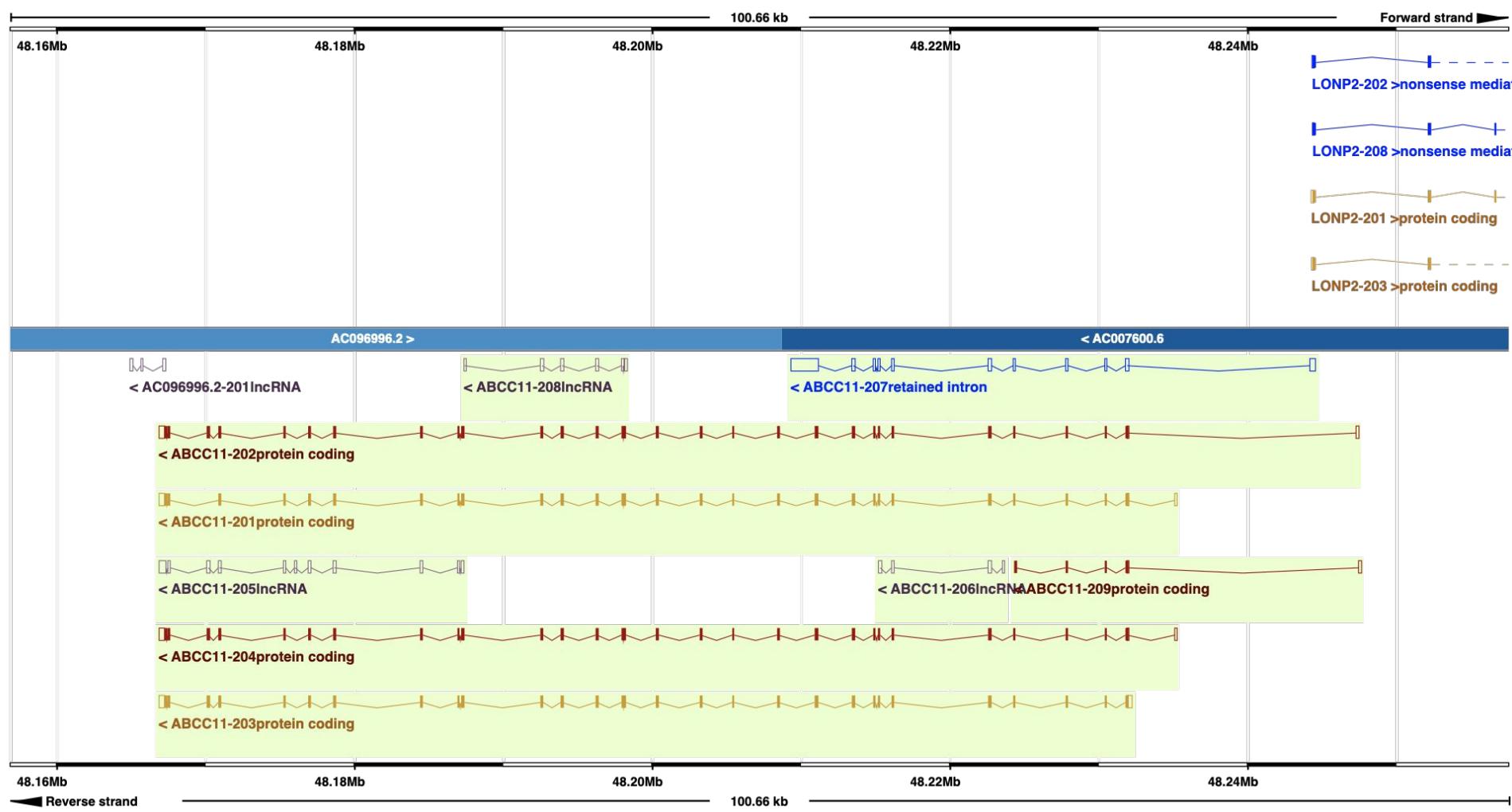
We're going to look at a region of the human genome,
[4:122868000-122946000](#), and manipulate the view to see
the data we're interested in.

Genes and transcripts in Ensembl

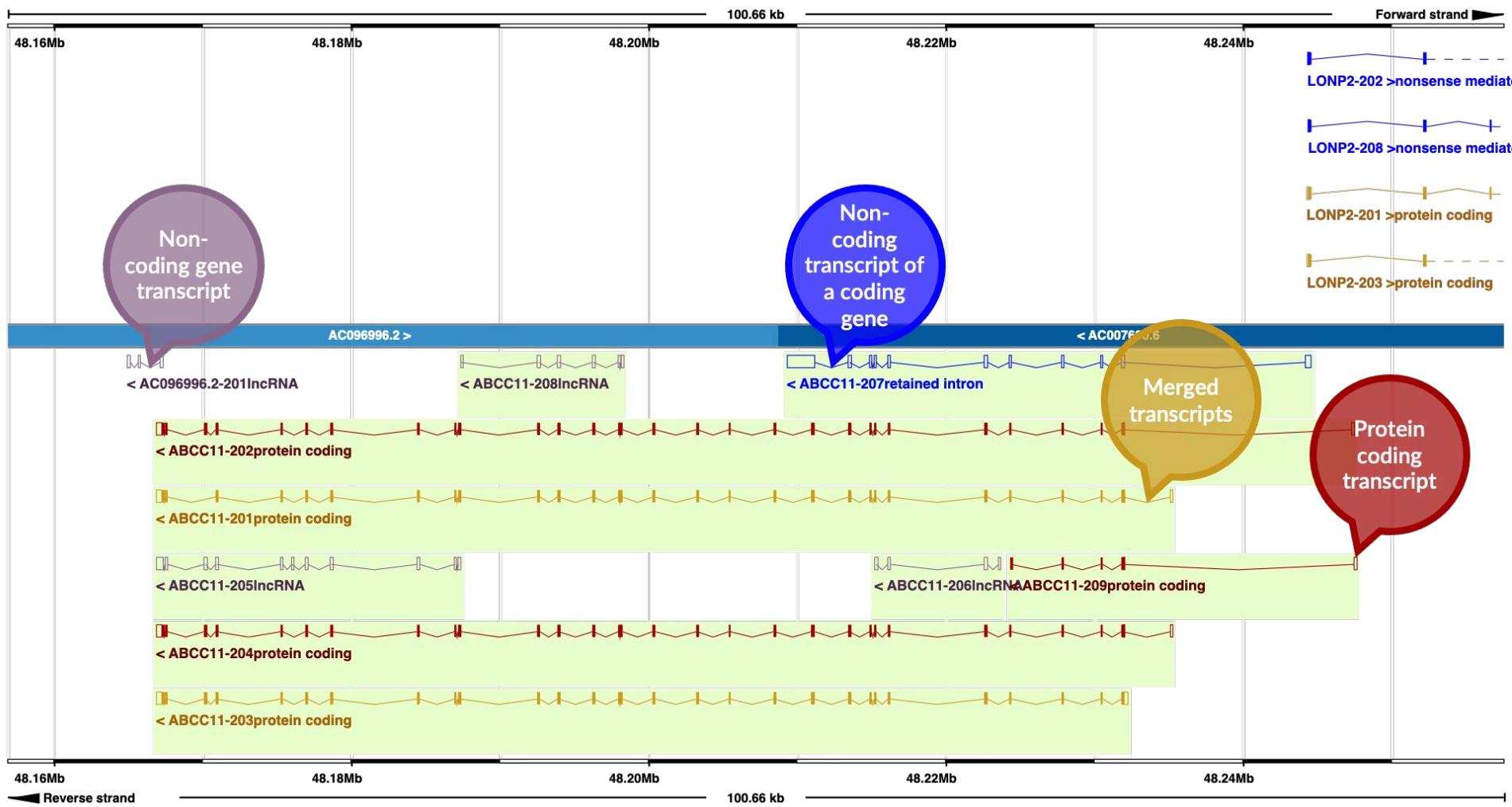
e!



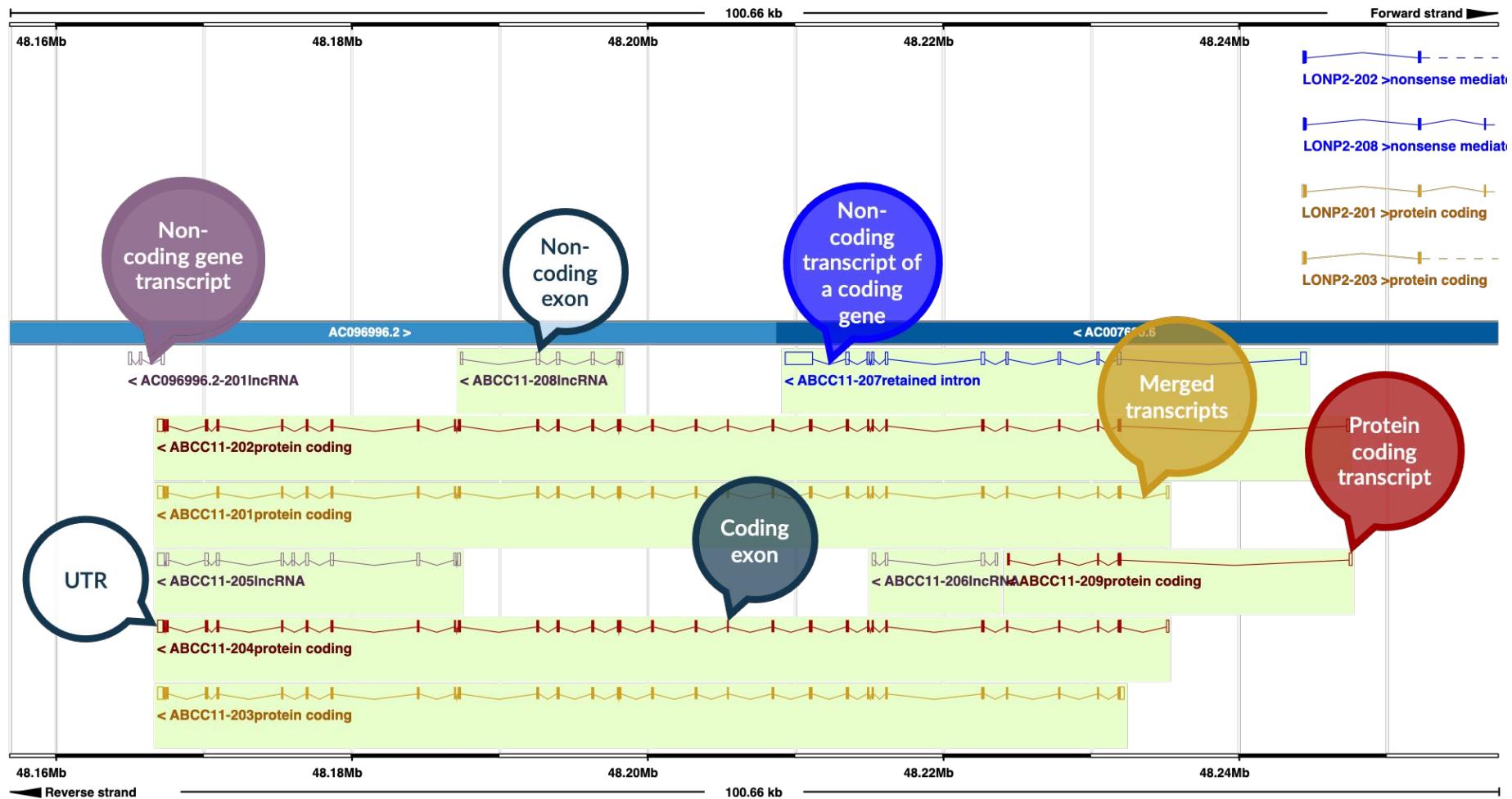
Gene view



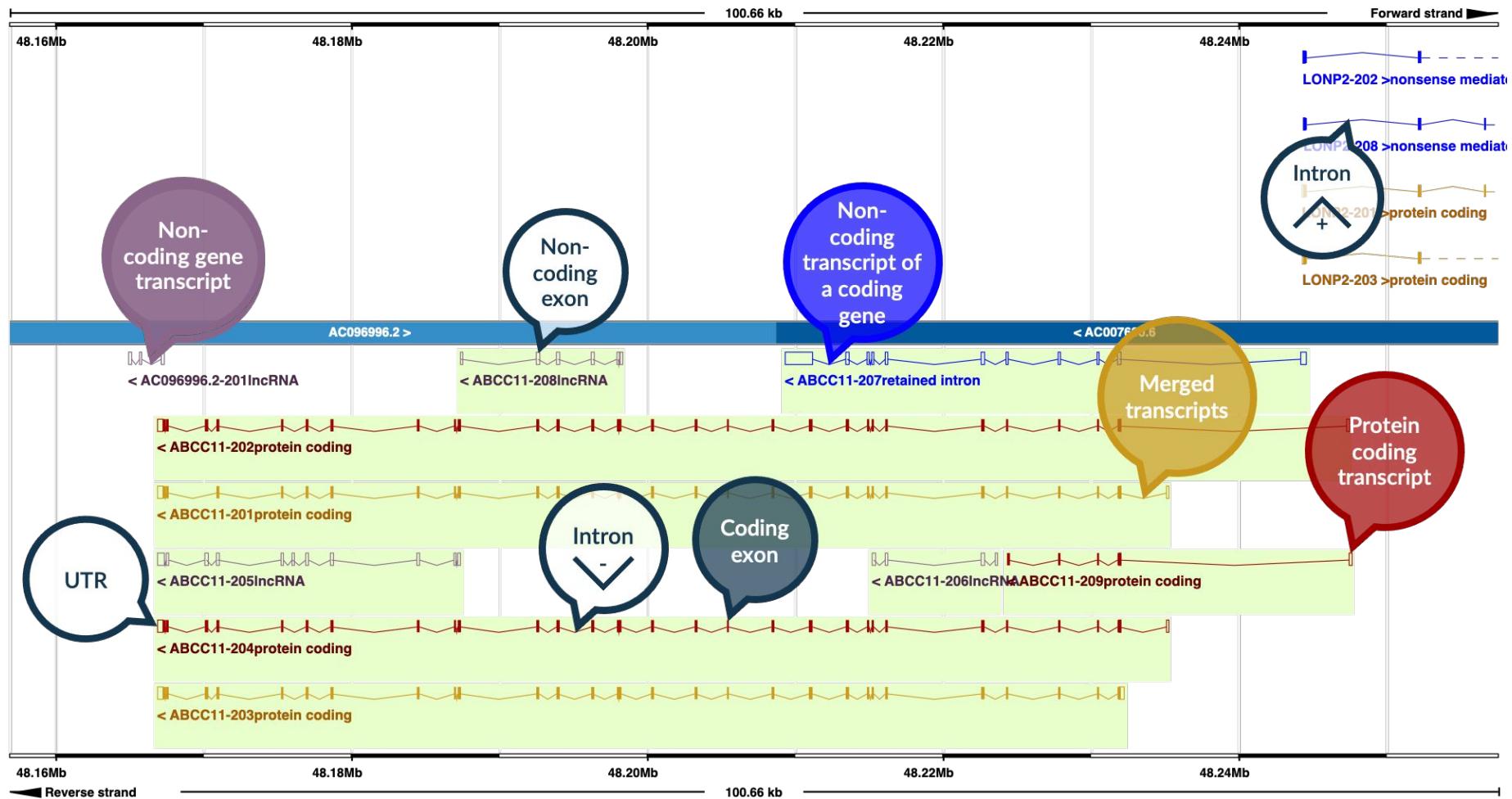
Gene view



Gene view



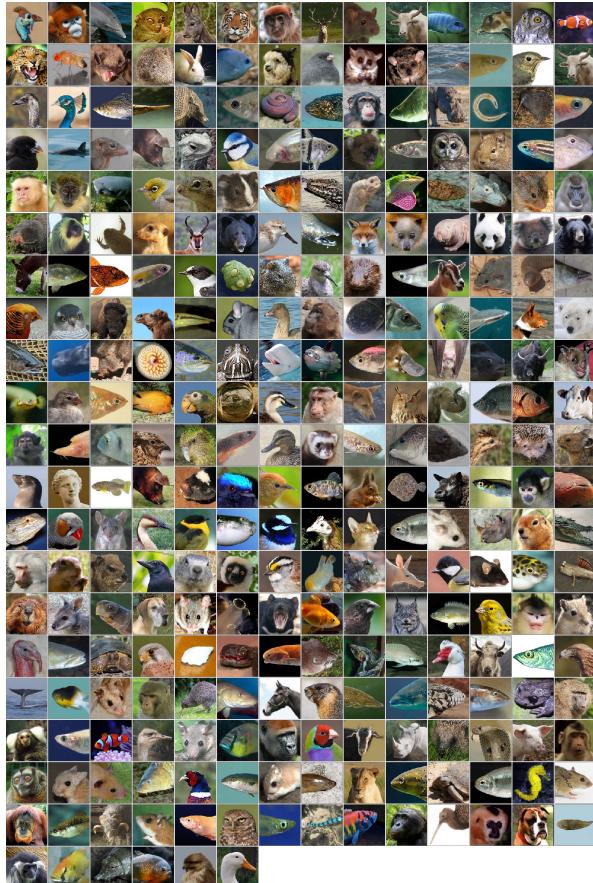
Gene view



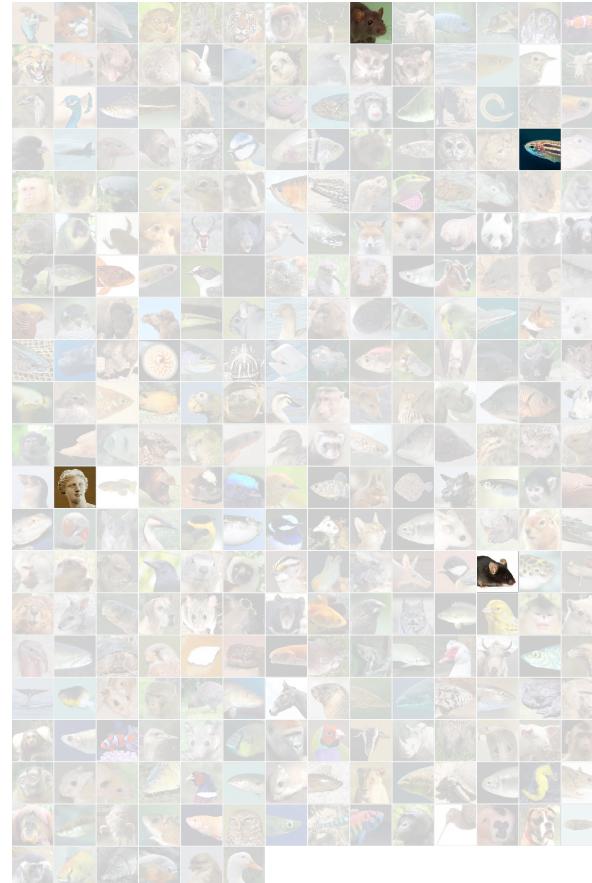
Automatic and manual annotation



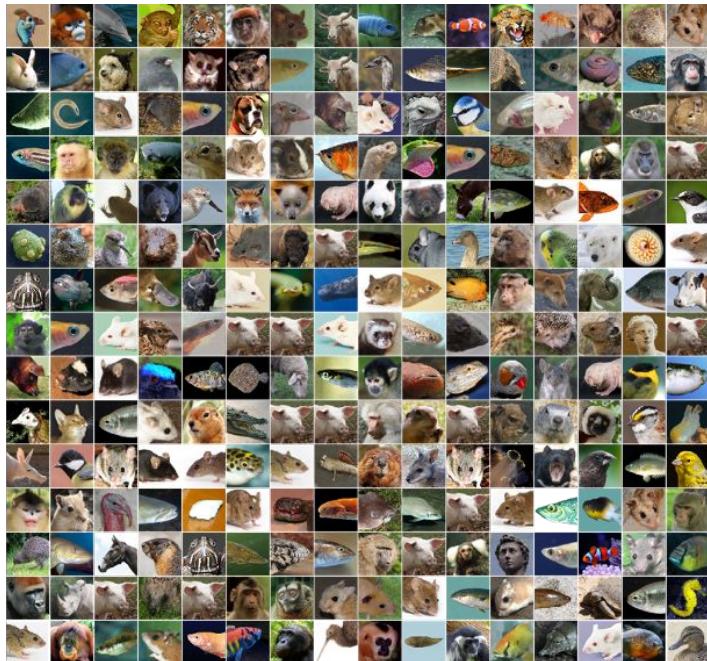
Automatic annotation



Manual annotation



Automatic annotation



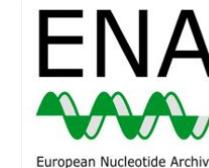
- Genome-wide determination using **automated pipeline**
- Predictions based on **experimental data**: known proteins, cDNAs plotted onto the genome using sequence matching
- One genome in two weeks
- Not ab initio ORFs prediction!

Biological evidence

Automatic annotation

- **INSDC** (International Nucleotide Sequence Database Collaboration)

- cDNAs
- ESTs
- RNA-seq



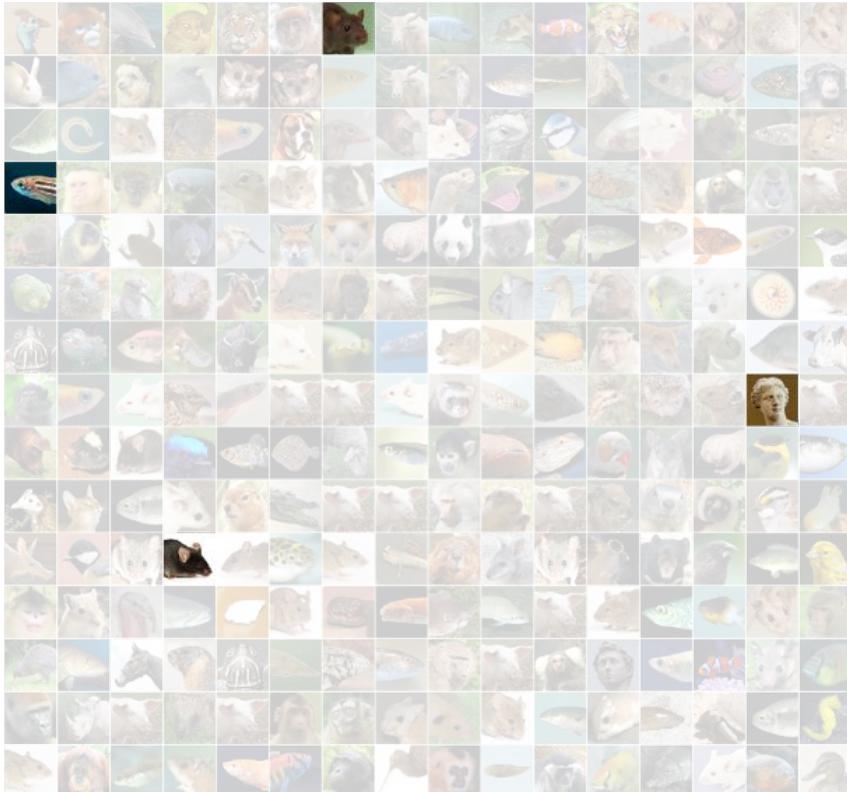
- **Protein sequence databases**

- Swiss-Prot: manually curated
- TrEMBL: unreviewed translations



- **Homology based inference:** e.g. predicting genes in by mapping cDNAs/proteins from to the genome

Manual annotation



- Genome-wide annotation
- Gene determination on a **case-by-case** basis by a person (**annotator**)
- Uses data from databases and papers
- One gene in half a day
- One genome in several years

Manual annotation

- Genome-wide annotation
- Gene determination on a case-by-case basis by a person (annotator)
- Uses data from databases and papers
- One gene in half a day
- One genome in several years
- RNA-seq transcriptome data
 - Illumina short reads
 - Long reads (PacBio and Oxford Nanopore)
- Transcript structure data
 - Intron spanning reads
 - CAGE transcription start sites
 - PolyA-Seq transcription ends
- Mass-Spec protein data
- Publications

Manual vs automatic

- Manual annotation is more comprehensive
 - More transcripts per gene, especially non-coding transcripts
 - More genes overall, especially non-coding genes
 - More biotypes
- Requires less evidence
- More accurate for difficult to annotate features:
 - UTRs
 - Splice sites
 - Single exon transcripts
 - Exceptions, such as immunoglobulins, stop codon readthroughs

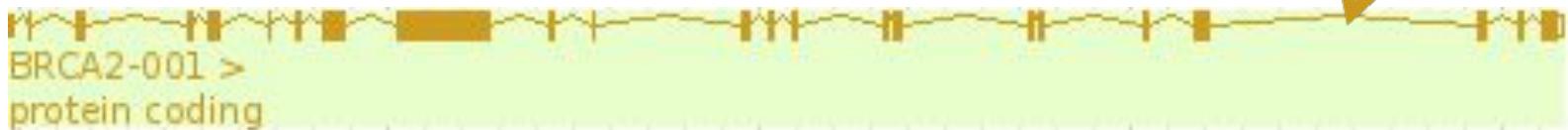
Golden transcripts

- Identical annotation



- Higher confidence and quality

Merged transcripts

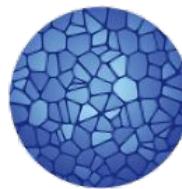




The GENCODE gene set is the merged set of Ensembl automatically annotated genes and Havana manually annotated genes



GENCODE is the default gene set used by major projects such as:



HUMAN
CELL
ATLAS

1000 Genomes Project



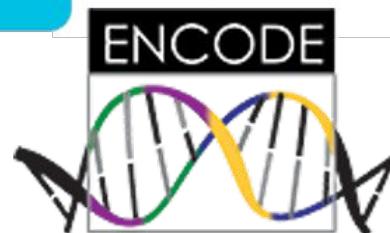
International
Cancer Genome
Consortium
for Medicine



THE CANCER GENOME ATLAS



ENCODE



Genomics
england



GTEx

Canonical transcript

Single, representative transcript identified at every locus

Selection for human coding genes:

- Highly conserved CDS (PhyloCSF)
- Highly expressed (RNAseq reads spanning introns, CAGE)
- Concordant with APPRIS P1 isoform
- Concordant with UniProt canonical isoform
- Longest CDS
- Covers the largest number of clinical variants
- Complete

MANE Select

MANE:Matched Annotation from the NCBI and EMBL-EBI

- MANE Select: one well-supported transcript for every protein-coding gene, agreed to be the most biologically relevant of that gene (based on expression, conservation and clinical variation)
- 100% identity (TSS, 5' UTR, CDS, 3' UTR, 3' end) between RefSeq (NM) and Ensembl (ENST) transcript

Where available

Ensembl Canonical = MANE Select

A single transcript chosen for a gene which is the **most conserved**, **most highly expressed**, has the **longest coding sequence** and is **represented in other key resources**, such as NCBI and UniProt.

Show/hide columns (1 hidden)									Filter	Export
Name	Transcript ID	bp	Protein	Biotype	CCDS	UniProt Match	RefSeq Match	Flags		
BRCA2-201	ENST00000380152.8	11954	3418aa	Protein coding	CCDS9344	P51587	NM_000059.4	MANE Select v0.93	Ensembl Canonical	GENCODE basic
BRCA2-210	ENST00000680887.1	11880	3418aa	Protein coding	CCDS9344	-	-		APPRIS P1	
BRCA2-206	ENST00000544455.6	11854	3418aa	Protein coding	CCDS9344	P51587	-		GENCODE basic	APPRIS P1
BRCA2-204	ENST00000530893.6	2011	481aa	Protein coding	-	AOA590UJ17	-	TSL:1	CDS 3' incomplete	
BRCA2-207	ENST00000614259.2	11763	2649aa	Nonsense mediated decay	-	-	-		TSL:2	
BRCA2-208	ENST00000665585.1	2598	438aa	Nonsense mediated decay	-	AOA590UJU6	-		CDS 5' incomplete	
BRCA2-202	ENST00000470094.1	842	186aa	Nonsense mediated decay	-	H0YE37	-	TSL:5	CDS 5' incomplete	
BRCA2-209	ENST00000666593.1	523	58aa	Nonsense mediated decay	-	AOA590UJ24	-		CDS 5' incomplete	
BRCA2-203	ENST00000528762.1	495	64aa	Nonsense mediated decay	-	H0YD86	-	TSL:4	CDS 5' incomplete	
BRCA2-205	ENST00000533776.1	523	No protein	Retained intron	-	-	-		TSL:3	

MANE Plus Clinical

MANE: Matched Annotation from the NCBI and EMBL-EBI

MANE Plus Clinical: second transcript where there are mutually exclusive clinically important exons ([43 additional transcripts](#))

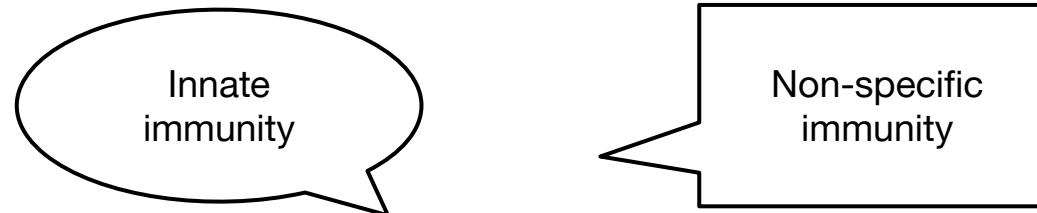
Ensembl stable IDs

- ENSG#####.# Ensembl Gene ID
- ENST#####.# Ensembl Transcript ID
- ENSP#####.# Ensembl Peptide ID
- ENSE#####.# Ensembl Exon ID
- For non-human species a suffix is added:
MUS (*Mus musculus*) for mouse ENSMUSG###
DAR (*Danio rerio*) for zebrafish: ENSDARG###

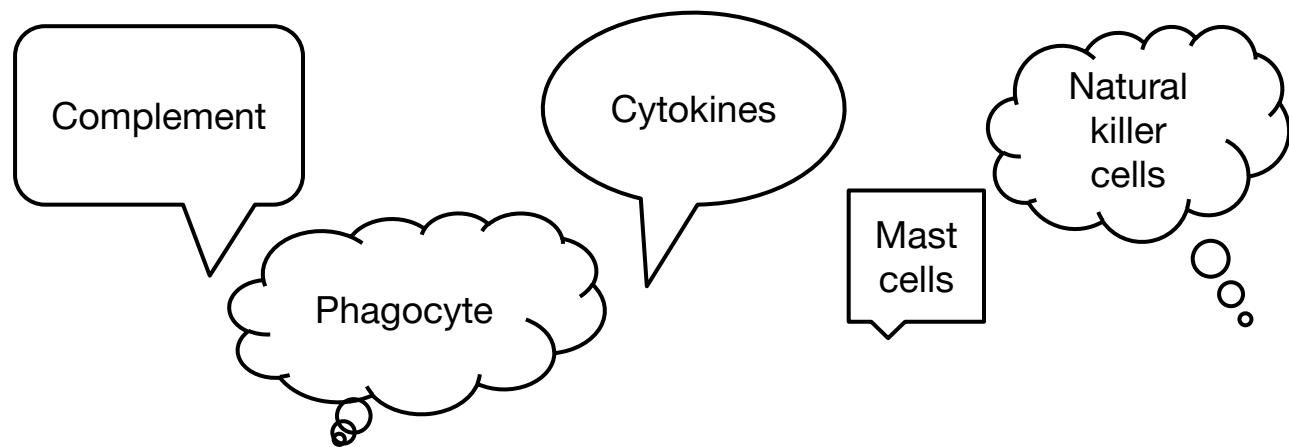
http://www.ensembl.org/info/genome/stable_ids/index.html

Why Gene Ontology (GO)?

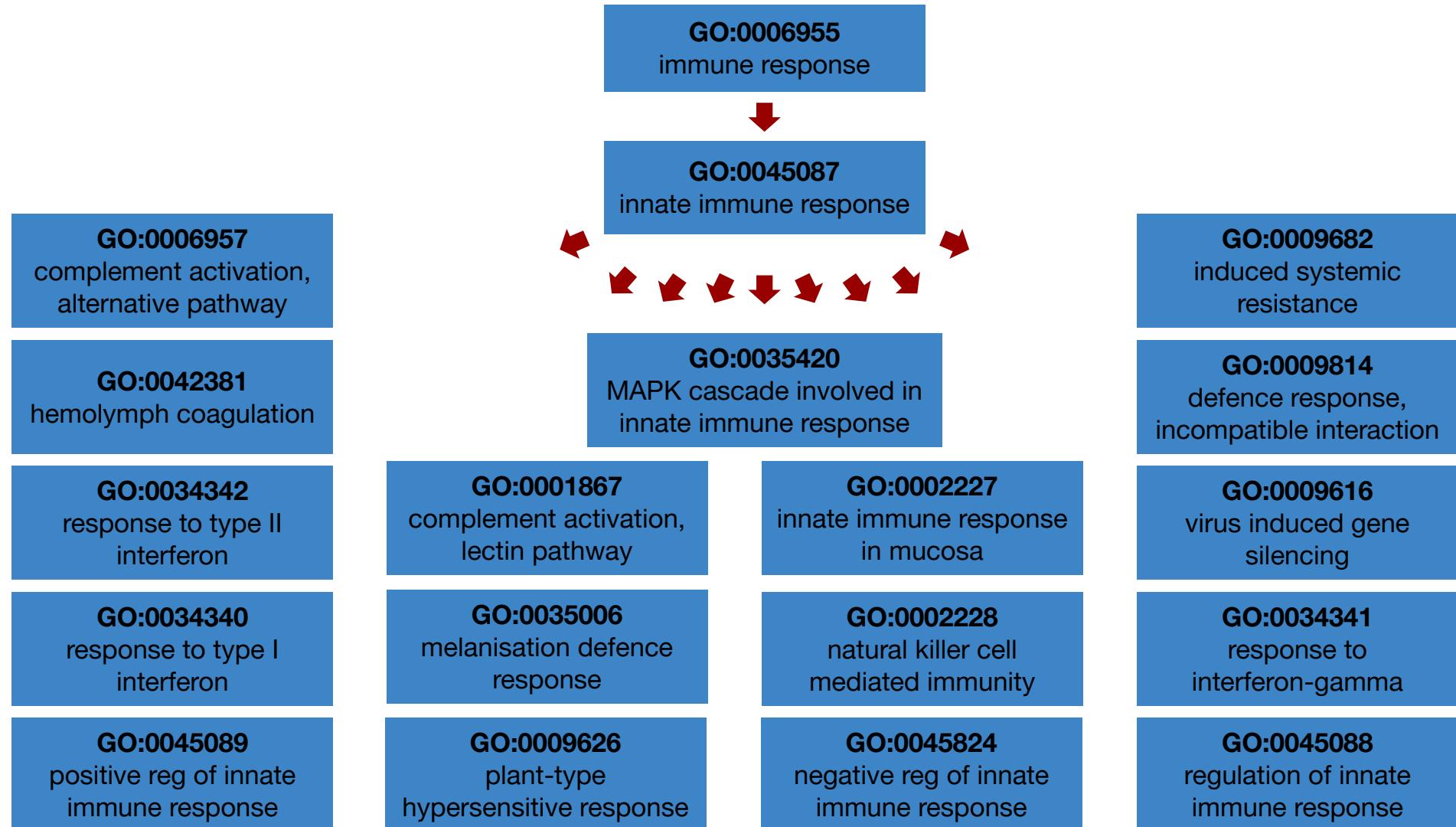
Multiple terms for
the same thing



Gene descriptions
too specific



GO terms are hierarchical



More information

The Ensembl gene annotation system.

Aken *et al.* 2016

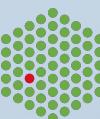
<http://europepmc.org/articles/PMC4919035>

Hands on

We're going to look at an Ensembl gene, *UQCRQ*, and find out information about it and its transcripts.



e! Ensembl's BioMart

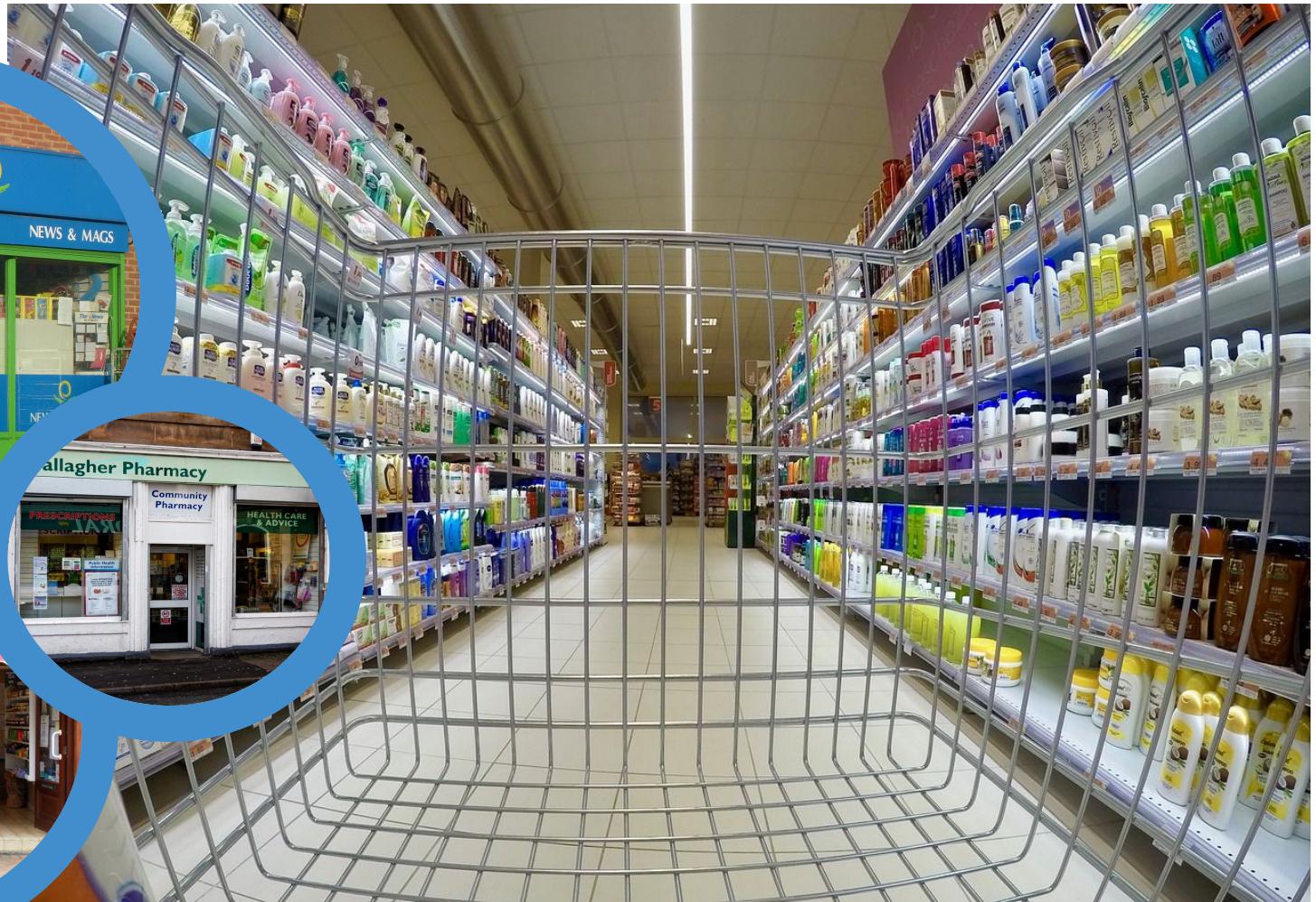


What is BioMart?

A tool in your browser to:

- Export data with no programming required
- Build queries with a few mouse clicks
- Generate custom data tables and sequence files

Why use BioMart?



Why use BioMart?

For things that would be time consuming/difficult with the Ensembl browser:

- Query multiple things at once:
 - ID conversions
 - Gene locations
 - Download sequences
-
- Export large amounts of data

Where can I find BioMart?

 [Login/Register](#)

[BLAST/BLAT](#) | [VEP](#) | [Tools](#) | [BioMart](#) | [Downloads](#) | [Help & Docs](#) | [Blog](#)

 [Search all species...](#) 

 [Login/Register](#)

[HMMER](#) | [BLAST](#) | [BioMart](#) | [Tools](#) | [Downloads](#) | [Documentation](#) | [Website help](#)

 [Search Ensembl Protists...](#) 

ensembl.org/biomart/martview

protists.ensembl.org/biomart/martview

For which genomes is BioMart available?

e!Ensembl

e!EnsemblFungi (some exceptions)

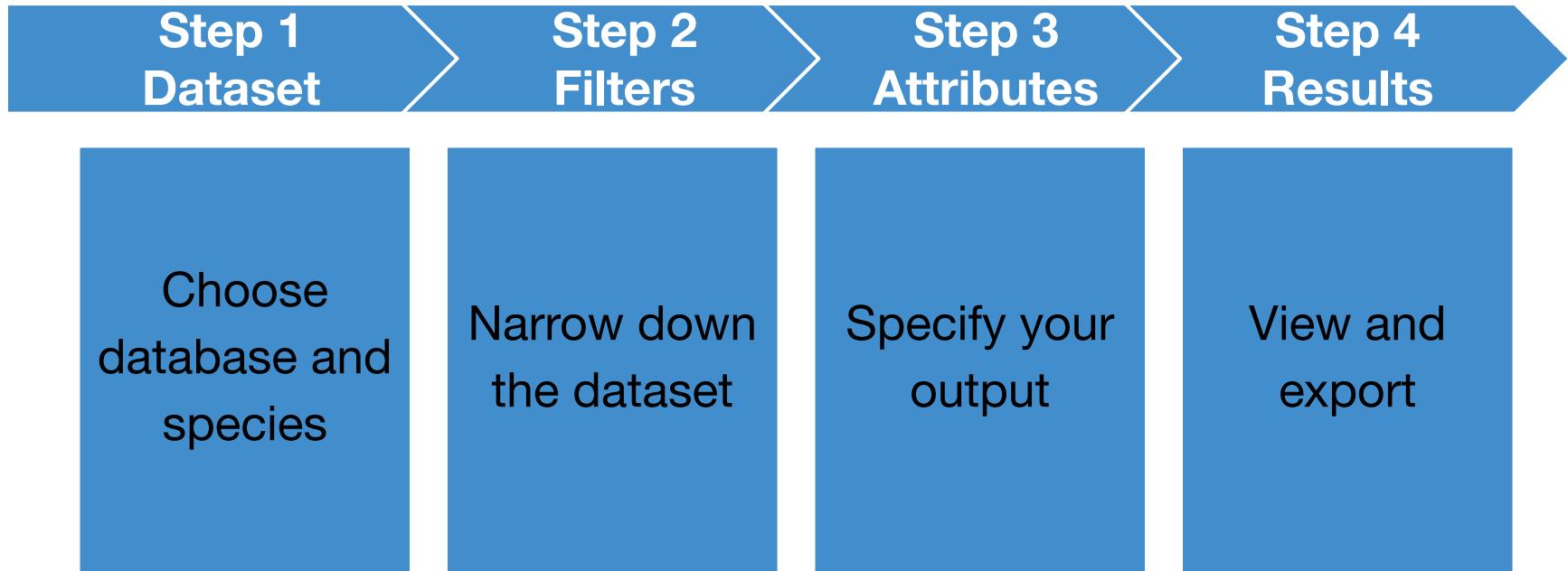
e!EnsemblMetazoa

e!EnsemblPlants

e!EnsemblProtists (some exceptions)

How do I use BioMart?

The four steps



How do I use BioMart?

Step 1 Dataset

- Define the database you want to search with your filters:
 - Genes
 - Variation
 - Regulation
 - Mouse strains
- Define the species

How do I use BioMart?

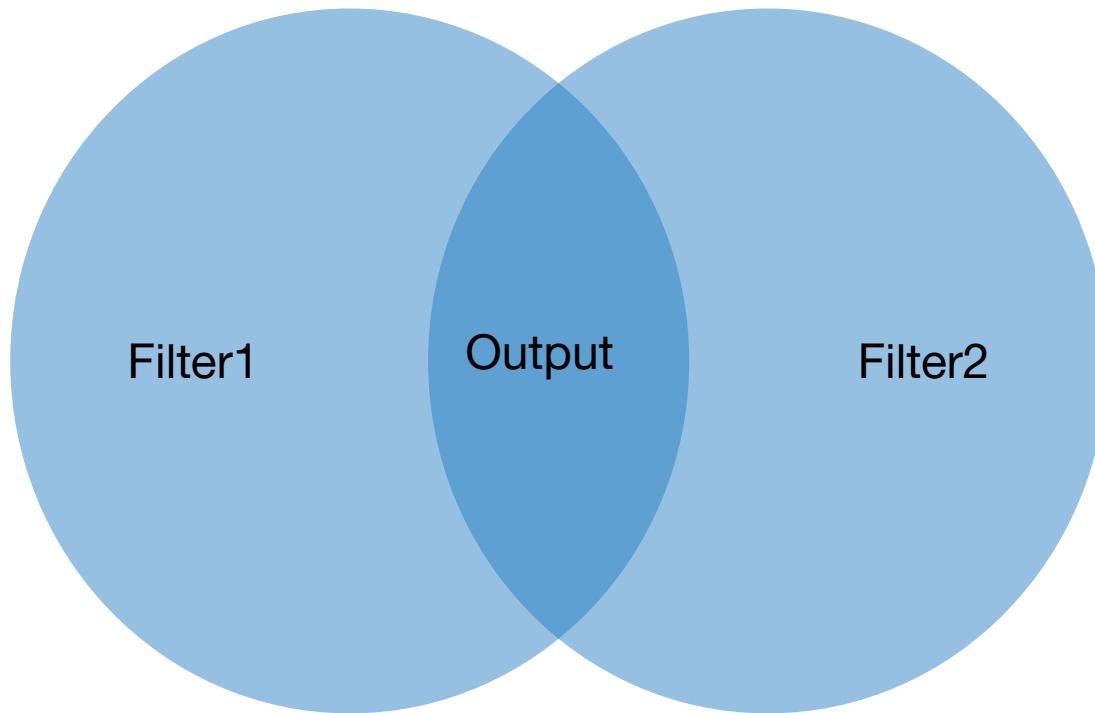
Step 2 Filters

Define a (large) set of genes/variants by combination of filters, e.g.:

- Region
- List of IDs
- Function (GO term)
- Phenotype

How do I use BioMart?

Step 2
Filters



How do I use BioMart?

Step 3 Attributes

Define the data you want to export (your output),
e.g.:

- IDs
- Features
- Variants
- Orthologues/paralogues
- Sequences

How do I use BioMart?

Step 4 Results

View and export data table/sequence in a number of formats:

- html
- csv
- tsv
- xls
- fasta

biomaRt: Bioconductor package for BioMart

Bioconductor provides tools for the analysis and comprehension of high-throughput genomic data using the R statistical programming language

- Easy installation:

```
source("http://bioconductor.org/biocLite.R")  
biocLite("biomaRt")
```

<http://www.bioconductor.org/packages/release/bioc/html/biomaRt.html>

<http://www.bioconductor.org/packages/release/bioc/vignettes/biomaRt/inst/doc/biomaRt.html>

More information

Kinsella *et al.*

Ensembl BioMarts: a hub for data retrieval across taxonomic space

Database: the Journal of Biological Databases and Curation (2011)

[europepmc.org/abstract/MED/21785142](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3117773/)

Smedley *et al.*

BioMart – biological queries made easy

BMC Genomics (2009) 10:22

[europepmc.org/abstract/MED/19144180](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2683333/)

Hands on

We're going to look at a set of six human genes, *ESPN*, *MYH9*, *USH1C*, *CISD2*, *THRIB* and *WHRN*, and find out:

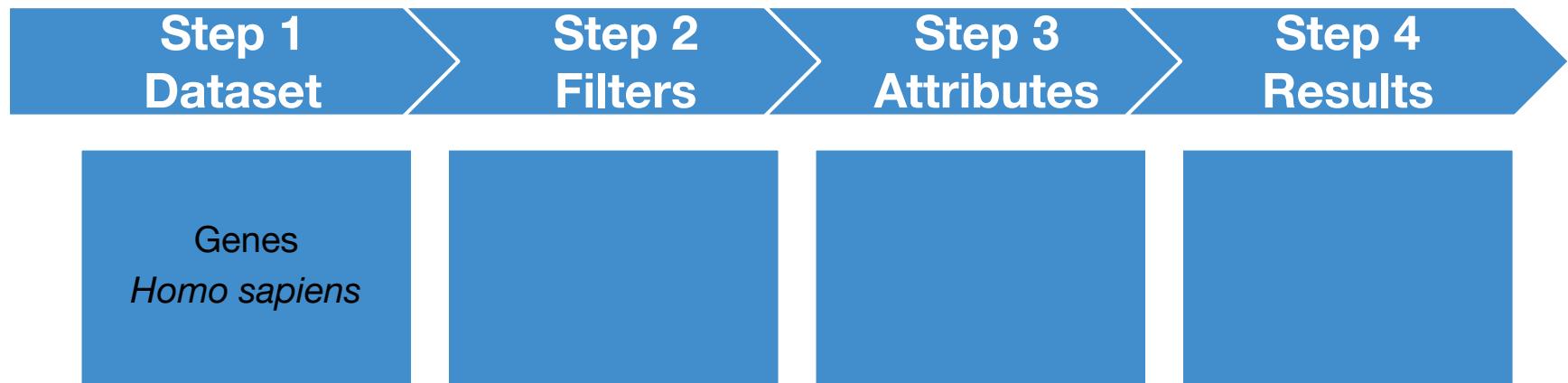
- Their NCBI gene IDs
- Their function via GO terms
- Their cDNA sequences



Hands on

We're going to look at a set of six human genes, *ESPN*, *MYH9*, *USH1C*, *CISD2*, *THR*B and *WHRN*, and find out:

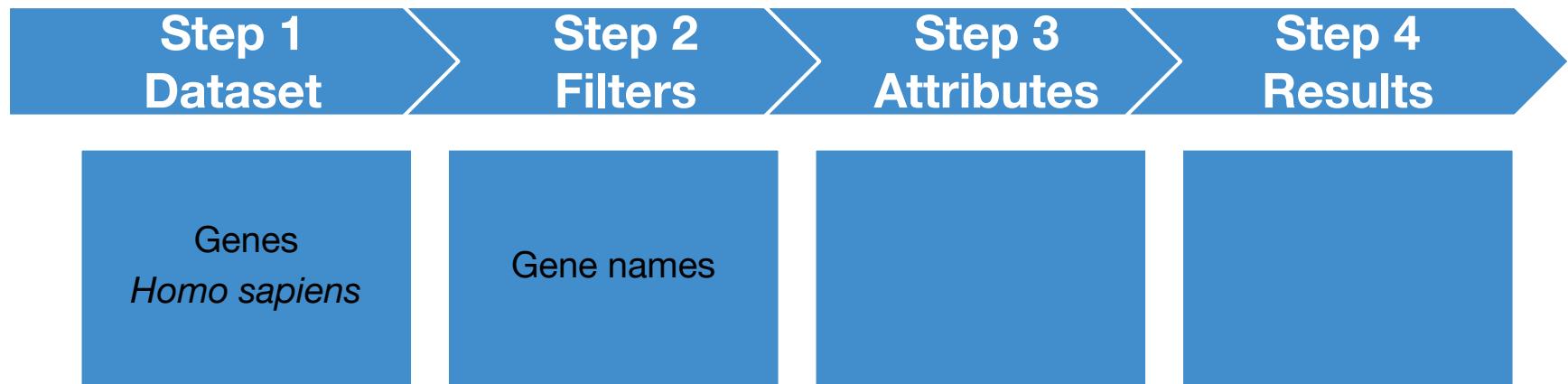
- Their NCBI gene IDs
- Their function via GO terms
- Their cDNA sequences



Hands on

We're going to look at a set of six human genes, *ESPN*, *MYH9*, *USH1C*, *CISD2*, *THR*B and *WHRN*, and find out:

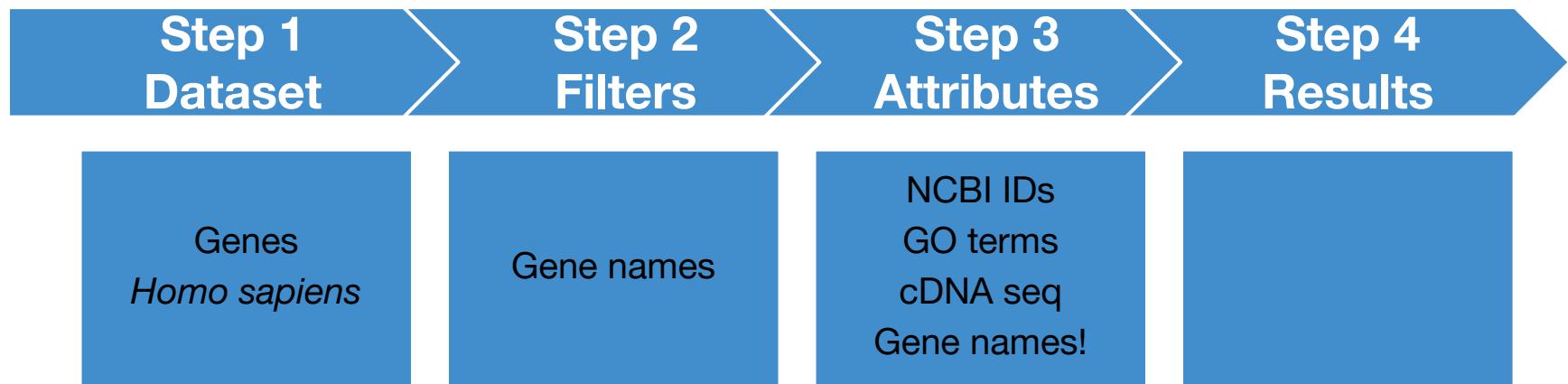
- Their NCBI gene IDs
- Their function via GO terms
- Their cDNA sequences



Hands on

We're going to look at a set of six human genes, *ESPN*, *MYH9*, *USH1C*, *CISD2*, *THR*B and *WHRN*, and find out:

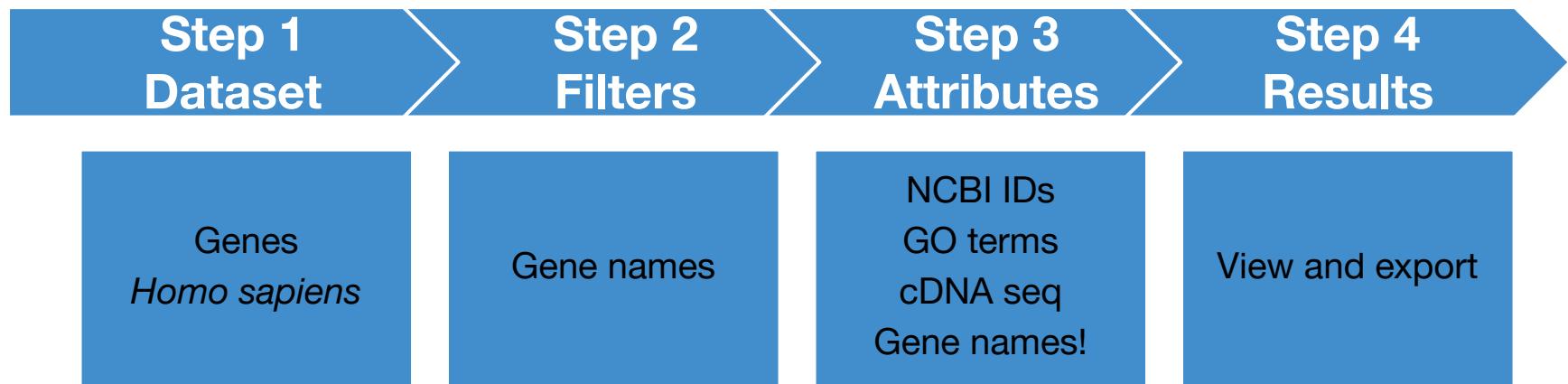
- Their NCBI gene IDs
- Their function via GO terms
- Their cDNA sequences



Hands on

We're going to look at a set of six human genes, *ESPN*, *MYH9*, *USH1C*, *CISD2*, *THR*B and *WHRN*, and find out:

- Their NCBI gene IDs
- Their function via GO terms
- Their cDNA sequences



Hands on

We're going to look at a set of six human genes, *ESPN*, *MYH9*, *USH1C*, *CISD2*, *THRΒ* and *WHRN*, and find out:

- Their NCBI gene IDs
- Their function via GO terms
- Their cDNA sequences

Wrap-up

Ensembl is a genome browser which integrates:

- Gene annotation
- Variation
- Comparative genomics
- Regulation

Ensembl REST API Workshop



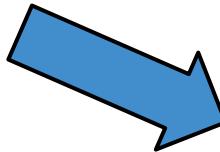
Ensembl Outreach

EMBL-EBI

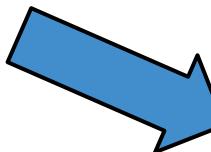
Structure



Presentation:
Background and theory
Code examples



Exercises:
Trying things out for yourself
Write your own code
Going beyond the demo



Demo:
Explaining and running the model answers

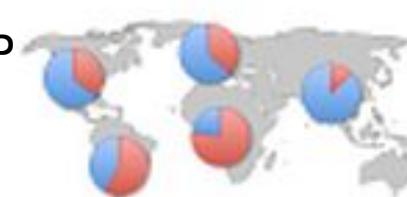
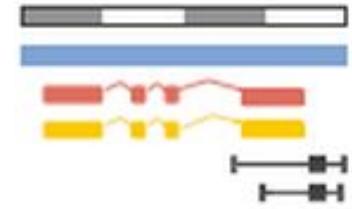
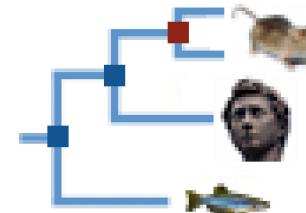
Course material

<http://training.ensembl.org/events/>

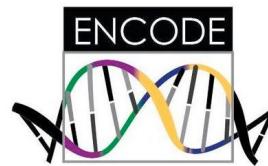
- Slides
- Notebooks in Python and R
 - Use whichever notebook you feel comfortable with
- When we demo the example answers we will use Python only
- Living Document

Ensembl features

- Genomes and gene builds for >300 species
- Variation data
- Compara (alignments, gene trees, homologues)
- Regulatory build
- BioMart (data export)
- Tools for data processing, e.g. VEP
- Display your own data
- Programmatic access via APIs
- Completely Open Source (FTP, GitHub)



Ve!P

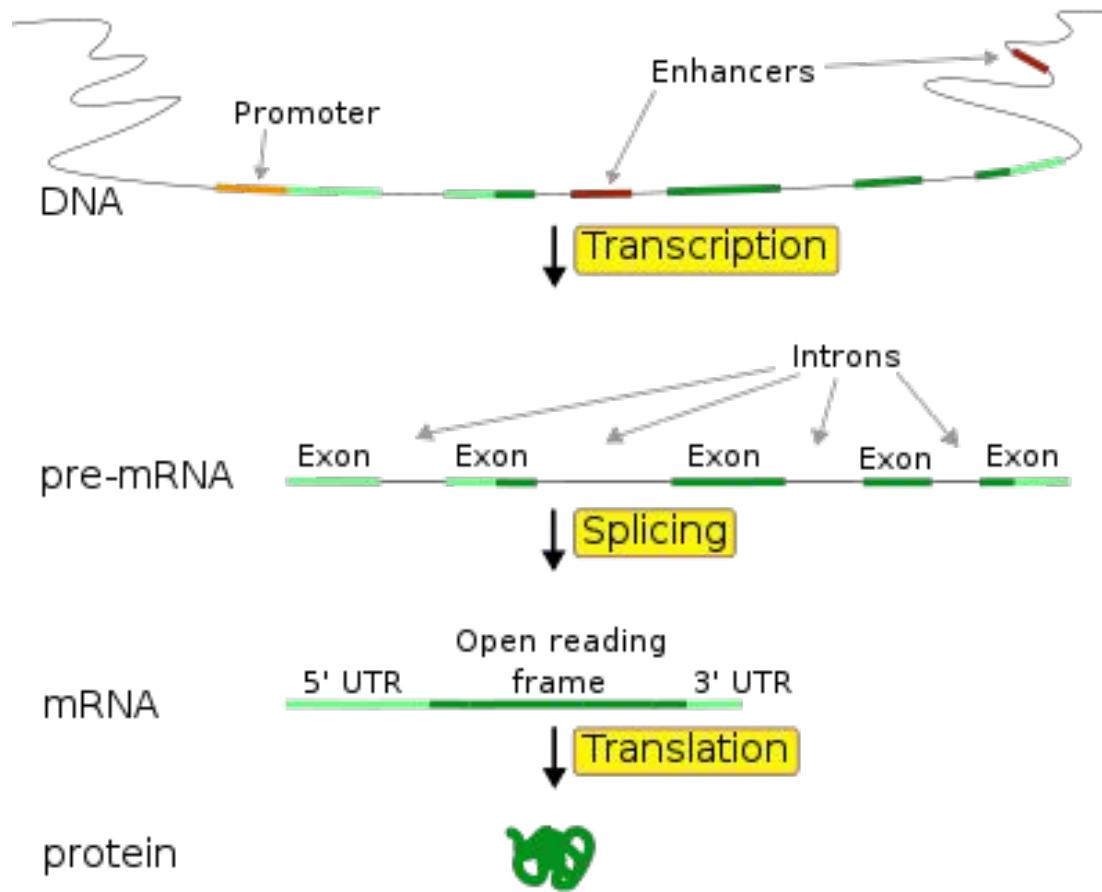


REST archives

Starting with release 87, there are REST archives (GRCh38 only). We will continue to provide archive services for up to five years, to match the Ensembl website archives.

<http://e87.rest.ensembl.org>

Ensembl data model

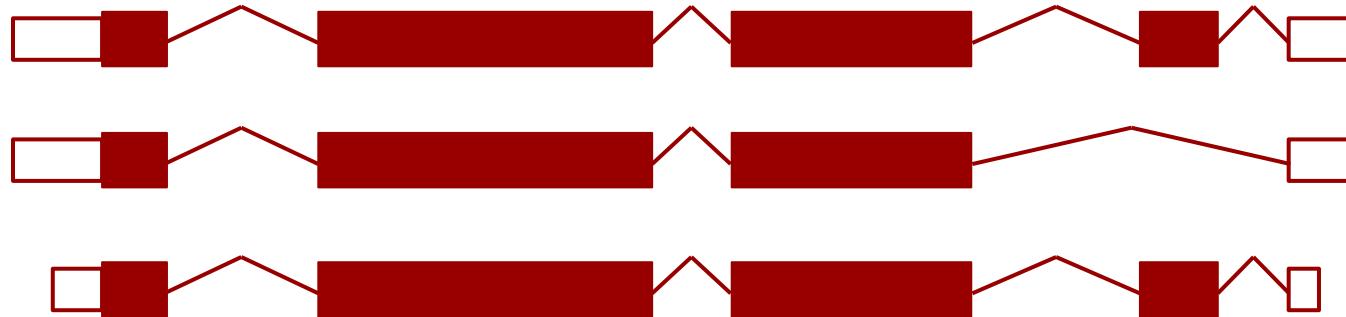


Ensembl data model

Primary feature types of Genes, Transcript, and Exons

A Gene is a set of alternatively spliced Transcripts

A Transcript is a set of Exons



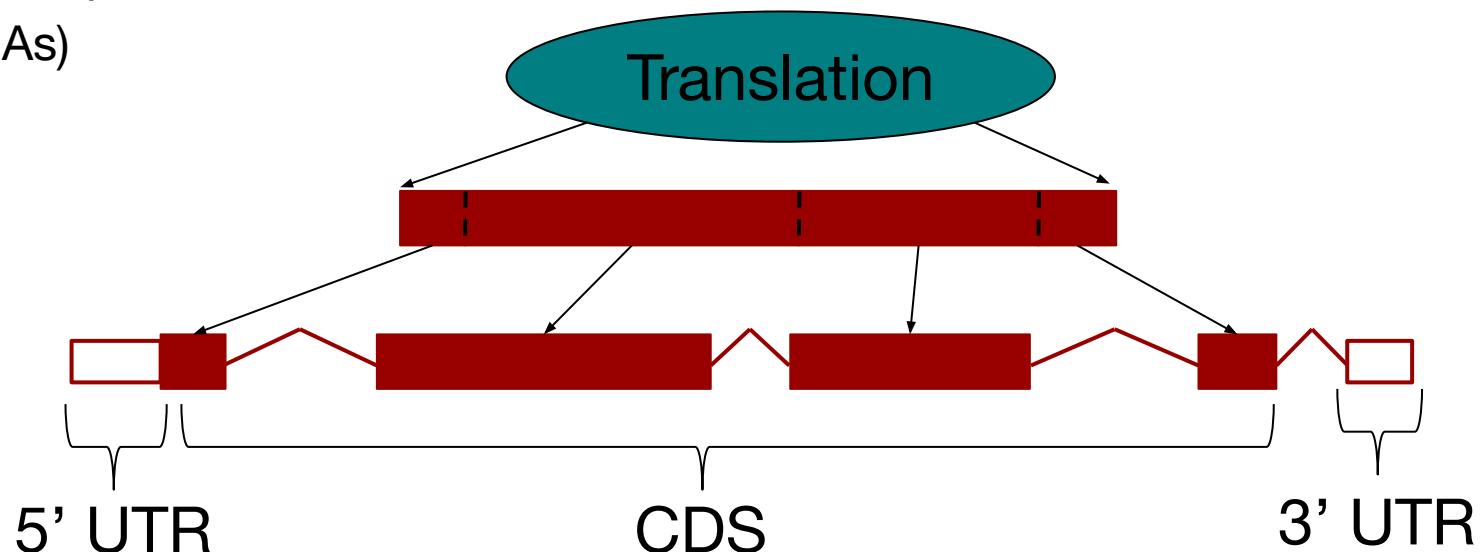
Ensembl data model

Translations are not Features.

A Translation object defines the UTR and CDS of a Transcript.

Peptides are not stored in the database, they are computed on the fly using Transcript objects.

Not all transcripts have a translation
(e.g. ncRNAs)

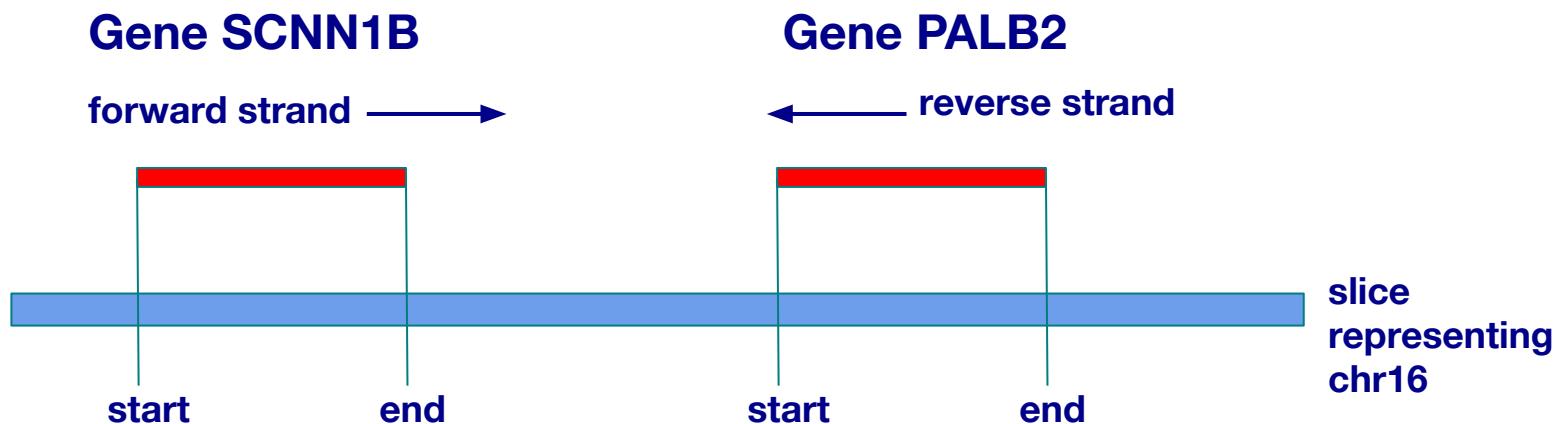


Features

Features have a defined location on the genome

Start and end are always plotted on the forward strand

$\text{start} < \text{end}$

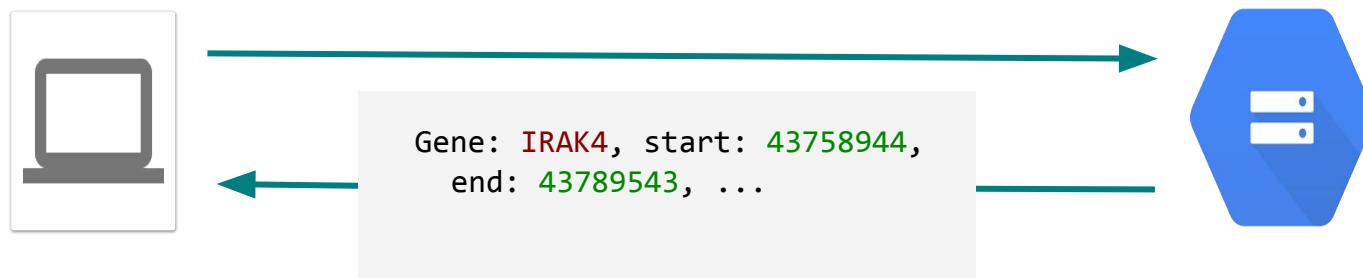


What is REST API?

REpresentational State Transfer. It describes how one system can communicate state with another.

Typically over HTTP(S), providing a machine readable, language agnostic method to access remote data or services.

<http://rest.ensembl.org/datalwant>



Ensembl REST API

- Language agnostic access to Ensembl datasets
- Only a fraction of the functionality of the Perl API is exposed

<http://rest.ensembl.org>

Ensembl REST API Endpoints

Archive

Resource	Description
<code>GET archive/id/:id</code>	Uses the given identifier to return the archived sequence
<code>POST archive/:id</code>	Retrieve the archived sequence for a set of identifiers

Comparative Genomics

Resource	Description
<code>GET genetree/:id/:id</code>	Retrieves a gene tree for a gene tree stable identifier
<code>GET genetree/member/:id/:id</code>	Retrieves the gene tree that contains the gene / transcript / translation stable identifier
<code>GET genetree/member/:symbol/:species/:symbol</code>	Retrieves the gene tree that contains the gene identified by a symbol
<code>GET alignment/:region/:species/:region</code>	Retrieves genomic alignments as separate blocks based on a region and species
<code>GET homology/:id/:id</code>	Retrieves homology information (orthologs) by Ensembl gene id
<code>GET homology/:symbol/:species/:symbol</code>	Retrieves homology information (orthologs) by symbol

Cross References

Resource	Description
<code>GET xrefs/:symbol/:species/:symbol</code>	Looks up an external symbol and returns all Ensembl objects linked to it. This can be a display name for a gene/transcript/translation, a synonym or an externally linked reference. If a gene's transcript is linked to the supplied symbol the service will return both gene and transcript (it supports transient links).
<code>GET xrefs/:id/:id</code>	Perform lookups of Ensembl Identifiers and retrieve their external references in other databases
<code>GET xrefs/:name/:species/:name</code>	Performs a lookup based upon the primary accession or display label of an external reference and returning the information

What Ensembl REST is and is not

- + HTTP access to Ensembl data
- + Stable service
- + Limited by network latency
- + Read only
- + Versioned with archives
- No mirrors
- Not an efficient data mining solution
- Incomplete coverage

What is an endpoint?

“In REST, the resource typically refers to some object or set of objects that are exposed at an API endpoint. /api/users/johnny. An endpoint by itself is just a reference to a uri that accepts web requests that may or may not be RESTful. /services/service.asmx.”

An endpoint is a particular output that you can get given a particular input.

It is a function that interacts with our database.

Endpoint documentation

Full documentation of all the endpoints is found at:

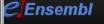
<http://rest.ensembl.org>

The documentation lists:

- All the endpoints grouped by function
- The required parameters for each endpoint
- Optional parameters
- Example code for using the endpoints

Functional groupings

- Archive
- Comparative Genomics
- Cross References
- Information
- Lookup
- Mapping
- Ontology & Taxonomy
- Sequence
- Variation, etc...

 Ensembl Endpoints User Guide Change Log About the Ensembl Project Contact Ensembl

Ensembl REST API Endpoints

Archive

Resource	Description
GET archive/id/:id	Uses the given identifier to return the archived sequence
POST archive/:id	Retrieve the archived sequence for a set of identifiers

Comparative Genomics

Resource	Description
GET genetree/:id/:id	Retrieves a gene tree for a gene tree stable identifier
GET genetree/member/:id/:id	Retrieves the gene tree that contains the gene / transcript / translation stable identifier
GET genetree/member/:symbol/:species/:symbol	Retrieves the gene tree that contains the gene identified by a symbol
GET alignment/:region/:species/:region	Retrieves genomic alignments as separate blocks based on a region and species
GET homology/:id/:id	Retrieves homology information (orthologs) by Ensembl gene id
GET homology/:symbol/:species/:symbol	Retrieves homology information (orthologs) by symbol

Cross References

Resource	Description
GET xrefs/:symbol/:species/:symbol	Looks up an external symbol and returns all Ensembl objects linked to it. This can be a display name for a gene/transcript/translation, a synonym or an externally linked reference. If a gene's transcript is linked to the supplied symbol the service will return both gene and transcript (it supports transient links).
GET xrefs/:id/:id	Perform lookups of Ensembl Identifiers and retrieve their external references in other databases
GET xrefs/:name/:species/:name	Performs a lookup based upon the primary accession or display label of an external reference and returning the information

Endpoint documentation

GET lookup/id/:id

Find the species and database for a single identifier e.g. gene, transcript, protein

Parameters

Required

Name	Type	Description	Default	Example Values
id	String	An Ensembl stable ID	-	ENSG00000157764

Optional

Name	Type	Description	Default	Example Values
callback	String	Name of the callback subroutine to be returned by the requested JSONP response. Required ONLY when using JSONP as the serialisation method. Please see the user guide .	-	randomlygeneratedname
db_type	String	Restrict the search to a database other than the default. Useful if you need to use a DB other than core	-	core otherfeatures
expand	Boolean(0,1)	Expands the search to include any connected features. e.g. If the object is a gene, its transcripts, translations and exons will be returned as well.	0	-

You must include the id in the URL in this position

Resource Information

Methods	GET
Response formats	json xml jsonp

You can choose to include these in the URL in the format: **parameter=option**

Sample code

[Example output](#)[Perl](#)[Python2](#)[Python3](#)[Ruby](#)[Java](#)[Curl](#)[Wget](#)

```
1. use strict;
2. use warnings;
3.
4. use HTTP::Tiny;
5.
6. my $http = HTTP::Tiny->new();
7.
8. my $server = 'http://rest.ensembl.org';
9. my $ext = '/lookup/id/ENSG00000157764?expand=1';
10. my $response = $http->get($server.$ext, {
11.     headers => { 'Content-type' => 'application/json' }
12. });
13.
14. die "Failed!\n" unless $response->{success};
15.
16.
17. use JSON;
18. use Data::Dumper;
19. if(length $response->{content}) {
```

Making a REST call in the browser

- The easiest way to make REST calls is to put URLs into the browser
- This can be used as a quick look-up
- This can help you to test the URLs in your scripts to see:
 - If they work
 - If you've included the correct parameters
 - What the output looks like

Pinging the database

Ping confirms that you have a connection to the database

<http://rest.ensembl.org/info/ping?content-type=application/json>

```
{  
  ping: 1  
}
```

Requesting a gene by ID

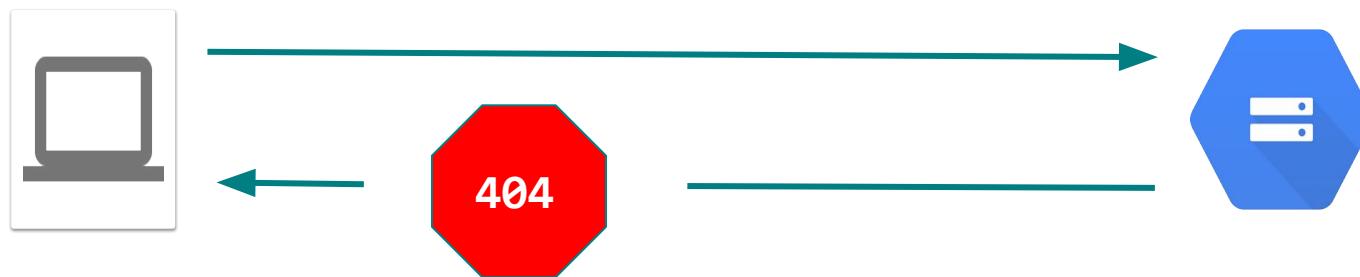
<http://rest.ensembl.org/lookup/id/ENSG00000157764?content-type=application/json>

```
{  
    "source": "ensembl_havana",  
    "object_type": "Gene",  
    "logic_name": "ensembl_havana_gene",  
    "version": 12,  
    "species": "homo_sapiens",  
    "description": "B-Raf proto-oncogene, serine/threonine kinase [Source:HGNC  
Symbol;Acc:HGNC:1097]",  
    "display_name": "BRAF",  
    "assembly_name": "GRCh38",  
    "biotype": "protein_coding",  
    "end": 140924764,  
    "seq_region_name": "7",  
    "db_type": "core",  
    "strand": -1,  
    "id": "ENSG00000157764",  
    "start": 140719327  
}
```

HTTP Status Codes

The server uses HTTP status codes to signal the request outcome

<http://rest.ensembl.org>thisdoesntexist>



HTTP Status Codes

Code	Name	Notes
200	OK	Request was a success
400	Bad Request	Occurs during exceptional circumstances such as the service is unable to find an ID. Check if the response Content-type or Accept was JSON. If so the JSON object is an exception hash with the message keyed under error
403	Forbidden	You are submitting far too many requests and have been temporarily forbidden access to the service. Wait and retry with a maximum of 15 requests per second.
404	Not Found	Indicates a badly formatted request. Check your URL

<https://github.com/Ensembl/ensembl-rest/wiki/HTTP-Response-Codes>

HTTP Status Codes

Code	Name	Notes
408	Timeout	The request was not processed in time. Wait and retry later
429	Too Many Requests	You have been rate-limited; wait and retry. The headers X-RateLimit-Reset, X-RateLimit-Limit and X-RateLimit-Remaining will inform you of how long you have until your limit is reset and what that limit was. If you get this response and have not exceeded your limit then check if you have made too many requests per second.
503	Service Unavailable	The service is temporarily down; retry after a pause
418	I'm a teapot	An April Fools joke added in 1998, who said computer scientists don't have a sense of humour?

<https://github.com/Ensembl/ensembl-rest/wiki/HTTP-Response-Codes>

Exercises

1. Find an endpoint which you can use to lookup information about a gene using its symbol.
2. Create a URL to find information about the gene *ESPN* in human.
3. Expand your results to include information about transcripts.

Answers

1. http://rest.ensembl.org/documentation/info/symbol_lookup
2. http://rest.ensembl.org/lookup/symbol/homo_sapiens/ESPN?content-type=application/json
3. http://rest.ensembl.org/lookup/symbol/homo_sapiens/ESPN?content-type=application/json;expand=1

Scripting around REST API calls

Scripting around calls allows you to:

- Extract specific bits of data from your REST call.
- Output in your preferred format.
- Link together calls for more complicated queries.
- Integrate your queries into a larger pipeline.

Language agnostic access

- REST APIs are designed to be accessed using any programming language.
- Calls can be made and decoded within any script.
- We have examples in Python, Perl and R.

Python modules

- To make requests in Python, you will need the `requests` package:
 - <http://docs.python-requests.org/en/master/user/install/> (not needed for this course, this is all set up in your Python Notebook)
- To decode JSON you will need the `JSON` package:
 - Should ship with standard Python installations
- You'll need `pprint` to print JSON in an easy to read way
 - Should ship with standard Python installations

```
import requests, sys, json
from pprint import pprint
```

R Libraries

- To make requests in R you will need the `httr` library
- To decode JSON you'll need the `jsonlite` package

```
library(httr)
library(jsonlite)
```

Requesting a gene by ID

<http://rest.ensembl.org/lookup/id/ENSG00000157764?content-type=application/json>

```
{  
    "source": "ensembl_havana",  
    "object_type": "Gene",  
    "logic_name": "ensembl_havana_gene",  
    "version": 12,  
    "species": "homo_sapiens",  
    "description": "B-Raf proto-oncogene, serine/threonine kinase [Source:HGNC  
Symbol;Acc:HGNC:1097]",  
    "display_name": "BRAF",  
    "assembly_name": "GRCh38",  
    "biotype": "protein_coding",  
    "end": 140924764,  
    "seq_region_name": "7",  
    "db_type": "core",  
    "strand": -1,  
    "id": "ENSG00000157764",  
    "start": 140719327  
}
```

Making a request – Python

- Make a string of the server (you'll use this multiple times)
- Make another string of the extension with all the parameters

```
import requests, sys

server = "http://rest.ensembl.org"
ext = "/lookup/id/ENSG00000157764?expand=1"

r = requests.get(server+ext, headers={"Accept": "application/json"})

pprint (r)
```

Making a request – R

- Make a string of the server (you'll use this multiple times)
- Make another string of the extension with all the parameters

```
library(httr)
library(jsonlite)

server <- "http://rest.ensembl.org"
ext <- "/lookup/id/ENSG00000157764"

r <- GET(paste(server, ext, sep = ""), accept("application/json"))

r
```

Error handling – Python

You should never assume a response will return correctly.

```
import requests, sys

server = "http://rest.ensembl.org"
ext = "/lookup/id/ENSG00000157764?expand=1"

r = requests.get(server+ext, headers={"Accept" : "application/json"})

if not r.ok:
    r.raise_for_status()
```

Check the response code returned by the server.

Error handling – R

You should never assume a response will return correctly.

```
library(httr)
library(jsonlite)

server <- "http://rest.ensembl.org"
ext <- "/lookup/id/ENSG00000157764"

r <- GET(paste(server, ext, sep = ""), content_type("application/json"))

r

stop_for_status(r)
```

Check the response code returned by the server.

HTTP Headers

HTTP allows the serving of different representations of a resource based on client preferences

Content-type and **Accept** headers are how servers and clients negotiate what format they will communicate with.

text/html, text/plain, application/json, image/png, etc.

HTTP Headers

Resource Information

Methods	GET
Response formats	fasta json seqxml text yaml jsonp

- The returned **content types** can be specified in the header as accept (you'll need to use content-type in URLs)
- Endpoint documentation pages list allowed content-types

<https://github.com/Ensembl/ensembl-rest/wiki/Output-formats>

Decoding the response – Python

- In most cases you'll be using JSON formatted responses
- Most languages have JSON parsers that return the data as a structure
- In Python pretty print (`pprint`) will give you a human readable format

```
decoded = r.json()  
  
pprint(decoded)
```

Decoding the response – R

- In most cases you'll be using JSON formatted responses
- Most languages have JSON parsers that return the data as a structure
- In R `prettify` will give you a human readable format

```
decoded = content(r, "text")
```

```
prettify(decoded)
```

Decoding JSON

<http://rest.ensembl.org/lookup/id/ENSG00000157764?content-type=application/json>

```
{  
    "source": "ensembl_havana",  
    "object_type": "Gene",  
    "logic_name": "ensembl_havana_gene",  
    "version": 12,  
    "species": "homo_sapiens",  
    "description": "B-Raf proto-oncogene, serine/threonine kinase [Source:HGNC  
Symbol;Acc:HGNC:1097]",  
    "display_name": "BRAF",  
    "assembly_name": "GRCh38",  
    "biotype": "protein_coding",  
    "end": 140924764,  
    "seq_region_name": "7",  
    "db_type": "core",  
    "strand": -1,  
    "id": "ENSG00000157764",  
    "start": 140719327  
}
```

Decoding JSON

- JSON is essentially a massive dictionary/hash/dataframe with keys and values.
- Sometimes a key may then contain another nested dictionary or list
 - Which may contain another
 - And another
 - And another
- Look at the json to work out what keys you need
- You can cycle through all keys in a dictionary with for loops

Helper function

- The helper function in your python script makes your life easier by:
 - Calling the request with the specified server, extension and content type.
 - Getting the status of a failed query
 - Decoding the JSON (if you've used JSON as your content type)
 - Returning the text (if you use any other content type)
- Add it to every script then just call it when you need to fetch an endpoint

Helper function: Python

```
def fetch_endpoint(server, request, content_type):
    """
    Fetch an endpoint from the server, allow overriding of default content-type
    """
    r = requests.get(server+request, headers={ "Accept" : content_type})

    if not r.ok:
        r.raise_for_status()
        sys.exit()

    if content_type == 'application/json':
        return r.json()
    else:
        return r.text
```

Helper function: R

```
Fetch_endpoint <- function(server, request, content_type){  
  """  
    Fetch an endpoint from the server, allow overriding of default content-type  
  """  
  r <- GET(paste(server, request, sep = ""), accept(content_type))  
  
  stop_for_status(r)  
  
  if (content_type == 'application/json'){  
    return (fromJSON(content(r, "text")))  
  } else {  
    return (content(r, "text"))  
  }  
}
```

Exercises

1. Write a script to lookup the gene called *ESPN* in human and print the results in JSON.

Feedback Survey

training.ensembl.org/events

- Feedback surveys help us to improve our courses for future participants
- You can opt to receive an e-certificate in the feedback
- This is a certificate that you have attended this course, it does not confer any qualification

Publications

<http://www.ensembl.org/info/about/publications.html>

Ensembl 2023

FJ Martin *et al*

<https://europepmc.org/article/MED/36318249>

Topic-specific publications mentioned throughout workshop

How is all this data organised?

- Ensembl browser sites

Main website, GRCh37, Ensembl Genomes, Archive!

- BioMart ‘DataMining tool’
- Ensembl Database (open source)

Perl-API, REST API, MySQL

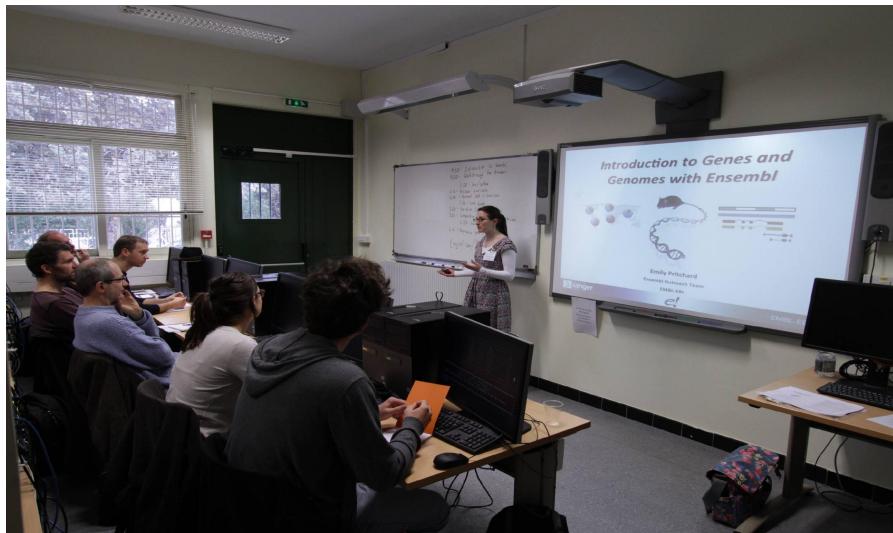
- FTP download site

<http://www.ensembl.org/info/data/ftp/index.html>

Recommend us to your friends

We can teach an Ensembl course at any institute for free (plus trainers' expenses in high income countries).

Email us: helpdesk@ensembl.org



Browser course

One day course on the Ensembl browser, aimed at wet-lab scientists.

REST API course

Half day course on the Ensembl REST API, aimed at bioinformaticians.

Train the Trainer course

One day course on delivering the Ensembl browser course.

training.ensembl.org/hosting

Help and documentation

Courses online



<http://www.ebi.ac.uk/training/online/subjects/11>

Tutorials www.ensembl.org/info/website/tutorials



Flash animations

www.youtube.com/user/EnsemblHelpdesk

<http://u.youku.com/Ensemblhelpdesk>



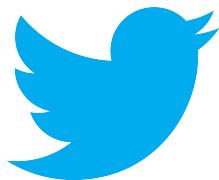
Email us helpdesk@ensembl.org

Ensembl public mailing lists dev@ensembl.org,
announce@ensembl.org

Follow us!



www.facebook.com/Ensembl.org



@Ensembl



www.ensembl.info

Ensembl 2023



Ensembl Acknowledgements

The Entire Ensembl Team

Fiona Cunningham, James E Allen, Jamie Allen, Jorge Alvarez-Jarreta, M Ridwan Amode, Irina M Armean, Olanrewaju Austine-Orimoloye, Andrey G Azov, If Barnes, Arne Becker, Ruth Bennett, Andrew Berry, Jyothish Bhai, Alexandra Bignell, Konstantinos Billis, Sanjay Boddu, Lucy Brooks, Mehrnaz Charkhchi, Carla Cummins, Luca Da Rin Fioretto, Claire Davidson, Kamalkumar Dodiya, Sarah Donaldson, Bilal El Houdaigui, Tamara El Naboulsi, Reham Fatima, Carlos Garcia Giron, Thiago Genez, Jose Gonzalez Martinez, Cristina Guijarro-Clarke, Arthur Gymer, Matthew Hardy, Zoe Hollis, Thibaut Hourlier, Toby Hunt, Thomas Juettemann, Vinay Kaikala, Mike Kay, Ilias Lavidas, Tuan Le, Diana Lemos, José Carlos Marugán, Shamika Mohanan, Aleena Mushtaq, Marc Naven, Denye N Ogeh, Anne Parker, Andrew Parton, Malcolm Perry, Ivana Piližota, Irina Prosovetskaia, Manoj Pandian Sakthivel, Ahamed Imran Abdul Salam, Bianca M Schmitt, Helen Schuilenburg, Dan Sheppard, José G Pérez-Silva, William Stark, Emily Steed, Kyösti Sutinen, Ranjit Sukumaran, Dulika Sumathipala, Marie-Marthe Suner, Michal Szpak, Anja Thormann, Francesca Floriana Tricomi, David Urbina-Gómez, Andres Veidenberg, Thomas A Walsh, Natalie Willhoft, Andrea Winterbottom, Elizabeth Wass, Marc Chakiachvili, Bethany Flint, Adam Frankish, Stefano Giorgetti, Leanne Haggerty, Sarah E Hunt, Garth R Ilsley, Jane E Loveland, Fergal J Martin, Benjamin Moore, Jonathan M Mudge, Matthieu Muffato, Emily Perry, Magali Ruffier, John Tate, David Thybert, Stephen J Trevanion, Sarah Dyer, Peter W Harrison, Kevin L Howe, Andrew D Yates, Daniel R Zerbino, Paul Flicek

European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Genome Campus, Hinxton, Cambridge CB10 1SD, UK

Funding



EMBL



Co-funded by the
European Union



UK Research
and Innovation

National Institute of
Allergy and
Infectious Diseases
(NIAID)

National Human
Genome Research
Institute (NHGRI)

Training materials

- Ensembl training materials are protected by a CC BY license 
- <http://creativecommons.org/licenses/by/4.0/>
- If you wish to re-use these materials, please credit Ensembl for their creation
- If you use Ensembl for your work, please cite our papers
- <http://www.ensembl.org/info/about/publications.html>