Ensembl runs API workshops worldwide, and for the first time we make one available online. Here you will find a collection of video lectures from the ‘Ensembl API workshop’ that took place at EMBL-EBI 22nd-24th May 2013, using Ensembl version 71. The exercises associated with the workshop are included to carry out at your leisure.

Note: the videos in this course will only run on Google Chrome, Internet Explorer and Safari.

Learning objectives:

- Understand how data is organised in the Ensembl database
- Understand how Ensembl assigns objects and methods
- Know how to search the API Doxygen documentation
- Be able to write Perl scripts that query the Ensembl database for various different kinds of data

How to take this course

This course makes use of lecture material from a three-day course that took place at EMBL-EBI in May 2013. This course is presented by Magali Ruffier from the Core team, Anja Thormann from Variation, Nathan Johnson from Regulation and Matthieu Muffato and Stephen Fitzgerald from Compara.

If you see this video icon, you will be able to watch a video of the lecture, presented in Mediasite [3] format. (Note that these videos will only run on Google Chrome [4], Internet Explorer or Safari.)
This course is made up of five modules:

1. Core
2. Variation
3. Regulation
4. Compara trees
5. Compara alignments

You can dip in and out of these modules, and pick and choose those of interest to you. However, we recommend that you start with the **Core module** as this provides an introduction to using the API, including how to access the documentation and how to connect to the Ensembl database.

The slides for each lecture are available as pdf files for you to study. Lecture links will open in a new tab and pdfs will open in your default pdf viewer (eg. Adobe Acrobat or Apple Preview).

The lectures are interspersed with exercises, allowing you to try out the techniques presented in the lectures. Sample scripts are provided for each of the exercises, giving an example of how you might have found the answer; remember that there's often many ways to do something - if your script is different to the sample script but still gets the correct data, then your script is correct.

This is a long course, intended to be taken over several days. The full course consists of 5.5 hours of video, with around 12-24 hours worth of exercises, depending on how quickly you are able to complete them.

This course is not intended to teach you Perl. Here's a [great introductory course](https://www.ebi.ac.uk/training/online) on Perl to brush up.

Find out about [hosting a similar workshop](https://www.ebi.ac.uk/training/online) at your institute.

**Installing the API**

Before you start this course, you'll need to install the Ensembl APIs. The course was taught using Ensembl version 71, which can be downloaded from here [6], following the instructions on the page.

There's a video tutorial on installing the APIs here:

If you wish to use an up-to-date version of Ensembl, you can download the latest APIs from here [7]. Note that some of the results you get from your scripts will have changed and there is a very small chance that API methods will have changed, so your scripts will need to be slightly different. If in doubt, check the documentation for your methods [8].
Core

The Core module is presented by Magali Ruffier. Magali works on our Core team; they are responsible for maintaining the Ensembl databases and the Core API.

The Core module covers basic access to the Ensembl database, as well as how to retrieve basic genome features, such as genes and transcripts, repeat features and sequences, and meta-data about the genomes themselves.

There are 71 min of video in this module and the exercises should take you between three and six hours in total.
Core introduction

This first presentation introduces all the Ensembl APIs, which use Object Oriented Perl, and how to access the API documentation and schema.

Click here to watch the video [10] (16 min)
Exercise

Exercise 1

a) Find documentation for the Exon class in the Ensembl core code base. Which method would you use to retrieve the DNA sequence for an exon? What is the return type for this method?

b) Can you find a table which stores stable ids for transcripts? Which table stores DNA sequence? How many columns does this table have?

Magali explains the answer to this question in this 1 min video. The documentation pages she refers to are at page a, page bi and page bii.

Accessing the Ensembl database

In this presentation you will see how to use the Ensembl Registry in your scripts to access the database.

Click here to watch the video (6 min)

Exercise 2
Create a script which uses the method load_registry_from_db to load all databases into the Registry and prints the names of the databases loaded.

*Hint: Have a look at the Doxygen documentation for the Registry object and method load_registry_from_db ([http://www.ensembl.org/info/docs/Doxygen/index.html](http://www.ensembl.org/info/docs/Doxygen/index.html)).*

Magali explains the answer to this question in this [1 min video](#), You can also download her [sample script](#) and [output](#).

### Coordinate systems and slices

This presentation demonstrates the different coordinate systems in the Ensembl database and how to access regions of the genome using slices.

[Click here to watch the video](#) (12 min).

[Click here to view the slides](#)

### Exercise

**Exercise 3**

(a) Fetch all chromosomes for human. Determine their number and print the name and length for each of them.

*The number of chromosomes is probably not what you would expect! Why is this?*

(b) Use the gene stable id 'ENSG00000101266' to fetch a slice surrounding this gene with 2kb of flanking sequence.

*Hint: use the Ensembl API documentation to find an appropriate method in SliceAdaptor class which retrieves a slice given a gene stable id (pay attention to the method's arguments).*
(c) Fetch the sequence of the first 10MB of chromosome 20 and write it to a file in FASTA format. Print the number of genes in this region.

*Hint: Slice objects inherit from Bio::Seq so can be written to file easily using*

Bio::SeqIO, e.g.:

```perl
my $output = Bio::SeqIO->new(-file=>'filename.fasta',
                          -format=>'Fasta');
$output->write_seq($slice);
```

Magali explains the answers to these questions in this [5 min video](https://www.ebi.ac.uk/training/online) [23]. You can download her sample scripts and outputs:

a) sample [script](https://www.ebi.ac.uk/training/online) [24] and [output](https://www.ebi.ac.uk/training/online) [25]

b) sample [script](https://www.ebi.ac.uk/training/online) [26] and [output](https://www.ebi.ac.uk/training/online) [27]

c) sample [script](https://www.ebi.ac.uk/training/online) [28] and [output](https://www.ebi.ac.uk/training/online) [29]

---

**Ensembl features**

This presentation demonstrates the different coordinate systems in the Ensembl database and how to access regions of the genome using slices.

[Click here to watch the video](https://www.ebi.ac.uk/training/online) [30] (5 min)

[Click here to view the slides](https://www.ebi.ac.uk/training/online) [31]

---

**Exercise**

**Exercise 4**

(a) Get all the repeat features from chromosome 20:1-500k. Print out the name and position of each on the chromosome and the total number.
Hint: create a slice, retrieve repeat features on this slice

(b) Find which genomic region the RefSeq dna entry NM_000059.3 was mapped to. Print the name of the region and coordinates of the alignment on the genome as well as the name of the region and coordinates of the alignment on the RefSeq dna entry. Print the score and percentage identity for the alignment.

Hint: use DnaAlignFeatureAdaptor; use the core schema documentation as a guide to appropriate methods which correspond to columns in dna_align_feature table

A list of useful Feature methods is in the Appendix (see page Recap and Appendix).

Magali explains the answers to these questions in this 4 min video [32]. You can download her sample scripts and outputs:

a) sample script [33] and output [34]

b) sample script [35] and output [36]

Genes, transcripts, exons and translations

In this presentation you will learn how genes, transcripts, exons and translations are linked together in the Ensembl database and how they can be accessed via the Perl API.

Click here to watch the video [37] (6 min)

Click here to view the slides [38]

Exercise

Exercise 5

(a) Fetch gene CSNK2A1 and print the number of its transcripts and exons.
(b) For the above gene, get all the transcripts and list the number of exons in each and the translations.

(c) Why do the exon numbers not match?

Magali explains the answers to these questions in this 3 min video [39]. You can download her sample script [40] and output [41].

Hint: use GeneAdaptor method fetch_by_display_label; remember that not all transcripts have a translation

External references

This presentation shows you how to access links to Ensembl features in other databases, such as Uniprot and RefSeq.

Click here to watch the video [42] (6 min)

Click here to view the slides [43]

Exercise

Exercise 6

Retrieve a list of GO term IDs and term names linked to the gene with stable id ‘ENSG00000139618’.

- Use get_all_DBLinks with an external database name argument to restrict the number of xrefs returned.
- Ontology term data such as term name and definition are stored outside of the core database. Create an OntologyTerm Adaptor with the help of the Registry method get_adaptor using arguments: ‘Multi’ (species), ‘Ontology’ (database type), ‘OntologyTerm’ (adaptor type).
- For all xrefs returned by get_all_DBLinks use the OntologyTerm Adaptor to fetch the relevant term and print
its accession and name (xref display_id is the same as term accession).

Magali explains the answers to these questions in this 3 min video. You can download her sample script and output.

Recap and appendix

This final Core presentation wraps up the key points in this module and where to go for help and documentation. The slides also contain an appendix with some useful methods.

Click here to watch the video (3 min)

Click here to view the slides

Variation
Anja Thormann presents the Variation module. Anja works for the Ensembl variation team, importing and processing variation data into the Ensembl database from various sources.

In this module, learn how to access sequence and structural variants in the database, and how to retrieve data about them, such as locations, alleles and population genetics.

This module comprises 57 min of video and approximately 3-6 hours of exercises.

Variation data in Ensembl

This first presentation introduces variation data in Ensembl and shows you how to access data about known variants, using variant IDs.
Exercise

Exercise 1

a) Retrieve source and variation class for the following variations in human:

- rs55710239
- rs56385407
- COSM998
- CI003207

b) For SNP rs1333049 in human, retrieve the following information for each of its alleles:

- Allele
- Frequency*
- Population name*
- Submitter name (‘handle’)*

*if exists

Anja explains the answers to these questions in this 2 min video. You can also download her sample script and expected output.

Variation features

Variation features, which are variants mapped to the genome, are introduced in this presentation.
Exercise

Exercise 2

a) Retrieve all variation in human located on chromosome 13 from 48985833 to 48987289. Get:

- Variant name
- Alleles (e.g. ‘A/C’)
- Reference allele (‘A’)
- Location (Chromosome:Start–End)

b) Retrieve genomic location and alleles for the following variants:

- rs7107418
- rs671
- rs17646946
- rs4988235

Anja explains the answers to these questions in this 3 min video. You can also download her sample script and expected output.

Structural variation

This presentation introduces structural variants in Ensembl.
Exercise

Exercise 3

a) For structural variation nsv428936 reported in dog get:
   - Structural variation class
   - Study name and description
   - Coordinates (check for breakpoint locations)

b) Get names and variation classes of its supporting evidences.

c) (Do a similar analysis with SV esv234231 in human.)

Anja explains the answers to these questions in this 3 min video. You can also download her sample script and expected output.

Variation consequences

This presentation shows how you can use the Ensembl APIs find out how variants affect genes and transcripts, including SO consequence terms.
Exercise

Exercise 4a

a) Fetch all transcript variations (germline and somatic) in transcript ENST0000000001008 in human and retrieve the following information:

- Variation name
- Consequence type (most severe)
- Amino acid change* 
- Position in cDNA* and position in translation*

b) In a second attempt filter for transcript variations of consequence type: 'missense_variant'

* if information exists

Anja explains the answers to these questions in this 4 min video [66]. You can also download her sample script [67] and expected output [68].

Variation consequences continued

This presentation tells you about some of the more detailed information you can get about variation consequences.

Click here to watch the video [69] (4 min)

Click here to view the slides [70]
Exercise

Exercise 4b

Retrieve all coding TranscriptVariationAlleles for transcript ENST00000001008 in human and get for each TVA:

- Allele string
- Codon change
- Amino acid change
- SIFT and PolyPhen predictions*

**HINT:** We are only interested in information for the alternate allele. You need to fetch TranscriptVariation objects first.

*if information exists

Anja explains the answers to these questions in this [2 min video](#). You can also download her [sample script](#) and [expected output](#).

Phenotype

In this presentation, you will find out about retrieving phenotype data from Ensembl.

[Click here to watch the video](#) (3 min)

[Click here to view the slides](#)

Exercise
Exercise 5

Find all phenotypes associated with the human SNP named rs12913832 and give:

- Phenotype description, source and p?value
- Find frequencies in 1000 Genomes populations for the (risk) allele associated with phenotype ‘Eye color’ [sic].

*Hint: You might find the perl split function useful: my ($name, $allele) = split(‘?’, ‘rs9894429?T’); 1000 genomes populations start with: ‘1000GENOMES’*

Anja explains the answers to these questions in this 2 min video [76]. You can also download her sample script [77] and expected output [73].

Linkage disequilibrium, variation sets and the VEP

This final Variation presentation looks at retrieving linkage disequilibrium data via the Ensembl API. It also introduces the variation sets and Variant Effect Predictor (VEP [78]), as well as providing a summary of all the variation studied.

Click here to watch the video [79] (5 min)

Click here to view the slides [80]

Exercises

Exercises 1

1. Regulatory Features: cell type specific data.

a) Using the human DB, fetch all the cell type specific regulatory features with stable ID 'ENSR00000623613'.
b) Print out the stable ID, bound_start/end and start/end values, name of the cell- and feature type for each.

*Hint: To get all the cell type specific RegulatoryFeatures use the fetch_all_by_stable_ID method.*

2. Regulatory Features: What RegulatoryFeatures are near the oncogene **BRCA2**?

a) Create a script which fetches all the RegulatoryFeatures within 1kb of the gene.

b) Print out their stable IDs, bound_start/end and start/end values, name of the cell and feature types.

*Hint: Use fetch_all_by_external_name with 'BRCA2' to get the gene object.*

*Look at the arguments for fetch_by_gene_stable_id, or use the Gene->feature_Slicemethod and Slice->expand methods.*


a) Now fetch just the ENSR00000623613 MultiCell feature.

b) Print out the display_label, start/end values of all the evidence features.

c) Compare with the start/end values of the regulatory feature itself.

*Hint: By default the fetch_by_stable_id method returns just the MultiCell features.*

Nathan explains the answers to these questions in this [11 min video](#). You can download his sample scripts and outputs:

1. sample [script](#) and [output](#)

2. sample [script](#) and [output](#)

3. sample [script](#) and [output](#)
The Regulation module is taught by Nathan Johnson. Nathan works for the Ensembl regulation team, who import data about gene regulation from sources such as ENCODE, and incorporate it into the Ensembl regulatory build, producing regulatory features and segmentation.

This module covers the different kinds of data that Ensembl use in the regulatory build, how this build takes place and how you can access it via the API.

There are 84 min of video in this module and the exercises will take 3-6 hours to complete.
The Ensembl regulatory build

The Ensembl regulatory build is introduced in this talk, including how the data is produced in the lab and how it is integrated to produce regulatory features and segmentation. How to access features and segments via the Ensembl API is introduced.

Click here to watch the video [89] (39 min)

Click here to view the slides [90]
Regulation DataSets

This presentation introduces some of the datasets in Ensembl regulation.

Click here to watch the video [91] (8 min)
Exercises

Exercises 2

1. DataSets

Datasets are a meta container for data, grouping the data obtained by an analysis (FeatureSets) to the underlying raw data (ResultSets).

a) Create a script which fetches all available DataSets for Human.

b) How many are there?

c) Now get the 'RegulatoryFeatures:MultiCell' data set and print the display label of the product feature set and all the supporting sets.

*Hint: Use the DataSetAdaptor methods.*

2. FeatureSets

Feature Sets hold processed data or features i.e. peak calls or the output of a high level analysis e.g. the Regulatory Build.

a) Print the name of the feature sets for the Human 'GM12878' cell type.

b) Print the name of the feature sets for the Human 'CTCF' feature type.

c) Is the Human FeatureSet 'VISTA enhancer set' associated to any cell type or feature type?

d) Trick question: Get the supporting data for the VISTA FeatureSet.

*Hint: Most adaptors have a fetch_by_name method.*

`DataSetAdaptor->fetch_by_product_FeatureSet` will fetch the Dataset containing the supporting/raw data for a FeatureSet.
Nathan explains the answers to these questions in this 7 min video [93]. You can download his sample scripts and outputs:

1. sample script [94] and output [83]

2. sample script [95] and output [96]

**AnnotatedFeatures**

In the presentation, access to the supporting evidence behind regulatory data is introduced.
Exercises

Exercises 3

1. Annotated Features

Annotated Features represents the results of an analysis of raw or processing signal data. These correspond to regions in the genome enriched for specific events (like TF binding or Histone Marks) i.e. they are 'peak calls'.

Compare the number of annotated features in the region Y:5000000-40000000 between the Human feature sets:

K562_DNase1_ENCODE_Duke_SWEml_R0025_D150

HepG2_DNase1_ENCODE_Duke_SWEml_R0025_D150

What are the differences and why?

2. Motif Features

Motif features represent putative binding sites based on alignments of PWMs from JASPAR. MotifFeatures are always associated to AnnotatedFeatures representing Transcription Factor (TF) Binding. More information about how we integrate these into the regulatory build process can be found here.

Get the 'motif' regulatory attributes associated to the Human Regulatory Feature ‘ENSR00001227187’. Print their properties.

*Hint: use 'motif' as a parameter for regulatory_attributes.*

Print the properties of the annotated features associated to the motif feature.

3. Binding Matrices and motif strength

Each MotifFeature is associated with a PWM, which are represented by the ‘BindingMatrix’ class. The MotifFeature score represents the relative binding affinity with respect to the PWM defined in the BindingMatrix.

Using the Motif feature obtained in exercise 2, get the associated Binding Matrix and print some details.

Check potential effect of changes in the sequence of the motif feature on the relative strength of that motif feature.
Check the GERP conservation scores along the motif. Compare with the JASPAR matrix.

Nathan explains the answers to these questions in this [11 min video](#). You can download his sample scripts and outputs:

1. sample [script](#) and [output](#)
2. sample [script](#) and [output](#)
3. sample [script](#) and [output](#)

**Compara trees**

This is the first of two comparative genomics modules; this one is presented by Matthieu Muffato. Matthieu is part of the comparative genomics team at Ensembl, comparing Ensembl genes pairwise to produce gene families, gene trees and infer gene homology.

In this module, you will learn how to get gene trees and homologous genes from the Ensembl database, including gene and protein alignments.

This module is made up of 72 min of video and approximately 2-4 hours of exercises.
Introduction and getting gene members

Ensembl comparative genomics is introduced and fetching gene tree members is shown in this presentation.

Click here to watch the video [107] (20 min)

Click here to view the slides [108]

Exercises

Exercises 1
1. Print the sequence of the Member corresponding to SwissProt protein O93279.

2. Find the Member(s) for the human ncRNA gene(s) *FAM41C*.

3. Find and print the sequence of all the peptide Members corresponding to the human protein-coding gene(s) *FRAS1*.

Matthieu explains the answers to these questions in this [6 min video](#) [109]. You can download his sample scripts and outputs:

1. sample [script](#) [110] and [output](#) [111]

2. sample [script](#) [112] and [output](#) [113]

3. sample [script](#) [114] and [output](#) [115]

## Alignments and families

This presentation shows you how to get alignments between genes, and how to extract protein families.

[Click here to watch the video](#) [116] (9 min)

[Click here to view the slides](#) [117]

## Exercises

### Exercises 2

1. Get the multiple alignment corresponding to the family with the stable id ENSFM00250000006121.

2. Get the families predicted for the human gene ENSG00000139618. What do you notice?
Matthieu explains the answers to these questions in this 9 min video [118]. You can download his sample scripts and outputs:

1. sample script [119] and output [120]

2. sample script [121] and output [122]

**Gene trees**

This presentation shows how gene/protein trees are produced in Ensembl, and how to fetch them via the Ensembl API.
Exercises

Exercises 3

1. Print the protein tree with the stable id ENSGT00390000003602.

2. Print all the members of the tree containing the human ncRNA gene ENSG00000238344.

3. Count the number of duplication events in the tree of the zebrafish protein-coding gene ENSDARG00000003399

Matthieu explains the answers to these questions in this 6 min video. You can download his sample scripts and outputs:

1. sample script and output
2. sample script and output
3. sample script and output

Homology inference

This presentation shows how homologues are inferred from gene trees in the Ensembl database. Learn how to access these homologues.

Click here to watch the video (6 min)
Exercises

Exercises 4

1. Get all the homologues for the human gene ENSG00000229314.

2. Count the number of “one2one” homologues between human and mouse.

3. Find the human orthologues of ENSMUSG00000004843 and ENSMUSG00000025746. For each homology, display the alignment and the dn value. Comment on the divergence.

Matthieu explains the answers to these questions in this 7 min video. You can download his sample scripts and outputs:

1. sample script and output
2. sample script and output
3. sample script and output

Compara alignments
This module will teach you how to access alignments in the Ensembl database, including how to find syntenic regions and conservation scores.

There are 50 min of video in this module, and the exercises are expected to take you around 2-4 hours.

The alignment database

This presentation introduces whole genome alignments in Ensembl and takes you through some of the tools for accessing alignments in the database.

During the talk Stephen takes you through a sample script which you can download here. [141]
Exercises

Exercises 1

1. 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.

   Hint: First you will need an adaptor of type "GenomeDB". Then use the fetch_all() method from the GenomeDB adaptor to bring back GenomeDB objects (these will be returned as an array-ref). Then get the name, assembly version and genebuild version from these GenomeDB objects.

2. A DnaFrag represents a top-level SeqRegion in the Compara database. Print all the DnaFrags for chimp.

   Hint: First you will need an adaptor of type DnaFrag. Then use the fetch_all_by_GenomeDB_region() from the adaptor to bring back all the dnafrags associated with a region (the region you want is "chromosome").

3. The MethodLinkSpeciesSet is a central component in the Compara database, it stores information connecting the various analyses (method_link_type) with a set of species (species_set).

   a) (ii) Print the total number of MethodLinkSpeciesSet entries stored in the database.
Hint: use the MethodLinkSpeciesSet adaptor fetch_all() method.

(iii) Print a unique list of method_link_types and a count of their number in the database.

Hint: With the MethodLinkSpeciesSet adaptor use the "method->type" method to get all the "method_type" entries in the database. In order to get a unique set of method types, use the returned values from the "method->type" method as keys in a hash (and increment this for the total number of entries of that type).

b) Print a list of the species and their internal ids (dbIDs) for the 12 eutherian mammal EPO alignments.

Hint: use the MethodLinkSpeciesSet adaptor fetch_by_method_link_type_species_set_name() method, and then the MethodLinkSpeciesSet (object) "species_set_obj()->genome_dbs" method (this brings back a list-ref of genome_db objects). The method_link_type is ‘EPO’ and the species_set_name is ‘mammals’.

Stephen explains the answers to these questions in this 12 min video [144]. You can download his sample scripts and outputs:

1. sample script [145] and output [146]
2. sample script [147] and output [148]
3a. sample script [149] and output [150]
3b. sample script [151] and output [152]

Alignment blocks

In this presentation, you are shown how to access alignment blocks in the Ensembl database.

Click here to watch the video [153] (9 min)

Click here to view the slides [154]

Exercise
**Exercise 2**

A GenomicAlignBlock represents an alignment between two or more regions of genomic DNA. Within these blocks every region of genomic DNA is represented by a GenomicAlign object.

a) Print the LASTZ-NET alignments for pig chromosome 15 with cow (using pig coordinates 105734307 and 105739335).

*Hint: use the MethodLinkSpeciesSet adaptor `fetch_by_method_link_type_registry_aliases()` method (eg. $mlss_adaptor->fetch_by_method_link_type_registry_aliases("LASTZ_NET", ["pig", "cow"]).*

you will need a *core* slice for pig chromosome 15.

get a GenomicAlignBlock adaptor and use the `fetch_all_by_MethodLinkSpeciesSet_Slice()` method to retrieve the genomic_align_block object(s).

use the GenomicAlignBlock restrict_between_reference_positions() method to "restrict" the genomic_align_block between base pairs 105734307 and 105739335.

finally print the alignment using the GenomicAlignBlock `get_SimpleAlign()` method [155] (use AlignIO to format the output alignment).

b) Change the above example so that it prints the 13-way eutherian mammal (EPO) multiple alignments.

*Hint: use the MethodLinkSpeciesSet adaptor `fetch_by_method_link_type_species_set_name()` method (the method_link_type is "EPO" and the species_set_name is "mammals").*

Stephen explains the answers to these questions in this 6 min video [156]. You can download his sample scripts and outputs:

a) sample script [157] and output [158]

b) sample script [159] and output [160]

**Your feedback**

Please tell us what you thought about this course. Your feedback is invaluable and helps us to improve our courses and thus enhance your learning experience.

**Get help and support on the Ensembl API**
Questions or any comments about Ensembl or this course? helpdesk [at] ensembl.org (Contact Helpdesk).

Ensembl information

- For more information on the data in Ensembl and where it comes from, have a look at our QuickTour [161], Browsing chordate genomes course [162] and Filmed browser course [163].
- See if your question has already been answered in the Ensembl FAQs [164].
- Have a look at the Ensembl Glossary [165].

Using the API

- View the full database structure for each of our modules: Core [166], Variation [9], Compara [106] and Regulation [88].
- Follow the tutorials for each module: Core [167], Variation [168], Compara [169] and Regulation [170].
- Complete documentation for all Ensembl API methods can be found on our Doxygen site [171].

Support

- If you are using Ensembl programmatically, our dev list [172] is a community of Ensembl developers where you can ask and answer questions.
- If you thought this course was useful, and you and your colleagues would like to experience it in the flesh, host an Ensembl API workshop [2] at your institution.

Contributors

This course was put together using talks and exercises by:

- Magali Ruffier
- Anja Thormann
- Nathan Johnson
- Matthieu Muffato
- Stephen Fitzgerald

We would like to thank the participants of the course, who allowed us to film.

A big thanks to Mark Adams from the EBI Training team for filming and processing the lectures.

The online course was compiled by Emily Pritchard.

Contributor profiles [173]
Emily Perry [1]

EMBL-EBI
Ensembl Outreach Project Leader

Emily is the Outreach Project Leader for Ensembl: she is responsible for the team that teaches workshops, creates training materials and help pages, manages social media, answers helpdesk queries and aids development of new tools for the resource. Emily started at EMBL-EBI as an Ensembl Outreach Officer in September 2012 and became the Project Leader in March 2015. Before working at EMBL-EBI, Emily did her PhD in molecular biology at the MRC Human Genetics Unit in Edinburgh, then worked for the University of Edinburgh's SCI-FUN group, touring Scottish secondary schools with an interactive science roadshow.

Funding

Ensembl is a joint project of the Wellcome Trust Sanger Institute and the European Bioinformatics Institute. The Wellcome Trust [174] provides majority funding for the project (grant numbers WT062023 and WT079643) with additional funding for specific project components from the National Human Genome Research Institute [175] (U01HG004695, U54HG004563 and U41HG006104), the BBSRC [176] (BB/I025506/1) and the European Molecular Biology Laboratory [177]. Additional support for specific project components as specified: "The research leading to these results has received funding from the European Community's Seventh Framework Programme [178](FP7/2007-2013) under grant agreement n° 222664. ("Quantomics" [179]). This Publication reflects only the author's views and the European Community is not liable for any use that may be made of the information contained herein"; "The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 200754 - the GEN2PHEN [180] project."; "The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 282510 - BLUEPRINT [181]." Rat genomics resources receive additional support as specified: "The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement N° HEALTH-F4-2010-241504 (EURATRANS)."
[173] https://www.ebi.ac.uk/training/online/glossary/profiles
[174] http://www.wellcome.ac.uk/