



Introduction to Biological Big Data Exercises

Two Day Version 2025-12



Licence

This manual is © 2021-25 Simon Andrews

This manual is distributed under the creative commons Attribution-Non-Commercial-Share Alike 2.0 licence. This means that you are free:

- to copy, distribute, display, and perform the work
- to make derivative works

Under the following conditions:

- Attribution. You must give the original author credit.
- Non-Commercial. You may not use this work for commercial purposes.
- Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a licence identical to this one.

Please note that:

- For any reuse or distribution, you must make clear to others the licence terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.
- Nothing in this license impairs or restricts the author's moral rights.

Full details of this licence can be found at

<http://creativecommons.org/licenses/by-nc-sa/2.0/uk/legalcode>



Introduction

In these exercises we are going to explore a number of different online resources to see what we can find out about one specific gene. This will give you the chance to play around in some of the most useful repositories of biological information and see how they might be able to help you in your future work.

The gene we are going to use as our example is human TEC – a protein tyrosine kinase.

Exercise 1: Genome Exploration

We are going to see what is known about the location, transcripts and structure of the TEC gene using two of the biggest genome browser sites.

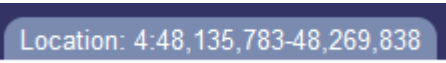
Ensembl

Go to www.ensembl.org and select the human genome

Use the search function to find the gene page for TEC (it should be ENSG00000135605)

- How many different transcripts are listed for TEC?
- How many encode a protein?
- Of the protein coding transcripts how many are likely to be removed by nonsense mediated decay?
- How long in both transcript base pairs and protein amino acids is the main splice form?

Location: 4:48,135,783-48,269,838

Click on the  tab at the top to take you to the genome browser view of this gene. This will display a track based view of a region of the genome. Many of the tracks shown will be present twice since there will be a separate track for the top and bottom genome strands. Tracks at the top, above the central blue line are top strand (running left to right), those under the blue line are bottom strand (running right to left). Tracks with no direction are drawn right at the bottom.

In the chromosome view you can use the controls to zoom in/out and move along the chromosome



When moving along a chromosome it's often useful to click on the double arrow to go into drag mode, and then drag in the main view to move it along.

- Is TEC close to the centromere of chr 4?
- Which genes flank TEC on either side? Answer this for both conventional protein coding genes and for small RNA genes.
- Look at the structure of the TEC transcripts. From which end (5' start or 3' end) are most of the splice variants most truncated?

[Configure this page](#)

You can click on [Configure this page](#) to select the tracks you want to see in the display. You'll get a main list of "Active Tracks" initially, and then you can select from a large number of tracks on the left which you can add.

Turn off some of the default tracks you're not using at the moment. Turning off unneeded tracks will make the loading of new views *much* quicker.

Add a track of CpG islands from the "Simple Features" section.

- Is there a CpG island in the promoter of TEC?

Move back to the TEC gene tab, and we can have a look at the range of species where an ortholog of TEC is found. Click on the Gene Tree from the menu on the left and have a look at the species where TEC has been found. You can see the species on the tree, and on the right a summary of the exon structure.

- In which placental mammals has TEC been observed? You will need to click on the placental group and expand the sub-tree to see the individual animals.
- Which phylogenetic group has a large deletion at the start of the gene (blank area on the exon structure picture)? Is this region missing from all animals in this group?

Click on the Orthologs item in the main left menu to get a table of all of the TEC orthologs. Find the ortholog for the crocodile (*Crocodylus porosus*)

- What type of ortholog is this (1-to-1 or 1-to-many)?
- How conserved is the DNA sequence (percentage identity)

Click on the Compare Regions link to open up a multi genome view of the two species around the TEC gene.

- Can you see conserved blocks (the light green shading) covering all of the exons of human TEC?

Expand the view to include the flanking genes in both species (drag a box in the Region Comparison section covering the genes you want to see and select to realign). You need to do this separately in both species.

- Do all of the surrounding genes have orthologs in the two species?
- Are there any obvious changes in structure between the transcripts in this region?

Go back to the main TEC gene tab and then select the page for the primary transcript (ENST00000381501.8)

- How many exons does this transcript have?
- How many of them form part of the coding sequence?



From the left hand menu select the Exons sequence view. Configure the page and change the options to turn off the highlighting of variants.

Find the start of the translated region (blue letters, should start with ATG).

- The Kozak consensus sequence is gcc Rcc ATG g (where the ATG is the first translated base). How well does the sequence at the start of the TEC CDS match this consensus?
- Does the coding sequence run through into the last exon?
- Which is the longest intron, and how long is it?

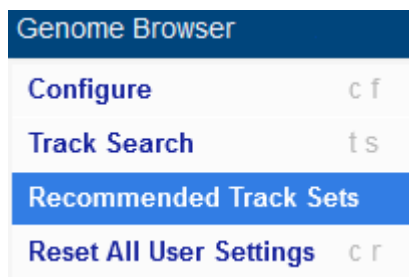
Move to the cDNA sequence section of the Transcript tab

- Download a fastA file of the TEC-201 transcript (turn off all of the other sequence types it offers to give you alongside the cDNA sequence)

UCSC Browser

Go to <http://genome.ucsc.edu/> and click the link to their genome browser in the main “Tools” section of the page (not from the menu at the top). You may be prompted to connect to a local mirror site – it’s fine (good indeed) to use the closest mirror.

Make sure you’re using the hg38 human genome, and find the TEC gene and have a look at the default information you get. You can be sure to get the default tracks by going to



then

Return to Default browser tracks.

The basic controls you need to know are:

1. You can click and drag in the main view to move left/right
2. You can hold control whilst dragging to select a region to zoom in to
3. You can zoom in out using the buttons at the top

zoom out 1.5x 3x 10x 100x

Zoom out 10X from the initial view of TEC

- Which genes can you see?
- How many splice variants do they have?

Look at the tracks showing conservation across species

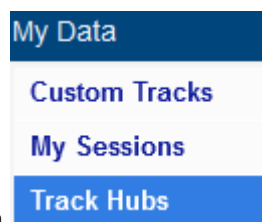


- Which part of which gene has the highest overall conservation?
- Do you observe a loss of conservation in intergenic regions (the bits between genes)?

You can right-click on a track to get some options for how it is displayed and to turn it off completely.

- Hide the RefSeq curated gene track
- Change the ENCODE cCRE track to being a 'squish' presentation

One of the biggest benefits to using UCSC is the amount of additional data you can incorporate from other sources. Tracks from other groups come from "track hubs" which you can connect to to see the additional information.



Press on **Track Hubs** to see the selection of available track hubs. Have a read down the list to see what sorts of information you can get at and add to your UCSC browser.

- Find the Methbase2 hub, which contains DNA Methylation data and connect to it. Your session will restart, so re-find the TEC gene and you should see a load of new methylation tracks.
- Zoom out from TEC and see if you can find a region near the gene where the methylation level drops substantially
- Is the extent of the unmethylated region the same in all of the tissues which are shown?

BioMart

Finally in this section we are going to have a look at the BioMart interface which allows you to do large scale queries against genome annotation information. Although there is a stand-alone site for this at <http://www.biomart.org/>, practically speaking the easiest way to use it is to use the implementation at Ensembl, <http://ensembl.org/biomart/martview/>.

In this section of the exercise we're going to get you to export a lot of information around all of the human genes on chromosome 4 (the chromosome the TEC gene is on).

To begin, go to the URL above and select that you wish to analyse Ensembl Genes, and that you want to work with the latest human genome.

Filters

The next step will be to pick the genes about which you wish to find information. To do this you will need to apply filters. You should automatically be taken to the filter section of BioMart as soon as you select the genome, but you can get back to this part at any time by clicking on the links on the left of the page.



Dataset
Human genes (GRCh38.p13)
Filters

The filters we would like you to apply are:

- We only want to use genes on chr 4 (under the REGION section)
- We only want to see protein coding genes (under the GENE section)
- We only want to see genes which have an ortholog in the platypus (under the MULTI SPECIES COMPARISON section)

After you've made these choices you can click on the "Count" button at the top of the left hand menu to see how many genes remain after applying the filter.

Dataset 566 / 67128 Genes
Human genes (GRCh38.p13)
Filters
Chromosome/scaffold: 4
Gene type: protein_coding
Orthologous Platypus Genes: Only

Note that the exact number you see here will change with every Ensembl release as either the human or platypus gene sets get updated. Don't worry if the number you see doesn't match exactly as long as it's similar and the filter descriptions look right.

Attributes

Next you need to say what it is that you want to know about the genes you have selected. Again there are many different options for this and you can see them by clicking on the attributes link on the left.

<input checked="" type="radio"/> Features	<input type="radio"/> Variant (Germline)
<input type="radio"/> Structures	<input type="radio"/> Variant (Somatic)
<input type="radio"/> Homologues (Max select 6 orthologues)	<input type="radio"/> Sequences

You can export information about:

- Annotation features – any of the pieces of annotation for a gene or transcript
- The Structure of a gene – how it breaks down into transcripts, exons and CDS regions
- Homologs – the equivalent Ensembl identifier for homologous genes in other species
- Variants – the details and properties for germline or somatic variants
- Sequences – the underlying sequence (genome, transcript, CDS exons etc)

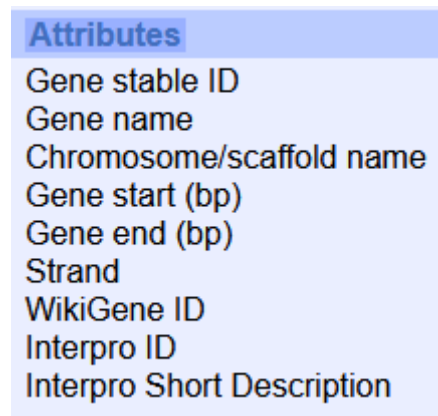
We are going to export only feature annotation information so you can choose Features under the attributes options.



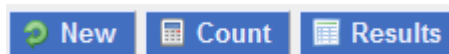
You can then see three additional groups of pieces of information you can add. You can pick as many of these as you like.

For our search the information we want to know is:

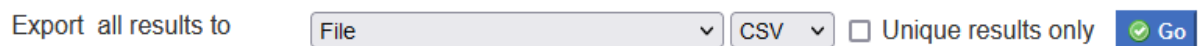
- The Ensembl Gene ID (Gene Stable ID)
- The Gene Name
- The gene's location (chromosome, start, end, strand)
- The WikiGene ID (under EXTERNAL)
- The Interpro ID and short description (under Proteins)



Once you've selected those you need to click on the Results button which is at the top left of the interface.



That should then show you a short preview of the data you've collected so you can check that it looks right. To download the full set of results you need to use the option at the top to save the results to a file.



If you save to a TSV or CSV file then you should be able to open the file directly in Excel to see all of the contents.

Sequence Query

Once you've completed the example above see if you can use BioMart to export out all of the spliced transcript (cDNA) sequences for the TEC gene. The sequences should be in fastA format and should just have the Ensembl transcript ID in their header. ie:

```
>ENST00000381501
ACTCTGGGGCGCTAGGCTCCGGACTCCGCGGCGCAGACTGCACCTCGCAGTCTCCCCAGG
TCCGCCCAGCAGCCGCGCTTCAGCCAGAATACTGGGATCTTCAGTGGCAGGAGGAGTAAT
CAGAAGACGGAGATGAATTTTAACACTATTTTGGAGGAGATTCTTATTTAAAGGTCACAG
CAGAAAAAGAAGACATCGCCCTTAACTACAAAGAGAGACTTTTTGTACTTACAAAGTCC
...
>ENST00000505452
AATACTGGGATCTTCAGTGGCAGGAGGAGTAATCAGAAGACGGAGATGAATTTTAACACT
```




ATTTTGGAGGAGATTCTTATTAAAAGGTCACAGCAGAAAAAGAAGACATCGCCCTTAAAC
TACAAAGAGAGACTTTTTGTACTTACAAAGTCCATGCTAACCTACTATGAGGGTCGAGCA
GAGAAGAAATACAGAAAGGGGTTTATTGATGTTTCAAAAATCAAGTGTGTGGAAATAGTG
AAGAATGATGATGGTGTTCATTCCCTGTCAAAATAAGTATCCATTTTCAGTCCACAAAGCAG
GGACCTGTGGGTGAAGAAGTTAAAAGAAGAAATAAAGAACAACAATAATATTATGATTAA
ATATCATCCTAAATTCTGGACAGATGGAAGTTATCAGTGTTGTAGACAAACTGAAAAATT
etc.



Exercise 2: Proteins, domains, structures

Protein Domains

Start by going to the SMART website <http://smart.embl-heidelberg.de> and look up the information for the TEC gene. For this you'll need to know the identifier for the TEC protein sequence. For TEC the protein access is P42680. You would have seen this in the Ensembl information for the TEC protein.

Name	Transcript ID	bp	Protein	Biotype	CCDS	UniProt Match
TEC-201	ENST00000381501.8	3661	631aa	Protein coding	CCDS3481	P42680

Put the accession code into the SMART search and see what domains are found for TEC. How many domains are there? Are they all globular domains or are there any disordered or low complexity linkers present?

For each domain read the information about it. Try to see whether it's a binding or catalytic domain, and if it's a binding domain see what it binds to (DNA, Protein, Cofactors etc).

On the domain plot you should see some vertical lines running down the image. What are they, and what is signified by their colour?

Now try to find Interpro domains which may contain details of specific binding or active site pockets. The interpro search is at <https://www.ebi.ac.uk/interpro/> but you'll need the TEC protein sequence to do the search, so go back to Ensembl to get this (it's only one sequence so you can do it from the main Ensembl web site not through BioMart). The interpro searches can take a little while to run so feel free to move on to the next part whilst this is running and come back to look at the results later.

Once you have the results have a look at the summary figure on the Overview and see whether you can see both large domains (such as you got from SMART and Pfam), but also smaller hits within those large domains.

To get more information click on the Entries tab at the top of the results to get a tabular view of the hits.

See if you can use these results to find where the active site of the Tyrosine kinase domain is and see what details are provided about it.

Protein Structures

Experimental Structures

We know that TEC is a multi-domain protein so it's unlikely that there will be a single structure covering all of the protein, but there may be structures for some of the individual domains.

Go to <https://www.rcsb.org/> and see what you can find. Search initially with the protein accession code for human TEC and see if you get any hits. If you do, which of the domain(s) do they cover.

Next you can try to be less specific and search for the TEC gene name to see if anything else comes up from other species. Just be aware that when searching you need to scroll down quite a long way in the search options to say that you're providing a gene name. Do you get any additional hits? Do they cover more domains than you saw before?



Have a look at the structures you've found. As well as the tabular and image information you can also select the 3D view of the protein and get an interactive view of the structure to play with.

You should have seen that there are still some domains for which you've seen no structural information. As a final search go to the "Advanced Search" option and use the sequence of the TEC protein to search against RCSB. Do you now get any additional hits. Why did these not come up before? Do they cover any domains which you didn't previously see?

Can you see any examples of structures which are complexed with either substrates or inhibitors to the catalytic active site. If so, see if the small molecule sits in the same position as you found the InterPro active site motif in the sequence.

Predicted Structures

In the experimental structures we found structures for some of the TEC domains, but always as isolated components, and not all of the sequence was covered.

Go to <https://alphafold.ebi.ac.uk> and look for the human TEC protein. Do you find a predicted structure for the whole protein? If so which parts are confidently predicted and which parts are more uncertain? Do these relate to the domain structure of the protein? You can make selections in the heatmap below the structure to highlight the relevant regions in both the 3D structure and the linear sequence.



Exercise 3: Reactions, Pathways, Functions

Biochemical Reactions for TEC

Before we get into more complex pathways or functional groups we can look at the reaction which is catalysed by TEC. This will be stored in the Rhea database, but it's not easy to find catalytic proteins there, only reactants and products. To find the correct reaction you either need to find the Enzyme Classification (EC) number of the protein, or go to a page which annotates the reaction directly.

One of the best resources for protein information is UniProt which collates information about all proteins and links out to a large number of other databases, including Rhea. If you look in the "General Identifiers" section of the Ensembl gene page you should find links to lots of other data sources, including UniProt.

Find the uniprot entry for TEC and you should see the reaction for its catalytic activity and a link to the relevant reaction in Rhea. Have a look and see what reactants and products are in the reaction which is catalysed.

You should also be able to find the EC number for the reaction.

Take a look at the Rhea reaction and see whether there are proteins listed which catalyse the reverse reaction to the TEC kinase. If you can't see any then look at the reactions involving O-phospho-L-tyrosyl-[protein] and see if you can find a corresponding reverse reaction that way.

Pathways

Reactome

Got to the Reactome database at www.reactome.org and search for TEC. You'll get many hits but you should be able to select just the human protein.

Expand the tree of hits to the pathway browser and see the range of pathways that this protein is involved in. You should see entries under both immune system and signal transduction.

Have a look at the pathway browser for the three major pathway branches which mention TEC and try to determine the following:

- Which protein complexes with TEC after IL3 stimulation. Which domain in the protein mediates this interaction?
- In the innate immune response which lipid is involved in the reaction involving TEC? What is the endpoint of the chain of reactions that this is part of?
- Under the signal transduction tree, which membrane complex does TEC (as well as several other tyrosine kinases) associate with?



KEGG

We can also look in KEGG to see which of its pathways TEC turns up in. From the main KEGG page at <https://www.genome.jp/kegg/> go to KEGG GENES and search for TEC, find it on the list of hits (it's not the top hit – it should be entry 7006).

From the gene page you should be able to link through to the two KEGG pathways which involve TEC. See which these are, and whether you can see the correspondence to the pathways you saw previously in reactome.

Gene Ontology

Finally, in this section we're going to take a look at the Gene Ontology and see what information it holds on TEC.

Go to the main Gene Ontology web site at <http://geneontology.org/> and search for TEC. You'll get a lot of hits, so add a filter to only look at the hits from humans. You should see a set of around 39 Gene Ontology hits.

From the list see if you can see a group which covers the main catalytic activity of TEC. Find the most specific catalytic category and then open the page for that category. Have a look at the Inferred Tree View to get a text representation of the hierarchical view from that category. Also try going to Graph Views and construct a figure in QuickGO so you can see the structure of the annotations from that category all the way back to the root of that part of the GO tree.

On the ontology hits you can filter by Ontology (aspect) – this allows you to see hits in the three branches of the tree

- P – Biological Process
- F – Molecular Function
- C – Cellular Component

Have a look to see which process categories are listed and see if they seem familiar from the pathway analysis you did before.

Look at the cellular component hits to see where TEC is found and acts. Try selecting some of the references so you can see the sources of information on which the assignments are based.



Exercise 4: Regulation and Interactions

Transcription Factors

UCSC Browser

Go back to UCSC browser, find TEC and then adjust the zoom to look more closely at the promoter. In the set of available tracks look under the regulation section and turn on the ENCODE cCRE regulation track if it's not there already. Look at the promoter region and see if ENCODE shows a promoter like signature in that region.

Now turn on the Re-MAP ChIP-Seq track in “squish” mode. This will add a track of features of ChIP-Seq peaks for transcription factors binding in this region. It will be quite large!

This sort of prediction can be useful, but is also prone to over-predicting, so just because you see a bound factor in an open region doesn't necessarily mean that you can be sure that that factor is actually regulating the gene.

JASPAR

One of the factors with a strong hit in TEC is Runx3. Take a look in the Jaspar database <http://jaspar.genereg.net> and find the entry for this transcription factor and see what the binding motif looks like.

Hocomoco

Find Runx3 in Hocomoco (<https://hocomoco13.autosome.org/>) and see what the motif there looks like. Is it the same as in Jaspar? If not, why not? Have a look at the subfamily of factors to which Runx3 belongs – how many species has this been detected in? How many other members of this family are there?

Factorbook

Find Runx3 in factorbook (<https://www.factorbook.org>). Have a look at the description they provide for it. One of the other datasets they provide is a graph to show in which tissues each factor is expressed. Have a look where Runx3 is highly expressed. Does this make sense with what you saw before about the pathways where TEC is important?

In ReMap2022 (<https://remap.univ-amu.fr>) find Runx3, and see how many experimental datasets they show which have measured this genome wide. Which cell type(s) were used. How many peaks were detected for this factor?

Interactions

Now we can look at some of the sources of information which detail the interactions which TEC has with other proteins or factors.

BioGRID

We can start with BioGRID (<https://thebiogrid.org>). Find TEC in their search and have a look at the list of interactions they have. You want to look at the evidence column on the right hand side of the list. Ideally you want to see low throughput studies (the lighter colours), and preferably for an interaction to have been confirmed by multiple studies.



Based on this how many interactions have solid evidence and what are they to. Do any of them fit with what you saw in the pathways section? Are there any proteins which you'd have expected to see which either don't appear or have less evidence than you might have expected?

If you have a look at the details of some of the interactions (click on the evidence count box) you will see that for some there are explanations of the function of the interaction. Have a look and see what some of these are.

String

String (<https://string-db.org>) has one of the nicest collections and presentation of interactions. Find human TEC and then look at the network of interactions they show. The number of lines connecting the points indicates the types of evidence used to link different proteins. In general the more lines the better!

Have a look at the set of proteins and click on them to see some information about them. Do you think that this set fits better with what you saw in the pathway databases for TEC than the results from BioGRID?

As well as the interaction network view you can also click on "Viewers" and look at the co-expression matrix for TEC and the proteins with which it is predicted to interact. How well do the occurrences of these proteins match? Is the co-occurrence better if you just look in human, or if you expand to other species?

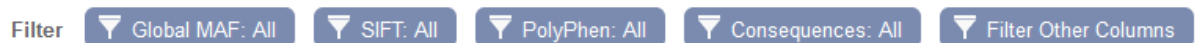


Exercise 5: Variant Exploration

Variants in Ensembl

As always, the easiest place to start looking for information is Ensembl. From the TEC gene page select the Variant Table view to bring up a list of all of the variants which have been recorded for this gene.

You will often find that the initial view of variants isn't particularly useful since the size of the table can be huge. There are some options above the table to filter the list to get to the subset of variants which are actually of interest to you.



The main filters you'd want to apply would firstly be on the consequences of the SNPs. Let's assume that we are only interested in SNPs which affect the protein sequence, either by changing or truncating it. To do this select the "Consequences" button and select "PTV or Missense".

Secondly, we can filter by the impact scores. There are two scores recorded by Ensembl, SIFT and PolyPhen. Both of these are measures calculated on the basis of the chemical change in the amino acids (ie a change from uncharged to acidic would be more important than between two uncharged amino acids), and also on the degree of conservation of the position being mutated – more conserved positions are likely to be more functionally important. You will find that these can be useful to get to the more interesting variants, but that some variants may not yet have a score calculated so will have a blank measure.

To cut down the full list apply filters to only see variants where both the SIFT and PolyPhen scores are above 0.6. This should get you a fairly small list.

For the variants which remain which position is mutated and what are the amino acid changes? If you look back at the results from the domain analysis can you see which of the domains are affected? Do any of them affect the catalytic site of the kinase?

dbSNP

The most interesting SNP in TEC is rs749636109. Find this variant in dbSNP (<https://www.ncbi.nlm.nih.gov/snp/>) and then click through from the search results to the SNP page.

Are there any reported clinical consequences for this SNP?

Look at the frequency with which the SNP has been observed. How prevalent is it in the population? In which geographic region(s) has it been observed?

ClinVar

The ClinVar database reports variants with clinical relevance. We know from our investigations above that the most significant SNPs in TEC are not associated with a clinical outcome. We can see if there are other variants which are more relevant though.



Go to <https://www.ncbi.nlm.nih.gov/clinvar/> and search for TEC. Are there any clinical variants which affect this gene? If so what sort of variants are they, and how specifically are they tied to TEC?

OMIM

On a slightly wider scale we can look to see what is known about the phenotypes which are associated with TEC by looking in the OMIM database (<https://www.omim.org>). Look up TEC and see what is known.

Are there any drugs which are known to target TEC? If so, what diseases are they used to treat?

Is there an knockout animal model for TEC? If so, which species and is it only TEC which is affected? What phenotypic effects are seen in the knockout?

Are there any papers which have described the structure and function of TEC in detail?

COSMIC

Rather than looking at naturally occurring variants within the general population we can instead look at somatic variants which appear in cancer cases. Go to <https://cancer.sanger.ac.uk> and look up TEC.

In which domain of the gene are the largest number of somatic mutations seen?

Which tissue types are the highest frequencies of mutation seen?

Which types of mutation (functional consequence) are most frequent?

Are there any specific publications associated with the role of TEC, or similar kinase in cancer?



Exercise 6: Finding data in GEO

Although you can find data directly in GEO, in most cases the route to getting to the data is finding a paper which describes an interesting piece of biology you want to pursue. We can search for interesting papers in the pubmed database (<https://pubmed.ncbi.nlm.nih.gov/>)

We are going to look for information relating to the Prox1 gene, specifically we'd like to find out what effect knocking this gene out in embryonic tissue has.

In pubmed search for "prox1 embryonic knockout transcriptome" and find a paper which is obviously based around RNA-Seq data (it may not be the top hit), and which includes all of these terms.

Follow the link to the paper and see if you can get the full text (probably as a PDF). See if they give a GEO accession (GSEXXXX) for the data they use. If they do is it data they created, or public data they re-analysed?

Search for the accession code you find in GEO (<https://www.ncbi.nlm.nih.gov/geo/>) and find the details for the dataset. Is the paper you found the original paper for this dataset? If not which paper first published it?

How many samples are included in this dataset? What experimental conditions do they represent and how many replicates of each condition are there?

Look at one of the Sample pages for this study in GEO. As well as raw data the authors are required to deposit quantitated data. What quantitation did they perform in this instance and what data have they provided?

Which type of sequencing platform was used to generate this sequence data?

From the main GEO study page follow the link through to the SRA run selector and look at the details of the files which have been deposited with this entry. How many runs (sequencing lanes) were produced for each sample (experiment)? Have a look at the first run in the list. How many reads (spots) are in this sample? How long is the read length?

From the GEO entry find the SRA database accession for this data. Take this and search for it in sra explorer <https://sra-explorer.info>. Can you see all of the runs you saw in the SRA run selector? Add the runs to your basket and then generate a list of download URLs where you could get the data if you wanted to download it all.

Finally put the GSE accession number into the text search (not the accession search) of <https://www.ebi.ac.uk/ena/>. Find the relevant study page and check that you can see the samples. See that you could click on the links to the individual fastq files to download them (but don't actually download them).



Exercise 7: Exploring FlowRepository

Here we will look at the data available in FlowRepository and download one of the available files and view it.

Go to <https://flowrepository.org>

Search for “PBMC” which is a type of standard blood sample (peripheral blood mononuclear cells), and includes T, B and NK cells.

From the original list of hits further filter for CD4 and find an experiment which describes the measurement of CD4 and IFN-g in monocytes. Look at the details for this submission.

Who submitted the sample?

What is the publication associated with this entry?

How many experimental conditions are in this dataset and what do they represent?

How complete is the metadata for this study (look at the MiFlowCyt score)?

Go to the downloads for this study and see which FCS files are available.

Exercise 8: Mass Spec Repositories

Protein Data in ProteomeExchange

Go to <http://www.proteomexchange.org/> and select the option to Access public data. You should see an interactive plot which shows you how many datasets have been deposited from different species, and using different types of spectrometer.

How many studies are there from Rat?

Use the search to find datasets coming from pituitary and find a dataset from human which profiled this in 2019.

Which of the underlying hosting databases contains the full dataset for this study?

Find the entry for this data in the underlying repository.

What type of Mass Spectrometer generated this data?

Which publication is associated with the data?

Have a look at the samples which were submitted as part of the study and see what information was recorded about each of them.

Click through to the FTP site associated with this data and check how many files you can see and what format they are in.



Metabolomic data in Metabolomics Workbench

Go to the main metabolomics workbench page at <https://www.metabolomicsworkbench.org/>

In the quick search at the top search for the accession ST000899.

You should find one match – click through so you can see it.

What was the purpose of this study? What biological material was collected for it? What experimental conditions does it contain and how many samples from each condition are there?

Select the option to Show Named Metabolites to see what compounds were detected in the study, remembering that these will be metabolites rather than proteins.

Note that the list of molecules is divided into sections based on the type of Ionisation which was used to detect them. Different molecules are efficiently detected using different types of ionisation (positive and negative) in different runs of the spectrometer.

From the list of metabolites find adipoylcarnitine.

Select it then press the button at the top to show the values for this metabolite. Then draw a bar graph

Bar graph by sample

Bar graph (samples)

of the levels in the different samples.

Remembering that the samples are in groups of 20, does it look like something interesting might be happening with this metabolite?

Boxplot

By factor

Go back and draw a boxplot by factor. Which of the conditions does this metabolite seem to be downregulated in?

We can do a larger scale analysis of the differential abundance of all metabolites between different conditions. Go back to the main study page and select **Perform statistical analysis**

We will construct a volcano plot of the results of a pairwise comparison of two conditions. This plots the p-value (y-axis) against the magnitude of change (x-axis). Select the volcano tool, and choose the Control as Group1 and Crohn disease as group 2 and run the analysis with the default options.

Have a look at the results, there is a table of results at the bottom of the page and you can click through to see the volcano plot and other summaries of the results. You should be able to see the adipoylcarnitine as an outlier along with several other metabolites.

Repeat the analysis on the same groups using the negative ion mode data instead of positive ion mode. Note how you get a different set of hits. In negative ion mode you should see that sucrose is a strong positive hit. Go back to the original full list of metabolites and find sucrose on the list and draw the barplot for it and check that you can see a strong increase in the crohn disease samples (21-40)



Exercise 9: Imaging Resources

BioImage Archive

First we're going to look at the more general data in the BioImage Archive.

Go to <https://www.ebi.ac.uk/bioimage-archive/>

Gel Images

Do a search for studies about Crispr deposited in 2020 (there should only be one).

Have a look at the gel images which are attached to this study.

Microscopy Movies

Do a search for "locomotor" and find the dataset which observes movement in wild type and mutant *C.elegans* worms.

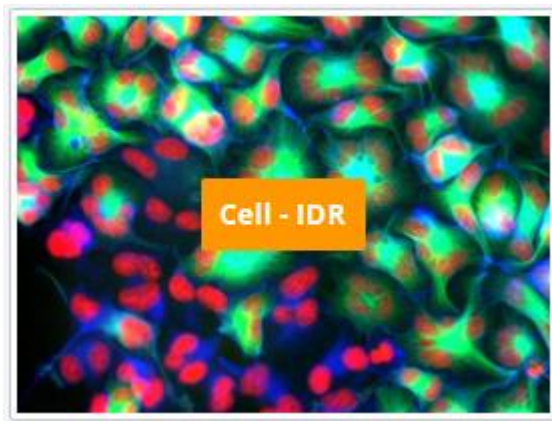
Look in the list of files, you should see a series of AVI movie files. Pick one (preferably a smallish one!), download it and see if you can view the movie.

Image Data Resource

We're going to look at the information and data we can get from studies in the image data resource.

Go to <https://idr.openmicroscopy.org/>

We're going to look at the cell level data since this is where most of the higher throughput data generally comes from.



Under Infection Studies look for study idr0081E. This is a study which looked at the ability of 1280 small molecule compounds to inhibit the activity of Human Adenovirus infection. It has high throughput data visualising the starting human cells and then the activity of the virus after treatment with each of the compounds.

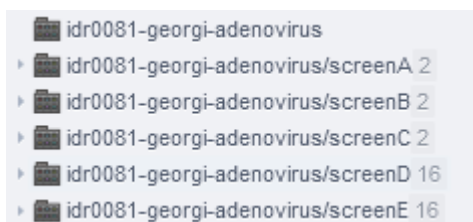


idr0081E Georgi F et al.



If you select the study and open the main data browser opened at the main study information.

On the left of the interface you have a list of all of the datasets in IDR. For this study there are several folders of information. At the top level you have the main idr0081 folder containing the overall information about the study. As you select that you should see some high level information appear in the sidebar on the right, but no image data.




The image data was collected in 5 distinct screens and these are listed as separate folders. You can select them in turn and look at the screen details on the right to see what they represent. You should see that screens A, B and C are all control experiments with either no real treatment or an artificial positive control. The actual screen data starts in screen D, so you should select this.

Under the screen D folder you should then see a set of 16 384-well plates. As you click on a plate you'll see a thumbnail of the images obtained from each well. Each image shows a small collection of human cells infected with Adenovirus and stained with two stains, Hoechst in blue which stains the human nuclei and GFP in green which represents the activity of the virus in these cells.



If you click on a well you will see the details of both the treatment in that well, ie which drug was added, and also the quantitated values which were extracted. You can see a measure for the mean and median virus intensity, and for the number of viral plaques (colonies).

Compound	
Added by: Public data	
Compound Name	Halofantrine hydrochloride 



Channels:	Hoechst:nuclei; GFP:infection
Max Nuclei Intensity:	52976
Total Nuclei Intensity:	16478844889
Mean Nuclei Intensity:	3928.862784
Number Of Nuclei:	36000
Max Virus Intensity:	62300
Total Virus Intensity:	11822913126
Mean Virus Intensity:	2818.802148
Median Virus Intensity:	1954
Number Of Plaques:	34
Number Of Infected Nuclei:	29725
Lesion Area:	2610960
Infection Index:	0.825694444

You should be able to glance over the plate and see some wells are mostly blue, indicating healthy human cells with little viral activity, some are green, indicating strong viral infection, and some are blank indicating that the treatment was likely toxic to the cells.

Have a look through the data and see if you can identify compounds which did both well and badly at preventing viral infection and check that the extracted values from the well agree with your assessment.

For any specific well you can also look at the image in more detail. Select an image from the plate and then at the top of the right hand sidebar there is an option to open the "Full Viewer". Try this and you can see an expanded version of the image. In the viewer you can zoom in more to see more details, and you can adjust the intensities of the two different stains to get a more balanced view of the image. You should be able to see distinct blue nuclei along with specific bright green viral plaques.

Exercise 10: Running Programs in Bash

- Note that you can't enter more commands in the terminal until you click on the cow to make it go away
- Read the man page to find out what the `-t 0` means

- Use **nano** to edit the file
- Change Mito to Mitochondrion in the **ID** and **AC** header lines
- Save the file and exit **nano**



- o Using `mv` rename the file from `Mito.dat` to `Mitochondrion.txt`

Copy the original `Mito.dat` file (the one at `~/seqmonk_genomes/Saccharomyces\cerevisiae/EF4/Mito.dat`) to the same filename in your `compare` directory

Run `diff` on `Mitochondrion.txt` and `Mito.dat` to see that it finds the edits you made to the file.

Exercise 12: Automation in Bash

Go into the `FastQ_Data` directory and look at one of the fastq files using `less`

- o Less is clever enough to realise that the file needs to be decompressed so you can just pass the fastq file to `less` directly
- o Now validate that one of the files can be successfully decompressed
 - Run `zcat` on the file, but...
 - Throw away the STDOUT output (using `> /dev/null`) so that you just see errors or warnings

Calculate the signatures of all of the fastq files using the `shasum` program

- o Start by running `shasum` on one fastq file to see how it works
- o Now run it on the entire contents of `FastQ_Data` using a wildcard (rather than a loop)
 - Write the results (STDOUT) to a file in your home directory using `>~/signatures.txt`

Run the `fastqc` program on all of the fastq.gz files (`*fastq.gz`)

Once the fastqc jobs have finished, run `multiqc .` (note the dot to specify it should run in the current directory) to assemble the fastqc output into a single report

Optional Exercise: Installing Software with BioConda

We have pre-installed the fastqc program for you on the systems you're working on, but if we hadn't you could have installed it from bioconda

Start by finding where fastqc is currently installed by running the following Bash command:

```
which fastqc
```

You should see that it's currently being accessed in `/usr/local/bin`.

Now you can install conda. You can follow the instructions at <https://conda-forge.org/download/>

You'll need to accept the license agreement, and you can allow it to activate the conda environment automatically. Once you've installed it you'll need to close the shell you had open and start another one for the changes to take effect. If it's worked you should see your command prompt to change to:

```
(base) student@ip-172-31-40-53:~$
```

..where the (base) at the start shows that conda is active.



Once it's running you can set up the channels it's going to use with:

```
conda config --add channels bioconda
conda config --add channels conda-forge
```

Once that's done you can install fastqc with:

```
conda install fastqc
```

You will see that it installs a ton of other things, this is the downside to conda, it's self contained and therefore duplicates lots of your system software to be sure it will work.

After the install has completed run:

```
which fastqc
```

..again, and you should see that you now have a new version of the program installed in your conda directory. If you want you can try reprocessing some of the fastq files to check the new version works.



Exercise 13: Interactive Data Analysis in R Notebooks

In this exercise we're going to give you a flavour of the sort of general data analysis which you can perform in an R notebook using the Tidyverse packages in R.

We're going to explore and analyse a small dataset where neutrophils were treated with different inhibitors and then their ability to invade cells was measured. The dataset has quantitative measures of invasion for multiple replicates of different treatments.

The structure of the input data file is:

Treatment	Invasion
DMSO	144.4393
DMSO	135.7167
DMSO	57.88828
TGX.221	99.61073
TGX.221	115.3576
etc.	etc.

Setting up the Notebook

Before we get into the actual analysis we're going to set up the template for the notebook and write a description of what we're going to do using simple markdown syntax.

Open rstudio (just type `rstudio` in a shell) and select `File > New File > R Notebook` from the menu. You will see a template notebook structure open.

You can immediately save the notebook in a file called `neutrophils.Rmd` in the `Big_Data` folder, which is where the data we're going to import is located.

Our analysis is going to be in four parts:

1. Loading in and looking at the data
2. Plotting our data
3. Calculating some summarised values from the data
4. Performing a statistical analysis

Adding a basic markdown structure

We're going to start by simply describing what we're going to do in this analysis. Delete everything below the header in the template notebook which opens (from "This is an [R Markdown] ...")

Use what you were shown about markdown syntax to make level 1 headers (underlined with equals signs) for the four sections we showed above and write a little bit of text to say what you're going to do in each one (just make it up!). In the loading section you can describe the conditions of the experiment and format them as a numbered list. The conditions are:



1. DMSO
2. TGX221
3. PI103
4. Akt1

Once you've added your markdown, save the document and compile it by selecting "Knit to HTML" from the drop down menu immediately above the editor. Check to see that the headings and list have rendered as you'd expect.

If you like you can modify the header of your document to add in the option to include a floating table of contents.

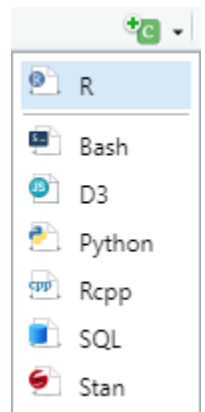
```
---
title: "Neutrophil Analysis"
output:
  html_document:
    df_print: paged
    toc: true
    toc_float: true
---
```

Save and re-render the document and check the table of contents shows up.

Now we have our basic document structure we can start adding in code. For each code block we want to add you need to put your cursor into the document at the point where you want to insert the code block and then use the menu above the editor to insert an R block.

You will see a block like this insert into the document.

```
19
20 {r}
21 |
22
```



You need to put your code between the upper and lower delimiters. When you want to run your code press the green play button at the top right. The output from the code will embed into the document just below the code block.

Before getting into the analysis we need to load the `tidyverse` package which we're going to be using. We're also going to load `rstatix` which is a library we'll later use for statistical testing. Create an initial code block just after the header section and write:

```
library(tidyverse)
library(rstatix)
```

You should see something like the following appear in the document:



```
Registered S3 methods overwritten by 'dbplyr':
  method      from
  print.tbl_lazy
  print.tbl_sql
-- Attaching packages -----
----- tidyverse 1.3.1 --
v ggplot2 3.3.3      v purrr 0.3.4
v tibble 3.1.2       v dplyr 1.0.6
v tidyr 1.1.3        v stringr 1.4.0
v readr 1.4.0        v forcats 0.5.1
-- Conflicts -----
----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
Registered S3 method overwritten by 'data.table':
  method      from
  print.data.table
```

Attaching package: "rstatix"

The following object is masked from "package:stats":

filter

Loading and Viewing Raw data

The data is contained in a file called `neutrophil_invasion.csv` (a comma separated value file). We're going to load this in using the `read_csv()` function from `tidyverse`.

Put a code block into your loading section containing:

```
read_csv("neutrophil_invasion.csv") -> data
data
```

This will load in the data and save it under the name 'data'. Putting `data` on its own prints out the contents of `data` so you see the table of results we're working with. You can use the controls under the table to see the remaining rows.

We can also try using a filtering option to have a look at all of the data we have for just one of the conditions.

Open a new code block and run:

```
data %>%
  filter(Treatment == "Akt1") %>%
  arrange(Invasion)
```

To see all of the Akt1 data and sort the rows from the lowest to highest Invasion values.



Plotting the data

Now we want to plot out the data to see what we're working with and what might be happening in the different conditions. To do this we're going to use the ggplot2 package (which is part of tidyverse) and is the most common plotting system used in R.

To create a basic stripchart view of the values for each condition create a new code block in the plotting section and run:

```
data %>%  
  ggplot(aes(x=Treatment, y=Invasion)) +  
  geom_jitter()
```

You might find that it's not very easy to tell which points come from which treatment, so we can help by adding colouring to the points, and making them slightly bigger.

```
data %>%  
  ggplot(aes(x=Treatment, y=Invasion, colour=Treatment)) +  
  geom_jitter(size=4)
```

See that ggplot automatically assigns colours and creates a colour legend next to the plot for us.

Have a look at the data and write into the document what you think you can see in the data.

We can also use a summarised representation of the data to visualise it. In a new code block we can draw a boxplot of the data.

```
data %>%  
  ggplot(aes(x=Treatment, y=Invasion)) +  
  geom_boxplot(fill="yellow")
```

See if this makes the trends in the data any clearer.

Calculating Summarised Values

We can see that there appear to be differences in the mean values for invasion in the different conditions, however we can also see that the spread of points is also somewhat different and it would be good to quantify this too.

We are therefore going to summarise the data by splitting it into groups based on the Treatment, and then calculating a mean and standard deviation for each subgroup.

To do this create a new code block in the summarisation section and run the following:

```
data %>%  
  group_by(Treatment) %>%  
  summarise(  
    mean=mean(Invasion),  
    stdev=sd(Invasion)  
  ) -> summarised_data  
  
summarised_data
```



This will calculate and store the mean and stdev for the invasion scores for each treatment and then will display it for us to see.

Which sample(s) show a marked shift in their mean value? How close are the StDev values for the treatment(s) which change?

We can also plot this summarisation out as a barplot. To do that we create a new block and use:

```
summarised_data %>%  
  ggplot(aes(x=Treatment, y=mean, ymin=mean-stdev, ymax=mean+stdev)) +  
  geom_col(fill="orange") +  
  geom_errorbar(size=1, width=0.4)
```

This will use our summarised values, where the height of the bars is the calculated mean, and the errorbars are the mean +/- the standard deviation.

Statistical Analysis

Finally we can do a statistical analysis of the different conditions. We're going to use an ANOVA test to compare all of the Treatments* and then a Tukey pos-hoc test to tell us which combinations of Treatments were the most significant.

* This is slightly shaky from a statistical point of view since the ANOVA assumes that all of your samples will have similar standard deviation, and ours shows a bit of a shift, but for the purposes of demonstration we're going to apply the test anyway. In real life we'd go and talk to a statistician to see whether the difference in StDev is big enough that it might invalidate the test results.

Firstly we're going to use an ANOVA to compare all of the conditions to see if there is any evidence of a difference in any of the means.

In your statistics section create a new block and run:

```
data %>%  
  anova_test(Invasion~Treatment) %>%  
  as_tibble()
```

The Invasion~Treatment means that what we're testing is the ability of the test to predict the Invasion level based on knowledge of the Treatment.

Look at the value for the p column. If this is below 0.05 then somewhere in the data there is a significant difference in the mean values. Does the value you see agree with what you'd expect having looked at the data?

We can finally go on to do a set of pairwise comparisons between all of the Treatments to determine which of them is going to be the most significant. Create a new block and run;

```
data %>%  
  tukey_hsd(Invasion~Treatment) %>%  
  arrange(p.adj)
```



Which pairs of samples have significant results ($p_{\text{adj}} < 0.05$). What do they have in common?

Once you've finished this you can re-render your notebook to create an HTML output.



Exercise 14: Python Application Development

In this section we're going to build a small python application. This is going to be a program which asks the user for the name of a human gene and then queries the Ensembl API for this name, retrieves the Ensembl Gene ID, and then uses this to get the cDNA sequence of all of the transcripts for that gene.

To do this we will be using code examples from both Ensembl's API documentation, but also using the BioPython project to parse the sequence files for us.

This is NOT intended to turn you into a python programmer in an hour or so, but should help illustrate how you can automate access to these kinds of data sources.

Overall Structure

We're going to develop this application in 4 stages

1. Collect a gene name from the user
2. Turn this gene name into an Ensembl ID
3. Get the cDNA sequences for the Ensembl ID
4. Parse the sequences and collect the names and lengths of the transcripts

Before we get into these, at the top of the script we need to load in the helper packages we're going to use later on:

```
import requests, sys
from Bio import SeqIO
import io
```

The first is used for the Ensembl API requests and the other two are used when parsing the fasta sequence data. After we've put these in you can save the file under the name `gene_stats.py`.

Asking the user for a gene name

To ask the user for a piece of data we can use the `input` function. This allows us to ask a question, wait for their answer and then store it so we can use it later.

```
query = input("Which gene? ")
```

When the query comes back it will have a newline character on the end (because they pressed return to submit it), and we need to remove that, which we can do with a method called `strip`.

```
query = query.strip()
```

If you want to check what they got you can print out any piece of data using the `print` function. You can pass multiple pieces of data to this separating them with commas.

```
print("They asked for", query)
```

Check that the script works this far and you can see the gene name they asked for.



Converting a gene name into an Ensembl ID

That was fairly simple, but now we get a little more complicated. We're going to use the Ensembl API web resource to query the Ensembl database for that gene and get back the gene (or possibly genes) which use that name. To do this we're going to adapt the instructions which Ensembl provide on their website (https://rest.ensembl.org/documentation/info/xref_external), they even give snippets of code in various languages to help you implement this for yourself. One of their examples is pretty much exactly what we want so we're going to adapt that:

**/xrefs/symbol/homo_sapiens/BRCA2?content-type=application
/json;external_db=HGNC**

Example output Perl Python2 **Python3** Ruby Java R Curl Wget

```
1. import requests, sys
2.
3. server = "https://rest.ensembl.org"
4. ext = "/xrefs/symbol/homo_sapiens/BRCA2?external_db=HGNC"
5.
6. r = requests.get(server+ext, headers={"Content-Type": "application/json"})
7.
8. if not r.ok:
9.     r.raise_for_status()
10.    sys.exit()
11.
12. decoded = r.json()
13. print(repr(decoded))
14.
```

In our case we need to change this slightly, firstly to insert the gene name we collected from the user, and secondly to extract the Ensembl gene id directly from the response rather than just printing the whole thing.

We start by putting in the server details:

```
server = "https://rest.ensembl.org"
```

..but then for the request we need to join our gene name into the query string:

```
ext = "/xrefs/symbol/homo_sapiens/"+query+"?external_db=HGNC"
```

Now we can make the request exactly as they suggest and check that it worked.

```
r = requests.get(server+ext, headers={"Content-Type": "application/json"})

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

The editor should automatically indent your code after the if statement. When you get to the end of the block you can use the backspace key to align your next statement with the left side of the screen.



Assuming it did work then we can get the decoded data and extract the first gene id. The code where we retrieve information from within decoded gets the first entry in the list (entry 0), and then gets the id value from that entry. This code needs to be back against the left side of the editor.ex

```
decoded = r.json()
gene_id = decoded[0]['id']
```

Finally we can print out the result to check it's working:

```
print("Query=", query, "ID=", gene_id)
```

If you query with BRCA2 you should get ENSG00000139618 back as the gene id.

Getting all of the cDNA sequences from the query gene

Now that we have an Ensembl gene ID we can use a different API to get all of the transcript sequences associated with that gene. For this we're going to use a different API, this one: https://rest.ensembl.org/documentation/info/sequence_id

This time none of the examples are exactly what we want, so we're going to have to build a query using the set of parameters we have at the top. The things we need to set are:

1. We're querying with a gene id
2. The type of sequence we want is cdna
3. We want to retrieve multiple sequences (one gene can have several transcripts)
4. We want fasta format data back

We're using the same server as before so we don't need to set that up again, but we're using a different query so we need to build that using the Ensembl ID we got back from the last query.

```
ext = "/sequence/id/"+gene_id+"?type=cdna&multiple_sequences=1"
```

So here we input our Ensembl ID and say that we want cdna sequence, and that we are happy to have multiple sequences returned.

We can now run the query, and say we want the output in fasta format. We can check that it worked in the same way as before:

```
r = requests.get(server+ext, headers={"Content-Type" : "text/x-fasta"})

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

If you want a quick check that this worked then you can print out the full response, but you probably don't want to leave this in your code once you know that it's working as you'll see that the full sequences can be pretty big! This code again needs to be out of the if block and back against the left of the editor.

```
print(r.text)
```



See if you can get this far and see the sequences coming back from Ensembl.

Parsing the fasta files and getting sequence details

For the final part we're going to use a sequence parser which is part of the biopython project (<https://biopython.org/>). Specifically we're going to use a package called SeqIO which deals with reading and writing sequences (<https://biopython.org/docs/1.75/api/Bio.SeqIO.html>).

The SeqIO parser is designed to read data from a file, and we've actually got it as a piece of text, so before we can parse it we need to use a little helper called `io.String` which makes a piece of text look like a file so SeqIO will read it.

```
s = io.StringIO(r.text)
```

Now we can use SeqIO to parse the data.

```
fasta = SeqIO.parse(s, "fasta")
```

Finally we can iterate through the sequences, pulling out the name and length and printing the details.

```
for f in fasta:
    name = f.id
    length = len(f)
    print("Found transcript", name, "with length", length)
```

This should get us our final program. Try running the program and trying out a few gene names to check it works.

Which gene? BRCA2

They asked for BRCA2

Query= BRCA2 ID= ENSG00000139618

Found transcript ENST00000544455.6 with length 11854

Found transcript ENST00000530893.6 with length 2011

Found transcript ENST00000380152.8 with length 11954

Found transcript ENST00000680887.1 with length 11880

Found transcript ENST00000614259.2 with length 11763

Found transcript ENST00000665585.1 with length 2598

Found transcript ENST00000528762.1 with length 495

Found transcript ENST00000470094.1 with length 842

Found transcript ENST00000666593.1 with length 523

Found transcript ENST00000533776.1 with length 523

Some example genes to try would be:

Sox2

Nanog

Fos

..but feel free to try your own favourites too.

