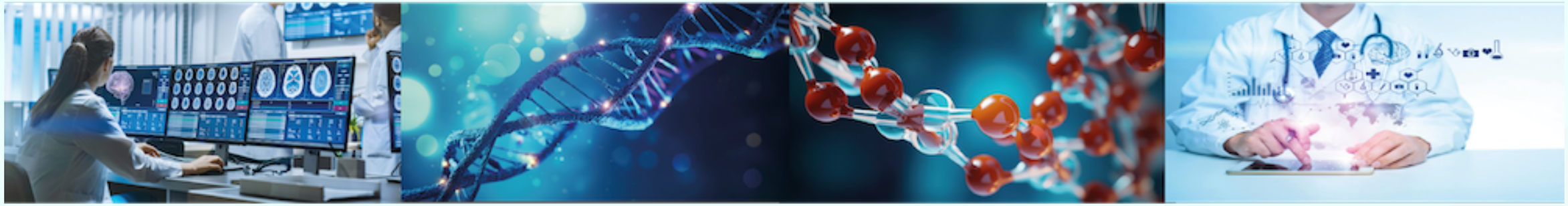


Programming for Biomedical Informatics

Lecture 5 - Mapping & Harmonisation

https://github.com/biomedical-informatics/pbi

https://biomedical-informatics.github.io/pbi-home/

Ian Simpson
ian.simpson@ed.ac.uk

THE UNIVERSITY of EDINBURGH
informatics

Mapping Entities in Biomedical Informatics

- There are many situations where you will need to link one or more type of data across different data sources
- Whilst it may at first appear trivial it is often complex due to a number of factors:
    - **synchronisation** - different data sources may use a common reference but those references may be from different releases of the data (e.g. SNPs aligned to a prior genome version)
    - **redundancy** - there is a lot of duplication of data, unfortunately both within and between resources
    - **deprecation** - accession identifiers have provenance, the one's you're working with may have changed and/or been removed. There is often a track history for these, but it isn't always straightforward to find
    - **re-annotation** - despite what you may have heard genomes and the data we map to them are not solved or completed, they are simply the latest iteration. That means mistakes are made and later corrected often leading to complex decisions that are hard to follow
    - **conflicts** - some resources simply disagree with each other on the nature of particular models and annotations. Often this is based on a fundamentally different approach to how sequences (in particular) are interpreted.
    - **quality** - depending on what you are trying to integrate or map between you may well be working with sources that have wildly different QC procedures and standardised methods. It's also particularly problematic if you're mapping between different species

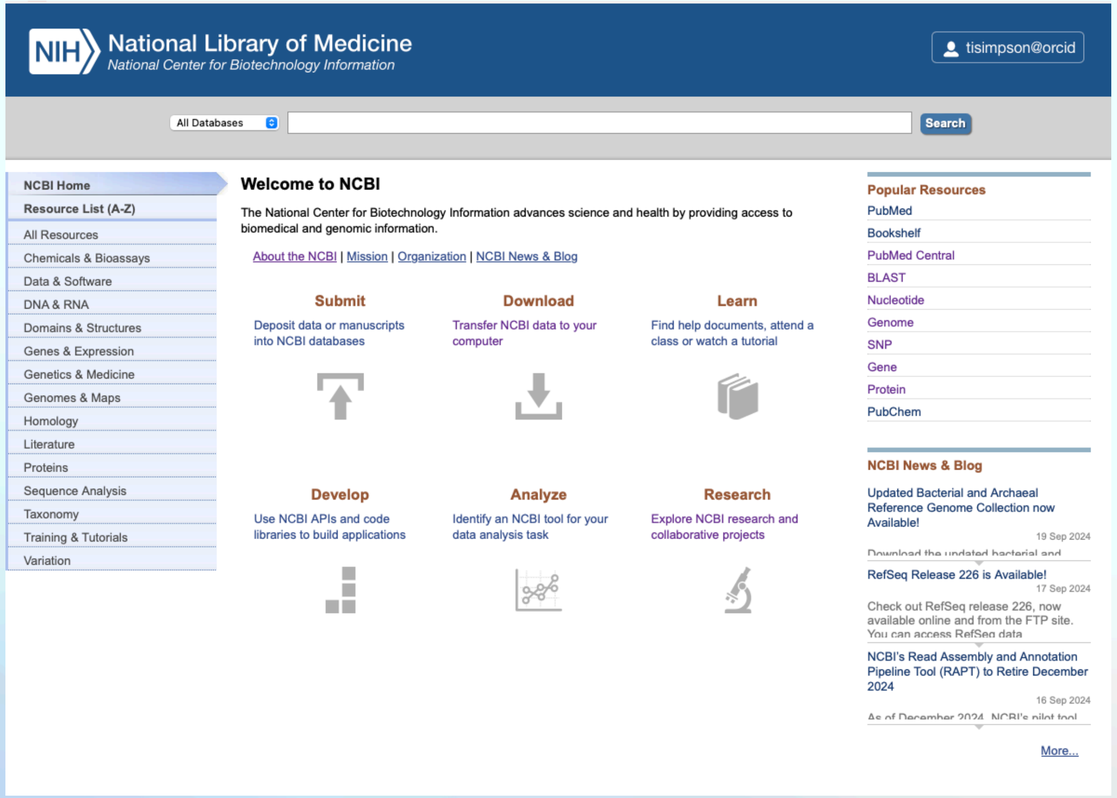
Mapping Entities in Biomedical Informatics

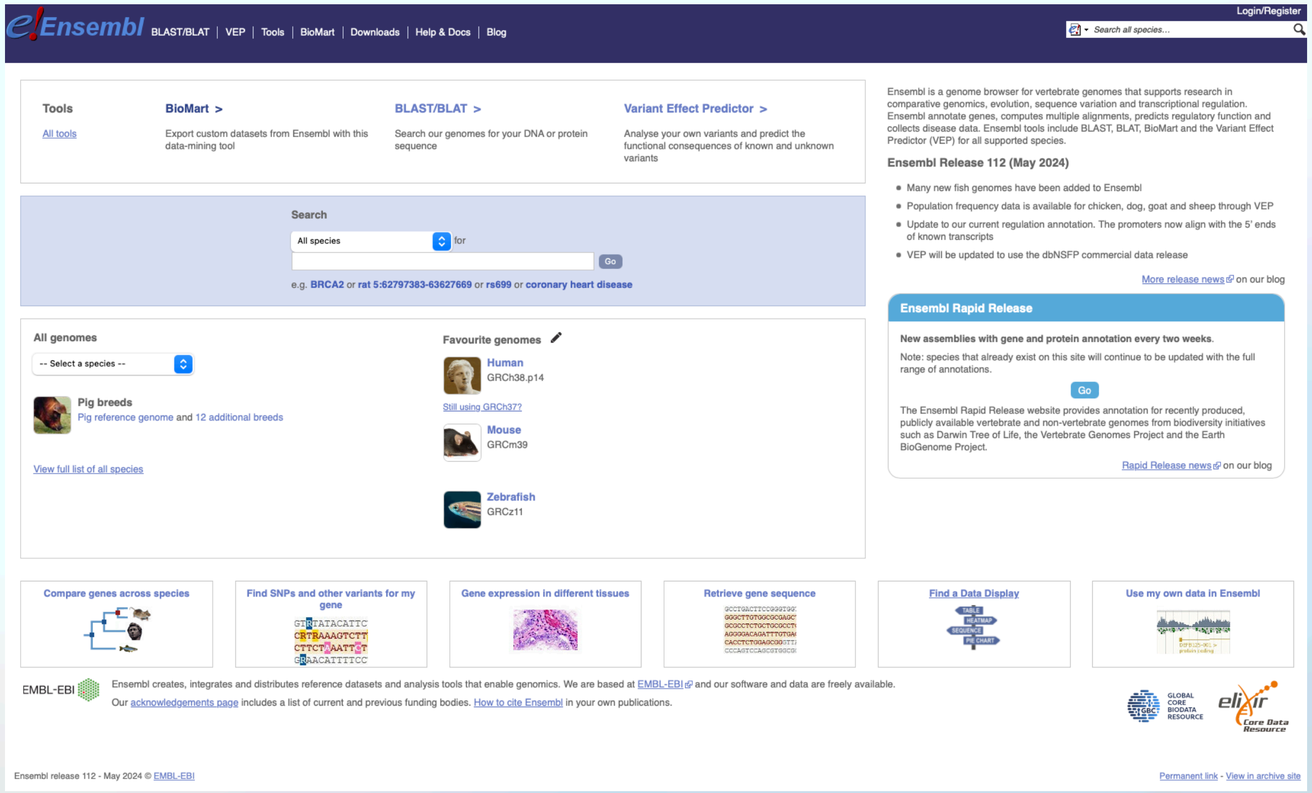
- We will use the two main international biomedical data organisations, NCBI-NLM (National Library of Medicine, US) and Ensembl (EMBL-EBI, UK) as examples
- NLM and Ensembl both contain large collections of databases for both molecular and non-molecular data, but they use two very different systems for mapping and harmonising data
- eUtilities (NLM) and BioMart (Ensembl)
- There are a number of Python libraries that have been developed or can be deployed to take advantage of these. For eUtils there is a nice implementation in BioPython, for BioMart access is best achieved through BioMart API endpoints, although several packages do implement core features.
- The bioservices project - https://bioservices.readthedocs.io/en/main/ aims to implement access to a wide range of biomedical data sources via python.

NCBI-NLM & EBI-Ensembl

NCBI (National Centre for Biotechnology Information) - https://www.ncbi.nlm.nih.gov/

Ensembl (EMBL-EBI) - https://www.ensembl.org/





NCBI-NLM Resources

Entrez Cross-Database Links

- Entrez is a search engine that connects multiple NCBI databases (e.g., Gene, Protein, PubChem, RefSeq, etc.) through cross-links, allowing users to move from one type of accession to another seamlessly.

E-utilities API

- NCBI's E-utilities API provides programmatic access to data. The `elink` tool is useful for mapping an accession from one database to linked records in other databases.

Gene ID to RefSeq

- NCBI Gene entries contain links to **RefSeq** entries allowing mapping between gene accessions and gold-standard protein or nucleotide sequences.

LinkOut

- Many NCBI resources have "LinkOut" sections, which provide links between accession numbers and other databases.

OMIM Links

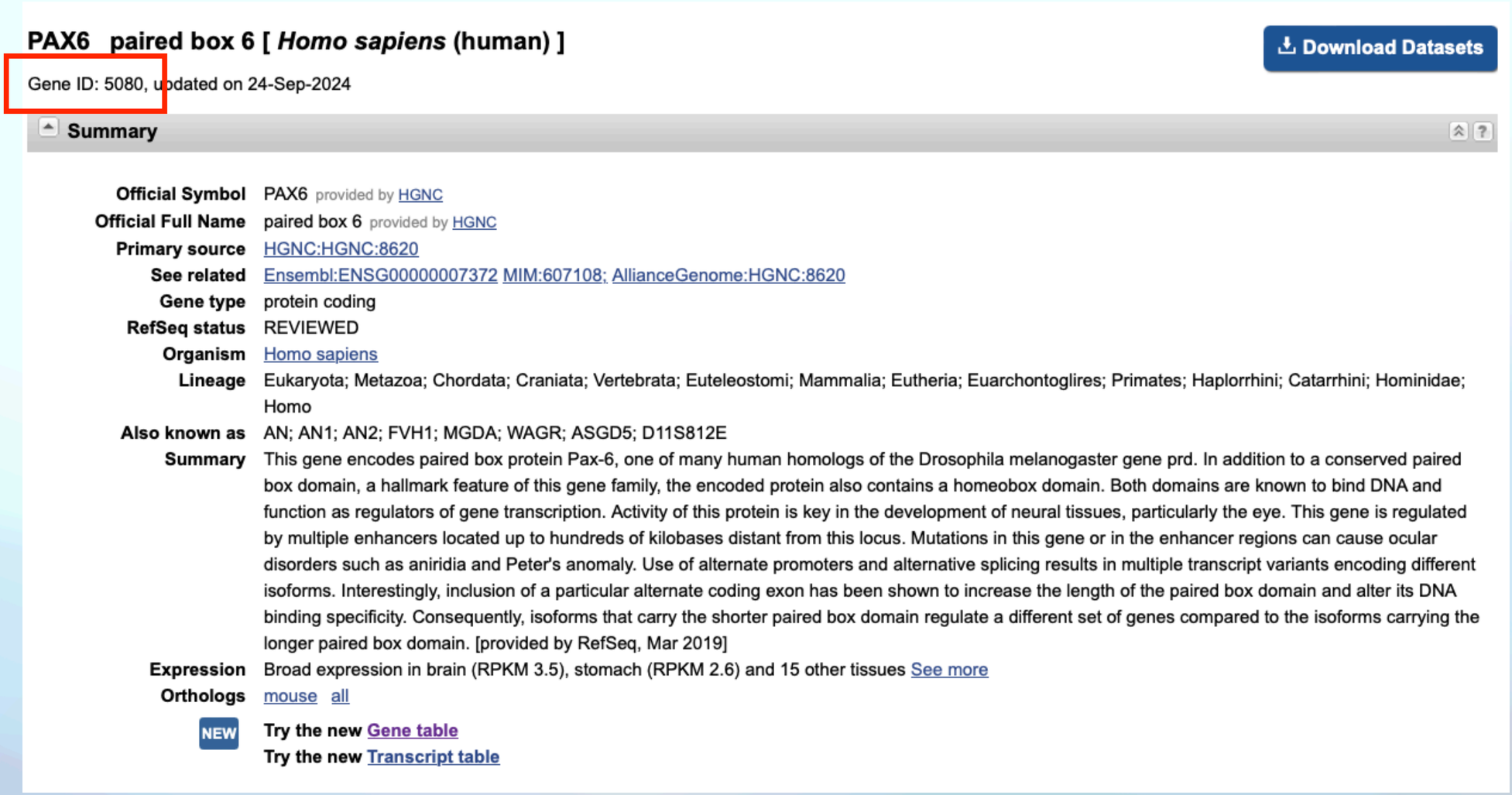
- For genes with medical relevance, Online Mendelian Inheritance in Man (OMIM) provides accession numbers that link to NCBI Gene, allowing mappings between clinical and sequence data.

- Sayers EW et al. Database resources of the national center for biotechnology information. Nucleic Acids Res. 2022 Jan 7;50(D1):D20-D26. doi: 10.1093/nar/gkab1112.

Database	Records	Description
Literature		
PubMed	33 027 761	Scientific and medical abstracts/citations
PubMed Central	7 325 415	Full-text journal articles
NLM Catalog	1 629 799	Index of NLM collections
Bookshelf	892 126	Books and reports
MeSH	348 370	Ontology used for PubMed indexing
Genomes		
Nucleotide	476 054 019	DNA and RNA sequences from GenBank and RefSeq
BioSample	19 473 659	Descriptions of biological source materials
SRA	15 919 320	High-throughput DNA/RNA sequence read archive
Taxonomy	2 492 889	Taxonomic classification and nomenclature catalog
Assembly	1 083 900	Genome assembly information
BioProject	536 242	Biological projects providing data to NCBI
Genome	64 815	Genome sequencing projects by organism
BioCollections	8 468	Museum, herbaria, and biorepository collections
Genes		
GEO Profiles	128 414 055	Gene expression and molecular abundance profiles
Gene	33 664 932	Collected information about gene loci
GEO DataSets	4 784 603	Functional genomics studies
PopSet	366 935	Sequence sets from phylogenetic/population studies
HomoloGene	141 268	Homologous gene sets for selected organisms
Clinical		
dbSNP	1 076 992 604	Short genetic variations
dbVar	7 117 914	Genome structural variation studies
ClinVar	1 071 071	Human variations of clinical significance
ClinicalTrials.gov	388 717	Registry of clinical studies and results database
MedGen	335 277	Medical genetics literature and links
GTR	77 498	Genetic testing registry
dbGaP	1 405	Genotype/phenotype interaction studies
Proteins		
Protein	968 236 913	Protein sequences from GenBank and RefSeq
Identical Protein Groups	448 096 579	Protein sequences grouped by identity
Protein Clusters	1 137 329	Sequence similarity-based protein clusters
Structure	181 772	Experimentally-determined biomolecular structures
Protein Family Models	179 133	Conserved domain architectures, HMMs, and BlastRules
Conserved Domains	62 852	Conserved protein domains
Chemicals		
PubChem Substance	284 180 803	Deposited substance and chemical information
PubChem Compound	110 628 849	Chemical information with structures, information and links
PubChem BioAssay	1 391 308	Bioactivity screening studies
BioSystems	983 968	Molecular pathways with links to genes, proteins and chemicals

NCBI-NLM Gene Entry

The NCBI Gene ID is a unique accession identifying genes



PAX6 paired box 6 [*Homo sapiens* (human)]

Download Datasets

Gene ID: 5080, updated on 24-Sep-2024

Summary

Official Symbol	PAX6 provided by HGNC
Official Full Name	paired box 6 provided by HGNC
Primary source	HGNC:HGNC:8620
See related	Ensembl:ENSG00000007372 MIM:607108; AllianceGenome:HGNC:8620
Gene type	protein coding
RefSeq status	REVIEWED
Organism	Homo sapiens
Lineage	Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini; Catarrhini; Hominidae; Homo
Also known as	AN; AN1; AN2; FVH1; MGDA; WAGR; ASGD5; D11S812E
Summary	This gene encodes paired box protein Pax-6, one of many human homologs of the Drosophila melanogaster gene prd. In addition to a conserved paired box domain, a hallmark feature of this gene family, the encoded protein also contains a homeobox domain. Both domains are known to bind DNA and function as regulators of gene transcription. Activity of this protein is key in the development of neural tissues, particularly the eye. This gene is regulated by multiple enhancers located up to hundreds of kilobases distant from this locus. Mutations in this gene or in the enhancer regions can cause ocular disorders such as aniridia and Peter's anomaly. Use of alternate promoters and alternative splicing results in multiple transcript variants encoding different isoforms. Interestingly, inclusion of a particular alternate coding exon has been shown to increase the length of the paired box domain and alter its DNA binding specificity. Consequently, isoforms that carry the shorter paired box domain regulate a different set of genes compared to the isoforms carrying the longer paired box domain. [provided by RefSeq, Mar 2019]
Expression	Broad expression in brain (RPKM 3.5), stomach (RPKM 2.6) and 15 other tissues See more
Orthologs	mouse all

NEW **Try the new Gene table**

Try the new Transcript table

NCBI-NLM Resources

NCBI RefSeq Gene

NG_008679.1

Genomic

1. **NG_008679.1 RefSeqGene**

Range	5001..38170
Download	GenBank, FASTA, Sequence Viewer (Graphics), LRG_720

NCBI RefSeq Transcript (mRNA) and Protein

NM_000280.6
NP_000271.1

1. **NM_000280.6 → NP_000271.1 paired box protein Pax-6 isoform a**

See identical proteins and their annotated locations for NP_000271.1

Status: REVIEWED

Description	Transcript Variant: This variant (1) encodes isoform a. Variants 1, 3, 6, 7, and 12-16 all encode the same isoform (a).
Source sequence(s)	M93650, Z83307, Z95332
Consensus CDS	CCDS31451.1
UniProtKB/Swiss-Prot	P26367, Q6N006, Q99413
UniProtKB/TrEMBL	B3KQG1, Q66SS1
Related	ENSP00000495109.1, ENST00000643871.1

Conserved Domains (2) summary

smart00351 Location:4 → 128	PAX; Paired Box domain
pfam00046 Location:214 → 267	Homeobox; Homeobox domain

NCBI RefSeq Transcript (mRNA) Entry

Homo sapiens paired box 6 (PAX6), transcript variant 1, mRNA

NCBI Reference Sequence: NM_000280.6

FASTA Graphics

Go to:

```
LOCUS       NM_000280               2701 bp    mRNA    linear   PRI 24-SEP-2024
DEFINITION  Homo sapiens paired box 6 (PAX6), transcript variant 1, mRNA.
ACCESSION   NM_000280
VERSION     NM_000280.6
KEYWORDS    RefSeq.
SOURCE      Homo sapiens (human)
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini;
            Catarrhini; Hominidae; Homo.
```

NCBI RefSeq Entries are **Curated**

```
COMMENT     REVIEWED REFSEQ: This record has been curated by NCBI staff. The
            reference sequence was derived from Z95332.1, M93650.1 and
            Z83307.1.

            On Dec 6, 2021 this sequence version replaced NM_000280.5.
```

NCBI-NLM LinkOut

National Library of Medicine
National Center for Biotechnology Information

LinkOut

About **For Libraries** **For Full Text Providers** **For Other Providers**

LinkOut is a service that allows organizations external to NCBI to add and update links to their own resources from individual records in NCBI databases such as PubMed. This service provides visitors convenient access to outside resources that are intended to extend, clarify, and supplement information found in NCBI databases.

Publishers, libraries, institutional repositories, and scientific databases are able to register as LinkOut Providers in order to connect their online resources to NCBI database records.

Interested in Becoming a LinkOut Provider?

Learn how by clicking on the appropriate provider type from the options below.

Full Text Providers

Publishers, Institutional Repositories, ...

Libraries

Other Providers

Databases, datasets, other resources

Examples of LinkOut Resources

- full-text publications,
- scientific databases,
- institutional repositories,
- consumer health information, and
- research tools

LinkOuts are Searchable via NCBI

Gene LinkOut

The following LinkOut resources are supplied by external providers. These providers are responsible for maintaining the links.

Chemical Information

FREE Interologous Interaction Database

Interologous Interaction Database

ORDER MilliporeSigma

Pax6 products

Medical

FREE MedlinePlus Health Information

PAX6 gene

Molecular Biology Databases

FREE Bgee database

PAX6 gene expression

FREE BioGPS

BioGPS

FREE BioGRID Open Repository of CRISPR Screens (ORCS)

BioGRID CRISPR Screen Phenotypes (16 hits/1275 screens)

FREE Domain Mapping of Disease Mutations

PAX6

FREE Eukaryotic Promoter Database

PAX6_1

FREE GlyGen glycoinformatics resource

GlyGen glycoinformatics resource

FREE Human Gene Mutation Database

Human Gene Mutation Database

FREE Human eFP Browser

Human eFP Browser

Human eFP Browser

Human eFP Browser

Ingenuity Pathways Analysis

Ingenuity Pathways Analysis

FREE InnateDB

InnateDB

FREE InterMine

InterMine

FREE Kyoto Encyclopedia of Genes and Genomes

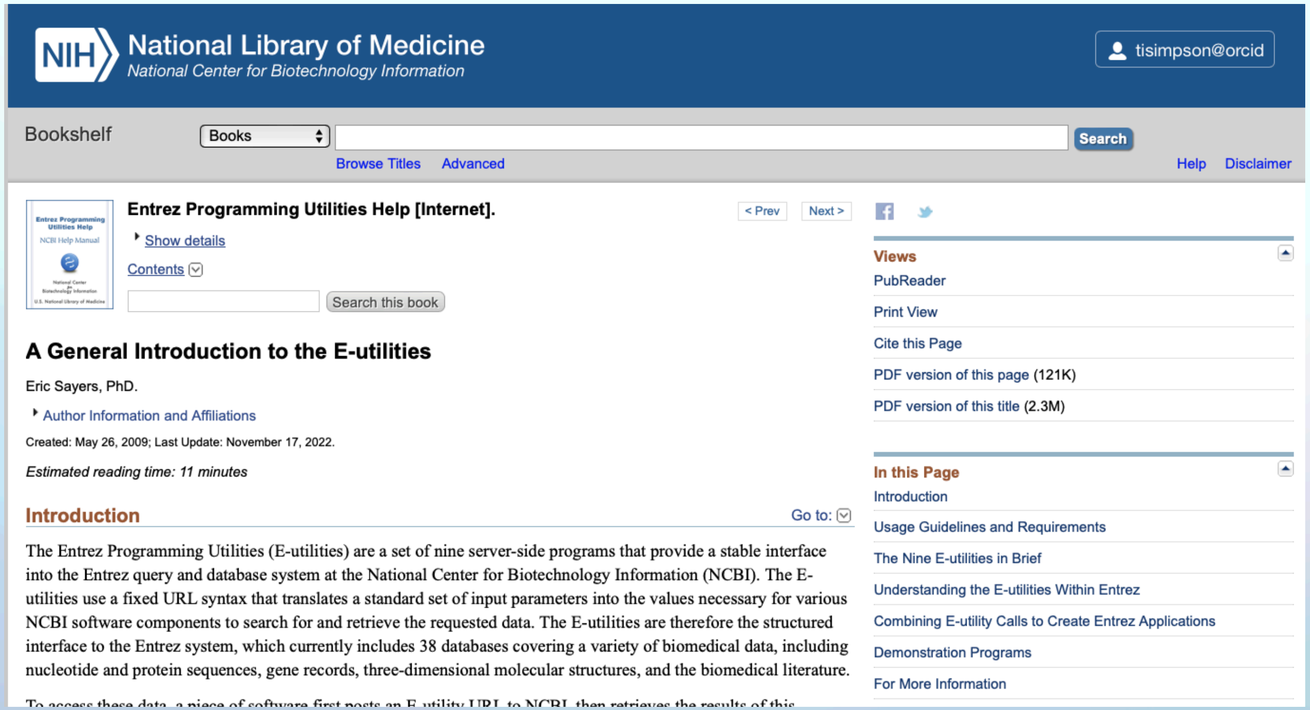
Kyoto Encyclopedia of Genes and Genomes

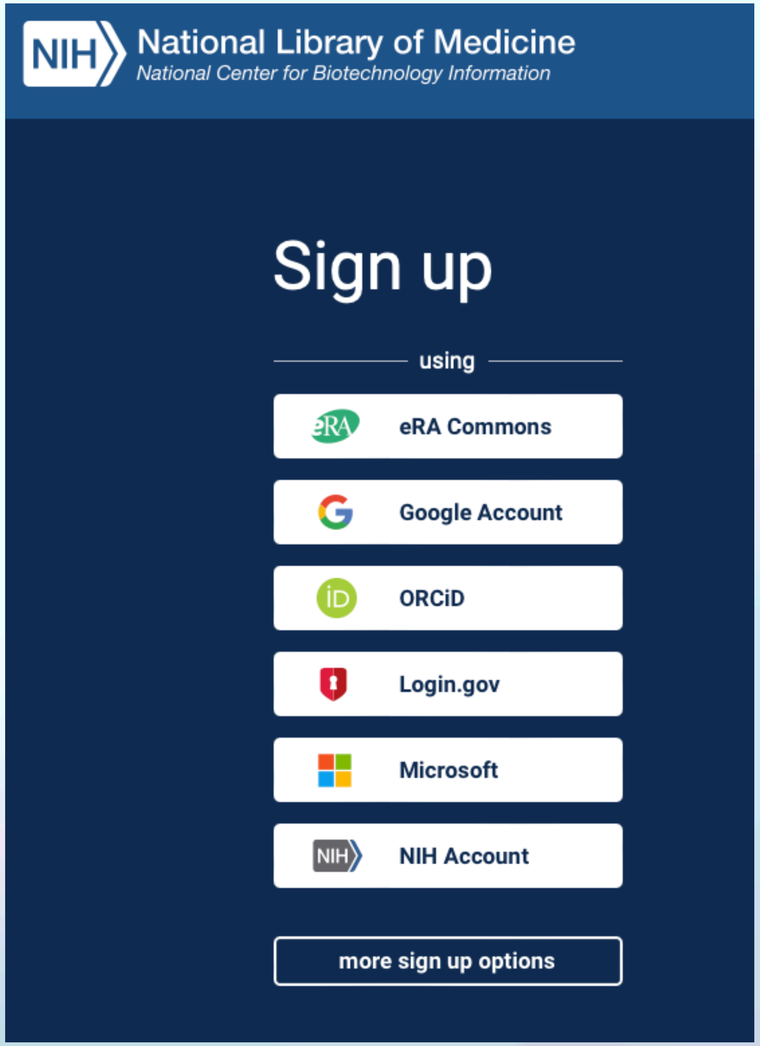
FREE OMA Browser: Orthologous MAtrix

OMA Browser: Orthologous MAtrix

Using NCBI API / eUtilities

- Please register (free) for an account with NCBI
  - https://account.ncbi.nlm.nih.gov/signup/
- Once you have this you can go to your account and find your API key
  - This allows you to send up to 10 requests/second
- NCBI have made an excellent "book" to explain how eUtilities works -
  - https://www.ncbi.nlm.nih.gov/books/NBK25497/
- It's platform independent so you can use whichever language you want to access





The NCBI-NLM eUtilities

- **EInfo (database statistics)** - eutils.ncbi.nlm.nih.gov/entrez/eutils/einfo.fcgi
  - Provides the number of records indexed in each field of a given database, the date of the last update of the database, and the available links from the database to other Entrez databases.
- **ESearch (text searches)** - eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi
  - Responds to a text query with the list of matching UIDs in a given database (for later use in ESummary, EFetch or ELink), along with the term translations of the query.
- **EPost (UID uploads)** - eutils.ncbi.nlm.nih.gov/entrez/eutils/epost.fcgi
  - Accepts a list of UIDs from a given database, stores the set on the History Server, and responds with a query key and web environment for the uploaded dataset.
- **ESummary (document summary downloads)** - eutils.ncbi.nlm.nih.gov/entrez/eutils/esummary.fcgi
  - Responds to a list of UIDs from a given database with the corresponding document summaries.
- **EFetch (data record downloads)** - eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi
  - Responds to a list of UIDs in a given database with the corresponding data records in a specified format.
- **ELink (Entrez links)** - eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi
  - Responds to a list of UIDs in a given database with either a list of related UIDs (and relevancy scores) in the same database or a list of linked UIDs in another Entrez database; checks for the existence of a specified link from a list of one or more UIDs; creates a hyperlink to the primary LinkOut provider for a specific UID and database, or lists LinkOut URLs and attributes for multiple UIDs.
- **EGQuery (global query)** - eutils.ncbi.nlm.nih.gov/entrez/eutils/egquery.fcgi
  - Responds to a text query with the number of records matching the query in each Entrez database.
- **ESpell (spelling suggestions)** - eutils.ncbi.nlm.nih.gov/entrez/eutils/espell.fcgi
  - Retrieves spelling suggestions for a text query in a given database.
- **ECitMatch (batch citation searching in PubMed)** - eutils.ncbi.nlm.nih.gov/entrez/eutils/ecitmatch.cgi
  - Retrieves PubMed IDs (PMIDs) corresponding to a set of input citation strings.

BioPython

- BioPython - https://biopython.org/
- Available via pip, conda, or source download
  - pip install biopython
  - conda install conda-forge::biopython
  - http://biopython.org/DIST/biopython-1.84.tar.gz
- The Biopython package provides a library **Bio.Entrez** that can be used to access the eUtils
- This is the simplest way to begin working with the eUtils using Python
- Once you are comfortable with the way the system (APIs) operate you may choose to simply use urlib (or similar) to code more flexibly with
- Biopython has excellent "cookbooks" which are code recipes to achieve common tasks

Accessing NCBI's Entrez databases

Entrez (https://www.ncbi.nlm.nih.gov/Web/Search/entrezfs.html) is a data retrieval
provides users access to NCBI's databases such as PubMed, GenBank, GEO, and m
can access Entrez from a web browser to manually enter queries, or you can use Bi
`Bio.Entrez` module for programmatic access to Entrez. The latter allows you for ex
PubMed or download GenBank records from within a Python script.

The `Bio.Entrez` module makes use of the Entrez Programming Utilities (also know
consisting of eight tools that are described in detail on NCBI's page at
https://www.ncbi.nlm.nih.gov/books/NBK25501/. Each of these tools corresponds
function in the `Bio.Entrez` module, as described in the sections below. This modu
that the correct URL is used for the queries, and that NCBI's guidelines for respons
are being followed.

The output returned by the Entrez Programming Utilities is typically in XML format
output, you have several options:

1. Use `Bio.Entrez`'s parser to parse the XML output into a Python object;
2. Use one of the XML parsers available in Python's standard library;
3. Read the XML output as raw text, and parse it by string searching and manipulation.

Simple BioPython eUtils Example

Converting Gene Symbols into NCBI Gene IDs

Import

- Import the `Entrez` library from `Bio` (Biopython) to access the NCBI databases.

API Key and Email

- Set the `api_key` and `email` for access to the Entrez service from your NCBI account.

Function - get_gene_ids()

- Takes a list of gene symbols and an optional organism parameter.
- Uses the `**Entrez.esearch()**` function to search for the gene symbol in the NCBI Gene database.
- Retrieves the first matching gene ID, restricted to the specified organism.

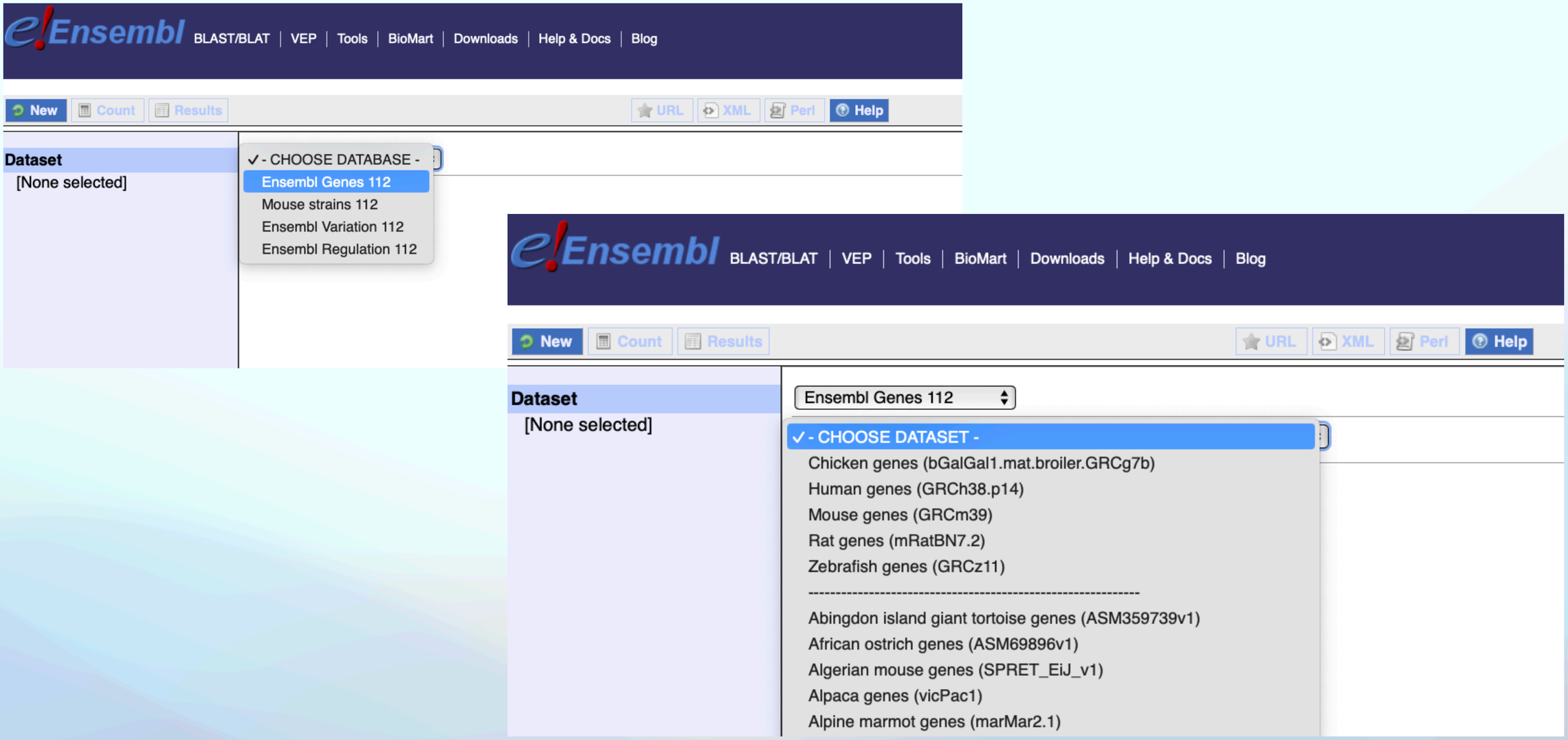
Details to Note

- specify the database (db)
- Use field keywords to focus the search (https://www.ncbi.nlm.nih.gov/books/NBK49540/) <- this is **super-useful**

```python
from Bio import Entrez

# load my API key from the file
with open('../../api_keys/ncbi.txt', 'r') as file:
    api_key = file.read().strip()

with open('../../api_keys/ncbi_email.txt', 'r') as file:
    email = file.read().strip()

Entrez.api_key = api_key
Entrez.email = email

def get_gene_ids(gene_symbols, organism="Homo sapiens"):
    """
    Convert a list of gene symbols into NCBI Gene IDs.

    Parameters:
    gene_symbols (list): List of gene symbols to search for.
    organism (str): Organism name to restrict search (default is "Homo sapiens").

    Returns:
    dict: A dictionary mapping gene symbols to NCBI Gene IDs.
    """
    gene_ids = {}
    for symbol in gene_symbols:
        handle = Entrez.esearch(db="gene", term=f"{symbol}[Gene] AND {organism}[Organism]", retmax=1)
        record = Entrez.read(handle)
        handle.close()

        if record["IdList"]:
            gene_ids[symbol] = record["IdList"][0]
        else:
            gene_ids[symbol] = None

    return gene_ids

# Example usage
gene_symbols = ["BRCA1", "TP53", "EGFR", "APOE", "TNF", "ESR1", "IL6", "VEGFA", "MTHFR", "FTO"]
gene_ids = get_gene_ids(gene_symbols)

# Print the gene symbol to NCBI Gene ID mapping
for symbol, gene_id in gene_ids.items():
    print(f"Gene Symbol: {symbol}, Gene ID: {gene_id}")
```

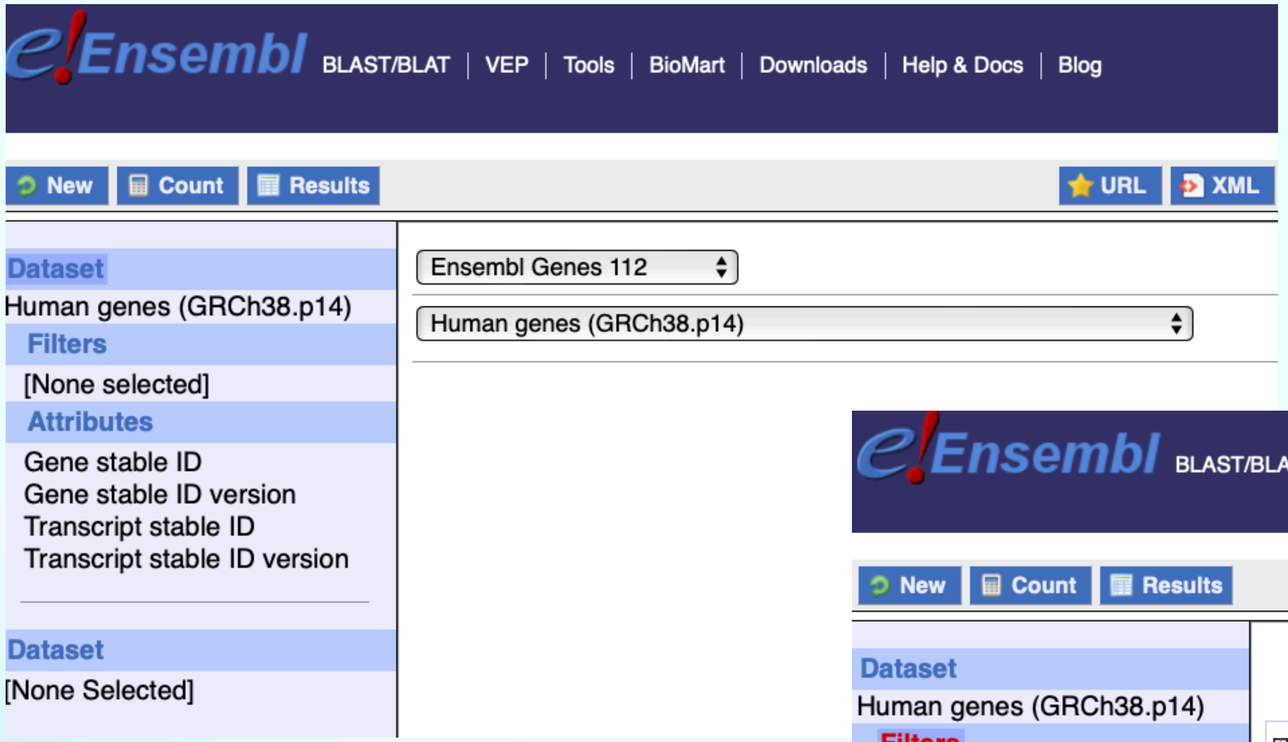
Using BioMart

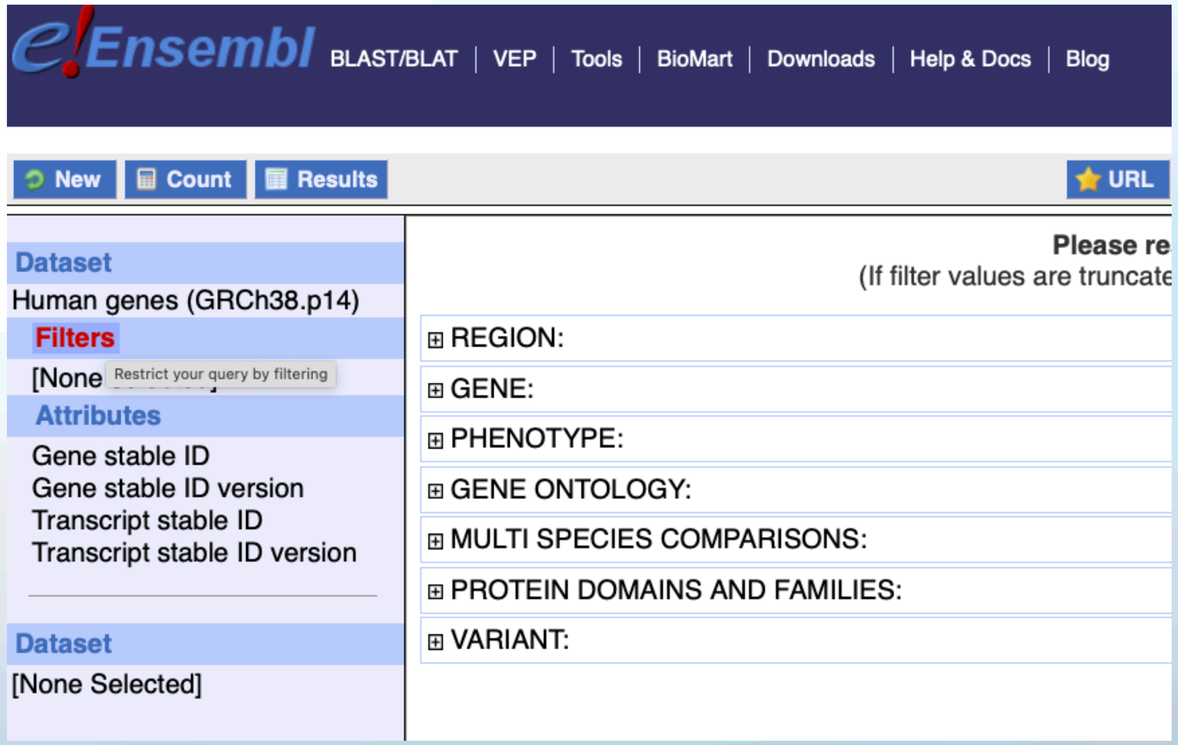
Select the database and dataset



Using BioMart

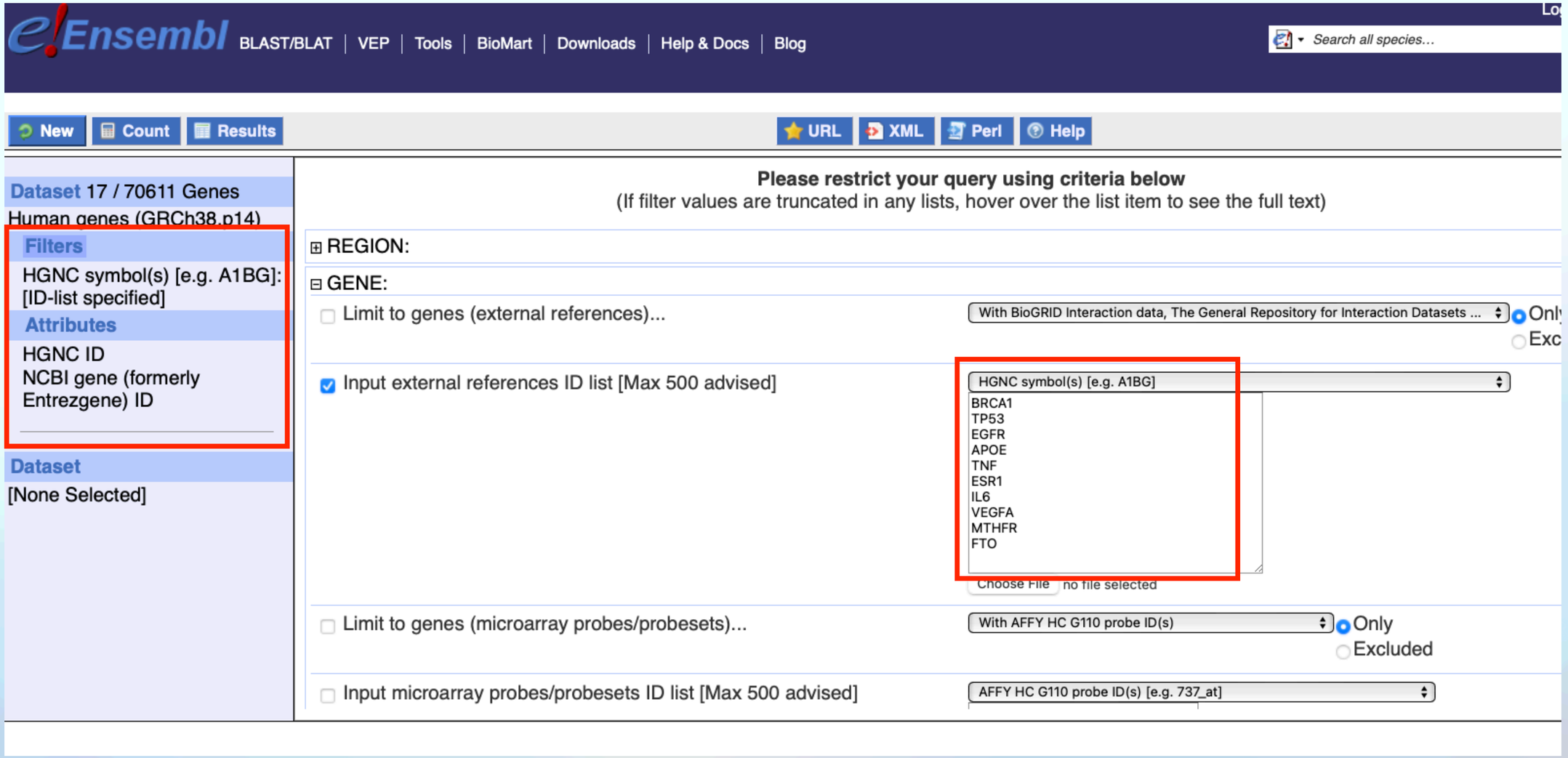
Configure the Filter (input) and Attributes (output)





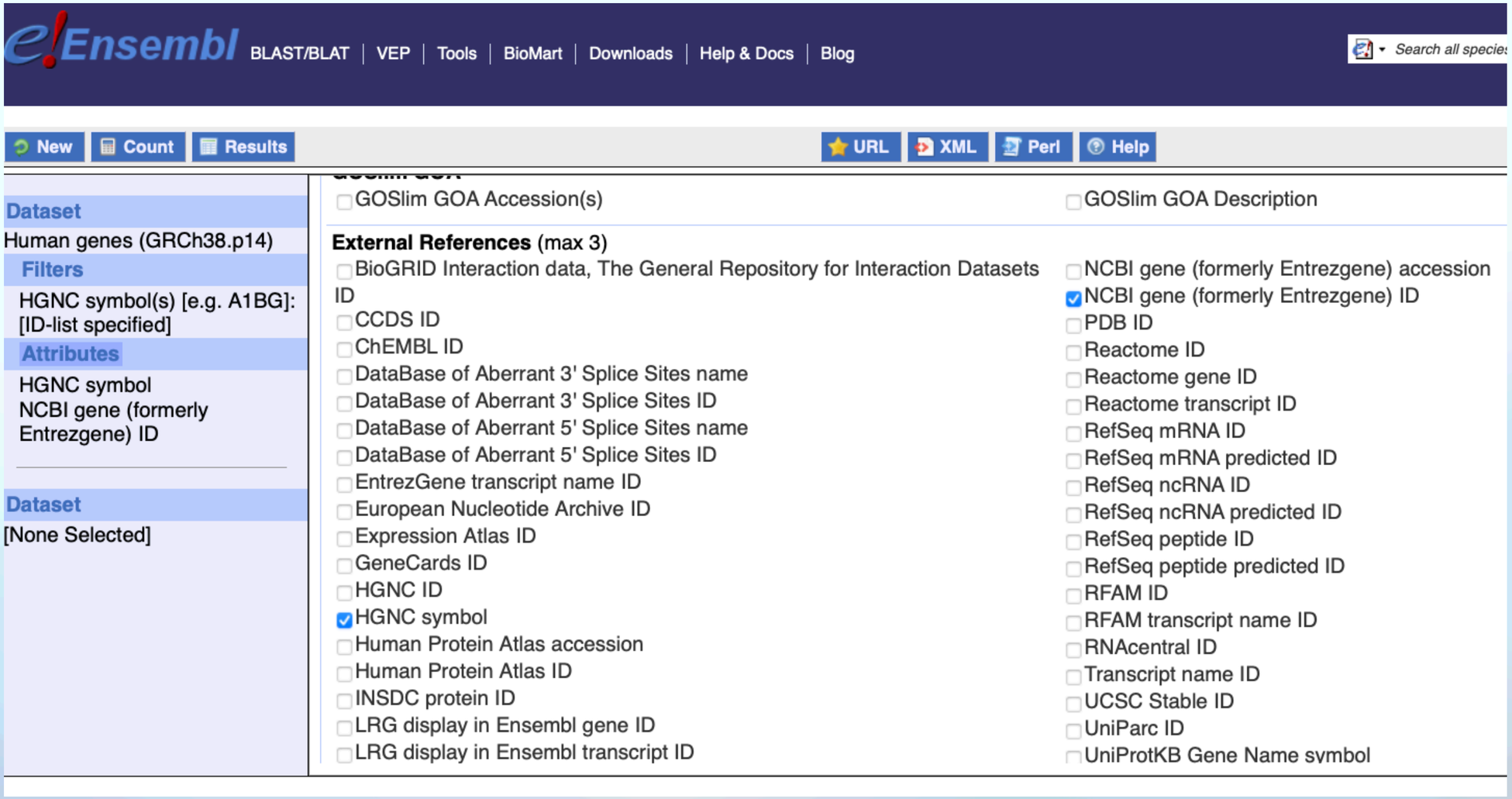
Using BioMart

Configure the Filter (input) and Attributes (output)

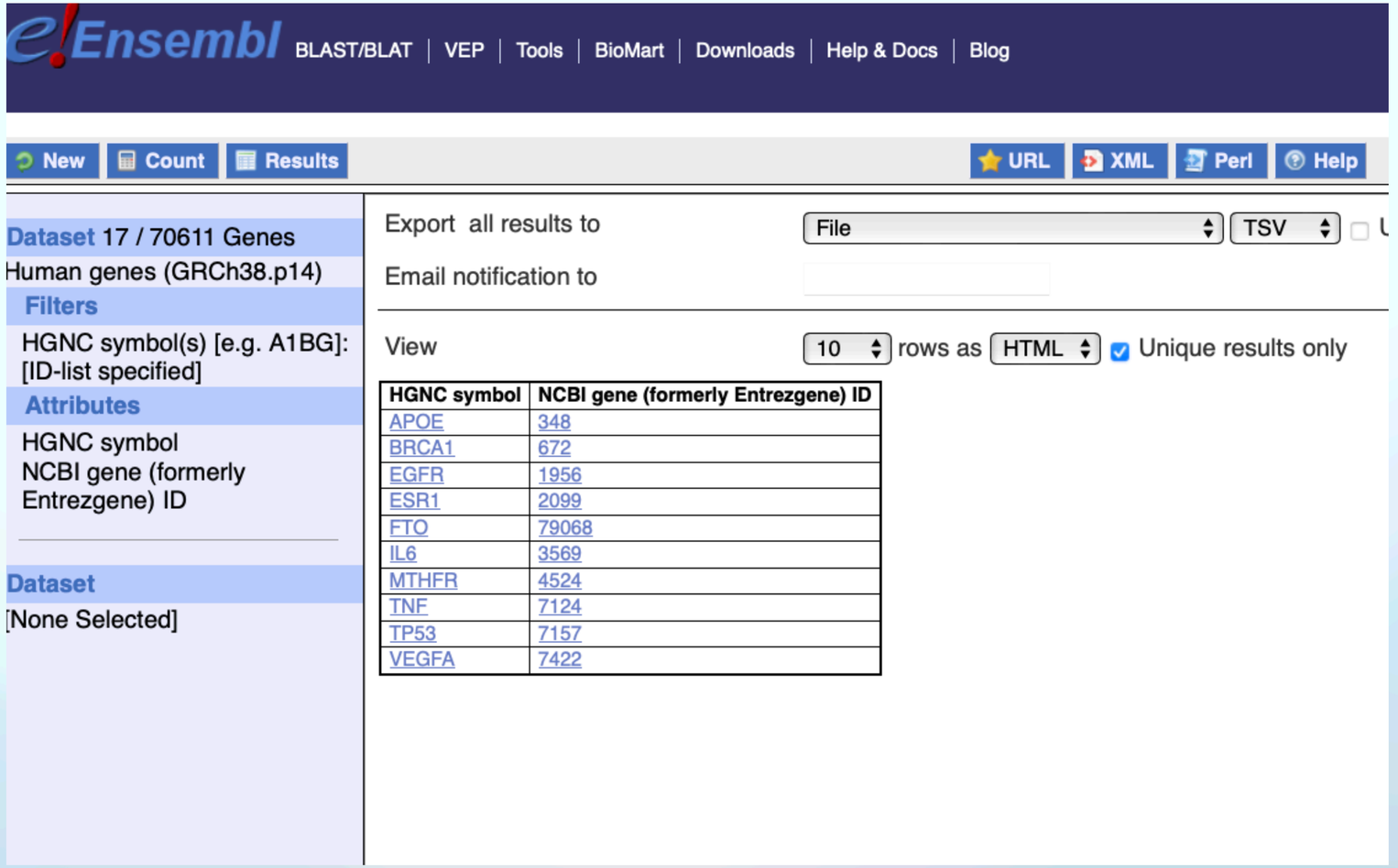


Using BioMart

Configure the Filter (input) and Attributes (output)



Using BioMart



HGNC symbol	NCBI gene (formerly Entrezgene) ID
APOE	348
BRCA1	672
EGFR	1956
ESR1	2099
FTO	79068
IL6	3569
MTHFR	4524
TNF	7124
TP53	7157
VEGFA	7422

Simple BioMart Example

Converting Gene Symbols into NCBI Gene IDs

Import

- Import the `requests` library to send an HTTP POST request to the Ensembl BioMart endpoint.

Function `get_ncbi_gene_ids()`

- Takes a list of gene symbols and an optional organism parameter (`hsapiens` is used for human).

Generates a BioMart XML query to request the HGNC symbols and NCBI Gene IDs

- Sends the query to Ensembl BioMart service and parses the response.
- Returns a dictionary mapping each gene symbol to its corresponding NCBI Gene ID.

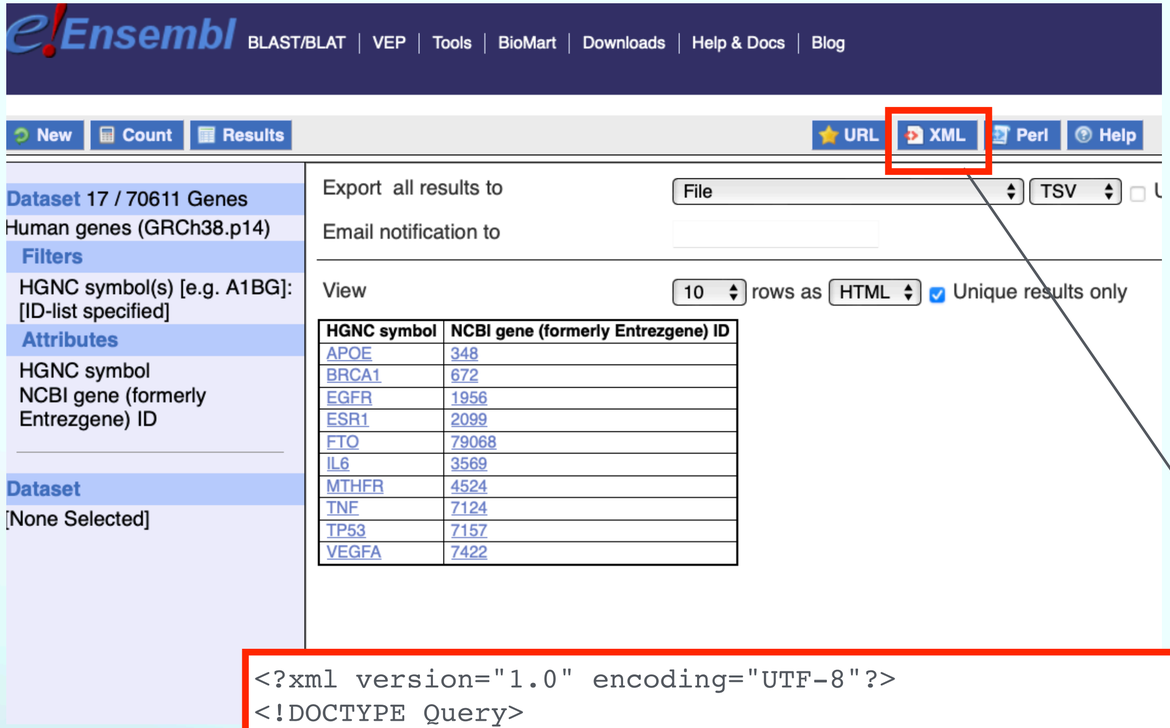
Details to Note

- The new piece for us here is the formatting of the query
- In BioMart the "filter" specifies which feature you are providing in the query. In this case it is a comma separated list of gene symbols
- In BioMart the attribute pair is the conversion, in this case gene symbol (HGNC) to NCBI (Entrez) Gene id
- You can specify the species, though the default is human
- **Practice on the Ensembl BioMart website to familiarise yourself with the way it works**

```python
import requests

def get_ncbi_gene_ids(gene_symbols, organism="hsapiens"):
    """
    Convert a list of gene symbols into NCBI Gene IDs using Ensembl BioMart.

    Parameters:
    gene_symbols (list): List of gene symbols to search for.
    organism (str): Organism prefix used by Ensembl BioMart (default is "hsapiens" for Homo sapiens).

    Returns:
    dict: A dictionary mapping gene symbols to NCBI Gene IDs.
    """
    # Prepare the XML query for BioMart
    query_xml = f"""<?xml version="1.0" encoding="UTF-8"?>
    <!DOCTYPE Query>
    <Query virtualSchemaName = "default" formatter = "TSV" header = "0" uniqueRows = "1" count = "">
        <Dataset name = "{organism}_gene_ensembl" interface = "default" >
            <Filter name = "hgnc_symbol" value = "{','.join(gene_symbols)}"/>
            <Attribute name = "hgnc_symbol" />
            <Attribute name = "entrezgene_id" />
        </Dataset>
    </Query>
    """

    # Send the request to Ensembl BioMart
    url = "https://www.ensembl.org/biomart/martservice"
    response = requests.post(url, data={"query": query_xml})

    # Parse the response
    gene_ids = {}
    if response.status_code == 200:
        for line in response.text.strip().split("\n"):
            symbol, gene_id = line.split("\t")
            gene_ids[symbol] = gene_id if gene_id != "" else None
    else:
        raise Exception(f"Error querying BioMart: {response.status_code}")

    return gene_ids

# Example usage
gene_symbols = ["BRCA1", "TP53", "EGFR", "APOE", "TNF", "ESR1", "IL6", "VEGFA", "MTHFR", "FTO"]
gene_ids = get_ncbi_gene_ids(gene_symbols)

# Print the gene symbol to NCBI Gene ID mapping
for symbol, gene_id in gene_ids.items():
    print(f"Gene Symbol: {symbol}, NCBI Gene ID: {gene_id}")
```

Simple BioMart Example

Converting Gene Symbols into NCBI Gene IDs



HGNC symbol	NCBI gene (formerly Entrezgene) ID
APOE	348
BRCA1	672
EGFR	1956
ESR1	2099
FTO	79068
IL6	3569
MTHFR	4524
TNF	7124
TP53	7157
VEGFA	7422

Notice the XML button (!) this is where you can find the details for the XML in the preceding code

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Query>
<Query  virtualSchemaName = "default" formatter = "TSV" header = "0" uniqueRows = "0" count = "" datasetConfigVersion = "0.6" >

   <Dataset name = "hsapiens_gene_ensembl" interface = "default" >
      <Filter name = "hgnc_symbol" value = "BRCA1,TP53,EGFR,APOE,TNF,ESR1,IL6,VEGFA,MTHFR,FTO"/>
      <Attribute name = "hgnc_id" />
      <Attribute name = "entrezgene_id" />
   </Dataset>
</Query>
```

Simpler(!) BioMart Example

Converting Gene Symbols into NCBI Gene IDs

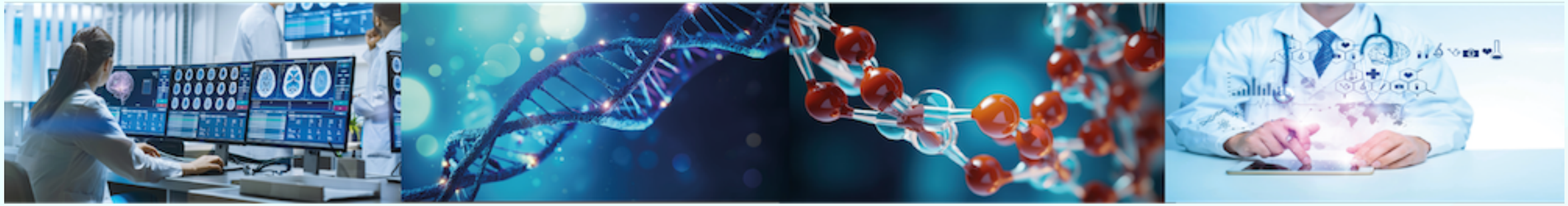
Import

- Import BiomartServer from biomart to interact with Ensembl's BioMart endpoint.

Function get_ncbi_gene_ids()

- Takes a list of gene symbols and an optional dataset parameter (hsapiens_gene_ensembl is used for humans).
- Connects to Ensembl's BioMart server.
- Accesses the specified dataset and sends a query for the given gene symbols, requesting the HGNC symbol and NCBI Gene ID attributes.
- Parses the response and stores the mapping in a dictionary.
- Returns the dictionary mapping each gene symbol to its corresponding NCBI Gene ID.

```python
from biomart import BiomartServer

def get_ncbi_gene_ids(gene_symbols, dataset="hsapiens_gene_ensembl"):
    """
    Convert a list of gene symbols into NCBI Gene IDs using Ensembl BioMart.

    Parameters:
    gene_symbols (list): List of gene symbols to search for.
    dataset (str): Ensembl BioMart dataset (default is "hsapiens_gene_ensembl" for Homo sapiens).

    Returns:
    dict: A dictionary mapping gene symbols to NCBI Gene IDs.
    """
    # Connect to the BioMart server
    server = BiomartServer("http://www.ensembl.org/biomart")
    server.verbose = False

    # Access the dataset
    mart = server.datasets[dataset]

    # Query the dataset
    response = mart.search({
        'filters': {
            'hgnc_symbol': gene_symbols
        },
        'attributes': [
            'hgnc_symbol', 'entrezgene_id'
        ]
    })

    # Parse the response
    gene_ids = {}
    for line in response.iter_lines():
        symbol, gene_id = line.decode().split("\t")
        gene_ids[symbol] = gene_id if gene_id != "" else None

    return gene_ids

# Example usage
gene_symbols = ["BRCA1", "TP53", "EGFR", "APOE", "TNF", "ESR1", "IL6", "VEGFA", "MTHFR", "FTO"]
gene_ids = get_ncbi_gene_ids(gene_symbols)

# Print the gene symbol to NCBI Gene ID mapping
for symbol, gene_id in gene_ids.items():
    print(f"Gene Symbol: {symbol}, NCBI Gene ID: {gene_id}")
```



Programming for Biomedical Informatics

Next Lecture this Thursday - “Data Integration & Summary Analysis”

Please Bring your Laptop!

Ask Questions on the Piazza Discussion Board

THE UNIVERSITY of EDINBURGH
informatics