LEK_5_NCBI

March 24, 2022

1 NCBI Databases

Name at least five different databases of the NCBI Entrez system. Write a short description for each named database.

1. PubMed database: contains millions of citations from life science journals. A great deal of these citations have abstracts, and millions have links to their full text articles (or both an abstract and a link to full text. PubMed is heavily linked to other core NCBI databases, thereby providing a crucial bridge between the data of molecular biology and the scientific literature.
2. Taxonomy database: provides links to all data for each taxonomic node, from superkingdoms to subspecies. The taxonomy database reflects sequence data from a great deal of formally described species. The Taxonomy Browser can be used to view the taxonomy tree or retrieve data from any of the Entrez databases for a particular organism or group.
3. RefSeq (Reference Sequence) database: allows a user access to non-redundant set of curated and computationally derived sequences for transcripts, proteins and genomic regions. RefSeq DNA and RNA sequences can be searched and retrieved from the Nucleotide database.
4. SRA (Sequence Read Archive) database: contains data generated by the latest generation of high-throughput nucleic acid sequencers.SRA stores raw sequence reads and alignments. Data are deposited into SRA as supporting evidence for a wide range of study types including de novo genome assemblies, GWAS, single nucleotide polymorphism and structural variation analysis, pathogen identification, transcript assembly, metagenomic community profiling and epigenetics.
5. Epigenomics database: collects data from studies examining epigenetic features such as post-translational modifications of histone proteins, genomic DNA methylation, chromatin organization and the expression of non-coding regulatory RNA. Raw data from these experiments, together with extensive metadata, are stored in the GEO and SRA databases. Data arepremapped to genomic coordinates, so users are not required to be familiar with or manipulate the raw data.

2 Biopython

Write down the example syntax for the EInfo, ESearch, ESummary, EFetch and ELink method as you did during your exercises. You can run all methods in one script.

[56]:
```
import Bio
Bio.__version__
```

[56]:
```
'1.79'
```

[57]:
```
from Bio import Entrez as ez
ez.email = 'biopython@gmail.com' # fake email address
```

2.1 EInfo

[58]:
```
# EInfo (to get a list of DBs)
record_1 = ez.read(ez.einfo())
print(type(record_1))
print(record_1.keys())
print(record_1['DbList'])
# print('\n'.join(sorted(record_1['DbList']))) # to make a string with each db␣
 ↪at a new line
```

```
<class 'Bio.Entrez.Parser.DictionaryElement'>
dict_keys(['DbList'])
['pubmed', 'protein', 'nuccore', 'ipg', 'nucleotide', 'structure', 'genome',
'annotinfo', 'assembly', 'bioproject', 'biosample', 'blastdbinfo', 'books',
'cdd', 'clinvar', 'gap', 'gapplus', 'grasp', 'dbvar', 'gene', 'gds',
'geoprofiles', 'homologene', 'medgen', 'mesh', 'ncbisearch', 'nlmcatalog',
'omim', 'orgtrack', 'pmc', 'popset', 'proteinclusters', 'pcassay', 'protfam',
'biosystems', 'pccompound', 'pcsubstance', 'seqannot', 'snp', 'sra', 'taxonomy',
'biocollections', 'gtr']
```

[89]:
```
# EInfo (e.g. for the Genome database)
record_2 = ez.read(ez.einfo(db = 'genome'))
print(type(record_2))
print(record_2.keys())
#print(record_2['DbInfo'])
for key in record_2['DbInfo'].keys():
    if key != 'FieldList':
        print(key, ':', record_2['DbInfo'][key])
```

```
<class 'Bio.Entrez.Parser.DictionaryElement'>
dict_keys(['DbInfo'])
DbName : genome
MenuName : Genome
Description : Genomic sequences, contigs, and maps
DbBuild : Build220322-0605.1
Count : 86051
LastUpdate : 2022/03/22 09:36
LinkList : [{'Name': 'genome_assembly', 'Menu': 'Assembly', 'Description':
'Related Assembly records', 'DbTo': 'assembly'}, {'Name': 'genome_bioproject',
'Menu': 'BioProject Links', 'Description': 'Related BioProjects', 'DbTo':
'bioproject'}, {'Name': 'genome_gene', 'Menu': 'Gene Links', 'Description':
'Related Gene', 'DbTo': 'gene'}, {'Name': 'genome_nuccore', 'Menu':
```

```
'Components', 'Description': 'Constituent nucleotide sequence', 'DbTo':
'nuccore'}, {'Name': 'genome_nuccore_samespecies', 'Menu': 'Other genomes for
species', 'Description': 'Other (nearly) complete genomes or particular genomic
segments for the species from DDBJ/EMBL/GenBank', 'DbTo': 'nuccore'}, {'Name':
'genome_protein', 'Menu': 'Protein Links', 'Description': 'Constituent protein
sequence', 'DbTo': 'protein'}, {'Name': 'genome_proteinclusters', 'Menu':
'Protein Cluster Links', 'Description': 'Related Protein Clusters', 'DbTo':
'proteinclusters'}, {'Name': 'genome_pubmed', 'Menu': 'PubMed Links',
'Description': 'PubMed articles cited by genome record', 'DbTo': 'pubmed'},
{'Name': 'genome_taxonomy', 'Menu': 'Taxonomy Links', 'Description': 'Taxonomy
sequences associated with genome record', 'DbTo': 'taxonomy'}]
```

2.2 ESearch

[74]:
```
# ESearch
'''Comment: I googled and found out that Badhamia lilacina is not very well␣
 ↪studied. So, I decided
to do my search against Badhamia (term 'Badhamia lilacina' gives only one␣
 ↪count'''
record_3 = ez.read(ez.esearch(db ='pubmed', term = 'Badhamia', retmax = 7))
print(type(record_3))
print(record_3.keys())
for key in record_3.keys():
    print(key, ':', record_3[key])
BadhamiaIDs = record_3['IdList']
print(BadhamiaIDs) # list of PubMed IDs
```

```
<class 'Bio.Entrez.Parser.DictionaryElement'>
dict_keys(['Count', 'RetMax', 'RetStart', 'IdList', 'TranslationSet',
'TranslationStack', 'QueryTranslation'])
Count : 7
RetMax : 7
RetStart : 0
IdList : ['24132078', '15132172', '21156594', '24221179', '7242651', '4939438',
'13490378']
TranslationSet : []
TranslationStack : [{'Term': 'Badhamia[All Fields]', 'Field': 'All Fields',
'Count': '7', 'Explode': 'N'}, 'GROUP']
QueryTranslation : Badhamia[All Fields]
['24132078', '15132172', '21156594', '24221179', '7242651', '4939438',
'13490378']
```

2.3 ESummary

[75]:
```
# ESummary
'''Here I use my list of IDs I created searching against "Badhamia", but I␣
 ↪limit it up to 3 records
only to descrease in size my potential output'''
for ID in BadhamiaIDs[:3]:
    jr_summary = ez.read(ez.esummary(db = 'pubmed', id=ID))
    #print(jr_summary[0].keys())
    for handle in jr_summary:
        print(f"Date of publication:\n{handle['PubDate']}\nFirst author:
 ↪\n{handle['AuthorList'][0]}\nTitle:\n{handle['Title']}\n")

# Here I am prinditng out dates of publications, the first authors for each␣
 ↪paper and the titles.
```

```
Date of publication:
2014 Apr
First author:
Aguilar M
Title:
Using environmental niche models to test the 'everything is everywhere'
hypothesis for Badhamia.

Date of publication:
2003 Jul-Aug
First author:
Haugen P
Title:
The molecular evolution and structural organization of self-splicing group I
introns at position 516 in nuclear SSU rDNA of myxomycetes.

Date of publication:
2003 Jan-Feb
First author:
Clark J
Title:
Biosystematics of the myxomycete Badhamia gracilis.
```

2.4 EFetch

[99]:
```
# EFetch
# https://medlineplus.gov/genetics/gene/sry/
'''Just for fun decided to play around with SRY gene - a gender-determining␣
 ↪one'''
# SRY: sex-determining region Y protein
```

```
record_4 = ez.read(ez.esearch(db = 'nucleotide', term = 'SRY[Gene Name] AND␣
 ↪RefSeq[Keyword]', retmax=2000, idtype = 'acc'))
#print(record_4)
counter_1 = 0
fetchList = []
for ID in record_4['IdList']:
    if 'NM_' in ID: # catches only mRNA
        counter_1 += 1
        fetch = ez.efetch(db = 'nucleotide', id = ID, rettype='fasta', retmode␣
 ↪= 'text')
        #readFetch = fetch.readline()
        readFetch = fetch.read()
        fetchList.append(readFetch)
        #print(readFetch)
#print(fetchList)
print(len(fetchList))
#print(f"Number of the NM_ records found: {counter_1}") # outputs only 23␣
 ↪records

for files in fetchList:
    with open('SRY.fasta', 'a+') as savedSRY:
        savedSRY.write(files) # now there is a SRY.fasta file in my current␣
 ↪directory
with open ('SRY.fasta', 'r') as f:
    first_line = next(f)
# printing out only the first line of the long fasta file to check that the␣
 ↪code is working all right
print(first_line)
```

```
23
>NM_003140.3 Homo sapiens sex determining region Y (SRY), mRNA
```

2.5 ELink

[85]:
```
# ELink
'''I will use my previously created list of IDs for Badhamia'''
print(BadhamiaIDs)
pmid = BadhamiaIDs[0] # get the first paper in my list PMID 24132078
print(pmid) # PMCID is 3960529)
record_5 = ez.read(ez.elink(dbfrom="pubmed", id=pmid))
#print(type(record_5)) # list
#print(record_5[0]) # getting into dictionary
print(record_5[0].keys())
print(f"Database: {record_5[0]['DbFrom']}, ID: {record_5[0]['IdList']}")
# LinkSetDb - contains my search results
```

```
print(f'Number of crosslinks found: {len(record_5[0]["LinkSetDb"])}') # I found␣
↪7 crosslinks!
# print(type(record_5[0]["LinkSetDb"])) # list
#print(record_5[0]["LinkSetDb"][0]) # dictionary
#print(record_5[0]["LinkSetDb"][0]['Link']) # list of dictionaries
counter_2 = 0
pmIDlist = []
for link in record_5[0]["LinkSetDb"][0]['Link']:
    #print(link['Id'])
    counter_2 += 1
    pmIDlist.append(link['Id'])
print(f'Number of PubMed IDs found: {counter_2}') # found 105 PubMed␣
↪crosslinked IDs.
print(pmIDlist)
print(f'E.g. this paper with ID {pmIDlist[1]} has in references my target␣
↪paper, ref. 27') # e.g this paper has a reference to my target paper ref. 27
```

```
['24132078', '15132172', '21156594', '24221179', '7242651', '4939438',
'13490378']
24132078
dict_keys(['ERROR', 'LinkSetDb', 'LinkSetDbHistory', 'DbFrom', 'IdList'])
Database: pubmed, ID: ['24132078']
Number of crosslinks found: 7
Number of PubMed IDs found: 105
['24132078', '28414791', '22402402', '25550302', '32649270', '22530009',
'26663217', '18522694', '19750951', '29533140', '25028491', '25232071',
'27128786', '26702956', '31586669', '23667494', '29024429', '29953506',
'20456231', '31613722', '31059141', '22962350', '17854485', '23497060',
'26321302', '33311142', '20524603', '34676941', '18836695', '24486990',
'19197466', '33171353', '31874965', '28919504', '29167159', '15132172',
'30903202', '22492409', '34913576', '24433389', '22818199', '28647618',
'33966339', '25451805', '27190206', '31055860', '17045493', '26240305',
'21262984', '31582156', '29337274', '32829066', '29248627', '26432394',
'17062415', '33090298', '26095958', '16367842', '33187474', '22776548',
'25300454', '17603517', '24018859', '23998707', '31358042', '27593700',
'24118115', '18761283', '22970129', '24987129', '17306538', '25732069',
'34624792', '30187122', '22421085', '32405644', '23869920', '27351595',
'31904883', '19168565', '22451600', '23167902', '27233443', '32399960',
'15212388', '30076328', '10633916', '22319123', '20943518', '22516430',
'30783861', '27539600', '17594425', '30945986', '28487259', '28187628',
'22112441', '26336224', '20594354', '17560828', '20525639', '20205541',
'22906180', '24838118', '34533412']
E.g. this paper with ID 28414791 has in references my target paper, ref. 27
```