Class 21: Searching the NCBI databases

April 9, 2019

Different ways of processing Entrez results

Every time we download data through the Entrez module, we can interact with the results in different ways. First, we can just use the handle we obtain as an ordinary file handle and just store or process the raw data provided by Entrez. Second, we can process the data with an appropriate, existing Biopython module. The latter will generally be preferable if an appropriate module exists. However, the various choices that are available may make things confusing.

As an example, we will again download the genbank record with the ID "KT220438", containing an influenza HA protein. We will consider four different ways of looking at the data. First, we use `retmode="text"` in `Entrez.efetch()` and just download the raw data and print it. We get a regular genbank file as output:

In [1]:
```
from Bio import Entrez, SeqIO
Entrez.email = "wilke@austin.utexas.edu" # put your email here

# Download sequence record for genbank id KT220438 (HA from influenza A)
# Using text mode
handle = Entrez.efetch(db="nucleotide", id="KT220438", rettype="gb", retmode="t
ext")
record = handle.read() # read file directly
print(record)
handle.close()
```

```
LOCUS       KT220438               1701 bp    cRNA    linear   VRL 20-JUL-2015
DEFINITION  Influenza A virus (A/NewJersey/NHRC_93219/2015(H3N2)) segment 4
            hemagglutinin (HA) gene, complete cds.
ACCESSION   KT220438
VERSION     KT220438.1
KEYWORDS    .
SOURCE      Influenza A virus (A/New Jersey/NHRC_93219/2015(H3N2))
  ORGANISM  Influenza A virus (A/New Jersey/NHRC_93219/2015(H3N2))
            Viruses; ssRNA viruses; ssRNA negative-strand viruses;
            Orthomyxoviridae; Influenzavirus A.
REFERENCE   1  (bases 1 to 1701)
  AUTHORS   Sitz,C.R., Thammavong,H.L., Balansay-Ames,M.S., Hawksworth,A.W.,
            Myers,C.A. and Brice,G.T.
  TITLE     GEISS Influenza Surveillance Response Program
  JOURNAL   Unpublished
REFERENCE   2  (bases 1 to 1701)
  AUTHORS   Sitz,C.R., Thammavong,H.L., Balansay-Ames,M.S., Hawksworth,A.W.,
            Myers,C.A. and Brice,G.T.
  TITLE     Direct Submission
  JOURNAL   Submitted (29-JUN-2015) Operational Infectious Diseases, Naval
            Health Research Center, 140 Sylvester Rd., San Diego, CA 92106, USA
COMMENT     ##Assembly-Data-START##
            Sequencing Technology :: Sanger dideoxy sequencing
            ##Assembly-Data-END##
FEATURES             Location/Qualifiers
     source          1..1701
                     /organism="Influenza A virus (A/New
                     Jersey/NHRC_93219/2015(H3N2))"
                     /mol_type="viral cRNA"
                     /strain="A/NewJersey/NHRC_93219/2015"
                     /serotype="H3N2"
                     /isolation_source="nasopharyngeal swab"
                     /host="Homo sapiens"
                     /db_xref="taxon:1682360"
                     /segment="4"
                     /lab_host="MDCK"
                     /country="USA: New Jersey"
                     /collection_date="17-Jan-2015"
     gene            1..1701
                     /gene="HA"
     CDS             1..1701
                     /gene="HA"
                     /function="receptor binding and fusion protein"
                     /codon_start=1
                     /product="hemagglutinin"
                     /protein_id="AKQ43545.1"
                     /translation="MKTIIALSYILCLVFAQKIPGNDNSTATLCLGHHAVPNGTIVKT
                     ITNDRIEVTNATELVQNSSIGEICDSPHQILDGENCTLIDALLGDPQCDGFQNKKWDL
                     FVERSKAYSNCYPYDVPDYASLRSLVASSGTLEFNNESFNWTGVTQNGTSSACIRRSS
                     SSFFSRLNWLTHLNYTYPALNVTMPNNEQFDKLYIWGVHHPGTDKDQIFLYAQSSGRI
                     TVSTKRSQQAVIPNIGSRPRIRDIPSRISIYWTIVKPGDILLINSTGNLIAPRGYFKI
                     RSGKSSIMRSDAPIGKCKSECITPNGSIPNDKPFQNVNRITYGACPRYVKHSTLKLAT
                     GMRNVPEKQTRGIFGAIAGFIENGWEGMVDGWYGFRHQNSEGRGQAADLKSTQAAIDQ
                     INGKLNRLIGKTNEKFHQIEKEFSEVEGRIQDLEKYVEDTKIDLWSYNAELLVALENQ
                     HTXDLTDSEMNKLFEKTKKQLRENAEDMGNGCFKIYHKCDNACIGSIRNGTYDHNVYR
                     DEALNNRFQIKGVELKSGYKDWILWISXAISCFLLCVALLGFIMWACQKGNIRCNICI
                     "
     mat_peptide     49..1035
                     /gene="HA"
                     /product="HA1"
     mat_peptide     1036..1698
                     /gene="HA"
                     /product="HA2"
```

We can also, as we have done before, process this file using the `SeqIO` module:

In [2]:
```
# Download sequence record for genbank id KT220438 (HA from influenza A)
# Using text mode
handle = Entrez.efetch(db="nucleotide", id="KT220438", rettype="gb", retmode="t
ext")
record = SeqIO.read(handle, "genbank") # parse with SeqIO
print(record)
handle.close()
```

```
ID: KT220438.1
Name: KT220438
Description: Influenza A virus (A/NewJersey/NHRC_93219/2015(H3N2)) segment 4 he
magglutinin (HA) gene, complete cds
Number of features: 5
/structured_comment=OrderedDict([('Assembly-Data', OrderedDict([('Sequencing Te
chnology', 'Sanger dideoxy sequencing')]))])
/date=20-JUL-2015
/molecule_type=cRNA
/data_file_division=VRL
/keywords=['']
/accessions=['KT220438']
/topology=linear
/taxonomy=['Viruses', 'ssRNA viruses', 'ssRNA negative-strand viruses', 'Orthom
yxoviridae', 'Influenzavirus A']
/source=Influenza A virus (A/New Jersey/NHRC_93219/2015(H3N2))
/organism=Influenza A virus (A/New Jersey/NHRC_93219/2015(H3N2))
/sequence_version=1
/references=[Reference(title='GEISS Influenza Surveillance Response Program', .
..), Reference(title='Direct Submission', ...)]
Seq('ATGAAGACTATCATTGCTTTGAGCTACATTCTATGTCTGGTTTTCGCTCAAAAA...TGA', IUPACAmbigu
ousDNA())
```

In addition to text mode, we can also download the data in XML mode. XML is a structured data format that allows for easy machine-processing of complex data files. If we just print the raw data, though, it doesn't look appealing:

In [3]:
```
# Download sequence record for genbank id KT220438 (HA from influenza A)
# Using XML mode
handle = Entrez.efetch(db="nucleotide", id="KT220438", rettype="gb", retmode="x
ml")
record = handle.read() # read file directly
print(record)
handle.close()
```

```
<?xml version="1.0" encoding="UTF-8"  ?>
<!DOCTYPE GBSet PUBLIC "-//NCBI//NCBI GBSeq/EN" "https://www.ncbi.nlm.nih.gov/d
td/NCBI_GBSeq.dtd">
<GBSet>
  <GBSeq>

    <GBSeq_locus>KT220438</GBSeq_locus>
    <GBSeq_length>1701</GBSeq_length>
    <GBSeq_strandedness>single</GBSeq_strandedness>
    <GBSeq_moltype>cRNA</GBSeq_moltype>
    <GBSeq_topology>linear</GBSeq_topology>
    <GBSeq_division>VRL</GBSeq_division>
    <GBSeq_update-date>20-JUL-2015</GBSeq_update-date>
    <GBSeq_create-date>20-JUL-2015</GBSeq_create-date>
    <GBSeq_definition>Influenza A virus (A/NewJersey/NHRC_93219/2015(H3N2)) seg
ment 4 hemagglutinin (HA) gene, complete cds</GBSeq_definition>
    <GBSeq_primary-accession>KT220438</GBSeq_primary-accession>
    <GBSeq_accession-version>KT220438.1</GBSeq_accession-version>
    <GBSeq_other-seqids>
      <GBSeqid>gb|KT220438.1|</GBSeqid>
      <GBSeqid>gi|887493048</GBSeqid>
    </GBSeq_other-seqids>
    <GBSeq_source>Influenza A virus (A/New Jersey/NHRC_93219/2015(H3N2))</GBSeq
_source>
    <GBSeq_organism>Influenza A virus (A/New Jersey/NHRC_93219/2015(H3N2))</GBS
eq_organism>
    <GBSeq_taxonomy>Viruses; ssRNA viruses; ssRNA negative-strand viruses; Orth
omyxoviridae; Influenzavirus A</GBSeq_taxonomy>
    <GBSeq_references>
      <GBReference>
        <GBReference_reference>1</GBReference_reference>
        <GBReference_position>1..1701</GBReference_position>
        <GBReference_authors>
          <GBAuthor>Sitz,C.R.</GBAuthor>
          <GBAuthor>Thammavong,H.L.</GBAuthor>
          <GBAuthor>Balansay-Ames,M.S.</GBAuthor>
          <GBAuthor>Hawksworth,A.W.</GBAuthor>
          <GBAuthor>Myers,C.A.</GBAuthor>
          <GBAuthor>Brice,G.T.</GBAuthor>
        </GBReference_authors>
        <GBReference_title>GEISS Influenza Surveillance Response Program</GBRef
erence_title>
        <GBReference_journal>Unpublished</GBReference_journal>
      </GBReference>
      <GBReference>
        <GBReference_reference>2</GBReference_reference>
        <GBReference_position>1..1701</GBReference_position>
        <GBReference_authors>
          <GBAuthor>Sitz,C.R.</GBAuthor>
          <GBAuthor>Thammavong,H.L.</GBAuthor>
          <GBAuthor>Balansay-Ames,M.S.</GBAuthor>
          <GBAuthor>Hawksworth,A.W.</GBAuthor>
          <GBAuthor>Myers,C.A.</GBAuthor>
          <GBAuthor>Brice,G.T.</GBAuthor>
        </GBReference_authors>
        <GBReference_title>Direct Submission</GBReference_title>
        <GBReference_journal>Submitted (29-JUN-2015) Operational Infectious Dis
eases, Naval Health Research Center, 140 Sylvester Rd., San Diego, CA 92106, US
A</GBReference_journal>
      </GBReference>
    </GBSeq_references>
    <GBSeq_comment>##Assembly-Data-START## ; Sequencing Technology :: Sanger di
deoxy sequencing ; ##Assembly-Data-END##</GBSeq_comment>
```

The advantage of XML mode is that there is the generic `Entrez.parse()` function that can parse XML files returned from `Entrez.efetch()`. Also, some modules in Biopython cannot work with files obtained in text mode, they can only work on files obtained in XML mode. The documentation will generally tell you for each module what kind of input it expects.

Reading the above example with `Entrez.parse()` gives us the following:

In [4]:
```
# Download sequence record for genbank id KT220438 (HA from influenza A)
handle = Entrez.efetch(db="nucleotide", id="KT220438", rettype="gb", retmode="x
ml")
parsed = Entrez.parse(handle)
record = list(parsed)[0] # We need to convert the parsed contents into a list.
Here we want just the 0th element.
handle.close()
print(record)
```

DictElement({'GBSeq_division': 'VRL', 'GBSeq_comment': '##Assembly-Data-START##
; Sequencing Technology :: Sanger dideoxy sequencing ; ##Assembly-Data-END##',
'GBSeq_references': [DictElement({'GBReference_reference': '1', 'GBReference_po
sition': '1..1701', 'GBReference_title': 'GEISS Influenza Surveillance Response
Program', 'GBReference_authors': ['Sitz,C.R.', 'Thammavong,H.L.', 'Balansay-Ame
s,M.S.', 'Hawksworth,A.W.', 'Myers,C.A.', 'Brice,G.T.'], 'GBReference_journal':
'Unpublished'}, attributes={}), DictElement({'GBReference_reference': '2', 'GBR
eference_position': '1..1701', 'GBReference_title': 'Direct Submission', 'GBRef
erence_authors': ['Sitz,C.R.', 'Thammavong,H.L.', 'Balansay-Ames,M.S.', 'Hawksw
orth,A.W.', 'Myers,C.A.', 'Brice,G.T.'], 'GBReference_journal': 'Submitted (29-
JUN-2015) Operational Infectious Diseases, Naval Health Research Center, 140 Sy
lvester Rd., San Diego, CA 92106, USA'}, attributes={})], 'GBSeq_locus': 'KT220
438', 'GBSeq_strandedness': 'single', 'GBSeq_update-date': '20-JUL-2015', 'GBSe
q_organism': 'Influenza A virus (A/New Jersey/NHRC_93219/2015(H3N2))', 'GBSeq_a
ccession-version': 'KT220438.1', 'GBSeq_source': 'Influenza A virus (A/New Jers
ey/NHRC_93219/2015(H3N2))', 'GBSeq_topology': 'linear', 'GBSeq_taxonomy': 'Viru
ses; ssRNA viruses; ssRNA negative-strand viruses; Orthomyxoviridae; Influenzav
irus A', 'GBSeq_other-seqids': ['gb|KT220438.1|', 'gi|887493048'], 'GBSeq_defin
ition': 'Influenza A virus (A/NewJersey/NHRC_93219/2015(H3N2)) segment 4 hemagg
lutinin (HA) gene, complete cds', 'GBSeq_create-date': '20-JUL-2015', 'GBSeq_le
ngth': '1701', 'GBSeq_primary-accession': 'KT220438', 'GBSeq_sequence': 'atgaag
actatcattgctttgagctacattctatgtctggttttcgctcaaaaaattcctggaaatgacaatagcacggcaacgc
tgtgccttgggcaccatgcagtaccaaacggaacgatagtgaaaacaatcacaaatgaccgaattgaagttactaatgc
tactgagctggttcagaattcctcaataggtgaaatatgcgacagtcctcatcagatccttgatggagaaaactgcaca
ctaatagatgctctattgggagaccctcagtgtgatggctttcaaaataagaaatgggacctttttgttgaacgaagca
aagcctacagcaactgctacccttatgatgtgccggattatgcctcccttaggtcactagttgcctcatccggcacact
ggagtttaacaatgaaagcttcaattggactggagtcactcaaaacggaacaagttctgcttgcataaggagatctagt
agtagtttctttagtagattaaattggttgacccacttaaactacacatacccagcattgaacgtgactatgccaaaca
atgaacaatttgacaaattgtacatttgggggttcaccacccgggtacggacaaggaccaaatcttcctgtatgctca
atcatcaggaagaatcacagtatctaccaaaagaagccaacaagctgtaatcccaaatatcggatctagacccagaata
agggatatccctagcagaataagcatctattggacaatagtaaaaccgggagacatactttgattaacagcacaggga
atctaattgctcctaggggttacttcaaaatacgaagtgggaaaagctcaataatgagatcagatgcacccattggcaa
atgcaagtctgaatgcatcactccaaatggaagcattcccaatgacaaaccattccaaaatgtaaacaggatcacatac
ggggcctgtcccagatatgttaagcatagcactctaaaattggcaacaggaatgcgaaatgtaccagagaaacaaacta
gaggcatatttggcgcaatagcgggtttcatagaaaatggttgggagggaatggtggatggttggtacggtttcaggca
tcaaaattctgagggaagaggacaagcagcagatctcaaaagcactcaagcagcaatcgatcaaatcaatgggaagctg
aatcgattgatcgggaaaaccaacgagaaattccatcagattgaaaaagaattctcagaagtagaaggaagaattcagg
accttgagaaatatgttgaggacactaaaatagatctctggtcatacaacgcggagcttcttgttgccctggagaacca
acatacarttgatctaactgactcagaaatgaacaaactgtttgaaaaaacaaagaagcaactgagggaaaatgctgag
gatatgggaaatggttgtttcaaaatataccacaaatgtgacaatgcctgcataggatcaataagaaatggaacttatg
accacaatgtgtacagggatgaagcattaaacaaccggttccagatcaagggagttgagctgaagtcagggtacaaaga
ttggatcctatggatttcctytgccatatcatgtttttgctttgtgttgctttgttggggttcatcatgtgggcctgc
caaaagggcaacattaggtgcaacatttgcatttga', 'GBSeq_feature-table': [DictElement({'GB
Feature_quals': [DictElement({'GBQualifier_name': 'organism', 'GBQualifier_valu
e': 'Influenza A virus (A/New Jersey/NHRC_93219/2015(H3N2))'}, attributes={}),
DictElement({'GBQualifier_name': 'mol_type', 'GBQualifier_value': 'viral cRNA'}
, attributes={}), DictElement({'GBQualifier_name': 'strain', 'GBQualifier_value
': 'A/NewJersey/NHRC_93219/2015'}, attributes={}), DictElement({'GBQualifier_na
me': 'serotype', 'GBQualifier_value': 'H3N2'}, attributes={}), DictElement({'GB
Qualifier_name': 'isolation_source', 'GBQualifier_value': 'nasopharyngeal swab'
}, attributes={}), DictElement({'GBQualifier_name': 'host', 'GBQualifier_value'
: 'Homo sapiens'}, attributes={}), DictElement({'GBQualifier_name': 'db_xref',
'GBQualifier_value': 'taxon:1682360'}, attributes={}), DictElement({'GBQualifie
r_name': 'segment', 'GBQualifier_value': '4'}, attributes={}), DictElement({'GB
Qualifier_name': 'lab_host', 'GBQualifier_value': 'MDCK'}, attributes={}), Dict
Element({'GBQualifier_name': 'country', 'GBQualifier_value': 'USA: New Jersey'}
, attributes={}), DictElement({'GBQualifier_name': 'collection_date', 'GBQualif
ier_value': '17-Jan-2015'}, attributes={})], 'GBFeature_intervals': [DictElemen
t({'GBInterval_to': '1701', 'GBInterval_from': '1', 'GBInterval_accession': 'KT
220438.1'}, attributes={})], 'GBFeature_key': 'source', 'GBFeature_location': '
1..1701'}, attributes={}), DictElement({'GBFeature_quals': [DictElement({'GBQua
lifier_name': 'gene', 'GBQualifier_value': 'HA'}, attributes={})], 'GBFeature_i
ntervals': [DictElement({'GBInterval_to': '1701', 'GBInterval_from': '1', 'GBIn

While this output may not seem useful, we now just have a set of nested dictionaries that we can interrogate using standard Python techniques:

In [5]:
```
print(list(record.keys())) # print out all the keys in the dictionary
```

```
['GBSeq_division', 'GBSeq_comment', 'GBSeq_references', 'GBSeq_locus', 'GBSeq_s
trandedness', 'GBSeq_update-date', 'GBSeq_organism', 'GBSeq_accession-version',
'GBSeq_source', 'GBSeq_topology', 'GBSeq_taxonomy', 'GBSeq_other-seqids', 'GBSe
q_definition', 'GBSeq_create-date', 'GBSeq_length', 'GBSeq_primary-accession',
'GBSeq_sequence', 'GBSeq_feature-table', 'GBSeq_moltype']
```

In [6]:
```
print(record['GBSeq_sequence']) # print out the sequence
```

```
atgaagactatcattgctttgagctacattctatgtctggttttcgctcaaaaaattcctggaaatgacaatagcacgg
caacgctgtgccttgggcaccatgcagtaccaaacggaacgatagtgaaaacaatcacaaatgaccgaattgaagttac
taatgctactgagctggttcagaattcctcaataggtgaaatatgcgacagtcctcatcagatccttgatggagaaaac
tgcacactaatagatgctctattgggagaccctcagtgtgatggctttcaaaataagaaatgggacctttttgttgaac
gaagcaaagcctacagcaactgctacccttatgatgtgccggattatgcctcccttaggtcactagttgcctcatccgg
cacactggagtttaacaatgaaagcttcaattggactggagtcactcaaaacggaacaagttctgcttgcataaggaga
tctagtagtagtttctttagtagattaaattggttgacccacttaaactacacatacccagcattgaacgtgactatgc
caaacaatgaacaatttgacaaattgtacatttgggggttcaccacccgggtacggacaaggaccaaatcttcctgta
tgctcaatcatcaggaagaatcacagtatctaccaaaagaagccaacaagctgtaatcccaaatatcggatctagaccc
agaataagggatatccctagcagaataagcatctattggacaatagtaaaaccgggagacatactttgattaacagca
cagggaatctaattgctcctaggggttacttcaaaatacgaagtgggaaaagctcaataatgagatcagatgcacccat
tggcaaatgcaagtctgaatgcatcactccaaatggaagcattcccaatgacaaaccattccaaaatgtaaacaggatc
acatacggggcctgtcccagatatgttaagcatagcactctaaaattggcaacaggaatgcgaaatgtaccagagaaac
aaactagaggcatatttggcgcaatagcgggtttcatagaaaatggttgggagggaatggtggatggttggtacggttt
caggcatcaaaattctgagggaagaggacaagcagcagatctcaaaagcactcaagcagcaatcgatcaaatcaatggg
aagctgaatcgattgatcgggaaaaccaacgagaaattccatcagattgaaaaagaattctcagaagtagaaggaagaa
ttcaggaccttgagaaatatgttgaggacactaaaatagatctctggtcatacaacgcggagcttcttgttgccctgga
gaaccaacatacarttgatctaactgactcagaaatgaacaaactgtttgaaaaaacaaagaagcaactgagggaaaat
gctgaggatatgggaaatggttgtttcaaaatataccacaaatgtgacaatgcctgcataggatcaataagaaatggaa
cttatgaccacaatgtgtacagggatgaagcattaaacaaccggttccagatcaagggagttgagctgaagtcagggta
caaagattggatcctatggatttcctytgccatatcatgtttttgctttgtgttgctttgttggggttcatcatgtgg
gcctgccaaaagggcaacattaggtgcaacatttgcatttga
```

In [7]:
```
features = record['GBSeq_feature-table'] # extract all the features
for feature in features: # loop over features and print feature key and feature
location
    print(feature['GBFeature_key'] + ": " + feature['GBFeature_location'])
```

```
source: 1..1701
gene: 1..1701
CDS: 1..1701
mat_peptide: 49..1035
mat_peptide: 1036..1698
```

Running search queries through Entrez

So far we have only downloaded specific records from Entrez. In addition to just downloading records, however, we can also run searches directly from python. Any query that you can do on the Entrez website (https://www.ncbi.nlm.nih.gov/ (https://www.ncbi.nlm.nih.gov/)) can also be executed directly from python. This allows you to find a large number of records all at once and process them in an automated fashion.

For example, below we will see how to automatically run and retrieve the results for the following search term: "influenza a virus texas h1n1 hemagglutinin complete cds". A direct link to the search results on the Entrez website is here: https://www.ncbi.nlm.nih.gov/nuccore/?term=influenza+a+virus+texas+h1n1+hemagglutinin+complete+cds (https://www.ncbi.nlm.nih.gov/nuccore/?term=influenza+a+virus+texas+h1n1+hemagglutinin+complete+cds)

(Note that in the following Python code, we limit the number of search hits returned to the first 10.)

In [8]:
```
# let's do a search for influenza H1N1 viruses from Texas
handle = Entrez.esearch(db="nucleotide",  # database to search
                        term="influenza a virus texas h1n1 hemagglutinin comple
te cds",  # search term
                        retmax=10  # number of results that are returned
                        )
record = Entrez.read(handle)
handle.close()

gi_list = record["IdList"] # list of genbank identifiers found
print(gi_list)
```

```
['1597259311', '1597259287', '1597258925', '1597258900', '1591440116', '1591440
105', '1591440078', '1590855198', '1590855179', '1590836043']
```

Note that even though NCBI is phasing out sequence GI numbers, (https://www.ncbi.nlm.nih.gov/news/03-02-2016-phase-out-of-GI-numbers/) for now the `esearch()` function still returns GI numbers (numerical sequence identifiers without version information).

We can download all the genbank records in the list of identifiers using the `Entrez.efetch()` function. This function provides us with a handle that needs to be processed with `SeqIO.parse()`. (We used `SeqIO.read()` previously, which reads a single record. `SeqIO.parse()` reads multiple records. See here (http://biopython.org/wiki/SeqIO) for details.)

In [9]:
```
handle = Entrez.efetch(db="nucleotide", id=gi_list, rettype="gb", retmode="text
")
records = SeqIO.parse(handle, "genbank")

for record in records:
    print(record.description)

handle.close() # important, close the handle only after you have iterated over
the records. Otherwise you will get an error!
```

```
Influenza A virus (A/Texas/12/2019(H1N1)) segment 4 hemagglutinin (HA) gene, co
mplete cds
Influenza A virus (A/Texas/07/2019(H1N1)) segment 4 hemagglutinin (HA) gene, co
mplete cds
Influenza A virus (A/Texas/06/2019(H1N1)) segment 4 hemagglutinin (HA) gene, co
mplete cds
Influenza A virus (A/Texas/05/2019(H1N1)) segment 4 hemagglutinin (HA) gene, co
mplete cds
Influenza A virus (A/Texas/157/2018(H1N1)) segment 4 hemagglutinin (HA) gene, c
omplete cds
Influenza A virus (A/Texas/143/2018(H1N1)) segment 4 hemagglutinin (HA) gene, c
omplete cds
Influenza A virus (A/Texas/02/2019(H1N1)) segment 4 hemagglutinin (HA) gene, co
mplete cds
Influenza A virus (A/Texas/148/2018(H1N1)) segment 4 hemagglutinin (HA) gene, c
omplete cds
Influenza A virus (A/Texas/147/2018(H1N1)) segment 4 hemagglutinin (HA) gene, c
omplete cds
Influenza A virus (A/Texas/156/2018(H1N1)) segment 4 hemagglutinin (HA) gene, c
omplete cds
```

As another example, let's search the "pubmed" database (database of scientific publications) for papers from 2015 written by "Wilke CO". The exact search term we need to use is the following: "Wilke CO[Author] AND 2015[Date - Publication]"

You can click here (http://www.ncbi.nlm.nih.gov/pubmed/?term=Wilke+CO%5BAuthor%5D+AND+2015%5BDate+-+Publication%5D) to see the result from that search online.

In [10]:
```
handle = Entrez.esearch(db="pubmed",  # database to search
                        term="Wilke CO[Author] AND 2015[Date - Publication]",
# search term
                        retmax=10  # number of results that are returned
                        )
record = Entrez.read(handle)
handle.close()

# search returns PubMed IDs (pmids)
pmid_list = record["IdList"]
print(pmid_list)
```

```
['26770819', '26468068', '26430238', '26397960', '26355089', '26275208', '26020
774', '25999509', '25787027', '25737813']
```

Just like with genes and genomes, we can download the records corresponding to these identifiers. They are references. We'll print the author(s), title, and reference (source).

In [11]:
```
from Bio import Medline
handle = Entrez.efetch(db="pubmed", id=pmid_list, rettype="medline", retmode="t
ext")
records = Medline.parse(handle)
for record in records:
    print(record['AU']) # author list
    print(record['TI']) # title
    print(record['SO']) # source (reference)
    print()
handle.close()
```

```
['Meyer AG', 'Spielman SJ', 'Bedford T', 'Wilke CO']
Time dependence of evolutionary metrics during the 2009 pandemic influenza viru
s outbreak.
Virus Evol. 2015 Jan;1(1). doi: 10.1093/ve/vev006. Epub 2015 Jan 1.

['Meyer AG', 'Wilke CO']
The utility of protein structure as a predictor of site-wise dN/dS varies widel
y among HIV-1 proteins.
J R Soc Interface. 2015 Oct 6;12(111):20150579. doi: 10.1098/rsif.2015.0579.

['Wilke CO']
Evolutionary paths of least resistance.
Proc Natl Acad Sci U S A. 2015 Oct 13;112(41):12553-4. doi: 10.1073/pnas.151739
0112. Epub 2015 Oct 1.

['Spielman SJ', 'Wilke CO']
Pyvolve: A Flexible Python Module for Simulating Sequences along Phylogenies.
PLoS One. 2015 Sep 23;10(9):e0139047. doi: 10.1371/journal.pone.0139047. eColle
ction 2015.

['Kerr SA', 'Jackson EL', 'Lungu OI', 'Meyer AG', 'Demogines A', 'Ellington AD'
, 'Georgiou G', 'Wilke CO', 'Sawyer SL']
Computational and Functional Analysis of the Virus-Receptor Interface Reveals H
ost Range Trade-Offs in New World Arenaviruses.
J Virol. 2015 Nov;89(22):11643-53. doi: 10.1128/JVI.01408-15. Epub 2015 Sep 9.

['Houser JR', 'Barnhart C', 'Boutz DR', 'Carroll SM', 'Dasgupta A', 'Michener J
K', 'Needham BD', 'Papoulas O', 'Sridhara V', 'Sydykova DK', 'Marx CJ', 'Trent
MS', 'Barrick JE', 'Marcotte EM', 'Wilke CO']
Controlled Measurement and Comparative Analysis of Cellular Components in E. co
li Reveals Broad Regulatory Changes in Response to Glucose Starvation.
PLoS Comput Biol. 2015 Aug 14;11(8):e1004400. doi: 10.1371/journal.pcbi.1004400
. eCollection 2015 Aug.

['Meyer AG', 'Wilke CO']
Geometric Constraints Dominate the Antigenic Evolution of Influenza H3N2 Hemagg
lutinin.
PLoS Pathog. 2015 May 28;11(5):e1004940. doi: 10.1371/journal.ppat.1004940. eCo
llection 2015 May.

['Kachroo AH', 'Laurent JM', 'Yellman CM', 'Meyer AG', 'Wilke CO', 'Marcotte EM
']
Evolution. Systematic humanization of yeast genes reveals conserved functions a
nd genetic modularity.
Science. 2015 May 22;348(6237):921-5. doi: 10.1126/science.aaa0769.

['Echave J', 'Jackson EL', 'Wilke CO']
Relationship between protein thermodynamic constraints and variation of evoluti
onary rates among sites.
Phys Biol. 2015 Mar 19;12(2):025002. doi: 10.1088/1478-3975/12/2/025002.

['Spielman SJ', 'Kumar K', 'Wilke CO']
Comprehensive, structurally-informed alignment and phylogeny of vertebrate biog
enic amine receptors.
PeerJ. 2015 Feb 17;3:e773. doi: 10.7717/peerj.773. eCollection 2015.
```

Problems

Problem 1

Use the following code to download the genbank record KT220438 in XML and parse it with the `Entrez.parse()` function:

In [12]:
```
# Download sequence record for genbank id KT220438 (HA from influenza A)
handle = Entrez.efetch(db="nucleotide", id="KT220438", rettype="gb", retmode="x
ml")
parsed = Entrez.parse(handle)
record = list(parsed)[0] # Convert the parsed contents into a list and take ele
ment 0.
handle.close()
```

Then:

(a) Print out the value for the key `GBSeq_definition`.

(b) Find the CDS feature and print out all its qualifiers. Note that qualifiers are provided under the keyword `GBFeature_quals`.

In [13]:
```
# Problem 1a

print(record['GBSeq_definition'])
```

```
Influenza A virus (A/NewJersey/NHRC_93219/2015(H3N2)) segment 4 hemagglutinin (
HA) gene, complete cds
```

In [14]:
```
# Problem 1b

features = record['GBSeq_feature-table'] # extract all the features
for feature in features: # loop over features and find CDS feature
    if feature['GBFeature_key']=='CDS':
        CDS_feature = feature
        break

qualifiers = CDS_feature['GBFeature_quals']
for q in qualifiers:
    print(q['GBQualifier_name'] + ": " + q['GBQualifier_value'])
```

```
gene: HA
function: receptor binding and fusion protein
codon_start: 1
transl_table: 1
product: hemagglutinin
protein_id: AKQ43545.1
translation: MKTIIALSYILCLVFAQKIPGNDNSTATLCLGHHAVPNGTIVKTITNDRIEVTNATELVQNSSIGE
ICDSPHQILDGENCTLIDALLGDPQCDGFQNKKWDLFVERSKAYSNCYPYDVPDYASLRSLVASSGTLEFNNESFNWTG
VTQNGTSSACIRRSSSSFFSRLNWLTHLNYTYPALNVTMPNNEQFDKLYIWGVHHPGTDKDQIFLYAQSSGRITVSTKR
SQQAVIPNIGSRPRIRDIPSRISIYWTIVKPGDILLINSTGNLIAPRGYFKIRSGKSSIMRSDAPIGKCKSECITPNGS
IPNDKPFQNVNRITYGACPRYVKHSTLKLATGMRNVPEKQTRGIFGAIAGFIENGWEGMVDGWYGFRHQNSEGRGQAAD
LKSTQAAIDQINGKLNRLIGKTNEKFHQIEKEFSEVEGRIQDLEKYVEDTKIDLWSYNAELLVALENQHTXDLTDSEMN
KLFEKTKKQLRENAEDMGNGCFKIYHKCDNACIGSIRNGTYDHNVYRDEALNNRFQIKGVELKSGYKDWILWISXAISC
FLLCVALLGFIMWACQKGNIRCNICI
```

Problem 2:

(a) Use an Entrez esearch query of the pubmed database to find out how many publications "Spielman SJ" wrote in 2015.

(b) From the results of part (a), compile a combined list of all the co-authors of "Spielman SJ" in 2015.

In [15]:
```
# Problem 2a

handle = Entrez.esearch(db="pubmed",  # database to search
                        term="Spielman SJ[Author] AND 2015[Date - Publication]"
,  # search term
                        retmax=10  # number of results that are returned
                        )
record = Entrez.read(handle)
handle.close()

# search returns PubMed IDs (pmids)
pmid_list = record["IdList"]
print("Publications found:", pmid_list)
print("Number of publications:", len(pmid_list))
```

```
Publications found: ['26770819', '26397960', '25737813', '25576365']
Number of publications: 4
```

In [16]:
```
# Problem 2b

from Bio import Medline
handle = Entrez.efetch(db="pubmed", id=pmid_list, rettype="medline", retmode="t
ext")
records = Medline.parse(handle)
coauthors = [] # start with empty list of coauthors
for record in records:
    au_list = record['AU']
    for author in au_list:
        if author != "Spielman SJ" and author not in coauthors:
            coauthors.append(author)
handle.close()
print('Co-authors of "Spielman SJ" in 2015:')
for author in coauthors:
    print(" ", author)
```

```
Co-authors of "Spielman SJ" in 2015:
  Meyer AG
  Bedford T
  Wilke CO
  Kumar K
```

If this was easy

Problem 3:

For larger searches, NCBI wants you to use the WebEnv method to download all your search results. This is explained in the Biopython tutorial here. (http://biopython.org/DIST/docs/tutorial/Tutorial.html#sec:entrez-webenv) Rewrite the influenza search from the section "Running search queries on through Entrez" in such a way that it uses the WebEnv method. For this downloading method, it makes sense to write all the results into a file and then read the results back in.

In [17]:
```
handle = Entrez.esearch(db="nucleotide",  # database to search
                        term="influenza a virus texas h1n1 hemagglutinin comple
te cds",  # search term
                        usehistory="y" # this switches on the WebEnv method
                        )
results = Entrez.read(handle)
handle.close()

# Because we ran the search with usehistory="y", the results now contain two ad
ditional
# pieces of information, the WebEnv session cookie and the QueryKey:
webenv = results["WebEnv"]
query_key = results["QueryKey"]

# We also get the number of search results:
count = int(results["Count"])

# We now download the results and store them in a local file
# Downloading happens in batches, let's download 20 results at a time:
batch_size = 20
out_handle = open("influenza_HA.gb", "w")
for start in range(0, count, batch_size):
    end = min(count, start+batch_size)
    print("Downloading records %i through %i" % (start+1, end))
    fetch_handle = Entrez.efetch(db="nucleotide", rettype="gb", retmode="text",
                                 retstart=start, retmax=batch_size,
                                 webenv=webenv, query_key=query_key)
    data = fetch_handle.read()
    fetch_handle.close()
    out_handle.write(data)
out_handle.close()
```

```
Downloading records 1 through 20
Downloading records 21 through 40
Downloading records 41 through 60
Downloading records 61 through 80
Downloading records 81 through 100
Downloading records 101 through 120
Downloading records 121 through 140
Downloading records 141 through 160
Downloading records 161 through 180
Downloading records 181 through 200
Downloading records 201 through 220
Downloading records 221 through 240
Downloading records 241 through 260
Downloading records 261 through 280
Downloading records 281 through 300
Downloading records 301 through 320
Downloading records 321 through 340
Downloading records 341 through 360
Downloading records 361 through 380
Downloading records 381 through 400
Downloading records 401 through 420
Downloading records 421 through 440
Downloading records 441 through 460
Downloading records 461 through 480
Downloading records 481 through 500
Downloading records 501 through 520
Downloading records 521 through 540
Downloading records 541 through 560
Downloading records 561 through 580
Downloading records 581 through 600
Downloading records 601 through 620
Downloading records 621 through 640
Downloading records 641 through 660
Downloading records 661 through 680
Downloading records 681 through 700
Downloading records 701 through 720
Downloading records 721 through 740
Downloading records 741 through 744
```

In [18]:
```
# We can now open the file we created and parse all the records we have previou
sly stored
in_handle = open("influenza_HA.gb", "r")
records = SeqIO.parse(in_handle, "genbank")

for record in records:
    print(record.description)

in_handle.close()
```

```
Influenza A virus (A/Texas/12/2019(H1N1)) segment 4 hemagglutinin (HA) gene, co
mplete cds
Influenza A virus (A/Texas/07/2019(H1N1)) segment 4 hemagglutinin (HA) gene, co
mplete cds
Influenza A virus (A/Texas/06/2019(H1N1)) segment 4 hemagglutinin (HA) gene, co
mplete cds
Influenza A virus (A/Texas/05/2019(H1N1)) segment 4 hemagglutinin (HA) gene, co
mplete cds
Influenza A virus (A/Texas/157/2018(H1N1)) segment 4 hemagglutinin (HA) gene, c
omplete cds
Influenza A virus (A/Texas/143/2018(H1N1)) segment 4 hemagglutinin (HA) gene, c
omplete cds
Influenza A virus (A/Texas/02/2019(H1N1)) segment 4 hemagglutinin (HA) gene, co
mplete cds
Influenza A virus (A/Texas/148/2018(H1N1)) segment 4 hemagglutinin (HA) gene, c
omplete cds
Influenza A virus (A/Texas/147/2018(H1N1)) segment 4 hemagglutinin (HA) gene, c
omplete cds
Influenza A virus (A/Texas/156/2018(H1N1)) segment 4 hemagglutinin (HA) gene, c
omplete cds
Influenza A virus (A/Texas/154/2018(H1N1)) segment 4 hemagglutinin (HA) gene, c
omplete cds
Influenza A virus (A/swine/Texas/A01785919/2019(H1N1)) segment 4 hemagglutinin
(HA) gene, complete cds
Influenza A virus (A/Texas/7939/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7923/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7921/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7920/2018(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7918/2018(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7917/2018(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7916/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7914/2018(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7913/2018(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7911/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7910/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7908/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7906/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7904/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7902/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7901/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7898/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7895/2018(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7949/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
complete cds
Influenza A virus (A/Texas/7947/2019(H1N1)) segment 4 hemagglutinin (HA) gene,
```