In-class worksheet 19

Apr 2, 2019

Introduction to Biopython

The Biopython package, available at biopython.org, (http://biopython.org) consists of a large set of helpful functions and tools to solve frequently encountered problems in computational biology. In particular, it has excellent functionality to download and analyze sequence data. It also has a useful module for carrying out analysis of protein structure.

Here, we will start by doing some basic sequence analysis. We will use the biopython modules `Entrez` and `SeqIO`:

In [1]:
```
from Bio import Entrez, SeqIO
```

`Entrez` provides a computational interface to the widely-use entrez online database provided by the National Institutes of Health: http://www.ncbi.nlm.nih.gov/pubmed (http://www.ncbi.nlm.nih.gov/pubmed). In this database you can find almost any information of interest in biological research.

As an example, let's look at the gene of a recent influenza virus. Specifically, we look at hemagglutinin (HA) from a 2009 swine-flu strain. It is listed by ID number `FJ966082.1`, and it can be found online here: http://www.ncbi.nlm.nih.gov/nuccore/FJ966082.1 (http://www.ncbi.nlm.nih.gov/nuccore/FJ966082.1)

We can download this record directly from python using the following Biopython code:

In [2]:
```
Entrez.email = "wilke@austin.utexas.edu" # put your email here

# Download sequence record for genbank id KT220438
# This is HA gene of Influenza A virus, strain A/NewJersey/NHRC_93219/2015(H3N2
)
handle = Entrez.efetch(db="nucleotide", id="KT220438", rettype="gb", retmode="t
ext")
record = SeqIO.read(handle, "genbank")
handle.close()

# The sequence record is now stored in the variable `record` and
# we can print it to see what it contains
print(record)
```

```
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')]))])
/accessions=['KT220438']
/source=Influenza A virus (A/New Jersey/NHRC_93219/2015(H3N2))
/data_file_division=VRL
/topology=linear
/date=20-JUL-2015
/keywords=['']
/organism=Influenza A virus (A/New Jersey/NHRC_93219/2015(H3N2))
/molecule_type=cRNA
/sequence_version=1
/references=[Reference(title='GEISS Influenza Surveillance Response Program', .
..), Reference(title='Direct Submission', ...)]
/taxonomy=['Viruses', 'ssRNA viruses', 'ssRNA negative-strand viruses', 'Orthom
yxoviridae', 'Influenzavirus A']
Seq('ATGAAGACTATCATTGCTTTGAGCTACATTCTATGTCTGGTTTTCGCTCAAAAA...TGA', IUPACAmbigu
ousDNA())
```

Since the record is a simple python object, we can access elements of it as usual:

In [3]:
```
print("- ID of the record:")
print(record.id)

print("\n- Brief description of the record:")
print(record.description)

print("\n- Annotations that come with the record (given as a python dictionary)
:")
print(record.annotations)

print("\n- The sequence in this record:")
print(record.seq)
```

```
- ID of the record:
KT220438.1

- Brief description of the record:
Influenza A virus (A/NewJersey/NHRC_93219/2015(H3N2)) segment 4 hemagglutinin (
HA) gene, complete cds

- Annotations that come with the record (given as a python dictionary):
{'structured_comment': OrderedDict([('Assembly-Data', OrderedDict([('Sequencing
Technology', 'Sanger dideoxy sequencing')]))]), 'accessions': ['KT220438'], 'so
urce': 'Influenza A virus (A/New Jersey/NHRC_93219/2015(H3N2))', 'data_file_div
ision': 'VRL', 'topology': 'linear', 'date': '20-JUL-2015', 'keywords': [''], '
organism': 'Influenza A virus (A/New Jersey/NHRC_93219/2015(H3N2))', 'molecule_
type': 'cRNA', 'sequence_version': 1, 'references': [Reference(title='GEISS Inf
luenza Surveillance Response Program', ...), Reference(title='Direct Submission
', ...)], 'taxonomy': ['Viruses', 'ssRNA viruses', 'ssRNA negative-strand virus
es', 'Orthomyxoviridae', 'Influenzavirus A']}

- The sequence in this record:
ATGAAGACTATCATTGCTTTGAGCTACATTCTATGTCTGGTTTTCGCTCAAAAAATTCCTGGAAATGACAATAGCACGG
CAACGCTGTGCCTTGGGCACCATGCAGTACCAAACGGAACGATAGTGAAAACAATCACAAATGACCGAATTGAAGTTAC
TAATGCTACTGAGCTGGTTCAGAATTCCTCAATAGGTGAAATATGCGACAGTCCTCATCAGATCCTTGATGGAGAAAAC
TGCACACTAATAGATGCTCTATTGGGAGACCCTCAGTGTGATGGCTTTCAAAATAAGAAATGGGACCTTTTTGTTGAAC
GAAGCAAAGCCTACAGCAACTGCTACCCTTATGATGTGCCGGATTATGCCTCCCTTAGGTCACTAGTTGCCTCATCCGG
CACACTGGAGTTTAACAATGAAAGCTTCAATTGGACTGGAGTCACTCAAAACGGAACAAGTTCTGCTTGCATAAGGAGA
TCTAGTAGTAGTTTCTTTAGTAGATTAAATTGGTTGACCCACTTAAACTACACATACCCAGCATTGAACGTGACTATGC
CAAACAATGAACAATTTGACAAATTGTACATTTGGGGGGTTCACCACCCGGGTACGGACAAGGACCAAATCTTCCTGTA
TGCTCAATCATCAGGAAGAATCACAGTATCTACCAAAAGAAGCCAACAAGCTGTAATCCCAAATATCGGATCTAGACCC
AGAATAAGGGATATCCCTAGCAGAATAAGCATCTATTGGACAATAGTAAAACCGGGAGACATACTTTTGATTAACAGCA
CAGGGAATCTAATTGCTCCTAGGGGTTACTTCAAAATACGAAGTGGGAAAAGCTCAATAATGAGATCAGATGCACCCAT
TGGCAAATGCAAGTCTGAATGCATCACTCCAAATGGAAGCATTCCCAATGACAAACCATTCCAAAATGTAAACAGGATC
ACATACGGGGCCTGTCCCAGATATGTTAAGCATAGCACTCTAAAATTGGCAACAGGAATGCGAAATGTACCAGAGAAAC
AAACTAGAGGCATATTTGGCGCAATAGCGGGTTTCATAGAAAATGGTTGGGAGGGAATGGTGGATGGTTGGTACGGTTT
CAGGCATCAAAATTCTGAGGGAAGAGGACAAGCAGCAGATCTCAAAAGCACTCAAGCAGCAATCGATCAAATCAATGGG
AAGCTGAATCGATTGATCGGGAAAACCAACGAGAAATTCCATCAGATTGAAAAAGAATTCTCAGAAGTAGAAGGAAGAA
TTCAGGACCTTGAGAAATATGTTGAGGACACTAAAATAGATCTCTGGTCATACAACGCGGAGCTTCTTGTTGCCCTGGA
GAACCAACATACARTTGATCTAACTGACTCAGAAATGAACAAACTGTTTGAAAAAACAAAGAAGCAACTGAGGGAAAAT
GCTGAGGATATGGGAAATGGTTGTTTCAAAATATACCACAAATGTGACAATGCCTGCATAGGATCAATAAGAAATGGAA
CTTATGACCACAATGTGTACAGGGATGAAGCATTAAACAACCGGTTCCAGATCAAGGGAGTTGAGCTGAAGTCAGGGTA
CAAAGATTGGATCCTATGGATTTCCTYTGCCATATCATGTTTTTTGCTTTGTGTTGCTTTGTTGGGGTTCATCATGTGG
GCCTGCCAAAAGGGCAACATTAGGTGCAACATTTGCATTTGA
```

We can also do things like extract the DNA sequence and translate it into a protein sequence:

In [4]:
```
# extract the sequence from the record
DNA_seq = record.seq

# translate into a protein sequence
protein_seq = DNA_seq.translate()
print(protein_seq)
```

```
MKTIIALSYILCLVFAQKIPGNDNSTATLCLGHHAVPNGTIVKTITNDRIEVTNATELVQNSSIGEICDSPHQILDGEN
CTLIDALLGDPQCDGFQNKKWDLFVERSKAYSNCYPYDVPDYASLRSLVASSGTLEFNNESFNWTGVTQNGTSSACIRR
SSSSFFSRLNWLTHLNYTYPALNVTMPNNEQFDKLYIWGVHHPGTDKDQIFLYAQSSGRITVSTKRSQQAVIPNIGSRP
RIRDIPSRISIYWTIVKPGDILLINSTGNLIAPRGYFKIRSGKSSIMRSDAPIGKCKSECITPNGSIPNDKPFQNVNRI
TYGACPRYVKHSTLKLATGMRNVPEKQTRGIFGAIAGFIENGWEGMVDGWYGFRHQNSEGRGQAADLKSTQAAIDQING
KLNRLIGKTNEKFHQIEKEFSEVEGRIQDLEKYVEDTKIDLWSYNAELLVALENQHTXDLTDSEMNKLFEKTKKQLREN
AEDMGNGCFKIYHKCDNACIGSIRNGTYDHNVYRDEALNNRFQIKGVELKSGYKDWILWISXAISCFLLCVALLGFIMW
ACQKGNIRCNICI*
```

Problems

Problem 1:

(a) Download the sequence record with the ID number `FJ966082` and print it out. What kind of a sequence is this?

(b) Print out the comments section of the annotation

(c) Translate the DNA sequence into a protein sequence and print it out

In [5]:
```
# Problem 1a

# you will need to import the appropriate modules for this to work
from Bio import Entrez, SeqIO

# always give Entrez your email
Entrez.email = "wilke@austin.utexas.edu"

# download sequence record for genbank id FJ966082
# This is HA gene of Influenza A virus, strain A/California/04/2009(H1N1)
handle = Entrez.efetch(db="nucleotide", id="FJ966082", rettype="gb", retmode="t
ext")
record = SeqIO.read(handle, "genbank")
handle.close()

# print the record
print(record)
```

```
ID: FJ966082.1
Name: FJ966082
Description: Influenza A virus (A/California/04/2009(H1N1)) segment 4 hemagglut
inin (HA) gene, complete cds
Database cross-references: BioProject:PRJNA37813
Number of features: 6
/structured_comment=OrderedDict([('FluData', OrderedDict([('Isolate', 'A/Califo
rnia/04/2009'), ('Subtype', 'H1N1'), ('Host_gender', 'M'), ('Host_age', '10'),
('Passage_history', 'CX'), ('Adamantane_resistance', 'resistant'), ('Zanamivir_
resistance', 'sensitive'), ('Oseltamivir_resistance', 'sensitive'), ('Country',
'USA'), ('State/Province', 'California'), ('Collection_day', '1'), ('Collection
_month', '4'), ('Collection_year', '2009'), ('EPI_isolate', 'GISAID EPI_ISL_295
73'), ('EPI_accession', 'EPI176470')]))])
/accessions=['FJ966082']
/source=Influenza A virus (A/California/04/2009(H1N1))
/data_file_division=VRL
/topology=linear
/date=02-SEP-2010
/keywords=['']
/organism=Influenza A virus (A/California/04/2009(H1N1))
/comment=GenBank Accession Numbers FJ966079-FJ966086 represent sequences
from the 8 segments of Influenza A virus
(A/California/04/2009(H1N1)).
Swine influenza A (H1N1) virus isolated during human swine flu
outbreak of 2009.  For more information, see http://www.cdc.gov/.
Some of the information does not have GenBank feature identifiers
and is being provided in the comment section.
/molecule_type=cRNA
/sequence_version=1
/references=[Reference(title='Emergence of a novel swine-origin influenza A (H1
N1) virus in humans', ...), Reference(title='Antigenic and genetic characterist
ics of swine-origin 2009 A(H1N1) influenza viruses circulating in humans', ...)
, Reference(title='Direct Submission', ...)]
/taxonomy=['Viruses', 'ssRNA viruses', 'ssRNA negative-strand viruses', 'Orthom
yxoviridae', 'Influenzavirus A']
Seq('ATGAAGGCAATACTAGTAGTTCTGCTATATACATTTGCAACCGCAAATGCAGAC...TAA', IUPACAmbigu
ousDNA())
```

In [6]:
```
# Problem 1b

print(record.annotations['comment'])
```

```
GenBank Accession Numbers FJ966079-FJ966086 represent sequences
from the 8 segments of Influenza A virus
(A/California/04/2009(H1N1)).
Swine influenza A (H1N1) virus isolated during human swine flu
outbreak of 2009.  For more information, see http://www.cdc.gov/.
Some of the information does not have GenBank feature identifiers
and is being provided in the comment section.
```

In [7]:
```
# Problem 1c

protein_seq = record.seq.translate()
print("Protein sequence:")
print(protein_seq)
```

```
Protein sequence:
MKAILVVLLYTFATANADTLCIGYHANNSTDTVDTVLEKNVTVTHSVNLLEDKHNGKLCKLRGVAPLHLGKCNIAGWIL
GNPECESLSTASSWSYIVETPSSDNGTCYPGDFIDYEELREQLSSVSSFERFEIFPKTSSWPNHDSNKGVTAACPHAGA
KSFYKNLIWLVKKGNSYPKLSKSYINDKGKEVLVLWGIHHPSTSADQQSLYQNADTYVFVGSSRYSKKFKPEIAIRPKV
RDQEGRMNYYWTLVEPGDKITFEATGNLVVPRYAFAMERNAGSGIIISDTPVHDCNTTCQTPKGAINTSLPFQNIHPIT
IGKCPKYVKSTKLRLATGLRNIPSIQSRGLFGAIAGFIEGGWTGMVDGWYGYHHQNEQGSGYAADLKSTQNAIDEITNK
VNSVIEKMNTQFTAVGKEFNHLEKRIENLNKKVDDGFLDIWTYNAELLVLLENERTLDYHDSNVKNLYEKVRSQLKNNA
KEIGNGCFEFYHKCDNTCMESVKNGTYDYPKYSEEAKLNREEIDGVKLESTRIYQILAIYSTVASSLVLVVSLGAISFW
MCSNGSLQCRICI*
```

Problem 2:

Print the record you downloaded under Problem 1 in FASTA format. This means that you need to first print a line starting with ">" plus some description of the record. Then you need to print a line containing the sequence in the record.

In [8]:
```
print(">", record.id, record.description)
print(record.seq)
```

```
> FJ966082.1 Influenza A virus (A/California/04/2009(H1N1)) segment 4 hemagglut
inin (HA) gene, complete cds
ATGAAGGCAATACTAGTAGTTCTGCTATATACATTTGCAACCGCAAATGCAGACACATTATGTATAGGTTATCATGCGA
ACAATTCAACAGACACTGTAGACACAGTACTAGAAAAGAATGTAACAGTAACACACTCTGTTAACCTTCTAGAAGACAA
GCATAACGGGAAACTATGCAAACTAAGAGGGGTAGCCCCATTGCATTTGGGTAAATGTAACATTGCTGGCTGGATCCTG
GGAAATCCAGAGTGTGAATCACTCTCCACAGCAAGCTCATGGTCCTACATTGTGGAAACACCTAGTTCAGACAATGGAA
CGTGTTACCCAGGAGATTTCATCGATTATGAGGAGCTAAGAGAGCAATTGAGCTCAGTGTCATCATTTGAAAGGTTTGA
GATATTCCCCAAGACAAGTTCATGGCCCAATCATGACTCGAACAAAGGTGTAACGGCAGCATGTCCTCATGCTGGAGCA
AAAAGCTTCTACAAAAATTTAATATGGCTAGTTAAAAAAGGAAATTCATACCCAAAGCTCAGCAAATCCTACATTAATG
ATAAAGGGAAAGAAGTCCTCGTGCTATGGGGCATTCACCATCCATCTACTAGTGCTGACCAACAAAGTCTCTATCAGAA
TGCAGATACATATGTTTTTGTGGGGTCATCAAGATACAGCAAGAAGTTCAAGCCGGAAATAGCAATAAGACCCAAAGTG
AGGGATCAAGAAGGGAGAATGAACTATTACTGGACACTAGTAGAGCCGGGAGACAAAATAACATTCGAAGCAACTGGAA
ATCTAGTGGTACCGAGATATGCATTCGCAATGGAAAGAAATGCTGGATCTGGTATTATCATTTCAGATACACCAGTCCA
CGATTGCAATACAACTTGTCAAACACCCAAGGGTGCTATAAACACCAGCCTCCCATTTCAGAATATACATCCGATCACA
ATTGGAAAATGTCCAAAATATGTAAAAAGCACAAAATTGAGACTGGCCACAGGATTGAGGAATATCCCGTCTATTCAAT
CTAGAGGCCTATTTGGGGCCATTGCCGGTTTCATTGAAGGGGGGTGGACAGGGATGGTAGATGGATGGTACGGTTATCA
CCATCAAAATGAGCAGGGGTCAGGATATGCAGCCGACCTGAAGAGCACACAGAATGCCATTGACGAGATTACTAACAAA
GTAAATTCTGTTATTGAAAAGATGAATACACAGTTCACAGCAGTAGGTAAAGAGTTCAACCACCTGGAAAAAAGAATAG
AGAATTTAAATAAAAAAGTTGATGATGGTTTCCTGGACATTTGGACTTACAATGCCGAACTGTTGGTTCTATTGGAAAA
TGAAAGAACTTTGGACTACCACGATTCAAATGTGAAGAACTTATATGAAAAGGTAAGAAGCCAGCTAAAAAACAATGCC
AAGGAAATTGGAAACGGCTGCTTTGAATTTTACCACAAATGCGATAACACGTGCATGGAAAGTGTCAAAAATGGGACTT
ATGACTACCCAAAATACTCAGAGGAAGCAAAATTAAACAGAGAAGAAATAGATGGGGTAAAGCTGGAATCAACAAGGAT
TTACCAGATTTTGGCGATCTATTCAACTGTCGCCAGTTCATTGGTACTGGTAGTCTCCCTGGGGGCAATCAGTTTCTGG
ATGTGCTCTAATGGGTCTCTACAGTGTAGAATATGTATTTAA
```

If this was easy

Problem 3:

Write a function that takes a sequence record as input and prints it out in FASTA format. Write the function in such a way that it breaks the sequence over multiple lines, such that each line contains at most 60 characters.

In [9]:
```
def print_fasta(record):
    print(">", record.id, record.description)
    seq = record.seq
    for i in range(0, len(seq), 60):
        print(seq[i:i+60])

print_fasta(record)
```

```
> FJ966082.1 Influenza A virus (A/California/04/2009(H1N1)) segment 4 hemagglut
inin (HA) gene, complete cds
ATGAAGGCAATACTAGTAGTTCTGCTATATACATTTGCAACCGCAAATGCAGACACATTA
TGTATAGGTTATCATGCGAACAATTCAACAGACACTGTAGACACAGTACTAGAAAAGAAT
GTAACAGTAACACACTCTGTTAACCTTCTAGAAGACAAGCATAACGGGAAACTATGCAAA
CTAAGAGGGGTAGCCCCATTGCATTTGGGTAAATGTAACATTGCTGGCTGGATCCTGGGA
AATCCAGAGTGTGAATCACTCTCCACAGCAAGCTCATGGTCCTACATTGTGGAAACACCT
AGTTCAGACAATGGAACGTGTTACCCAGGAGATTTCATCGATTATGAGGAGCTAAGAGAG
CAATTGAGCTCAGTGTCATCATTTGAAAGGTTTGAGATATTCCCCAAGACAAGTTCATGG
CCCAATCATGACTCGAACAAAGGTGTAACGGCAGCATGTCCTCATGCTGGAGCAAAAAGC
TTCTACAAAAATTTAATATGGCTAGTTAAAAAAGGAAATTCATACCCAAAGCTCAGCAAA
TCCTACATTAATGATAAAGGGAAAGAAGTCCTCGTGCTATGGGGCATTCACCATCCATCT
ACTAGTGCTGACCAACAAAGTCTCTATCAGAATGCAGATACATATGTTTTTGTGGGGTCA
TCAAGATACAGCAAGAAGTTCAAGCCGGAAATAGCAATAAGACCCAAAGTGAGGGATCAA
GAAGGGAGAATGAACTATTACTGGACACTAGTAGAGCCGGGAGACAAAATAACATTCGAA
GCAACTGGAAATCTAGTGGTACCGAGATATGCATTCGCAATGGAAAGAAATGCTGGATCT
GGTATTATCATTTCAGATACACCAGTCCACGATTGCAATACAACTTGTCAAACACCCAAG
GGTGCTATAAACACCAGCCTCCCATTTCAGAATATACATCCGATCACAATTGGAAAATGT
CCAAAATATGTAAAAAGCACAAAATTGAGACTGGCCACAGGATTGAGGAATATCCCGTCT
ATTCAATCTAGAGGCCTATTTGGGGCCATTGCCGGTTTCATTGAAGGGGGGTGGACAGGG
ATGGTAGATGGATGGTACGGTTATCACCATCAAAATGAGCAGGGGTCAGGATATGCAGCC
GACCTGAAGAGCACACAGAATGCCATTGACGAGATTACTAACAAAGTAAATTCTGTTATT
GAAAAGATGAATACACAGTTCACAGCAGTAGGTAAAGAGTTCAACCACCTGGAAAAAAGA
ATAGAGAATTTAAATAAAAAAGTTGATGATGGTTTCCTGGACATTTGGACTTACAATGCC
GAACTGTTGGTTCTATTGGAAAATGAAAGAACTTTGGACTACCACGATTCAAATGTGAAG
AACTTATATGAAAAGGTAAGAAGCCAGCTAAAAAACAATGCCAAGGAAATTGGAAACGGC
TGCTTTGAATTTTACCACAAATGCGATAACACGTGCATGGAAAGTGTCAAAAATGGGACT
TATGACTACCCAAAATACTCAGAGGAAGCAAAATTAAACAGAGAAGAAATAGATGGGGTA
AAGCTGGAATCAACAAGGATTTACCAGATTTTGGCGATCTATTCAACTGTCGCCAGTTCA
TTGGTACTGGTAGTCTCCCTGGGGGCAATCAGTTTCTGGATGTGCTCTAATGGGTCTCTA
CAGTGTAGAATATGTATTTAA
```

Problem 4:

Write a function that takes a DNA sequence as input, translates the sequence in all three reading frames, and counts the number of stop codons found in each translation. Remember that stop codons are indicated by a "*".

Note that biopython doesn't like to translate sequences whose length is not a multiple of three, so you will have to pad the sequence with trailing Ns to avoid a warning or error.

Hint: The solution to this problem will be simpler if you first write a function that can count the stop codons in one sequence.

In [10]:
```
def count_stop(aa_seq):
    count = 0
    for c in aa_seq:
        if c == "*":
            count += 1
    return count

def translate_all_frames(seq):
    # frame 1
    count1 = count_stop(seq.translate())

    # frame 2
    seq2 = seq[1:]+"N"
    count2 = count_stop(seq2.translate())

    # frame 3
    seq3 = seq[2:]+"NN"
    count3 = count_stop(seq3.translate())

    # return all three counts
    return (count1, count2, count3)

counts = translate_all_frames(record.seq)
print(counts)
```

```
(1, 42, 33)
```