# Biopython I: Working with Sequence Files

Bioinformatics data is heavy on strings (sequences) and various types of tab delimited tables, as well as some key:value pairs such as GenBank records (field header: field contents). There are also some complex data structures such as multiple alignments, phylogenetic trees, etc. **BioPython** is a collection of Python modules that provide functions to deal with Bioinformatics data types and functions for useful computing operations (reverse complement a DNA string, find motifs in protein sequences, access web servers, etc.) as well as ‘wrappers’ that provide interfaces to run other software (both via webservers and installed on your local computer) and work with the output.

Biopython is not included in the standard modules that are installed with the Python program, it must be downloaded separately from <http://biopython.org/wiki/Download>

Biopython also uses supporting modules **scipy** and **numpy**. On some systems install of Biopython will also require setting up a C compiler. On Mac OSX systems, the XCode suite and command line utilities are required.

Most of the following examples come from the Biopython Tutorial

<http://biopython.org/DIST/docs/tutorial/Tutorial.html>

It will be handy to have a couple of example data files on hand from the tutorial. You can grab them with the linux **wget** command (note, this is not a Python command)

wget <https://raw.githubusercontent.com/biopython/biopython/master/Doc/examples/ls_orchid.fasta>

wget <http://biopython.org/SRC/biopython/Tests/GenBank/NC_005816.gb>

wget <http://biopython.org/SRC/biopython/Tests/GenBank/NC_005816.fna>

A proper Python way to download a file from a url uses the **urllib** module:

>>> import urllib2

>>> response = urllib2.urlopen('https://raw.githubusercontent.com/biopython/biopython/master/Doc/examples/ls\_orchid.fasta')

>>> html = response.read()

>>> out = open("ls\_orchid.fasta", "w")

>>> out.write(html)

#### Sequences

Biopython creates special data types (objects) which include useful methods. You can put a DNA or protein sequence into an ordinary string like this:

>>> my\_dna = 'GATCAACG'

>>> print(my\_dna)

GATCAACG

But the **Bio.Seq** module has a **Seq** data type that provides many useful functions such as reverse\_complement(), GC(), etc.

**Bio** is a huge module that contains many sub-modules, which in turn have their own functions and methods. Rather than importing all of **Bio**, which would define thousands of functions, it is more convenient to import just one module such as **Bio.Seq** using the **from *x* import *y*** syntax. There are a number of nice functions (methods) built into Seq objects, such as length: len(), and reverse\_complement().

>>> from Bio.Seq import Seq

>>> pr

>>> test

Seq('TTTCCGATAC', Alphabet())

>>> print(test)

TTTCCGATAC

>>> print len(test)

9

>>> print(test.reverse\_complement())

GTATCGGAAA

Another set of handy tools is available in the SeqUtils module, including GC content: GC()

>>> from Bio.SeqUtils import GC

>>> print GC(test)

The slightly more complex **SeqRecord** class creates sequence objects that contain attributes for identifiers and features from GenBank, EMBL and other database formats in addition to the sequence itself. <http://biopython.org/wiki/SeqRecord>

DNA and protein sequences are the most common data type in bioinformatics, and FASTA is the standard file format for these sequences. BioPython uses the **Bio.SeqIO** module to read and write FASTA files. **SeqIO** can read a multi-sequence FASTA file and access its headers and sequences and store them in a **SeqRecord** object. **SeqIO** can also read many other important file formats such as Illumina FASTQ, GenBank and EMBL records, and Clustal multiple alignment files (see <http://biopython.org/wiki/SeqIO> for details).

Here is a handy (non python) code to download Genbank sequences in Genbank and FASTA format using curl and the GenBank web API. We will get to the python version of this API shortly.

i= NM\_000518.4

curl -s "https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nucleotide&id=${i}&rettype=gb&retmode=txt">$i.gbk

curl -s "https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nucleotide&id=${i}&rettype=fasta\_cds\_aa&retmode=txt">$i.fasta

curl -s "https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nucleotide&id=${NM\_000518.4}&rettype=fasta\_cds\_aa&retmode=txt">$NM\_000518.4.fasta

For a single sequence stored in FASTA format in your current working directory,**SeqIO.read** will create a SeqRecord object. The syntax uses **open** on the filename and then specifies the file type, which is "fasta" in this case. Then you can print the **.seq** attribute of the SeqRecord that has been created.

>>> from Bio import SeqIO

>>> seqRec1 = SeqIO.read(open('NM\_000518.4.fasta'), 'fasta')

>>> print(seqRec1)

>>> print(seqRec1.seq)

>>> print len((seqRec1.seq))

If you apply this same method to a Genbank record, you will see information for the **.name**, **.description**, and **.dbxrefs** attributes.

>>> record = SeqIO.read("NM\_000518.4.gbk", "genbank")

>>> record.description

For a FASTA file containing multiple sequences, use **SeqIO.parse** with a **for** loop. Then you can access the identifier (.id) and the sequence string (.seq) for each sequence.

>>> for seq\_record in SeqIO.parse('ls\_orchid.fasta', 'fasta'):

print(seq\_record.id)

print(repr(seq\_record.seq))

Or you could use a **for** loop to write each sequence in the FASTA file into a list of SeqRecords where it is easy to access them by the list index number.

>>> my\_seqlist = []

>>> for seq\_record in SeqIO.parse('ls\_orchid.fasta', 'fasta'):

my\_seqlist.append(seq\_record)

>>> my\_seqlist[0]

Biopython also has modules to access Genbank and other databases and directly download sequences by web services. **Bio.Entrez.efetch** is the module to access Genbank at the NCBI. It works together with **SeqIO** and uses a very similar syntax.

>>> from Bio import Entrez

>>> Entrez.email='[your.name@nyumc.org](mailto:your.name@nyumc.org)'

>>> temp = Entrez.efetch(db="nucleotide",rettype="gb",id="NM\_000518")

>>> out = open("NM\_000518.fasta",'w')

>>> gbseq = SeqIO.read(temp, "genbank")

>>> SeqIO.write(gbseq,out,"fasta")

>>> temp.close()

>>> out.close()

>>> print(gbseq)

>>> print(gbseq.seq)

Homework Assignment

1. Using **Bio.Entrez.efetch** and **SeqIO,** download from GenBank, the mRNA sequences for the human genes HBA1(NM\_000558) and HBA2 (NM\_000517) . Print the sequence ID, name, and description of these sequence records.

**NM\_000558**

ORIGIN

1 actcttctgg tccccacaga ctcagagaga acccaccatg gtgctgtctc ctgccgacaa

61 gaccaacgtc aaggccgcct ggggtaaggt cggcgcgcac gctggcgagt atggtgcgga

121 ggccctggag aggatgttcc tgtccttccc caccaccaag acctacttcc cgcacttcga

181 cctgagccac ggctctgccc aggttaaggg ccacggcaag aaggtggccg acgcgctgac

241 caacgccgtg gcgcacgtgg acgacatgcc caacgcgctg tccgccctga gcgacctgca

301 cgcgcacaag cttcgggtgg acccggtcaa cttcaagctc ctaagccact gcctgctggt

361 gaccctggcc gcccacctcc ccgccgagtt cacccctgcg gtgcacgcct ccctggacaa

421 gttcctggct tctgtgagca ccgtgctgac ctccaaatac cgttaagctg gagcctcggt

481 ggccatgctt cttgcccctt gggcctcccc ccagcccctc ctccccttcc tgcacccgta

541 cccccgtggt ctttgaataa agtctgagtg ggcggca

Summary: The human alpha globin gene cluster located on chromosome

16 spans about 30 kb and includes seven loci:pseudozeta

- mu - pseudoalpha-1 - alpha-2 - alpha-1 - theta - 3'. The alpha-2

(HBA2) and alpha-1 (HBA1) coding sequences are identical. These

genes differ slightly over the 5' untranslated regions and the

introns, but they differ significantly over the 3' untranslated

regions. Two alpha chains plus two beta chains constitute HbA,

which in normal adult life comprises about 97% of the total

hemoglobin; alpha chains combine with delta chains to constitute

HbA-2, which with HbF (fetal hemoglobin) makes up the remaining 3%

of adult hemoglobin. Alpha thalassemias result from deletions of

each of the alpha genes as well as deletions of boHBA1;

some nondeletion alpha thalassemias have also been reported.

[provided by RefSeq, Jul 2008].

NM\_000517

Summary: The human alpha globin gene cluster located on chromosome

16 spans about 30 kb and includes seven loci: 5'- zeta - pseudozeta

- mu - pseudoalpha-1 - alpha-2 - alpha-1 - theta - 3'. The alpha-2

(HBA2) and alpha-1 (HBA1) coding sequences are identical. These

genes differ slightly over the 5' untranslated regions and the

introns, but they differ significantly over the 3' untranslated

regions. Two alpha chains plus two beta chains constitute HbA,

which in normal adult life comprises about 97% of the total

hemoglobin; alpha chains combine with delta chains to constitute

HbA-2, which with HbF (fetal hemoglobin) makes up the remaining 3%

of adult hemoglobin. Alpha thalassemias result from deletions of

each of the alpha genes as well as deletions of both HBA2 and HBA1;

some nondeletion alpha thalassemias have also been reported.

[provided by RefSeq, Jul 2008].

ORIGIN

1 actcttctgg tccccacaga ctcagagaga acccaccatg gtgctgtctc ctgccgacaa

61 gaccaacgtc aaggccgcct ggggtaaggt cggcgcgcac gctggcgagt atggtgcgga

121 ggccctggag aggatgttcc tgtccttccc caccaccaag acctacttcc cgcacttcga

181 cctgagccac ggctctgccc aggttaaggg ccacggcaag aaggtggccg acgcgctgac

241 caacgccgtg gcgcacgtgg acgacatgcc caacgcgctg tccgccctga gcgacctgca

301 cgcgcacaag cttcgggtgg acccggtcaa cttcaagctc ctaagccact gcctgctggt

361 gaccctggcc gcccacctcc ccgccgagtt cacccctgcg gtgcacgcct ccctggacaa

421 gttcctggct tctgtgagca ccgtgctgac ctccaaatac cgttaagctg gagcctcggt

481 agccgttcct cctgcccgct gggcctccca acgggccctc ctcccctcct tgcaccggcc

541 cttcctggtc tttgaataaa gtctgagtgg gcagca

1. Read the sequence records from a list of GenBank IDs in a text file (seq\_id.list) into a Python list, and import them using **Bio.Entrez.efetch** into a Python list of SeqRecords. Print the sequence name and the length of each of these sequences.

>>> from Bio import Entrez

>>> Entrez.email='Moosun.Kim@nyumc.org'

>>> temp = Entrez.efetch(db="nucleotide",rettype="gb",id="NM\_000518")

>>> out = open("NM\_000518.fasta",'w')

>>> gbseq = SeqIO.read(temp, "genbank")

>>> SeqIO.write(gbseq,out,"fasta")

1

>>> temp.close()

>>> out.close()

>>> print(gbseq)

ID: NM\_000518.5

Name: NM\_000518

Description: Homo sapiens hemoglobin subunit beta (HBB), mRNA

Number of features: 23

/molecule\_type=mRNA

/topology=linear

/data\_file\_division=PRI

/date=05-JUL-2018

/accessions=['NM\_000518']

/sequence\_version=5

/keywords=['RefSeq']

/source=Homo sapiens (human)

/organism=Homo sapiens

/taxonomy=['Eukaryota', 'Metazoa', 'Chordata', 'Craniata', 'Vertebrata', 'Euteleostomi', 'Mammalia', 'Eutheria', 'Euarchontoglires', 'Primates', 'Haplorrhini', 'Catarrhini', 'Hominidae', 'Homo']

/references=[Reference(title='Heterogeneity between Two alpha Subunits of alpha2beta2 Human Hemoglobin and O2 Binding Properties: Raman, (1)H Nuclear Magnetic Resonance, and Terahertz Spectra', ...), Reference(title='Direct observation of conformational population shifts in crystalline human hemoglobin', ...), Reference(title='A phased SNP-based classification of sickle cell anemia HBB haplotypes', ...), Reference(title='Genetic Basis and Genetic Modifiers of beta-Thalassemia and Sickle Cell Disease', ...), Reference(title='Upstream open reading frames cause widespread reduction of protein expression and are polymorphic among humans', ...), Reference(title='Beta-Thalassemia', ...), Reference(title='Sickle Cell Disease', ...), Reference(title='The structure of hemoglobin Indianapolis [beta112(G14) arginine]. An unstable variant detectable only by isotopic labeling', ...), Reference(title='Human beta-globin messenger RNA. III. Nucleotide sequences derived from complementary DNA', ...), Reference(title="Complete 3' noncoding region sequences of rabbit and human beta-globin messenger RNAs", ...)]

/comment=REVIEWED REFSEQ: This record has been curated by NCBI staff. The

reference sequence was derived from AK311825.1 and BU661647.1.

This sequence is a reference standard in the RefSeqGene project.

On Jun 13, 2018 this sequence version replaced NM\_000518.4.

Summary: The alpha (HBA) and beta (HBB) loci determine the

structure of the 2 types of polypeptide chains in adult hemoglobin,

Hb A. The normal adult hemoglobin tetramer consists of two alpha

chains and two beta chains. Mutant beta globin causes sickle cell

anemia. Absence of beta chain causes beta-zero-thalassemia. Reduced

amounts of detectable beta globin causes beta-plus-thalassemia. The

order of the genes in the beta-globin cluster is 5'-epsilon --

gamma-G -- gamma-A -- delta -- beta--3'. [provided by RefSeq, Jul

2008].

Publication Note: This RefSeq record includes a subset of the

publications that are available for this gene. Please see the Gene

record to access additional publications.

[ECO:0000332]

COMPLETENESS: full length.

/structured\_comment=OrderedDict([('Evidence-Data', OrderedDict([('Transcript exon combination', 'AU139015.1, SRR5189652.246572.1')]))])

Seq('ACATTTGCTTCTGACACAACTGTGTTCACTAGCAACCTCAAACAGACACCATGG...CAA', IUPACAmbiguousDNA())

>>> print(gbseq.seq)

ACATTTGCTTCTGACACAACTGTGTTCACTAGCAACCTCAAACAGACACCATGGTGCATCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAAGGTGAACGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAGGCTGCTGGTGGTCTACCCTTGGACCCAGAGGTTCTTTGAGTCCTTTGGGGATCTGTCCACTCCTGATGCTGTTATGGGCAACCCTAAGGTGAAGGCTCATGGCAAGAAAGTGCTCGGTGCCTTTAGTGATGGCCTGGCTCACCTGGACAACCTCAAGGGCACCTTTGCCACACTGAGTGAGCTGCACTGTGACAAGCTGCACGTGGATCCTGAGAACTTCAGGCTCCTGGGCAACGTGCTGGTCTGTGTGCTGGCCCATCACTTTGGCAAAGAATTCACCCCACCAGTGCAGGCTGCCTATCAGAAAGTGGTGGCTGGTGTGGCTAATGCCCTGGCCCACAAGTATCACTAAGCTCGCTTTCTTGCTGTCCAATTTCTATTAAAGGTTCCTTTGTTCCCTAAGTCCAACTACTAAACTGGGGGATATTATGAAGGGCCTTGAGCATCTGGATTCTGCCTAATAAAAAACATTTATTTTCATTGCAA

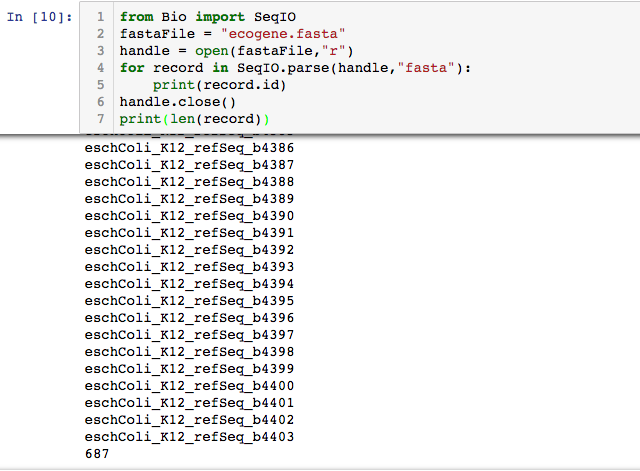
>>> len(gbseq)

628

1. Import a large set of sequences from a Fasta file: ecogene.fasta
   1. How many sequences are in this file?

4294

687



* 1. Using Biopython, read the sequence ID, name, and description for the first Sequence Record. What do you get? Compare to the metadata available for the GenBank records above.

from Bio import SeqIO

records = list(SeqIO.parse("ecogene.fasta", "fasta"))

print("Found %i records" % len(records))

print("The last record")

last\_record = records[-1] #using Python’s list tricks

print(last\_record.id)

print(repr(last\_record.seq))

print(len(last\_record))

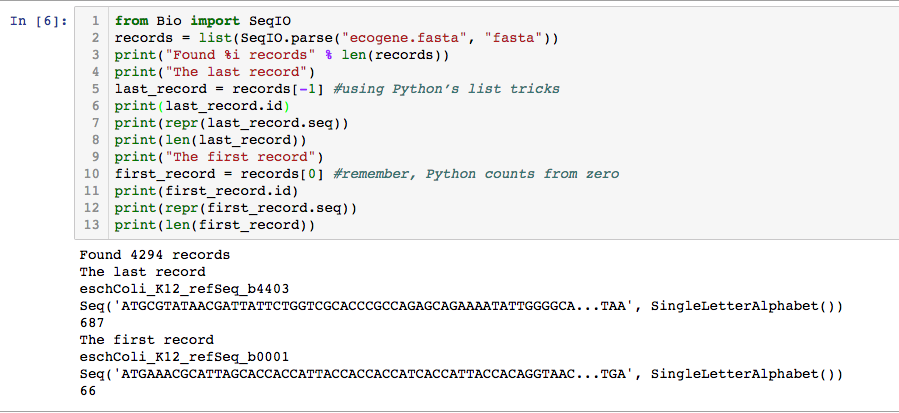
print("The first record")

first\_record = records[0] #remember, Python counts from zero

print(first\_record.id)

print(repr(first\_record.seq))

print(len(first\_record))



* 1. What is the total length of all of the sequences in the file (just the DNA, not headers)

4130063

* 1. Make a new FASTA file with just the sequences >= 300 bp in length
     1. Show the Python code that you used



* 1. Make a new FASTA file with just the sequences with %GC > 60
     1. Show the Python code that you used

from Bio import SeqIO

from Bio.SeqUtils import GC

fa = open('cbbtx.fa', 'r')

out = open('cb300.fa', 'w')

bases = 0

seqlist = list(SeqIO.parse(fa, "fasta"))

for temp in seqlist:

x = len(temp.seq)

bases += x

print bases

list300=[]

for temp in seqlist:

if len(temp) > = 300:

list300.append(temp)

SeqIO.write(list300, "cb300.fa", "fasta")

seq\_id.list

NM\_001301326.1

NM\_001122995.2

NM\_001044583.1

NM\_001044550.1

NM\_001122994.2

NM\_001244641.1

NM\_001297634.1

NM\_001297633.1

NM\_001297632.1

KJ729036.1

KF280282.1

KF280281.1

KF280280.1

NM\_001244668.1

NM\_001244667.1

NM\_001244640.1

NM\_001243579.1

NM\_214269.2

KJ686126.1

NM\_001130735.1