

Principles of Computer Science II

Introduction to BioPython

Ioannis Chatzigiannakis

Sapienza University of Rome

Lecture 4



BioPython Functionality and Tools

- ▶ Provides a range of tools to parse bioinformatics files into Python
 1. FASTA
 2. Pubmed and Medline
 3. ExPASy
 4. SCOP
 5. SwissProt
 6. PDB
- ▶ Tools for Sequence Alignment
- ▶ Tools for Data Clustering
- ▶ Interface to common bioinformatics programs
 1. Blast
 2. Clustalw



Installing BioPython

```
pip3 install numpy, Bio
```



Sequence Objects

```
from Bio.Seq import Seq  
  
my_seq = Seq("GATCGATGGGCCTATATAGGATCGAAAATCGC")
```

- ▶ Very similar functionality to strings.

```
my_seq == "ACGT"  
False
```

```
print(my_seq[4])  
G
```

- ▶ Like Python strings, you can do substrings

```
my_seq.seq[4:12]  
Seq('GCTGTAGTAAG')
```

- ▶ With additional tools

```
another_seq.count("G")  
9
```



Extracting Codon from Sequence Objects

- ▶ Like Python strings, you can do slices with a start, stop and stride.
- ▶ For example, we can get the first, second and third codon positions of this DNA sequence

```
my_seq[0::3]
Seq('GCTGTAGTAAG')
```

- ▶ Reverse the sequence

```
my_seq[::-1]
Seq('CGCTAAAAGCTAGGATATATCCGGGTAGCTAG')
```



Concatenating Sequence Objects

- ▶ Concatenating sequences

```
protein_seq = Seq("EVRNAK")
dna_seq = Seq("ACGT")
protein_seq + dna_seq
Seq('EVRNAKACGT')
```

- ▶ Working with lists of sequences

```
list_of_seqs = [Seq("ACGT"), Seq("AACC"), Seq("GGTT")]
concatenated = Seq("")
for s in list_of_seqs:
    concatenated += s
```

```
concatenated
Seq('ACGTAACCGTT')
```



Nucleotide sequences and (reverse) complements

- ▶ Nucleotide sequences

```
my_seq = Seq("AGTACACTGGT")
my_seq.complement()
Seq('TCATGTGACCA')
```

- ▶ Sequence Complement

```
my_seq.reverse_complement()
Seq('ACCAGTGTACT')
```



Mutable sequence object

- ▶ Sequence object is Immutable

```
my_seq[4] = 'G'
```

```
-----
TypeError Traceback (most recent call last)
<ipython-input-37-bf58366622a8> in <module>
----> 1 my_seq[4] = 'G'
TypeError: 'Seq' object does not support item assignment
```

- ▶ Instead we use a Mutable Sequence Object

```
from Bio.Seq import MutableSeq
mutable_seq = MutableSeq("GCCATTGTAATGGGCCGCTGAAAGGGTGCCCGA")
mutable_seq[4] = 'A'
```

- ▶ We need to convert a Mutable Sequence Object to a Sequence object to take advantage of the available tools.

```
new_seq = mutable_seq.toseq()
new_seq
Seq('GCCAATGTAATGGGCCGCTGAAAGGGTGCCCGA')
```



Sequence Records Objects

- ▶ Allows higher level features associated with a sequence:
 1. Identifier, Sequence name, description.
 2. Features,
 3. Annotations.
- ▶ Allows to import records from
 1. FASTA
 2. GenBank
 3. ...

```
from Bio.SeqRecord import SeqRecord

simple_seq = Seq("GATC")
simple_seq_r = SeqRecord(simple_seq)
simple_seq_r.id = "AC12345"
simple_seq_r.description = "Made up an example sequence"
simple_seq_r.annotations["evidence"] = "None."
```



Importing FASTA record

- ▶ Example using NC_005816.fna
https://raw.githubusercontent.com/biopython/biopython/master/Tests/GenBank/NC_005816.fna
- ▶ Contains 94 records
- ▶ The SeqIO package allows to parse sequences from files:

```
from Bio import SeqIO
record = SeqIO.read("NC_005816.fna", "fasta")

record.id
'gi|45478711|ref|NC_005816.1|'

record.description
'gi|45478711|ref|NC_005816.1| Yersinia pestis biovar Microtus str.'

record.seq
Seq('TGTAACGAACGGTGCAATAGTGATCCACACCCAACGCCTGAAATCAGATCCAGG...CTG')
```



Importing FASTA record with multiple sequences

- ▶ Example using Cypridiodioideae (a subfamily of orchids)
https://raw.githubusercontent.com/biopython/biopython/master/Doc/examples/ls_orchid.fasta
- ▶ Contains 94 records

```
from Bio import SeqIO
record = SeqIO.read("ls_orchid.fasta", "fasta")

-----
ValueError                                Traceback (most recent call last)
<ipython-input-8-b14fde270420> in <module>
----> 1 record = SeqIO.read("ls_orchid.fasta", "fasta")

~/local/lib/python3.8/site-packages/Bio/SeqIO/__init__.py
in read(handle, format, alphabet)
    660         next(iterator)
--> 661         raise ValueError("More than one record found in handle")
    662     except StopIteration:

ValueError: More than one record found in handle
```



Importing FASTA record with multiple sequences

- ▶ Example using Cypridiodioideae (a subfamily of orchids)
https://raw.githubusercontent.com/biopython/biopython/master/Doc/examples/ls_orchid.fasta
- ▶ Contains 94 records
- ▶ The SeqIO package allows to parse sequences from files:

```
for seq_record in SeqIO.parse("ls_orchid.fasta", "fasta"):
    print(seq_record.id)
    print(repr(seq_record.seq))
    print(len(seq_record))

gi|2765658|emb|Z78533.1|CIZ78533
Seq('CGTAACAAGGTTTCGGTAGGTGAACCTGCGGAAGGATCATTGATGAGACCGTGG...CGC')
740
gi|2765657|emb|Z78532.1|CCZ78532
Seq('CGTAACAAGGTTTCGGTAGGTGAACCTGCGGAAGGATCATTGTTGAGACAACAG...GGC')
753
...
```



Importing GenBank records

- ▶ Example using *Cypripedioideae* (a subfamily of orchids)
https://raw.githubusercontent.com/biopython/biopython/master/Doc/examples/ls_orchid.gb
- ▶ Contains 94 records
- ▶ The SeqIO package allows to parse sequences from files:

```
for seq_record in SeqIO.parse("ls_orchid.gb", "genbank"):
    print(seq_record.id)
    print(repr(seq_record.seq))
    print(len(seq_record))
```

```
Z78533.1
Seq('CGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTGATGAGACCGTGG...CGC')
740
Z78532.1
Seq('CGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTGTTGAGACAACAG...GGC')
753
...
```



Iterating using next()

```
record_iterator = SeqIO.parse("ls_orchid.gb", "genbank")
first_record = next(record_iterator)
print(first_record)
ID: Z78533.1
Name: Z78533
Description: C.irapeanum 5.8S rRNA gene and ITS1 and ITS2 DNA
Number of features: 5
/molecule_type=DNA
/topology=linear
/data_file_division=PLN
/date=30-NOV-2006
/accessions=['Z78533']
/sequence_version=1
/gi=2765658
/keywords=['5.8S ribosomal RNA', '5.8S rRNA gene', 'internal transcribe
/source=Cypripedium irapeanum
/organism=Cypripedium irapeanum
/taxonomy=['Eukaryota', 'Viridiplantae', 'Streptophyta', 'Embryophyta',
/reference=[Reference(title='Phylogenetics of the slipper orchids (Cyp
Seq('CGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTGATGAGACCGTGG...CGC')
```



Connecting with biological databases

- ▶ Connects to a biological database and retrieves records.
- ▶ Biopython can extract information from the following databases:
 1. Entrez (and PubMed) from the NCBI
 2. ExPASy
 3. SCOP



NCBI Entrez

- ▶ A large set of web-based tools.
 - ▶ Look up a specific record nucleotide, protein, mRNA, EST, PubMed, structure,...
 - ▶ Search for matches to a gene or disease name
 - ▶ Download sequence and other data associated with a nucleotide or protein
- ▶ Sometimes we need to automate the process
- ▶ Use Entrez to select and return the items of interest, rather than download, parse, and select.
- ▶ We need to be careful with Entrez resources.
- ▶ Supply our email address.
- ▶ Avoid multiple requests on the same minute/hour.
- ▶ ...our computer might be blocked.



Retrieving records from GenBank using their GI numbers

► In FASTA format:

```
from Bio import Entrez
Entrez.email = "ichatz@diag.uniroma1.it"
with Entrez.efetch(
    db="nucleotide", rettype="fasta", retmode="text", id="6273291"
) as handle:
    seq_record = SeqIO.read(handle, "fasta")

print("%s with %i features" % (seq_record.id, len(seq_record.features)))
# AF191665.1 with 0 features

with Entrez.efetch(
    db="nucleotide", rettype="gb", retmode="text", id="6273291"
) as handle:
    seq_record = SeqIO.read(handle, "gb")
```



Retrieving multiple records

```
with Entrez.efetch(
    db="nucleotide", rettype="gb", retmode="text", id="6273291,6273290,"
) as handle:
    for seq_record in SeqIO.parse(handle, "gb"):
        print("%s %s..." % (seq_record.id, seq_record.description[:50]))
        print(
            "Sequence length %i, %i features, from: %s"
            % (
                len(seq_record),
                len(seq_record.features),
                seq_record.annotations["source"],
            )
        )
```

```
AF191665.1 Opuntia marenae rpl16 gene; chloroplast gene for c...
Sequence length 902, 3 features, from: chloroplast Grusonia marenae
AF191664.1 Opuntia clavata rpl16 gene; chloroplast gene for c...
Sequence length 899, 3 features, from: chloroplast Grusonia clavata
AF191663.1 Opuntia bradtiana rpl16 gene; chloroplast gene for...
Sequence length 899, 3 features, from: chloroplast Grusonia bradtiana
```

