

INTRODUCTION TO PYTHON BIOPYTHON SUPPLEMENT CHEATSHEET

SEQUENCE FILE INPUT/OUTPUT

For file input/output, load the SeqIO or/and AlignIO modules from BioPython:

```
from Bio import SeqIO
from Bio import AlignIO
```

Function	What it does	Usage syntax
SeqIO.read()	Read a single sequence from a file	SeqIO.read("filename", "format")
SeqIO.parse()	Read multiple sequences from a file	list(SeqIO.parse("filename", "format"))
AlignIO.read()	Read a single alignment from a file	AlignIO.read("filename", "format")
AlignIO.parse()	Read multiple alignments from a file	list(AlignIO.parse("filename", "format"))
SeqIO.write()	Write sequences to a file	SeqIO.write(seq_record(s), "filename", "format")
SeqIO.convert()	Convert file of sequence(s) to a different format	SeqIO.convert("infile", "informat", "outfile", "outformat")
AlignIO.convert()	Convert file of alignment(s) to a different format	AlignIO.convert("infile", "informat", "outfile", "outformat")

BIOPYTHON SEQ OBJECTS

To use and/or manipulate Seq objects, load these modules:

```
from Bio.Seq import Seq
from Bio.Alphabet import * # usually not needed, except to access/change .alphabet
```

Defining a Seq object

```
my_seq = Seq("ACGT") # no alphabet, defaults to Alphabet()
my_seq = Seq("ACGT", alphabet = IUPACAmbiguousDNA)
```

Useful Seq object attributes

```
my_seq.seq # returns the sequence (you can re-cast to str)
my_seq.alphabet # returns the alphabet
```

Useful Seq object methods

```
my_seq.transcribe()
my_seq.back_transcribe()
my_seq.translate()
my_seq.complement()
my_seq.reverse_complement()
my_seq.tomutable() # allows you to change the Seq.seq attribute
```

BIOPYTHON SEQRECORD OBJECTS

To use and/or manipulate Seq objects, load these modules:

```
from Bio.Seq import Seq
from Bio.Alphabet import * # usually not needed, except to access/change .alphabet
from Bio.SeqRecord import SeqRecord
```

Defining a SeqRecord object

```
my_seq = Seq("ACGT") # SeqRecords are defined with Seq objects
my_seqrecord = SeqRecord( my_seq, id = "my_id")
```

Useful SeqRecord object attributes

```
my_seqrecord.seq
my_seqrecord.id
my_seqrecord.description # Useful for GenBank sequence records
my_seqrecord.annotations # Useful for GenBank sequence records
my_seqrecord.features    # Useful for GenBank sequence records
```

EXAMPLE: SAVING SEQUENCES TO A FILE

```
from Bio import SeqIO
from Bio.Seq import Seq
from Bio.SeqRecord import SeqRecord

# Sequences we want to save
sequences = {"id1": "ACGTACGT", "id2": "GCTAGCTA", "id3": "TACCGAT"}

# Empty list to store SeqRecord objects
records = []

# Populate records list
for seq_id in sequences:
    seq_rec = SeqRecord( Seq(sequences[seq_id]), id = seq_id )
    records.append( seq_rec )

# Save records to file
SeqIO.write(records, "outputfile.fasta", "fasta")
```