BIOPYTHON

```
from Bio.Seq import Seq              REVERSE
DNA = Seq("ATGCTGGGATATTGA")         INIT
mRNA = DNA.transcribe()              TRANSCRIBE
protein = mRNA.translate()           TRANSLATE
y = x [ : : -1 ]                     REVERSE
x.reverse_complement()               REVERSE COMPLEMENT
D1 = DNA[4:7]                        SLICE
D = D1 + D2                          ADD
y = x[::2]                           PASS
from Bio.SeqUtils import GC          GC CONTENT
GC(x)
```

OPEN FILE

```
from Bio import SeqIO
fastaFile =  "sequences.fasta"
sequences = SeqIO.parse(open(fastaFile),'fasta')            FASTA
for record in sequences:                                    READ
    print(record.id, record.seq)                            ID, Seq
from Bio import SeqIO
fastqFile = "reads.fastq"                                   FASTQ
data = SeqIO.parse(fastqFile,"fastq")
for record in data:
    print(record)       ID, Name, Description, Number of features, letter annotation, Seq
```

PHRED SCORE FASTQ

$$S_{PHRED} = -10 * \log_{10} (P_{Error})$$

$$P_{Error}=10^\wedge[-S_{PHRED} / 10]$$

```
fastq = sys.argv[1]                   read in the FASTQ file name
data = SeqIO.parse(fastq,"fastq")
sum_p = [0] * 2000  #initialize a list of probabilities
for record in data:
  for i,Q in enumerate(record.letter_annotations["phred_quality"]):
         p_err = 10**(-float(Q)/10.0)
         sum_p[i] += p_err                    PHRED to probability
```

DOWNLOAD DATA

bash

```
wget http://hendrixlab.cgrb.oregonstate.edu/teaching/sequences.fasta
```

Biopython

```
from Bio import Entrez
Entrez.email = "example@oregonstate.edu"
p_handle = Entrez.efetch(db="protein", id='4507341', rettype="gb", retmode="text")          BY ID
```

READ

```
rec = Entrez.read(Entrez.esearch(db="protein", term="NP_003173"))      ←
p_handle = Entrez.efetch(db="protein", id=rec["IdList"][Choose index],
         rettype="fasta")
print(p_handle.read( ) )
```

MOTIF PATTERN

```
from Bio.Seq import Seq
from Bio import SeqUtils
pattern = Seq("ACG")
sequence = Seq("ATGCCGAT")
results = SeqUtils.nt_search(str(sequence),pattern)                               SEARCH
        or
results_rc = SeqUtils.nt_search(str(sequence),pattern.reverse_complement())
print(results)    → ['ACG', 7, 31, 43]from Bio.Seq import Seq                      REVERSE
DNA = Seq("ATGCTGGGATATTGA")                                                      INIT
mRNA = DNA.transcribe()                                                           TRANSCRIBE
protein = mRNA.translate()                                                        TRANSLATE
```