

# Biyoinformatiğe Giriş Ödev 1

Mahire Zühal Özdemir

19360859015

**Adım 1:** Python versiyonunu kontrol etme:

```
PS C:\Users\zuhal\Desktop\bioinformaticProjects> python --version
Python 3.9.5
```

**Adım 3:** Biopython versiyonunu kontrol etme:

```
>>> import Bio
>>> print(Bio.__version__)
1.81
```

**Fasta dosyası :** Kaydettiğimiz fasta dosyası ve çıktısı

```
from Bio.SeqIO import parse
from Bio.Seq import Seq
from Bio.Data import IUPACData

file=open("example.fasta")
records=parse(file,"fasta")
for record in records:
    #print(record)
    print("ID: %s" % (record.id))
    print("Name: %s" % record.name)
    print("Description: %s" % (record.description))
    print("Annotations: %s" % (record.annotations))
    print("Sequence Data: %s" % (record.seq))
```

```
ID: sp|P25730|FMS1_ECOLI
Name: sp|P25730|FMS1_ECOLI
Description: sp|P25730|FMS1_ECOLI CS1 fimbrial subunit A precursor (CS1 pilin)
Annotations: {}
Sequence Data: MKLKKITIGAMALATLFATMGASAVEKTISVTASVDPTVLLQSDGSAFPNSVALTYPANVNFHAHTINTVHTNDSKGVVVKLSADPVLNVLNPTLQIPVSVNFAGKPLSTTGITIDSNLNFASSGVNKSSTQKLSIHADA
ID: sp|P15488|FMS3_ECOLI
Name: sp|P15488|FMS3_ECOLI
Description: sp|P15488|FMS3_ECOLI CS3 fimbrial subunit A precursor (CS3 pilin)
Annotations: {}
Sequence Data: MLKIKYLLIGLSLSSAMSSYSLAAGPTLTKELALNVLSPAALDATWAPODNLTLSNTGVSNTLVGVLTSNTSIDTVSIASNTVSDTSKNGTVTFHAHETNNSASFATTISTONANITLDKNAGNTIVKTTNGSQLPTNPLKFITT
```

```
ID: sp|P25730|FMS1_ECOLI
Name: sp|P25730|FMS1_ECOLI
Description: sp|P25730|FMS1_ECOLI CS1 fimbrial subunit A precursor (CS1 pilin)
Annotations: {}
Sequence Data: MKLKKITIGAMALATLFATMGASAVEKTISVTASVDPTVLLQSDGSAFPNSVALTYPANVNFHAHTINTVHTNDSKGVVVKLSADPVLNVLNPTLQIPVSVNFAGKPLSTTGITIDSNLNFASSGVNKSSTQKLSIHADATVGGALTAGQYQGLVSIILTKSTTTTTTKGT
ID: sp|P15488|FMS3_ECOLI
Name: sp|P15488|FMS3_ECOLI
Description: sp|P15488|FMS3_ECOLI CS3 fimbrial subunit A precursor (CS3 pilin)
Annotations: {}
Sequence Data: MLKIKYLLIGLSLSSAMSSYSLAAGPTLTKELALNVLSPAALDATWAPODNLTLSNTGVSNTLVGVLTSNTSIDTVSIASNTVSDTSKNGTVTFHAHETNNSASFATTISTONANITLDKNAGNTIVKTTNGSQLPTNPLKFITTEGNEHLVSNVYRANITITSTINGGGTKKGTDDKK
```

**!!!Alphabet modülü Bioypthon'ın 1.77'inci versiyonuyla birlikte kaldırılmıştır.O yüzden "record.seq.alphabet" kısmı çalışmadı.**

**Sequence modülü :** Sequence oluşturma

```
seq1=Seq("AGCT")  
print(seq1)
```

Output:

```
AGCT
```

Alphabet modülünde en sık karşılaştığım hata aşağıdaki gibiydi.Bazı fonksiyonlar ve parametrelerin kaldırılması ve versiyon uyumsuzluğu bu hataya neden oldu.

***ImportError: Bio.Alphabet has been removed from Biopython. In many cases, the alphabet can simply be ignored and removed from scripts.***

**IUPACData Sınıfı:**Aşağıdaki kod IUPACProtein alfabesinin olası harflerini verir.

```
#IUPACDATA  
print(IUPACData.protein_letters)
```

```
ACDEFGHIKLMNPQRSTVWY
```

**Temel İşlemler:** Sekanslar üzerinde temel işlemler. Buradaki birçok özellik python dilinin string işlemlerine benzer.

```
#Basic Operarions
print("Basic Operations")
seq_string=Seq("AGCTAGCT")
print("first value seq[0] => ", seq_string[0])
print("first two values seq[0:2] => ", seq_string[0:2])
print("all values seq[:] => ", seq_string[:])
print("length len(seq_string) => ", len(seq_string))
print("count of a value seq_string.count('A') => ", seq_string.count('A'))
```

```
Basic Operations
first value seq[0] =>  A
first two values seq[0:2] =>  AG
all values seq[:] =>  AGCTAGCT
length len(seq_string) =>  8
count of a value seq_string.count('A') =>  2
```

**İki Sekansı Toplama:** Sekanslar üzerinde temel işlemler.

```
#add two sequence
seq1=Seq("AGCT")
#seq2=Seq("TCGA", generic_dna) bu kullanım 1.81'de kaldırılmıştır
seq2=Seq("TCGA")
print("add two sequence seq1+seq2 => ", seq1+seq2)
```

```
add two sequence seq1+seq2 =>  AGCTTCGA
```

**Protein ve Dna'yı toplama:** İki farklı türe ait sekans toplama işlemi yeni versiyonda pdf'teki gibi gerçekleşmiyor. Biopython 1.81 versiyonunda sekans türleri kontrol edilmiyor. O yüzden toplama işlemini normal string toplar gibi yapıyor.

```
#add different type protein
seq3=Seq("AGCTACATTAGC")
seq4=Seq("AGCUACGUGAUA")
print("add two sequence seq1+seq2 => ", seq3+seq4)
```

```
add two sequence seq1+seq2 =>  AGCTACATTAGCAGCUACGUGAUA
```

**Birden çok sekansı liste içinde toplama:** Sekanslar önce listeye alınır sonra liste elemanları tek tek toplanır.

```
#adding many sequences
print("Add two or more sequences [Seq('AGCT'),Seq('CTAT')] => ")
list=[Seq("AGCT"),Seq("CTAT"),Seq("TGCA"),Seq("CAGT")]
for i in list:
    print(i)

final_seq=Seq("")
for i in list:
    final_seq+=i
print(final_seq)
```

```
AGCT
CTAT
TGCA
CAGT
AGCTCTATTGCACAGT
```

**Case İşlemleri:** Sekanslar içerisindeki nükleotitler için arama yaparken dikkat edilmesi gereken bir hususta case sensitive halidir. 'A', 'a'ya eşit değildir.

```
#change case of sequence
rna=Seq("agctagtat")
print("case sensitivity => ", 'A' in rna)
print("case sensitivity => ", 'a' in rna)
print("case sensitivity => ", 'GCT' in rna)
print("change case of sequence rna.upper() => ", rna.upper())
```

```
case sensitivity => False
case sensitivity => True
case sensitivity => False
change case of sequence rna.upper() => AGCTAGTAT
```

**Find İşlemleri:** Sekans içerisinde nükleotit arar. Find fonksiyonu bize indis numarası verir.

```
#to find letter in sequence
protein_seq=Seq("AGCUAGAGCGAU")
print("to find letter in sequence seq.find('A') => ", protein_seq.find('A')) #ilk olarak G nükleotidini bulduğu indis
print("to find letter in sequence seq.find('AG') => ", protein_seq.find('AG'))
print("to find letter in sequence seq.find('AU') => ", protein_seq.find('AU'))
```

```
to find letter in sequence seq.find('A') => 0
to find letter in sequence seq.find('AG') => 0
to find letter in sequence seq.find('AU') => 10
```

**Split İşlemleri:** Sekansı nükleotide göre ayırır.

```
#split operation
protein_seq1=Seq("AGUACACUGGU")
print("to perform splitting operation seq.split('A') => ", protein_seq1.split('A'))
```

```
to perform splitting operation seq.split('A') => [Seq(''), Seq('GU'), Seq('C'), Seq('CUGGU')]
```

**Strip İşlemleri:** Sekansın başındaki ve sonundaki boşlukları siler.

```
#strip operations in sequence
protein_seq2=Seq("    AGCT    ")
print("to perform strip operations in the sequence seq.strip() => ", protein_seq2.strip())
```

```
to perform strip operations in the sequence seq.strip() => AGCT
```