

## DNA

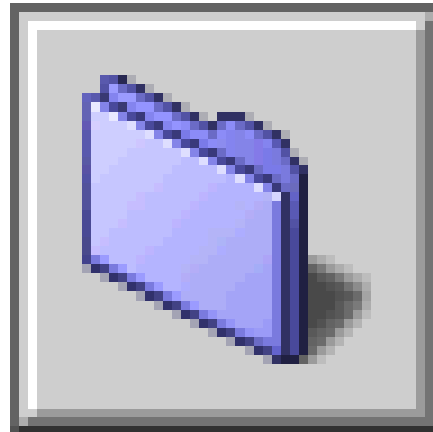
ATGCTAGCTGCTCGA  
TGCTAGCATCGCT

## RNA

AUCGCUCGCUAGCUA  
CGACUACGACUCG

## PROTEIN

HIKKLLPPSSDDEECC  
MMNNKKYYTTREW



**Bio**

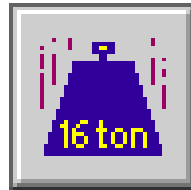


Python 2.2

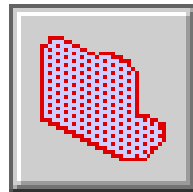
31 items, 1.96 GB available



Python IDE



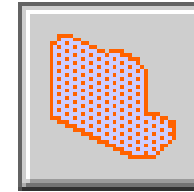
PythonInterpreter



Lib

Lib

214 items, 1.96 GB available

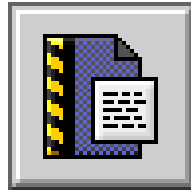


rver.py

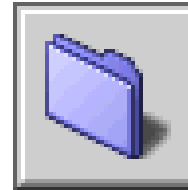
site-packages

site-packages

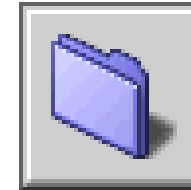
16 items, 1.96 GB available



README



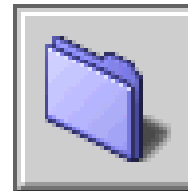
RNG



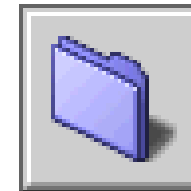
Martel



Bio



Doc



Tests

**BioPython**  
**Computational Nirvana**  
**for a**  
**Molecular Biologist**

**A biopython Seq object has **two** important attributes:**

---

1. **data** -- as the name implies, this is the actual sequence **data string** of the sequence.
2. **alphabet** -- an object describing what the **individual characters** making up the string ``mean'' and how they should be interpreted.

## Let's invite some friends to the party

```
>>> from Bio.Alphabet import IUPAC
```

```
>>> my_alpha = IUPAC.unambiguous_dna
```

```
>>> from Bio.Seq import Seq
```

```
>>> my_seq = Seq('GATCGATGGGCCTATATAGGATCGAAAAT', my_alpha)
```

```
>>> print my_seq
```

```
Seq('GATCGATGGGCCTATATAGGATCGAAAAT', IUPACUnambiguousDNA())
```

## Manipulate Like any other string

```
>>> my_seq[4:12]
```

```
Seq('GATGGGCC', IUPACUnambiguousDNA())
```

```
>>> len(my_seq)
```

```
29
```

```
>>> new_seq = my_seq[0:5]
```

```
>>> print new_seq
```

```
Seq('GATCG', IUPACUnambiguousDNA())
```

```
>>> my_seq + new_seq
```

```
Seq('GATCGATGGGCCTATATAGGATCGAAAATGATCG', IUPACUnambiguousDNA())
```

```
>>> my_seq[5]
```

```
?
```

```
>>> my_seq == new_seq
```

```
?
```

## So what ?

```
>>> protein_seq = Seq('EVRNAK', IUPAC.protein)
```

```
>>> dna_seq = Seq('ACGT', IUPAC.unambiguous_dna)
```

```
>>> protein_seq + dna_seq
```

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in ?
```

```
File "/usr/local/lib/python1.6/site-packages/Bio/Seq.py", line 42, in __add__
```

```
    raise TypeError, ("incompatible alphabets", str(self.alphabet),
```

```
TypeError: ('incompatible alphabets', 'IUPACProtein()', 'IUPACUnambiguousDNA()')
```

```
>>> my_seq.tostring()
```

```
'GATCGATGGGCCTATATAGGATCGAAAATCGC'
```

```
>>> my_seq[5] = 'G'
```

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in ?
```

```
AttributeError: 'Seq' instance has no attribute '__setitem__'
```

**We (Molecular Biologists) love**  
**Central Dogma of Biology**

**DNA**



**RNA**



**Protein**

## DNA --> RNA

```
>>> from Bio.Tools import Transcribe
```

```
>>> transcriber = Transcribe.unambiguous_transcriber
```

```
>>> my_rna_seq = transcriber.transcribe(my_seq)
```

```
>>> print my_rna_seq
```

```
Seq('GAUCGAUGGGCCUAUAUAGGAUCGAAAAU', IUPACUnambiguousRNA())
```

```
>>> transcriber.back_transcribe(my_rna_seq)
```

```
Seq('GATCGATGGGCCTATATAGGATCGAAAAT', IUPACUnambiguousDNA())
```

## DNA --> Proteins

```
>>> from Bio.Tools import Translate
```

```
>>> standard_translator = Translate.unambiguous_dna_by_id[1]
```

```
>>> mito_translator = Translate.unambiguous_dna_by_id[2]
```

```
>>> my_seq = Seq('GCCATTGTAATGGGCCGCTGAAAG', IUPAC.unambiguous_dna)
```

```
>>> standard_translator.translate(my_seq)
```

```
Seq('AIVMGR*KG', IUPACProtein())
```

```
>>> mito_translator.translate(my_seq)
```

```
Seq('AIVMGRWKG', IUPACProtein())
```

```
>>> standard_translator.translate_to_stop(my_seq)
```

```
Seq('AIVMGR', IUPACProtein())
```

## But a Cell hasn't found a way for

DNA <---- Protein

```
>>> my_protein = Seq('AVMGRWKGGRAAG', IUPAC.protein)
```

```
>>> standard_translator.back_translate(my_protein)
```

```
Seq('GCTGTTATGGGTCGTTGGAAGGGTGGTCGTGCTGCTGGT', IUPACUnambiguousDNA())
```

**The End**

## If you insist on changing

```
>>> mutable_seq = my_seq.tomutable()
```

```
>>> print mutable_seq
```

```
MutableSeq('GATCGA_TGGGCCTATATAGGATCGAAAAT', IUPACUnambiguousDNA())
```

```
>>> mutable_seq[5] = 'T'
```

```
>>> print mutable_seq
```

```
MutableSeq('GATCGT_TGGGCCTATATAGGATCGAAAAT', IUPACUnambiguousDNA())
```

```
>>> mutable_seq.remove('T')
```

```
>>> print mutable_seq
```

```
MutableSeq('GACGTTGGGCCTATATAGGATCGAAAAT', IUPACUnambiguousDNA())
```

```
>>> mutable_seq.reverse()
```

```
>>> print mutable_seq
```

```
MutableSeq('TAAAAGCTAGGATATATCCGGGTTGCAG', IUPACUnambiguousDNA())
```