Biopython

- What is Biopython?
- How do I get it to run on my computer?
- What can it do?
Biopython

- Biopython is a set of Python modules useful in bioinformatics

- Features include:
  - Parsing files in different database formats
  - Interfaces into programs like Blast, Entrez and PubMed
  - A sequence class (can transcribe, translate, invert, etc)
  - Code for handling alignments of sequences
  - Clustering algorithms

- Useful tutorials at http://biopython.org
Making Biopython available on your computer

- http://biopython.org/DIST/docs/install/installation.html
- The more recent your computer, the better your chances
- Runs on Windows, MacOSX, and Linux
Why isn’t it on the course computers?

• Regrettably, we won’t be able to use Biopython on the course machines

• The problem may be too-old versions of other software

• I was able to install it on OSX and Linux machines in my lab

• Try it on your own machines—if they are recent it is quite likely to work

• There will be no Biopython homework questions
>>> from Bio.Seq import Seq  # the sequence class
>>> my_seq = Seq("AGTACACTGGT")
>>> my_seq.alphabet
Alphabet()
>>> print my_seq.tostring()
AGTACACTGGT
More functionality than a plain string

```python
>>> my_seq
Seq('AGTACACTGGT', Alphabet())
>>> my_seq.complement()
Seq('TCATGTGACCA', Alphabet())
>>> my_seq.reverse_complement()
Seq('ACCAGTGTACT', Alphabet())
```
A sequence in a specified alphabet

```python
>>> from Bio.Seq import Seq
>>> from Bio.Alphabet import IUPAC
>>> my_seq = Seq('AGTACACTGGT', IUPAC.unambiguous_dna)
>>> my_seq
Seq('AGTACACTGGT', IUPACUnambiguousDNA())
```
Transcribe

```python
>>> from Bio import Transcribe
>>> my_dna = Seq("GATCGATGGGCTATATAGGATCGAAAATCGC",
...               IUPAC.unambiguous_dna)
>>> transcriber = Transcribe.unambiguous_transcriber
>>> my_rna = transcriber.transcribe(my_seq)
>>> print my_rna_seq
Seq('GAUCGAUGGGCCUAUAUAGGAUCGAAAAUCGC', IUPACUnambiguousRNA())
# also possible to reverse transcribe, translate
```
FASTA database file named "ls_orchid.fasta":

```python
from Bio import SeqIO
handle = open("ls_orchid.fasta")
for seq_record in SeqIO.parse(handle, "fasta") :
    print seq_record.id
    print seq_record.seq
    print len(seq_record.seq)
handle.close()
```

```
>gi|2765658|emb|Z78533.1|CIZ78533 C.irapeanum 5.8S rRNA gene
CGTAACAGATTCCCCTGAGGTGAACCTGCGGAAGGATCATTGATGAGACCGTGGAATTTACGATCGAGTG
AATCCGGAGGACCAGTCTACTAGCTCACCGGCGGCGATTTGCTCCCGTGGATCCTGTGGTTGG
```

```
Seq('CGTAACAAGGTTCCGTAGGTGAACCTGCGGAAGGATCATTGATGAGACCGTGGAATAACGATCGAGTG
AATCCGGAGGACCAGTCTACTAGCTCACCGGCGGCGATTTGCTCCCGTGGATCCTGTGGTTGG',
    SingleLetterAlphabet())
```

740
Searching GenBank

from Bio import GenBank
gi_list = GenBank.search_for("Opuntia AND rpl16")

# gi_list will be a list of all of the GenBank
# identifiers that match our query:
print gi_list
['6273291', '6273290', '6273289', '6273287',
 '6273286', '6273285', '6273284']
Searching GenBank

```
ncbi_dict = GenBank.NCBIDictionary("nucleotide", "genbank")
gb_record = ncbi_dict[gi_list[0]]
print gb_record

LOCUS          AF191665   902 bp   DNA       PLN       07-NOV-1999
DEFINITION    Opuntia marenae rpl16 gene; chloroplast gene for chloroplast
              product, partial intron sequence.
ACCESSION     AF191665
VERSION       AF191665.1 GI:6273291
...```
How would I use Biopython?

- Browse the documentation and become familiar with its capabilities
- When writing a bioinformatics program, keep Biopython in mind
- Prefer it to writing your own code for:
  - Defining and handling sequences and alignments
  - Parsing database formats
  - Interfacing with databases
- Biopython is not a program itself; it’s a collection of tools for Python bioinformatics programs
- You don’t have to use it all: pick out one or two elements to learn first
Code re-use

- If someone has written solid code that does what you need, use it
- Don’t "re-invent the wheel" unless you’re doing it as a learning project
- Python excels as a "glue language" which can stick together other peoples’ programs, functions, classes, etc.