Why do we want to use Python for bioinformatics?

Some examples:

3d Modeling of proteins and protein dynamics:

https://pymol.org/2/

**PyMOL 2.3**

PyMOL is a user-sponsored molecular visualization system on an open-source foundation.

**Machine Learning** tools:

scikit-learn
keras
Tensorflow
Pytorch

**Biopython**

https://biopython.org  Biopython 1.73

The **Biopython Project** is an international association of developers of freely available Python (http://www.python.org) tools for computational biology. Similarly, there exist BioPerl and BioJava Projects.

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

# create a sequence object
my_seq = Seq('CATGTAGACTAG')

# print out some details about it
print 'seq %s is %i bases long' % (my_seq, len(my_seq))
print 'reverse complement is %s' % my_seq.reverse_complement()
print 'protein translation is %s' % my_seq.translate()

### OUTPUT

seq CATGTAGACTAG is 12 bases long
reverse complement is CTAGTCTACATG
protein translation is HVD*

Use the SeqIO module for reading or writing sequences as SeqRecord objects. For multiple sequence alignment files, you can alternatively use the AlignIO module.

Tutorial online:
The main Biopython releases have lots of functionality, including:

- The ability to parse bioinformatics files into Python utilizable data structures, including support for the following formats:
  - Blast output – both from standalone and WWW Blast – Clustalw
  - FASTA
  - GenBank (NCBI annotated collection of public DNA sequences)
  - PubMed and Medline
  - ExPASy files, like Enzyme and Prosite (Protein DB)
    
    ExPasy is a bioinformatics portal operated by the Swiss Institute of Bioinformatics
  - UniGene
  - SwissProt (Proteic sequences DB)

- Files in the supported formats can be iterated over record by record or indexed and accessed via a Dictionary interface.

- Code to deal with popular on-line bioinformatics destinations such as:
  - NCBI – Blast, Entrez (search engine on several biomed DB) and PubMed services
    
    NCBI=National Center for Biotechnology Information (Bethesda, Maryland)
  - ExPASy – Swiss-Prot and Prosite entries,
    as well as Prosite searches

- Interfaces to common bioinformatics programs such as:
  - Standalone Blast from NCBI
  - Clustalw alignment program
  - EMBOSS command line tools (for pairwise sequence alignment)

- A standard sequence class that deals with sequences, ids on sequences, and sequence features.

- Tools for performing common operations on sequences, such as translation, transcription and weight calculations.

- Code to perform classification of data using k Nearest Neighbors, Naive Bayes or Support Vector Machines.

- Code for dealing with alignments, including a standard way to create and deal with substitution matrices.
- Code making it easy to split up parallelizable tasks into separate processes.
- GUI-based programs to do basic sequence manipulations, translations, BLASTing, etc.
- Extensive documentation and help with using the modules, including a tutorial, on-line wiki documentation, the web site, and the mailing list.
- Integration with BioSQL, a sequence database schema also supported by the BioPerl and BioJava projects.

Some bioinformatic topics (and corresponding functions) covered by the tutorial.

3 Sequence objects
Sequences and Alphabets
Nucleotides, Aminoacids, Open alphabets …

4 Sequence annotation objects

https://biopython.org/wiki/SeqRecord
(wiki-entries and often published articles to explain the implemented functionalities)

>>> from Bio.SeqRecord import SeqRecord
>>> help(SeqRecord)

Most of the sequence file format parsers in BioPython can return SeqRecord objects

SeqIO system will only return SeqRecord objects.

5 Sequence Input/Output
5.1 Parsing or Reading Sequences
5.1.1 Reading Sequence Files

5.3 Parsing sequences from the net
5.3.1 Parsing Gen Bank records from the net
5.3.2 Parsing Swiss Prot sequences from the net

5.6 Low level FASTA and FASTQ parsers

6 Multiple Sequence Alignment objects
6.1 Parsing or Reading Sequence Alignments

7 BLAST
7.1 Running BLAST over the Internet
7.2 Running BLAST locally

9 Accessing NCBI’s Entrez

NCBI’s Entrez databases is a data retrieval system that provides users access to NCBI’s databases such as PubMed, GenBank, GEO, and many others.

You can access Entrez from a web browser to manually enter queries, or you can use Biopython’s Bio.Entrez module for programmatic access to Entrez. The latter allows you for example to search PubMed or download GenBank records from within a Python script.

10 Swiss-Prot and ExPASy
10.1 Parsing Swiss-Prot files

11 Going 3D: The PDB module
11.1 Reading and writing crystal structure files

11.2.6 Extracting a specific Atom/Residue/Chain/Model from a Structure

12 Bio.PopGen: Population genetics

Bio.PopGen is a Biopython module supporting population genetics, available in Biopython 1.44 onwards.

GenePop (http://genepop.curtin.edu.au/) is a popular population genetics software package supporting Hardy-Weinberg tests, linkage disequilibrium, population differentiation, basic statistics, $F_{st}$ and migration estimates, among others.

13 Phylogenetics with Bio.Phylo
13.1 Demo: What’s in a Tree?

15 Cluster analysis
15.1 Distance functions

16 Supervised learning methods
16.1 The Logistic Regression Model

Note the supervised learning methods described in this chapter all require Numerical Python (numpy) to be installed.

17 Graphics including Genome Diagram

19 Bio.phenotype: analyse phenotypic data
19.1 Phenotype Microarrays