

Computing for Medicine: Phase 3, Seminar 2 Project

Based on slides by Michelle Craig

Seminar 2 Project: Sequence Analysis

- The project is posted handout is posted:
 - <https://c4m-uoft.github.io/seminars/Seminar2Project.pdf>
- Packages used:
 - Biopython (included in your Anaconda distribution)
- Installation instructions:
 - <https://c4m-uoft.github.io/>

GLOB

Python's glob module

- <https://docs.python.org/3/library/glob.html>
- Used to find files whose names match a given pattern.
- Symbols used:
 - * (matches zero or more characters)
 - ? (matches exactly one character)
 - [] (matches one character contained within the brackets)

Demo

Example directory contains the following files:

a.txt, apple.txt, b.jpg, banana.txt, carrot.txt, carrot.jpg

```
>>> glob.glob('*.txt')
```

```
['a.txt', 'apple.txt', 'banana.txt', 'carrot.txt']
```

Demo

Example directory contains the following files:

a.txt, apple.txt, b.jpg, banana.txt, carrot.txt, carrot.jpg

```
>>> glob.glob('*jpg')
```

```
['b.jpg', 'carrot.jpg']
```

```
>>> glob.glob('?.txt')
```

```
['a.txt']
```

Demo

Example directory contains the following files:

a.txt, apple.txt, b.jpg, banana.txt, carrot.txt, carrot.jpg

```
>>> glob.glob('?.*')
```

```
['a.txt', 'b.jpg']
```

Demo

```
>>> glob.glob('a*')
```

```
['a.txt', 'apple.txt']
```

```
>>> glob.glob('*a*')
```

```
['a.txt', 'apple.txt', 'banana.txt', 'carrot.jpg', 'carrot.txt']
```

```
>>> glob.glob('[ab].*')
```

```
['a.txt', 'b.jpg']
```

```
>>> glob.glob('[bc]*.txt')
```

```
['banana.txt', 'carrot.txt']
```


BIOPYTHON

Python's biopython module

- The starter code in `calculate_consensus.py` uses the biopython module (Bio).
- To complete this project, you should read the starter code and aim to understand what that biopython code is doing.
- You will need to model part of your solution to `find_mutations.py` on the starter code provided in `calculate_consensus.py`.

Demo

```
filename =
```

```
"EBOV_REDC502_MinION_GUI_Conakry_2015-07-13.reads.fasta"
```

```
# Open the file containing the input reads
```

```
handle = open(filename, "rU")
```

```
# Iterate over the input reads and save their sequence in a list
```

```
read_sequences = []
```

```
for record in SeqIO.parse(handle, "fasta"):
```

```
    read_sequences.append(str(record.seq))
```

Upcoming Seminar

Seminar 3: Frank Rudzick

Date: Tuesday November 20, 2018; 4-6pm

Location: DSC Innovation Lab, Gerstein Library

Topic: Natural Language Processing in Clinical Medicine

Profile: <http://www.cs.toronto.edu/~frank/>