CSE 111 Bio: Program Design I
Lecture 13: BLAST, while loops

Bob Sloan (CS) & Rachel Poretsky (Bio)
University of Illinois, Chicago
October 10, 2017
Grace Hopper Celebration of Women in Computing

Apply in Spring, probably April, if you want to be part of the UIC Team for Grace Hopper 2018
Announcements

• Project 1 out today. Due 1 week from Friday
In the evolution of life on Earth, the three domains of life—Archaea, Bacteria, and Eukarya—branch from a single point. (credit: modification of work by Eric Gaba)
Figure 12.3
Sequence comparisons

• Analysis of relationships
  – Phylogenetic tree reconstruction
  – Detection of conserved regions

• Prediction of function
  – Genes of similar sequence often encode proteins of similar function
  – Sequences of unknown function can be queried against a database of sequences with known functions
Similarity, Homology <-> Function

• Generally, if two sequences are similar, they have:
  – A common ancestor – homology
  – A common structure
  – A similar function
Comparing amino acid (protein) vs. nucleic acid (DNA) sequences— which gives you a better idea functional similarity?

A. Amino acid
B. DNA
C. Same information
D. I have no idea
• Within the cell, where does energy to power reactions come from? ATP
• ATP is a small, relatively simple molecule, but within its bonds contains the potential for a quick burst of energy that can be harnessed to perform cellular work.
• This molecule can be thought of as the primary energy currency of cells in the same way that money is the currency that people exchange for things they need.
• ATP is used to power the majority of energy-requiring cellular reactions.
ATP

• Cells cannot store significant amounts of free energy. Excess free energy would result in an increase of heat in the cell, which would denature enzymes and other proteins, and thus destroy the cell.
• Cells use ATP to store energy safely and release it for use only as needed
• ATP functions as a rechargeable battery:
  • When ATP is broken down, energy is released
  • For example, in the mechanical work of muscle contraction, ATP supplies energy to move the contractile muscle proteins.
(a) The electron transport chain is a set of molecules that supports a series of oxidation-reduction reactions.

(b) ATP synthase is a complex, molecular machine that uses an $H^+$ gradient to regenerate ATP from ADP.
The Basic Local Alignment Search Tool (BLAST) for comparing gene and protein sequences against others in public databases.

BLAST is a set of sequence comparison algorithms used to search databases for optimal local alignments to a query.
LOOPS: MOSTLY WHILE LOOPS
Recall: Sequential coding elements

- Generally Python statements run one at a time, in order we write them in

- Assignment statements
- Function calls

- But if, for, and while change execution order
def spamify(L):
    '''Add "n spam" to every string in L.'''
    newL = []
    for s in L:
        newL.append(s + "n spam")
    return newL

For loop goes directly over the list L

>>> spamify(['eggs', 'sausage', 'oatmeal'])
['eggsn spam', 'sausagen spam', 'oatmealn spam']

Green eggs n spam!
Review: two types of for loop

```python
def spamCount(S):
    '''Count occurrences of "spam" in input S.'''
    counter = 0
    for i in range(len(S)):
        if S[i:i+4] == "spam":
            counter = counter + 1
    return counter

>>> spamCount("gspamtspammspamn")
3
```

For loop goes over indices into string S
def mystery(n):
    for d in range(2, n):
        if n % d == 0:
            return False
    return True

mystery(6) returns:

A. False
B. True
C. False False True
D. False True False True False
E. No clue
A mystery…

def mystery(n):
    for d in range(2, n):
        if n % d == 0:
            return False
    return True

What if you don’t want to exit?
Print, not return

What if you want to return the list of all the results of the if test?
A mystery...returning a list

def mystery1(n):
    myList = []
    for d in range(2, n):
        if n % d == 0:
            myList = myList+[False]
        else:
            myList = myList+[True]
    return myList
Iteration (Iterative code elements)

- Repeat
  - for: do once for each letter in a string
    - And other generalizations we will see later
  - while loops
    - repeat until some condition is met
  - Lots of uses. One coming up for us: In gene finding, we will need to say, among other things "while I'm not too close to the end of this string, see if a gene starts in this position"
While Loops

while <condition>:
    <body>

- While condition is True, execute body statements. Once it is False, continue to next section of code
- Condition must evaluate to either True or False
- Colon and indentation are required
Example

```python
x = 3
while x < 6:
    print(x)
    x = x + 1
```

- x starts as 3; while condition true; 3 is printed
- x becomes 4
- while condition still true
- 4 printed
- x becomes 5
- while condition still true
- 5 is printed
- x becomes 6
- while condition false
- 6 not printed
x = 8
while x > 1:
    print(x)
    x = x // 2
A. 8, 4, 2, 1
B. 8, 4, 2
C. 4, 2, 1
D. 4, 2
Writing While Loops

<initialization>
while <condition>:
    <body>
        <increment or advance>
        - Initialize: Set up variable(s)
        - Condition: How long should loop run?
        - Body: What loop does
        - Increment: Change variable(s) (make sure loop will eventually stop!)
Example

```python
x = 8
while x > 1:
    print(x)
    x = x // 2
```
Example

```
x = 8
while x > 1:
    print(x)
    x = x // 2
```
Example

```
x = 8
while x > 1:
    print(x)
    x = x // 2
```
Example

```
x = 8
while x > 1:
    print(x)
    x = x // 2
```
Example

```
x = 8
while x > 1:
    print(x)
    x = x // 2
```
How many times will this loop print

\[
x = "Microbes rule"
while len(x) > 9:
    print(x)
    x = x + "!"
\]

A. 12  
B. 9  
C. 3  
D. 4  
E. It will print forever
What value is stored in variable z when code finishes executing?

\[
\begin{align*}
x &= 1 & A. & 2 \\
y &= 2 & B. & 3 \\
z &= 0 & C. & 4 \\
\text{while } x \leq 3: & D. & 6 \\
\quad z &= z + y & E. & 8 \\
\quad x &= x + 1
\end{align*}
\]
What value is stored in variable \( z \) when code finishes executing?

\[
x = 1 \\
z = 1 \\
while x <= 3:
\begin{align*}
  z &= z + x \\
  x &= x + 1
\end{align*}
\]

A. 2  
B. 3  
C. 4  
D. 7  
E. 9
What value is stored in variable z when code finishes executing?

\[
x = 5 \\
y = 1 \\
z = 3
\]

while \( x > 0 \):
  \[
z = z + y \\
x = x - 1
\]

A. 2  
B. 3  
C. 4  
D. 8  
E. 9
What value is stored in variable z when code finishes executing?

```python
x = 0
z = 0
while x < 3:
    x = x + 1
    z = z + 2
```

A. 2  
B. 3  
C. 4  
D. 6  
E. 8