Working on exercises (a few notes first)

Comments

Sometimes you want to make a comment in the Python code, to remind you what's going on.

Python ignores everything from a # to the end of the line. Feel free to write anything you want.

```
# This is ignored by Python.

print "Hello, Cape Town"  # And so is this.

# n = 0
# for line in open("filename"):
#   print n, line
#   n = n + 1
```

IDLE helps out

To comment many lines:
Select some lines then choose "Format" from the pull-down menu. Choose "Comment Out Region" from the list that appears.

To uncomment many lines:
Use "Format" -> "Uncomment Region"

To move lines to the right:
Use "Format" -> "Indent Region"

To move lines to the left, "Dedent Region"

Python and division

Python integers are "closed" under division

That's a special way of saying that an integer divided by another integer using Python will always return an integer.

"integers" are a special way of saying "whole numbers," that is, numbers without a fraction.

Mathematicians have lots of special names!

(And so do biologists. And programmers. ☺)

Python rounds down

When the result is a fraction, Python rounds it down to the next smallest integer

```
>>> 20 / 10
2
>>> 15 / 10
1
>>> 10 / 10
1
>>> 9 / 10
0
>>>
```

How to fix it

The author of Python now says this behavior was a mistake. It should work like people expect.

Instead, you need to convert one of the integers into a floating point number

```
>>> 15 / float(10)
1.5
>>>
```

More examples

```
>>> 20 / float(10)
2.0
>>> 15 / float(10)
1.5
>>> 10 / float(10)
1.0
>>> 9 / float(10)
0.90000000000000002
>>>
```

Why do you need to know about this?

Yesterday's assignment asked you to find all sequences with more than 50% GC content

```
>>> G = 120
>>> C = 33
>>> length = 400
>>> (G + C) / length
0
>>> (G + C) / float(length)
0.38250000000000001
>>>
```

Exercise 6

Look again at sequences.seq from yesterday.

Did your program assume only A, T, C, and G?

For this exercise, count the number of sequences in that file which have some other letter.

What might those mean?

How does it affect your %GC calculations?

Exercise 7

What are the extra letters?

Write a program to list which letters in the data file sequences.seq are not A, T, C, or G. It should only list each letter once.

Hint for Exercise 7

```
# Start with a list of unknown letters.
# (This is empty because at the start there are
# no unknown letters.)
unknown_letters = []

for each sequence in the data file:
    for each letter in the sequence:
        if letter not in "ATCG":
            # it isn't an A, T, C, or G:
            if letter not in unknown_letters:
                # it isn't in the list of unknown letters
                append it to the list of unknown letters

print the list of unique letters
```

Exercise 8

Search by molecular weight

From experiments you believe your DNA sequence has a molecular weight between 224245 and 226940. You think it might be in the database from yesterday, sequences.seq.

For each sequence in that file which have a molecular weight in that range, print the molecular weight and the sequence.

You might need the data table on the next page.

Molecular weights

A = 347.0
C = 323.0
B = 336.0
D = 344.0
G = 363.0
H = 330.666666667
K = 342.5
M = 335.0
N = 338.75
S = 343.0
R = 355.0
T = 322.0
W = 334.5
V = 344.333333333
Y = 322.5
X = 338.75

I put a copy of these weights
in the file
/usr/coursehome/dalke/weights.txt

And the molecular weight
of water is 18.0.

(Why did I give you that?)

Exercise 9

Verify that your program actually works.
How? Here are some possibilities:

Compare your results with others'.

Create a small sequence file by hand where you know the weight. Compare your manual calculations to what your program found.

Find a web site which does this. Compare.

Find a published table of weights. Compare.

....

Exercise 10

Modify your program from Exercise 8 to ask the user for the filename to use, the lower weight, and the upper weight. You will need the `float` function to convert the string from `raw_input` into a number.

Redo exercises 8 and 9 with the file /usr/coursehome/dalke/many_sequences.seq Your program must get the filename and ranges from the user via `raw_input` and not by modifying your program.

Exercise 11

If you finish all the previous assignments:

Use the many_sequences.seq file as the input file for the programs you wrote yesterday.

Help other people learn Python.