Homework Assignment 3 – Python Code & PDB Files

DUE in class, Oct. 3

Write a Python program to identify certain protein secondary structural segments. You will use a protein we will study later (dihydrofolate reductase). Some useful python code and coding tips are provided below to get you started.

Start by downloading the full PDB file (1RA2.pdb) from the *compbio2/Shared/* directory on kirin to a subdirectory in your home directory on the Mac. Include your name in a comments section at the top of your Python code. You can use the templates below or your own knowledge of Python in your program. Your program should do the following analysis:

1) Determine the mean i to i+3 Alpha Carbon – Alpha Carbon distance (along the chain) and have the code print out this number to the screen.
2) Write the residue index (i) and i -> i+3 CA distance to a file (not a redirect from the screen).
3) Plot the residue index vs. CA distance data, label the plot (axes, title, name and secondary structure as described below).
4) Identify the secondary structure of protein segments of >4 residues with CA separation <6.5 Å using PyMOL.

Please name your program file with *your* JHED ID (e.g. for me it would be *pflemin1.py* or *hmwk3_pflemin1.py*). The program must run as-is for credit.

1. Send a copy of the python script by email to Dr. Fleming (pat.fleming@jhu.edu).
2. Also, attach a ps (or png or pdf) copy of the plot of residue index vs. CA distance. In the subtitle of your plot indicate the secondary structure you determined in step 4, above and your name.

Your name must appear in at least three places: file name with .py extension, header or comment section of the python file, subtitle of your plot. No name, no credit.

Example code for calculating distances was in a lab exercise.

In class you learned how to iterate through a list, e.g.

```
for i in range(0,len(list)):
    print(list[i])
```

In this assignment you will need to iterate through a list and assign the i and i+3 items, e.g.

```
for i in range(0,len(list)-3):
    var1 = float(list[i])
    var2 = float(list[i+3])
```

Below is the template code:

```
# Define a function or module
def readpdb():
```

```
# Open the pdb file with read only permission
file = open('1RA2.pdb','r')

# Read in all the lines in the pdb file
temp = file.readlines()

# Close the pdb file once you are done extracting information
file.close()

# Confirm that you read in the file successfully
print("There are ",len(temp)," lines in your file")
print(temp[0])
print(temp[1])

# Remove the end of line character '\n' from the file.
#   strip removes all white space at the beginning and end
#   of the line, including end of line character
# Notice that we do two operations simultaneously in this loop:
#   strip the line and append the stripped line to array temp2
# We do not include any arguments with the strip command because
#   we want the default which is white space
temp2 = []
for i in range(0,len(temp)):
    temp2.append(temp[i].strip())

# Split each line into the words (splits by white space).
# Store each word string as an element in a list
A = []
for i in range(0,len(temp)):
    A.append(temp2[i].split())

# Check that first and second indexed items are what you expect
print(A[0])
print(A[1])

# Separate the ATOM information from list A
# Notice that we use two indices to look at both a specific line
# of array A [i] and a specific item [0] within that line.
AtomInfo = []
for i in range(0,len(A)):
        if A[i][0] == "ATOM":
            AtomInfo.append(A[i])

#Check it
print(AtomInfo[0])
print(AtomInfo[1])

# Separate the Alpha Carbon information from the AtomInfo list
# and put in a list or array
# Your code

# Determine the mean i->i+3 Alpha Carbon Cartesian distance separation
# using the coordinates of each
# Your code

# Print out to a file i and i->i+3 distance separation
# It should look like this,
# 1      10.785
# 2       9.864
# 3      10.504
# etc.
# Your code

# Plot the i and distance values in xmgrace.
# Label the plot (name in title) and email a postscript or PDF copy.
```

```
    # Find segments of >4 residues with CA separation of <6.5 Angstroms.
    # Using pymol identify the secondary structure of these segments.
    # Write in the subtitle of the plot the answer to the following question:
    #   What secondary structure type are these segments?

# Call the module
readpdb()
```