

Genomics data analysis in Python (DSCI511) Fall 2018

Things you should be familiar with

1. print() function
2. input() function
3. conditional statements – if, elif, else
4. boolean expressions
5. logical operators
6. type() function and objects
7. try and except
8. Membership operators
9. len() function
10. modules
11. if `__name__ == '__main__':`
12. range() function
13. looping through strings
14. indexing strings
15. slicing strings
16. str.find() method
17. str.replace() method
18. str.translate() method
19. string parsing
20. reading from files (read(), readline(), for loops)
21. writing to files
22. proper handling of exceptions raised when opening files
23. creating lists
24. indexing lists
25. slicing lists
26. traversing lists
27. list.append() method
28. list.sort() method

29. `str.split()` method and creating lists from strings
30. `str.rstrip()` method
31. matrices
32. creating dictionaries
33. key-value pairs and assignment
34. searching for keys and values in dictionaries
35. `dict.get()` method
36. creating tuples
37. difference between tuples, lists, and dictionaries
38. `tuple.count()` method
39. `tuple.index()` method
40. simple list comprehensions

Note: There won't be any questions related to subprocesses, command line arguments, or regular expressions.

Sample questions (expect 5 questions with a similar level of difficult to those below)

1. Write a function, `revcomp(sequence)`, that reverse complements a DNA string. The function should work when imported into another program and should not include an input prompt.
2. Write a function, `cat(input_file1, input_file2, output)`, that concatenates two files into a new file. The function should exit gracefully if the files can't be opened and should not leave any files open.
3. Write a function, `double_matrix(matrix)`, that doubles the values of each element within a matrix and returns the results as a new matrix.
4. Write a function, `kmer(sequence, k)` that computes the number of occurrences of all k-mers of length k in a sequence.
5. Write a function, `fasta_revcomp(input_file, output_file)`, that reverse complements all sequences within a fasta file.