Introduction to Script Programming

Final Project

Exercise 1 (REMOVING DUPLICATES)

In this problem you are given a file with 20 columns and several rows. The rows contain information regarding interactions between proteins but don't worry as no biological background is needed and the problem is purely scripting.
The interactions are between 2 proteins A and B and the 20 columns are the following:

***Organism**:* the organism in which the interaction is found

protein_id_A: unique accession number (ID) of the 1st protein (protein A). Each protein can have a unique ID.

Name_A: full name of protein A

GeneA: gene that codes for protein A

motif_A: a regular expression that specified the amino acids involved in the interaction in protein A. Each letter is an amino acid, the dot can be any character and the number in {} is the number of characters (ie I.{3}V.{3}L.{2}VL.{6}L). ***Each protein can have 1 or more motifs***

fixed_residues_A: number of amino acid letters in the motif of protein A. For I.{3}V.{3}L.{2}VL.{6}L it is 6. So there are 6 amino acids that interact and the amino acids represented by dots don't interact.

prop_cost_A: this is the multiple alignment cost and can be disregarded.

error_rate_A: the error rate percentage of the motif found in the sequence (this is a percentage: 0, 0.05, 0.1, 0.15 or 0.2). Each motif can have different error rates.

cost_A: the number of amino acids different between the motif and protein sequence. For instance if the motif is I.{3}V and the sequence is IEKLV the cost will be 0 because you have I at the first position, 3 characters and then V, the error rate will also be 0 in this case. However, if the sequence was IEKLE, the cost would be 1, because the V is replaced by E.

Valid_A: if protein A has a 3D structure (PDB) or not. (1 if there is a structure and 0 if not)

protein_id_B: same description as before but for protein B
Name_B: same description as before but for protein B
GeneB: same description as before but for protein B
motif_B: same description as before but for protein B
fixed_residues_B: same description as before but for protein B

prop_cost_B: same description as before but for protein B
error_rate_B: same description as before but for protein B
cost_B: same description as before but for protein B
Valid_B: same description as before but for protein B

Valid: this should be 1 if and only if both Valid_A and Valid_B are 1, and 0 otherwise.

The problem with this data is the duplications and this can be of two types.

1. Row duplication, the same row is repeated more than once in the data. So exact same information copied twice or more.
2. Interaction duplication. This is the more challenging. In fact you might have a row with the interaction information between A and B (A-B) that has the same information as another interaction (B-A).
   For example, assume A is protein P41232 and B is protein A0JPA0 in one row (interaction1: P41232 - A0JPA0). If there is another row that has protein A to be A0JPA0 and protein B to be P41232 (interaction 2: A0JPA0 - P41232), then interaction2 **IS CONSIDERED A DUPLICATE IF THE FOLLOWING 4 CONDITIONS ARE TRUE:**
   - The motif_A of interaction 1 is the same as motif_B of interaction 2
   - The motif_B of interaction 1 is the same as motif_A of interaction 2
   - The error_rate_A and cost_A of interaction 1 are the same as error_rate_B and cost_B of interaction 2
   - The error_rate_B and cost_B of interaction 1 are the same as error_rate_A and cost_A of interaction 2

You need to write a script that takes as input argument the data file “TransmembraneData” and process it to identify all duplicate interactions, and keep only one occurrence of each interaction. Your script should do the following:

- Print on the terminal to total number of initial interactions
- Keep a backup of the original file
- Write all unique interactions to a tab-delimited file named “uniqueInteractions.txt”. Be careful here, you should keep one entry for each duplicated interaction as well. For instance if A-B and B-A are in the data, you should keep either A-B or B-A in the uniqueInteractions.txt file (again assuming the 4 conditions mentioned above are met). So here unique means one occurrence of each interaction.
- Write all duplicated (ie removed interactions) to another tab-delimited file named “duplicateInteractions.txt”
- Print on the terminal the number of unique interactions
- Print on the terminal the number of duplicated interactions.
- You script will be graded based on the correctness as well as time and memory efficiency!

Exercise 2 (Counting Interactions)

You are given a set of txt files that contain information regarding protein interactions. These files are in PDBSum.zip and have the following format:

```
List of atom-atom interactions across protein-protein interface
---------------------------------------------------------------


                PDB code: 4jkv   Chains A }{ B
                -------------------------------



Hydrogen bonds
--------------|

       <----- A T O M   1 ----->       <----- A T O M   2 ----->

       Atom Atom Res  Res              Atom Atom Res  Res
        no. name name no.  Chain          no. name name no.  Chain  Distance
  1.    2091  NE2 GLN  351    A    <-->  5611  OG  SER  358    B      3.12
  2.    2448  OH  TYR  399    A    <-->  5939  OH  TYR  399    B      2.27
  3.    2448  OH  TYR  399    A    <-->  5939  OH  TYR  399    B      2.27
  4.    2576  OH  TYR  417    A    <-->  5623  OH  TYR  359    B      2.83
  5.    2576  OH  TYR  417    A    <-->  5623  OH  TYR  359    B      2.83

Non-bonded contacts
-------------------

       <----- A T O M   1 ----->       <----- A T O M   2 ----->

       Atom Atom Res  Res              Atom Atom Res  Res
        no. name name no.  Chain          no. name name no.  Chain  Distance
  1.    1974  CE1 TYR  337    A    <-->  5616  CB  TYR  359    B      3.70
  2.    2005  CE1 HIS  340    A    <-->  5621  CE2 TYR  359    B      3.74
  3.    2080  CE2 TYR  350    A    <-->  5611  OG  SER  358    B      3.71
  4.    2081  CZ  TYR  350    A    <-->  5610  CB  SER  358    B      3.79
  5.    2081  CZ  TYR  350    A    <-->  5611  OG  SER  358    B      3.70
  6.    2082  OH  TYR  350    A    <-->  5610  CB  SER  358    B      3.47
  7.    2082  OH  TYR  350    A    <-->  5611  OG  SER  358    B      3.89
  8.    2082  OH  TYR  350    A    <-->  5612  N   TYR  359    B      3.76
  9.    2082  OH  TYR  350    A    <-->  5619  CD2 TYR  359    B      3.24
 10.    2091  NE2 GLN  351    A    <-->  5611  OG  SER  358    B      3.12
```

The file can have 1 or more section: “Hydrogen bonds” , “Non-bonded contacts”, “ Bisulphide bonds”.

WE ARE ONLY INTERESTED IN “Non-bonded contact”

This section provides information regarding interacting atoms in each amino acids. Each line has two similar parts, one for ATOM 1 and one for its interacting complement ATOM 2.

The first column (Atom no.) is the atom number

The second column (Atom name) is the atom name

The third column (Res name) is the amino acid symbol

The forth column (Res no.) is the amino acid number

The fifth column (Chain) is the Chain in the amino acid is found in.

Based on the above figure:
TYR at position 337 in ATOM 1 interacts with TYR at position 359 in ATOM 2
HIS at position 340 in ATOM 1 interacts with TYR at position 359 in ATOM 2
TYR at position 350 in ATOM 1 interacts with SER at position 358 in ATOM 2
TYR at position 350 in ATOM 1 interacts with TYR at position 359 in ATOM 2
GLN at position 351 in ATOM 1 interacts with SER at position 358 in ATOM 2

You need to write a script that will identify for each Amino Acid (Res Name) in Atom 1, its complement amino acid (Res. Name) in Atom 2 and count the number of interactions. Be careful to take into account the amino acid number as well (Res no.).

For instance, based on the figure above,
TYR - TYR : 2 (1 from TYR 337 – TYR 359 and 1 from TYR 350 – TYR 359
HIS - TYR: 1 (from HIS 340 – TYR 359)
TYR – SER: 1 (from TYR 350 – SER 358)
GLN – SER: 1 (from GLN 351 – SER 358)

Notice here that we only take the amino acid and not the atoms when counting. TYR 350 and SER 358 had multiple atoms interacting (5) but we only count it as one interaction. So basically you should take only the unique amino acid (Res name.) and number (Res no.) combination.

Your script should read and process all the txt files provided at ounce. It should print on the terminal the number of files processed.
The script should also compute the interaction number of each amino acid (in all files together and not in each file separate) and write them to a single file "interactionStatistics.txt" with the number of interactions for each Amino Acid (residue). Your file should be also tab delimited and should look like this:

	Ala	Arg	Asn	Acid	Cys	Acid	Gln	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	230	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Arg	210	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Asn	10	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Acid	140	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Cys	540	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Acid	321	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Gln	443	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Gly		…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
His	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Ile	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Leu	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Lys	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Met	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Phe	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Pro	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Ser	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Thr	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Trp	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Tyr	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Val	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

NB: The … should be actual numbers!