PDB File Analysis Using Shell Scripting:

Project Overview:

This project aims to automate the retrieval and analysis of PDB (Protein Data Bank) files using Linux shell scripting. By leveraging common Unix command-line tools such as curl, grep, awk, cut, and sort, we performed structured operations on biological data files to extract meaningful insights.

The core tasks involved:

- Batch downloading PDB structures from the RCSB database using curl
- Filtering out invalid files that were not found on the server
- Parsing atomic-level information, particularly alpha carbon (CA) atoms

Counting amino acid frequencies while avoiding artificial inflation due to multiple models in a single file

This work showcases the utility of shell scripting in bioinformatics and structural biology, where researchers often deal with large sets of PDB files and need fast, reproducible methods for batch processing.

Initialization:

First we create a working directory called “STRUCTURE”:

mkdir -p STRUCTURE

Then we change the directory we are working on:

cd STRUCTURE

Task 1: Download multiple .pdb files using curl and remove the ones that are invalid or not found in the PDB database.

By using the curl command we change the identifier in the presented URL:

curl “http://files.rcsb.org/view/1W[0-9][A-Z].pdb “ -o structure_#1#2.pdb

This command attempts to fetch a range of PDB entries using a wildcard pattern. It creates .pdb files named by combining characters matched in the URL.

```
joudy@DESKTOP-U2QRDRD:~/STRUCTURE$ curl "http://files.rcsb.org/view/1W[0-9][A-Z].pdb" -o structure_#1#2.pdb

[1/260]: http://files.rcsb.org/view/1W0A.pdb --> structure_0A.pdb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 2432k    0 2432k    0     0   406k      0 --:--:--  0:00:05 --:--:--  503k

[2/260]: http://files.rcsb.org/view/1W0B.pdb --> structure_0B.pdb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 2628k    0 2628k    0     0   173k      0 --:--:--  0:00:15 --:--:--  122k

[3/260]: http://files.rcsb.org/view/1W0C.pdb --> structure_0C.pdb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 1527k    0 1527k    0     0  95898      0 --:--:--  0:00:16 --:--:-- 90144

[4/260]: http://files.rcsb.org/view/1W0D.pdb --> structure_0D.pdb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  897k    0  897k    0     0  83637      0 --:--:--  0:00:10 --:--:-- 95682

[5/260]: http://files.rcsb.org/view/1W0E.pdb --> structure_0E.pdb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  340k    0  340k    0     0  73225      0 --:--:--  0:00:04 --:--:-- 82869

[6/260]: http://files.rcsb.org/view/1W0F.pdb --> structure_0F.pdb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  349k    0  349k    0     0   147k      0 --:--:--  0:00:02 --:--:--  147k
```

```
[7/260]: http://files.rcsb.org/view/1W0G.pdb --> structure_0G.pdb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  349k    0  349k    0     0   249k      0 --:--:--  0:00:01 --:--:--  249k

[8/260]: http://files.rcsb.org/view/1W0H.pdb --> structure_0H.pdb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  205k    0  205k    0     0   248k      0 --:--:-- --:--:-- --:--:--  248k

[9/260]: http://files.rcsb.org/view/1W0I.pdb --> structure_0I.pdb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  602k    0  602k    0     0   245k      0 --:--:--  0:00:02 --:--:--  245k

[10/260]: http://files.rcsb.org/view/1W0J.pdb --> structure_0J.pdb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 2078k    0 2078k    0     0   164k      0 --:--:--  0:00:12 --:--:--  188k

[11/260]: http://files.rcsb.org/view/1W0K.pdb --> structure_0K.pdb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 1961k    0 1961k    0     0   157k      0 --:--:--  0:00:12 --:--:--  167k

[12/260]: http://files.rcsb.org/view/1W0L.pdb --> structure_0L.pdb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   260  100   260    0     0   1277      0 --:--:-- --:--:-- --:--:--  1280
```
…

We notice that the file structure_0L.pdb of size 260 and all files of same size contain the following:

```
joudy@DESKTOP-U2QRDRD:~/STRUCTURE$ cat structure_0L.pdb
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
  <head>
    <title>404 Not Found</title>
  </head>
  <body>
    <h1>Not Found</h1>
    <p>The requested URL was not found on this server.</p>
    <hr>
    <address>RCSB PDB</address>
  </body>
</html>
```

Therefore, in order to get the files that do not exist in the database (that are not found in the database, those of size 260)

we can use the *grep* command: **grep -il 'not found' *.pdb | xargs rm**

or we can use *awk* command: **ls -l | awk '$5 == 260 {print $NF}' | xargs rm**

Task 2: List amino acids (excluding 'UNK') by counting how often their CA (alpha carbon) atoms appear in all PDB files.

grep -h '^ATOM.*CA' *.pdb |grep -v 'UNK'| cut -c 18-20| sort| uniq -c| sort -nr

EXPLANATION:

*grep -h '^ATOM.*CA' *.pdb :* get the lines starting with "ATOM" and containing "CA".
-h: hides the name of the file
grep -v 'UNK' : display all lines except the ones with 'UNK'
cut -c 18-20 : get the column with the 3 letters amino acid name
sort: sort the amino acids in alphabetical order.
uniq -c: count number of occurrences of each unique line of amino acids
sort -nr: sort amino acids by numeric value and in reverse order.

```
joudy@DESKTOP-U2QRDRD:~/STRUCTURE$ grep -h '^ATOM.*CA' *.pdb |grep -v 'UNK'| cut -c 18-20| sort| uniq -c| sort -nr
  17250 LEU
  16678 ALA
  15930 GLY
  14992 VAL
  13116 GLU
  11805 ASP
  11623 SER
  11422 LYS
  11062 THR
  10992 ILE
  10551 ARG
   9220 PRO
   8723 ASN
   7520 PHE
   7244 GLN
   6997 TYR
   4108 HIS
   3813 MET
   2877 TRP
   2720 CYS
```

We noticed that some files contain several Models as the example shown below:

```
joudy@DESKTOP-U2QRDRD:~/STRUCTURE$ grep -A 10 'MDL' structure_9N.pdb
NUMMDL    20
AUTHOR    M.EKKELENKAMP,M.G.M.HANSSEN,S.-T.D.HSU,A.DE JONG,D.MILATOVIC,
AUTHOR   2 J.VERHOEF,N.A.J.VAN NULAND
REVDAT   7   15-NOV-23 1W9N    1       REMARK LINK   ATOM
REVDAT   6   02-MAY-18 1W9N    1       JRNL   REMARK
REVDAT   5   13-JUL-11 1W9N    1       VERSN
REVDAT   4   24-FEB-09 1W9N    1       VERSN
REVDAT   3   03-APR-07 1W9N    1       HETATM
REVDAT   2   21-APR-05 1W9N    1       LINK
REVDAT   1   01-APR-05 1W9N    0
JRNL        AUTH   M.B.EKKELENKAMP,M.HANSSEN,S.T.DANNY HSU,A.DE JONG,
--
ENDMDL
MODEL        2
HETATM    1  C   2OP A   1       8.537  26.287  -2.682  1.00 10.75           C
HETATM    2  O   2OP A   1       9.225  26.945  -3.461  1.00 10.89           O
HETATM    3  CB  2OP A   1       9.380  26.769  -0.376  1.00 11.39           C
HETATM    4  OHN 2OP A   1       7.112  26.027  -0.764  1.00 11.41           O
HETATM    5  CA  2OP A   1       8.179  26.817  -1.302  1.00 11.16           C
HETATM    6  HB1 2OP A   1      10.289  26.801  -0.955  1.00 11.48           H
HETATM    7  HB2 2OP A   1       9.354  25.854   0.199  1.00 11.70           H
HETATM    8  HB3 2OP A   1       9.352  27.616   0.296  1.00 11.38           H
HETATM    9  H   2OP A   1       6.278  26.519  -0.837  1.00 11.39           H
--
```

Continuing with this example, we notice that within a file, the same amino acid is counted several times if there is more than 1 model, i.e if there are 20 models, the frequency of a specific amino acid would increase by 20 instead of 1.

The amino acids are similar in every model within a file may only vary when it comes to coordinates, as we observed and compared in output of the following command:

```
joudy@DESKTOP-U2QRDRD:~/STRUCTURE$ grep -B 400 'ENDMDL' structure_9N.pdb | grep '^ATOM.*CA'
ATOM     65  CA  LYS A   6      23.553 -12.294   1.573  1.00  4.51           C
ATOM    109  CA  ILE A   9      15.534 -12.075   5.687  1.00  3.26           C
ATOM    128  CA  LYS A  10      13.196 -14.931   6.558  1.00  3.16           C
ATOM    150  CA  ALA A  11       9.442 -14.398   6.975  1.00  3.25           C
ATOM    169  CA  LYS A  13       6.597 -13.069   2.450  1.00  3.96           C
ATOM    191  CA  LYS A  14       3.556 -15.107   3.428  1.00  4.48           C
ATOM    213  CA  LEU A  15       0.797 -12.506   2.996  1.00  3.83           C
ATOM    232  CA  CYS A  16       2.805 -10.085   5.151  1.00  2.19           C
ATOM    242  CA  ARG A  17       4.605  -8.178   2.393  1.00  3.96           C
ATOM    266  CA  GLY A  18       2.543  -5.113   3.225  1.00  4.13           C
ATOM    273  CA  PHE A  19       1.607  -4.666  -0.411  1.00  2.82           C
ATOM    305  CA  LEU A  21      -3.691  -4.227   2.166  1.00  3.21           C
ATOM    336  CA  CYS A  23      -6.853  -3.030  -2.058  1.00  3.51           C
ATOM    346  CA  GLY A  24      -8.423  -5.837  -4.035  1.00  5.01           C
ATOM    353  CA  CYS A  25     -11.150  -6.245  -1.448  1.00  5.27           C
ATOM    363  CA  HIS A  26     -14.759  -5.113  -1.296  1.00  7.10           C
ATOM    381  CA  PHE A  27     -16.031  -7.272   1.541  1.00  8.71           C
ATOM    412  CA  GLY A  29     -16.404  -3.800   6.180  1.00 12.06           C
ATOM    419  CA  LYS A  30     -19.511  -1.761   6.830  1.00 13.70           C
ATOM    441  CA  LYS A  31     -22.459  -1.697   9.199  1.00 15.24           C
ATOM     65  CA  LYS A   6       1.400  17.859  -6.550  1.00  4.51           C
ATOM    109  CA  ILE A   9       6.521  13.951  -1.569  1.00  3.26           C
ATOM    128  CA  LYS A  10       8.712  10.865  -1.383  1.00  3.16           C
ATOM    150  CA  ALA A  11       7.423   9.895   2.047  1.00  3.25           C
ATOM    169  CA  LYS A  13       4.019   7.657  -1.417  1.00  3.96           C
ATOM    191  CA  LYS A  14       2.932   4.068  -2.176  1.00  4.48           C
ATOM    213  CA  LEU A  15       4.040   0.910  -0.302  1.00  3.83           C
ATOM    232  CA  CYS A  16       4.114   2.676   3.048  1.00  2.19           C
ATOM    242  CA  ARG A  17       0.491   2.314   4.190  1.00  3.96           C
ATOM    266  CA  GLY A  18      -0.184   1.856   0.502  1.00  4.13           C
ATOM    273  CA  PHE A  19       0.412  -1.835  -0.037  1.00  2.82           C
ATOM    305  CA  LEU A  21      -3.805  -4.610   2.053  1.00  3.21           C
ATOM    336  CA  CYS A  23      -6.722  -3.074  -2.145  1.00  3.51           C
ATOM    346  CA  GLY A  24      -8.383  -6.056  -3.755  1.00  5.01           C
ATOM    353  CA  CYS A  25     -11.349  -6.029  -1.390  1.00  5.27           C
ATOM    363  CA  HIS A  26     -14.672  -4.220  -0.907  1.00  7.10           C
ATOM    381  CA  PHE A  27     -16.843  -4.811  -3.948  1.00  8.71           C
ATOM    412  CA  GLY A  29     -22.174  -4.875  -6.318  1.00 12.06           C
ATOM    419  CA  LYS A  30     -24.126  -4.388  -9.517  1.00 13.70           C
ATOM    441  CA  LYS A  31     -24.873  -0.674  -9.229  1.00 15.24           C
```

So we need to add to the beginning of our initial answer, a command that considers all files, if a file contains several models, consider only the first one (i.e., content before the first ENDMDL).

This is the command:

```
for file in *.pdb; do
  if grep 'ENDMDL' "$file"; then
    awk '/ENDMDL/{exit} 1' "$file"
  else
    cat "$file"
  fi
done | grep -h '^ATOM.*CA' | grep -v 'UNK' | cut -c 18-20 | sort | uniq -c | sort -nr
```

Explanation:

for loop: iterates over all .pdb files in the current directory.
if statement uses *grep 'ENDMDL' "$file"*: checks if the file contains the ENDMDL string.
If *ENDMDL is found*: awk is used to print everything up to the first ENDMDL.
If *ENDMDL is not found*: *cat* is used to print the entire file.

After processing each file according to the presence of multiple models delimited by the first occurrence of ENDMDL, the rest of the pipeline (grep, cut, sort, uniq -c, sort -nr) processes the combined output to count the amino acid residues as in the original script.

This is the final output:

```
joudy@DESKTOP-U2QRDRD:~/STRUCTURE$ for file in *.pdb; do
  if grep 'ENDMDL' "$file"; then
    awk '/ENDMDL/{exit} 1' "$file"
  else
    cat "$file"
  fi
done | grep -h '^ATOM.*CA' | grep -v 'UNK' | cut -c 18-20 | sort | uniq -c | sort -nr
  14901 LEU
  14060 ALA
  13746 GLY
  12906 VAL
  10686 GLU
  10180 SER
   9992 ASP
   9656 ILE
   9637 THR
   8858 LYS
   8660 ARG
   7810 PRO
   7589 ASN
   6824 PHE
   6212 TYR
   6001 GLN
   3684 HIS
   3233 MET
   2432 TRP
   2138 CYS
```

The final script combines careful file selection and model filtering with a streamlined pipeline for amino acid frequency extraction. This work can serve as a reusable template for analyzing residue composition across any group of protein structures in PDB format.