ASSIGNMENT #2 – Regular Expressions – BIOL550 (5 Questions; 35 PTS)

Regular expressions (*aka* regexes) are motifs/patterns designed to match specific strings of characters in text files. These matches can then be captured or excluded/ignored depending on the desired objective.

Regular expressions can vary in complexity, from very simple to highly complex. When designing a regular expression, I recommend starting with something broad and simple that matches all the characters that you are looking for. This often leads to capturing extra characters and/or patterns that were not desired. If so, you can modify/complexify your expression as needed make it more specific and remove undesired matches.

Files commonly used in bioinformatics can be found as binary files or as human-readable ASCII text files. Text files in bioinformatics include formats commonly found in programming (*e.g.* .json, .xml, .tsv) and custom formats designed specifically to store biological data (*e.g.* .fastq, .gbk, .vcf). These texts files can be further designed for human readability or to facilitate parsing with regular expressions and/or scripting/programming languages.

In this assignment, we will practice how to use regular expressions by parsing several text files often encountered in bioinformatics. We'll learn more about these text files in the second half of the semester. ***Feel free to use online regular expression tools like Regexr (https://regexr.com/) to help you design your regular expressions.***

The input files for this assignment are available Blackboard as A2_F22_files.tar.gz. Note that while we will be using files that are relevant to bioinformatics in the questions below, the goal here is to test your understanding of regular expressions, not biology/bioinformatics. ***If you are not in biology or lack the relevant background pertaining to these files, feel free to ask questions about them.***

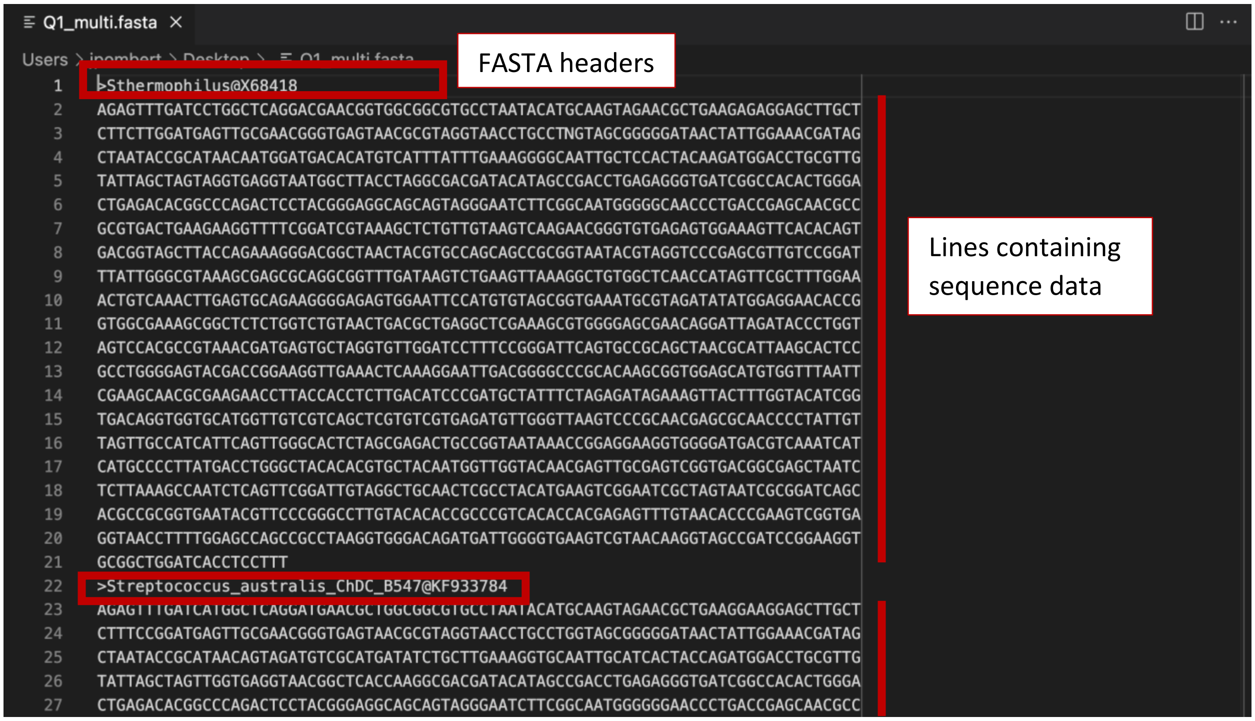
What to do:

a) Download the file A2_F22_files.tar.gz from Blackboard to your local computer
b) In your account on the class server, create a folder named: ~/Assignment/Assignment_2
c) Upload A2_F22_files.tar.gz from your local computer to your ~/Assignment/Assignment_2 folder on the class server
d) Untar the A2_F22_files.tar.gz file archive and work from your account on the class server.
e) You can test your Perl Compliant Regular Expressions (PCRE) directly on the input files with grep -P ## See question 1 below for an example of how to do so.
f) Write the regular expressions you designed for each question in a single text file that includes your name (*e.g.* ***Pombert_JF_regexes.txt***). Make sure to use regular expressions, *i.e.* **DO NOT** use the output examples as input files for grep. Use a code editor to save your regexes; Microsoft Word will likely corrupt your regexes!
g) Submit the text file containing your regular expressions to Blackboard.

7 PTS. QUESTION 1

FASTA is a standard file format for biological sequences, it can contain one
## or more sequences. Each sequence is identified by a header starting with >.
## This header is followed by one or more line(s) containing the corresponding
## sequence: https://www.ncbi.nlm.nih.gov/genbank/fastaformat/

1. Create a regular expression that will return only the **sequence lines**; *i.e.* lines that do not matches the fasta headers from the file Q1_multi.fasta.
2. Test it with grep on Q1_multi.fasta. Write the output in Q1_myHeaders.txt; e.g.
   grep -P 'regular expression' Q1_multi.fasta > Q1_myHeaders.txt
3. You can compare your output to that of Q1_desired_output.txt with ***wc -l*** and ***diff***:
   wc -l Q1_myHeaders.txt Desired_outputs/Q1_desired_output.txt
   diff -s Q1_myHeaders.txt Desired_outputs/Q1_desired_output.txt



FASTA headers

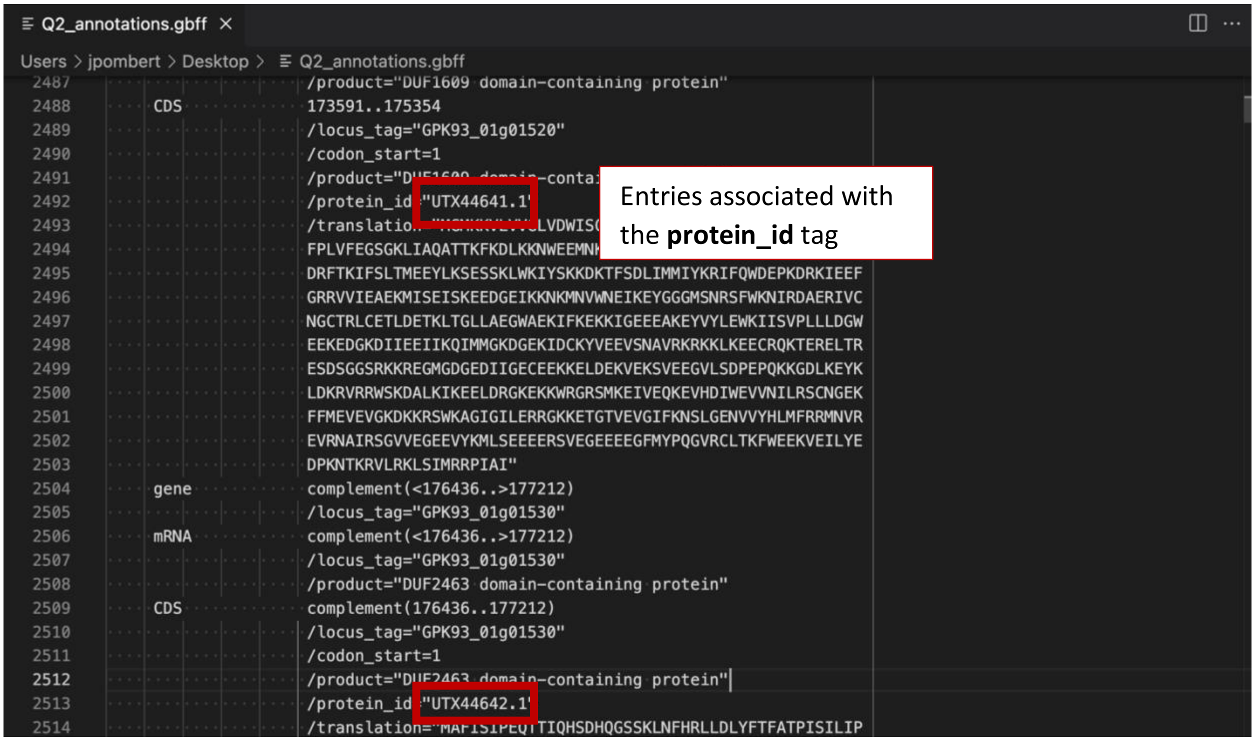
Lines containing sequence data

```
>Sthermophilus@X68418
AGAGTTTGATCCTGGCTCAGGACGAACGGTGGCGGCGTGCCTAATACATGCAAGTAGAACGCTGAAGAGAGGAGCTTGCT
CTTCTTGGATGAGTTGCGAACGGGTGAGTAACGCGTAGGTAACCTGCCTNGTAGCGGGGGATAACTATTGGAAACGATAG
CTAATACCGCATAACAATGGATGACACATGTCATTTATTTGAAAGGGGCAATTGCTCCACTACAAGATGGACCTGCGTTG
TATTAGCTAGTAGGTGAGGTAATGGCTTACCTAGGCGACGATACATAGCCGACCTGAGAGGGTGATCGGCCACACTGGGA
CTGAGACACGGCCCAGACTCCTACGGGAGGCAGCAGTAGGGAATCTTCGGCAATGGGGGCAACCCTGACCGAGCAACGCC
GCGTGACTGAAGAAGGTTTTCGGATCGTAAAGCTCTGTTGTAAGTCAAGAACGGGTGTGAGAGTGGAAAGTTCACACAGT
GACGGTAGCTTACCAGAAAGGGACGGCTAACTACGTGCCAGCAGCCGCGGTAATACGTAGGTCCCGAGCGTTGTCCGGAT
TTATTGGGCGTAAAGCGAGCGCAGGCGGTTTGATAAGTCTGAAGTTAAAGGCTGTGGCTCAACCATAGTTCGCTTTGGAA
ACTGTCAAACTTGAGTGCAGAAGGGGAGAGTGGAATTCCATGTGTAGCGGTGAAATGCGTAGATATATGGAGGAACACCG
GTGGCGAAAGCGGCTCTCTGGTCTGTAACTGACGCTGAGGCTCGAAAGCGTGGGGAGCGAACAGGATTAGATACCCTGGT
AGTCCACGCCGTAAACGATGAGTGCTAGGTGTTGGATCCTTTCCGGGATTCAGTGCCGCAGCTAACGCATTAAGCACTCC
GCCTGGGGAGTACGACCGGAAGGTTGAAACTCAAAGGAATTGACGGGGCCCGCACAAGCGGTGGAGCATGTGGTTTAATT
CGAAGCAACGCGAAGAACCTTACCACCTCTTGACATCCCGATGCTATTTCTAGAGATAGAAAGTTACTTTGGTACATCGG
TGACAGGTGGTGCATGGTTGTCGTCAGCTCGTGTCGTGAGATGTTGGGTTAAGTCCCGCAACGAGCGCAACCCCTATTGT
TAGTTGCCATCATTCAGTTGGGCACTCTAGCGAGACTGCCGGTAATAAACCGGAGGAAGGTGGGGATGACGTCAAATCAT
CATGCCCCTTATGACCTGGGCTACACACGTGCTACAATGGTTGGTACAACGAGTTGCGAGTCGGTGACGGCGAGCTAATC
TCTTAAAGCCAATCTCAGTTCGGATTGTAGGCTGCAACTCGCCTACATGAAGTCGGAATCGCTAGTAATCGCGGATCAGC
ACGCCGCGGTGAATACGTTCCCGGGCCTTGTACACACCGCCCGTCACACCACGAGAGTTTGTAACACCCGAAGTCGGTGA
GGTAACCTTTTGGAGCCAGCCGCCTAAGGTGGGACAGATGATTGGGGTGAAGTCGTAACAAGGTAGCCGATCCGGAAGGT
GCGGCTGGATCACCTCCTTT
>Streptococcus_australis_ChDC_B547@KF933784
AGAGTTTGATCATGGCTCAGGATGAACGCTGGCGGCGTGCCTAATACATGCAAGTAGAACGCTGAAGGAAGGAGCTTGCT
CTTTCCGGATGAGTTGCGAACGGGTGAGTAACGCGTAGGTAACCTGCCTGGTAGCGGGGGATAACTATTGGAAACGATAG
CTAATACCGCATAACAGTAGATGTCGCATGATATCTGCTTGAAAGGTGCAATTGCATCACTACCAGATGGACCTGCGTTG
TATTAGCTAGTTGGTGAGGTAACGGCTCACCAAGGCGACGATACATAGCCGACCTGAGAGGGTGATCGGCCACACTGGGA
CTGAGACACGGCCCAGACTCCTACGGGAGGCAGCAGTAGGGAATCTTCGGCAATGGGGGGAACCCTGACCGAGCAACGCC
```

7 PTS. QUESTION 2

In genomics, several formats are available to store biological annotations.
## These annotations include where genes are located and information about
## the products they encode (if known). Some of the most common genome
## annotation formats include those from the NCBI GenBank (.gb/.gbff) and
## EMBL (.embl) databases as well as the Genome File Format (.gff/.gff3)
## version 3

1. Create a regular expression that matches all entries associated with the **protein_id** tags from the file Q2_annotations.gbff and returns the corresponding lines with grep (see Q2_desired_output.txt for example).
2. Test it with grep on Q2_annotations.gbff. Write the output in Q2_myProtein_IDs.txt.



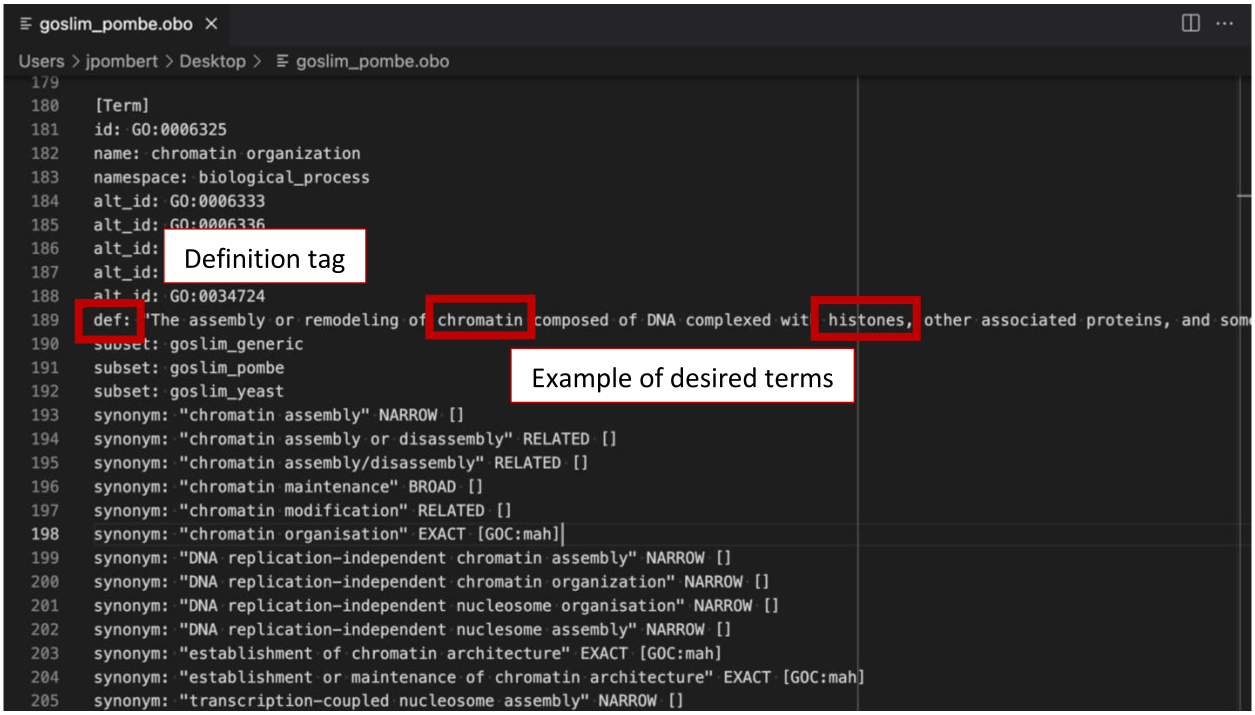
```
                     /product="DUF1609 domain-containing protein"
     CDS             173591..175354
                     /locus_tag="GPK93_01g01520"
                     /codon_start=1
                     /product="DUF1609 domain-conta
                     /protein_id="UTX44641.1"
                     /translatio            LVDWIS
                     FPLVFEGSGKLIAQATTKFKDLKKNWEEMNK
                     DRFTKIFSLTMEEYLKSESSKLWKIYSKKDKTFSDLIMMIYKRIFQWDEPKDRKIEEF
                     GRRVVIEAEKMISEISKEEDGEIKKNKMNVWNEIKEYGGGMSNRSFWKNIRDAERIVC
                     NGCTRLCETLDETKLTGLLAEGWAEKIFKEKKIGEEEAKEYVYLEWKIISVPLLLDGW
                     EEKEDGKDIIEEIIKQIMMGKDGEKIDCKYVEEVSNAVRKRKKLKEECRQKTERELTR
                     ESDSGGSRKKREGMGDGEDIIGECEEKKELDEKVEKSVEEGVLSDPEPQKKGDLKEYK
                     LDKRVRRWSKDALKIKEELDRGKEKKWRGRSMKEIVEQKEVHDIWEVVNILRSCNGEK
                     FFMEVEVGKDKKRSWKAGIGILERRGKKETGTVEVGIFKNSLGENVVYHLMFRRMNVR
                     EVRNAIRSGVVEGEEVYKMLSEEEERSVEGEEEEGFMYPQGVRCLTKFWEEKVEILYE
                     DPKNTKRVLRKLSIMRRPIAI"
     gene            complement(<176436..>177212)
                     /locus_tag="GPK93_01g01530"
     mRNA            complement(<176436..>177212)
                     /locus_tag="GPK93_01g01530"
                     /product="DUF2463 domain-containing protein"
     CDS             complement(176436..177212)
                     /locus_tag="GPK93_01g01530"
                     /codon_start=1
                     /product="DUF2463 domain-containing protein"
                     /protein_id="UTX44642.1"
                     /translation="MAFISIPEQTTIQHSDHQGSSKLNFHRLLDLYFTFATPISILIP
```

Entries associated with the **protein_id** tag

7 PTS. QUESTION 3

Gene ontologies (GOs) are used in bioinformatics to help standardize
## nomenclature pertaining to biological functions of various molecules
## such as proteins and ribonucleic acids (RNAs). The two main formats are the
## Open Biomedical Ontology (OBO) and the Ontology Web Language (OWL).
## http://geneontology.org/docs/ontology-documentation/

1. Create a regular expression that matches all lines with the **def:** tag and which contains at least one of the following terms: **chromatin**, **telomere**, or **histone** from Q3_goslim_pombe.obo (see Q3_desired_output.txt for desired output).
   ## pombe = *Schizosaccharomyces pombe* (fission yeast); an important model organism in
   ## biology
2. Test your regular expression with grep on Q3_ goslim_pombe.obo. Write the output in Q3_myGOs.txt.



```

[Term]
id: GO:0006325
name: chromatin organization
namespace: biological_process
alt_id: GO:0006333
alt_id: GO:0006336
alt_id:
alt_id:
alt_id: GO:0034724
def: "The assembly or remodeling of chromatin composed of DNA complexed wit histones, other associated proteins, and som
subset: goslim_generic
subset: goslim_pombe
subset: goslim_yeast
synonym: "chromatin assembly" NARROW []
synonym: "chromatin assembly or disassembly" RELATED []
synonym: "chromatin assembly/disassembly" RELATED []
synonym: "chromatin maintenance" BROAD []
synonym: "chromatin modification" RELATED []
synonym: "chromatin organisation" EXACT [GOC:mah]
synonym: "DNA replication-independent chromatin assembly" NARROW []
synonym: "DNA replication-independent chromatin organization" NARROW []
synonym: "DNA replication-independent nucleosome organisation" NARROW []
synonym: "DNA replication-independent nucleosome assembly" NARROW []
synonym: "establishment of chromatin architecture" EXACT [GOC:mah]
synonym: "establishment or maintenance of chromatin architecture" EXACT [GOC:mah]
synonym: "transcription-coupled nucleosome assembly" NARROW []
```

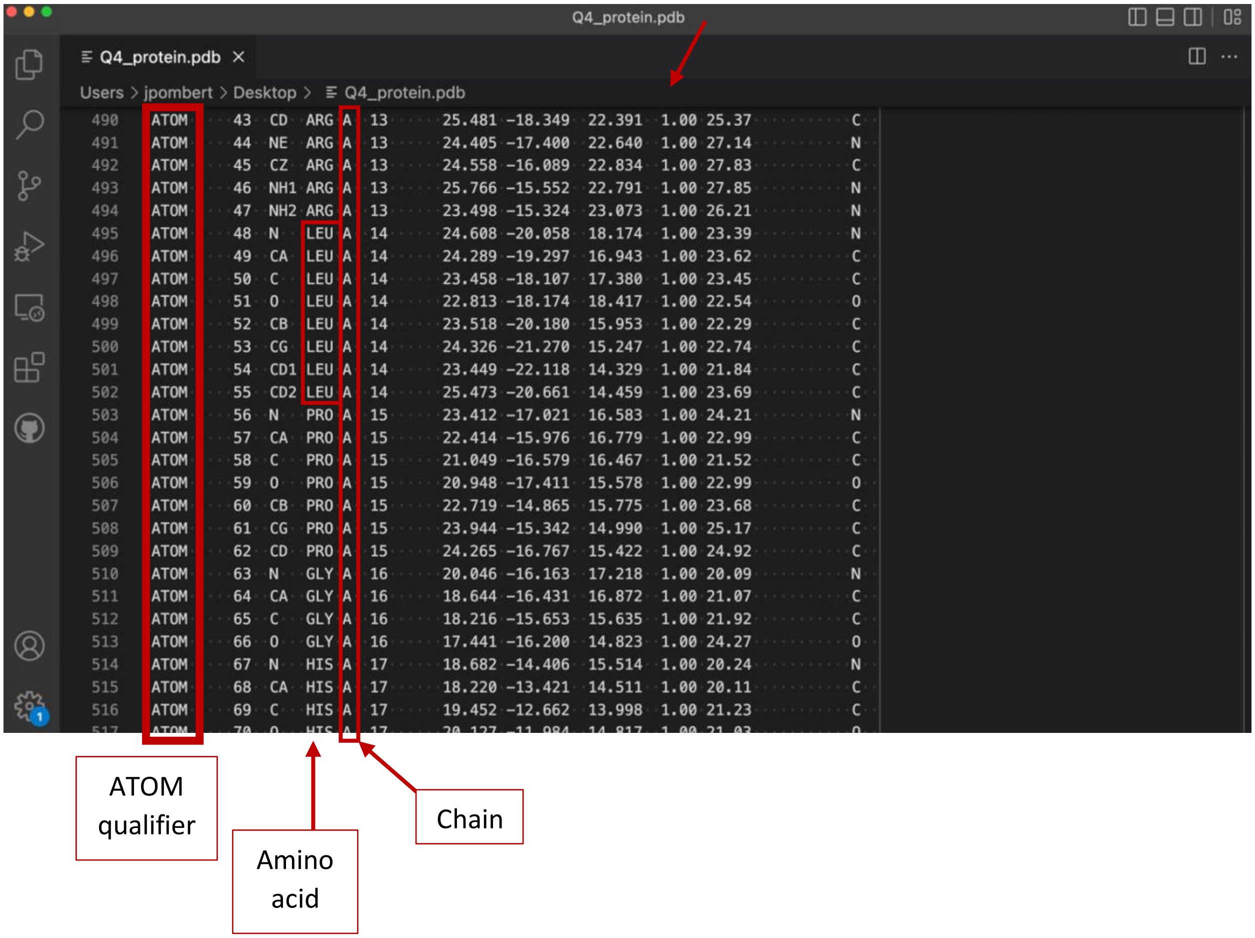
Definition tag

Example of desired terms

7 PTS. QUESTION 4

```
## Protein 3D structures are often stored using the Protein Data Bank (PDB)
## eponym PDB format (https://www.rcsb.org/). This legacy format was later
## expanded into the PDBx/mmCIF (macromolecular Crystallographic
## Information File) format to accommodate larger molecules. Both file formats
## can be visualized using various protein 3D structure viewers like ChimeraX
## (https://www.cgl.ucsf.edu/chimerax/)
```

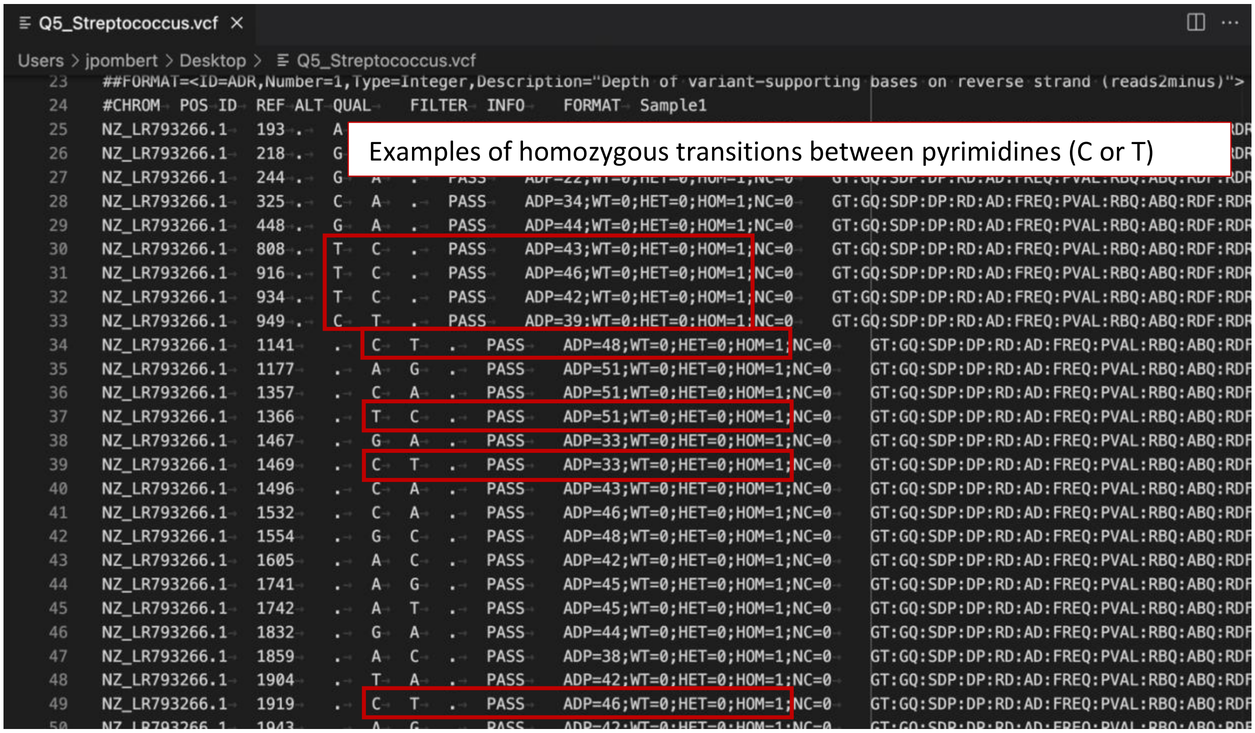
1. Create a regular expression that matches only lines starting with the **ATOM** qualifier and which contain the leucine amino acid (**LEU**) located on chain **A** in Q4_protein.pdb. See lines from Q4_desired_output.txt for the desired outcome.
2. Test your regular expression with grep on Q4_protein.pdb. Write the output in Q4_myLeucine_residues.txt.



```
ATOM     43  CD  ARG A  13      25.481 -18.349  22.391  1.00 25.37           C
ATOM     44  NE  ARG A  13      24.405 -17.400  22.640  1.00 27.14           N
ATOM     45  CZ  ARG A  13      24.558 -16.089  22.834  1.00 27.83           C
ATOM     46  NH1 ARG A  13      25.766 -15.552  22.791  1.00 27.85           N
ATOM     47  NH2 ARG A  13      23.498 -15.324  23.073  1.00 26.21           N
ATOM     48  N   LEU A  14      24.608 -20.058  18.174  1.00 23.39           N
ATOM     49  CA  LEU A  14      24.289 -19.297  16.943  1.00 23.62           C
ATOM     50  C   LEU A  14      23.458 -18.107  17.380  1.00 23.45           C
ATOM     51  O   LEU A  14      22.813 -18.174  18.417  1.00 22.54           O
ATOM     52  CB  LEU A  14      23.518 -20.180  15.953  1.00 22.29           C
ATOM     53  CG  LEU A  14      24.326 -21.270  15.247  1.00 22.74           C
ATOM     54  CD1 LEU A  14      23.449 -22.118  14.329  1.00 21.84           C
ATOM     55  CD2 LEU A  14      25.473 -20.661  14.459  1.00 23.69           C
ATOM     56  N   PRO A  15      23.412 -17.021  16.583  1.00 24.21           N
ATOM     57  CA  PRO A  15      22.414 -15.976  16.779  1.00 22.99           C
ATOM     58  C   PRO A  15      21.049 -16.579  16.467  1.00 21.52           C
ATOM     59  O   PRO A  15      20.948 -17.411  15.578  1.00 22.99           O
ATOM     60  CB  PRO A  15      22.719 -14.865  15.775  1.00 23.68           C
ATOM     61  CG  PRO A  15      23.944 -15.342  14.990  1.00 25.17           C
ATOM     62  CD  PRO A  15      24.265 -16.767  15.422  1.00 24.92           C
ATOM     63  N   GLY A  16      20.046 -16.163  17.218  1.00 20.09           N
ATOM     64  CA  GLY A  16      18.644 -16.431  16.872  1.00 21.07           C
ATOM     65  C   GLY A  16      18.216 -15.653  15.635  1.00 21.92           C
ATOM     66  O   GLY A  16      17.441 -16.200  14.823  1.00 24.27           O
ATOM     67  N   HIS A  17      18.682 -14.406  15.514  1.00 20.24           N
ATOM     68  CA  HIS A  17      18.220 -13.421  14.511  1.00 20.11           C
ATOM     69  C   HIS A  17      19.452 -12.662  13.998  1.00 21.23           C
ATOM     70  O   HIS A  17      20.127 -11.984  14.817  1.00 21.03           O
```

7 PTS. QUESTION 5 ## The Variant Calling Format (VCF) is a standardized format used to store ## information about genetic differences between a query and a reference ## dataset. Differences can include single nucleotide variations as well as larger ## insertions and/or deletions.
## https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3137218/

1. Create a regular expression that matches all lines with **homozygous transitions (HOM=1)** between **pyrimidines (C -> T, T -> C)** from Q5_Streptococcus.vcf (see Q5_desired_output.txt for desired output). ## In this exercise, a dataset from a reference *Streptococcus* genome was mapped against a new clinical isolate…
2. Test it with grep and write the output in Q5_mySNPs.txt.



```
##FORMAT=<ID=ADR,Number=1,Type=Integer,Description="Depth of variant-supporting bases on reverse strand (reads2minus)">
#CHROM  POS ID  REF ALT QUAL    FILTER  INFO    FORMAT  Sample1
NZ_LR793266.1   193 .   A                                                                                       RDR
NZ_LR793266.1   218 .   G                                                                                       RDR
NZ_LR793266.1   244 .   G   A   .   PASS    ADP=22;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF:RDR
NZ_LR793266.1   325 .   C   A   .   PASS    ADP=34;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF:RDR
NZ_LR793266.1   448 .   G   A   .   PASS    ADP=44;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF:RDR
NZ_LR793266.1   808 .   T   C   .   PASS    ADP=43;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF:RDR
NZ_LR793266.1   916 .   T   C   .   PASS    ADP=46;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF:RDR
NZ_LR793266.1   934 .   T   C   .   PASS    ADP=42;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF:RDR
NZ_LR793266.1   949 .   C   T   .   PASS    ADP=39;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF:RDR
NZ_LR793266.1   1141    .   C   T   .   PASS    ADP=48;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1177    .   A   G   .   PASS    ADP=51;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1357    .   C   A   .   PASS    ADP=51;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1366    .   T   C   .   PASS    ADP=51;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1467    .   G   A   .   PASS    ADP=33;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1469    .   C   T   .   PASS    ADP=33;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1496    .   C   A   .   PASS    ADP=43;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1532    .   C   A   .   PASS    ADP=46;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1554    .   G   C   .   PASS    ADP=48;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1605    .   A   C   .   PASS    ADP=42;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1741    .   A   G   .   PASS    ADP=45;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1742    .   A   T   .   PASS    ADP=45;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1832    .   G   A   .   PASS    ADP=44;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1859    .   A   C   .   PASS    ADP=38;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1904    .   T   A   .   PASS    ADP=42;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1919    .   C   T   .   PASS    ADP=46;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
NZ_LR793266.1   1943    .   A   G   .   PASS    ADP=42;WT=0;HET=0;HOM=1;NC=0    GT:GQ:SDP:DP:RD:AD:FREQ:PVAL:RBQ:ABQ:RDF
```

Examples of homozygous transitions between pyrimidines (C or T)