Python sample codes
Codes written in Python to solve real time problems in PhD and Postdoctoral projects

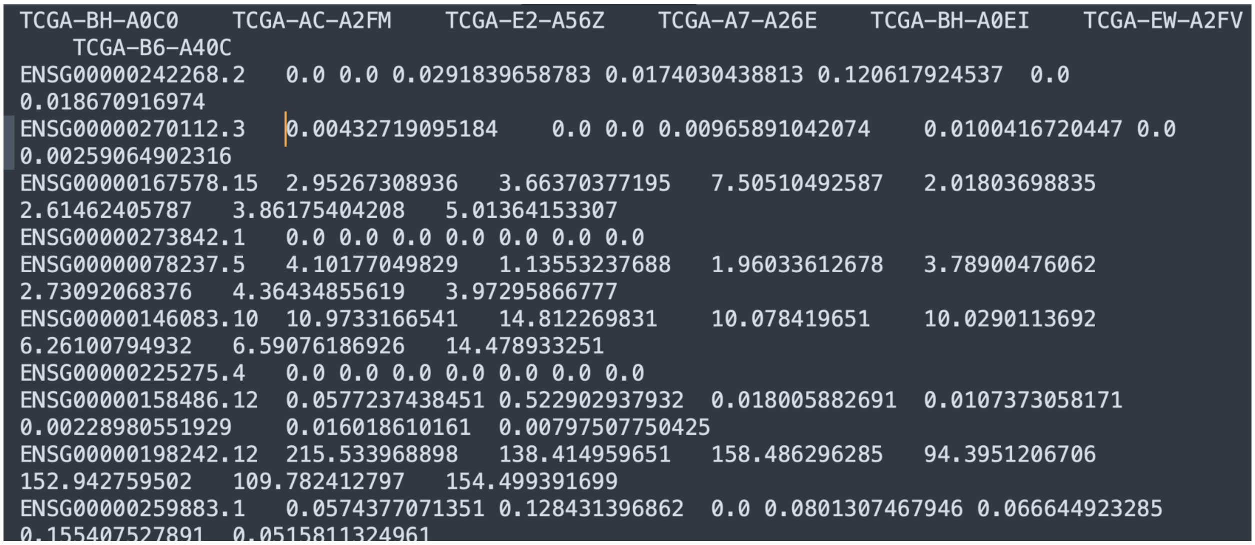
List of codes:

Code-1: #A code to make matrix of gene expression data of 20000 genes from 10000 patient samples downloaded from TCGA
Code-2: # A code to find a match in multiple files (e.g. 4 files here) and output as matrix of 1 (present) and 0 (absent)
Code-3: # A code to print variants specific to a patient sample and variants common to all samples
Code-4: # A code to convert GO IDS to gene symbols and count number of genes belong to each GO id
Code-5: # A code to calculate H-bond distance and H-bond angle from a directory containing n number of files and a master file in another current working directory
Code-6: #A code to convert file containing multiple molecules into files of each molecule
Code-7: #A code to replace string (****) in .mol files to the molecule name
Code-8: #A code to convert molar mass to molecular formula
Code-9: #A code to replace three letter amino acid name (vcf file) to one letter code, e.g Tyr222Cys to T222C
Code 10: # A code to get average of multiple columns in a file using two methods as:
#1. pandas python package
#2 .using dict without help of any python package

Note: These are just few sample codes. I have written many more codes in Python. Currently, I am learning Web application development and its deployment using Python Flask.

Code-1: #A code to make matrix of gene expression data of 20000 genes from 10000 patient samples downloaded from TCGA

```
import glob
path = "/Users/apple/Desktop/TCGA/All-patient-files/*/*"
f2 = open("out.txt","w")
f3=open("gdc_sample_sheet.2018-08-06.tsv","r").read().split("\n")
mfile = f3
dict={}
for filename in glob.glob(path):
    # print (filename)
    splitfilename  = filename.split("/")
    splitfilename2 = splitfilename[9].split(".")
    print (splitfilename2[0])
    f1=open("{0}".format(filename),"r").read().split("\n")
    List = f1
    for line in List:
        # print ("***********")
        splitline = line.split("\t")
        key = splitline[0]
        # print (key)
        value = splitline[1]
        if key in dict:
            fetchvalue = dict.get(key)
            value = fetchvalue + "\t" + value
            dict[key]=value
        else:
            dict[key]=value
    for line2 in mfile:
        # print ("###########")
        splitmline = line2.split("\t")
        splitmline2=splitmline[1].split(".")
        if splitmline2[0] == splitfilename2[0]:
            # print (splitmline[1])
            f2.write(splitmline[5])
            f2.write("\t")
f2.write("\n")
for key in dict:
    # print (key,dict[key])
    f2.write(key)
    f2.write("\t")
    f2.write(dict[key])
    f2.write("\n")
```

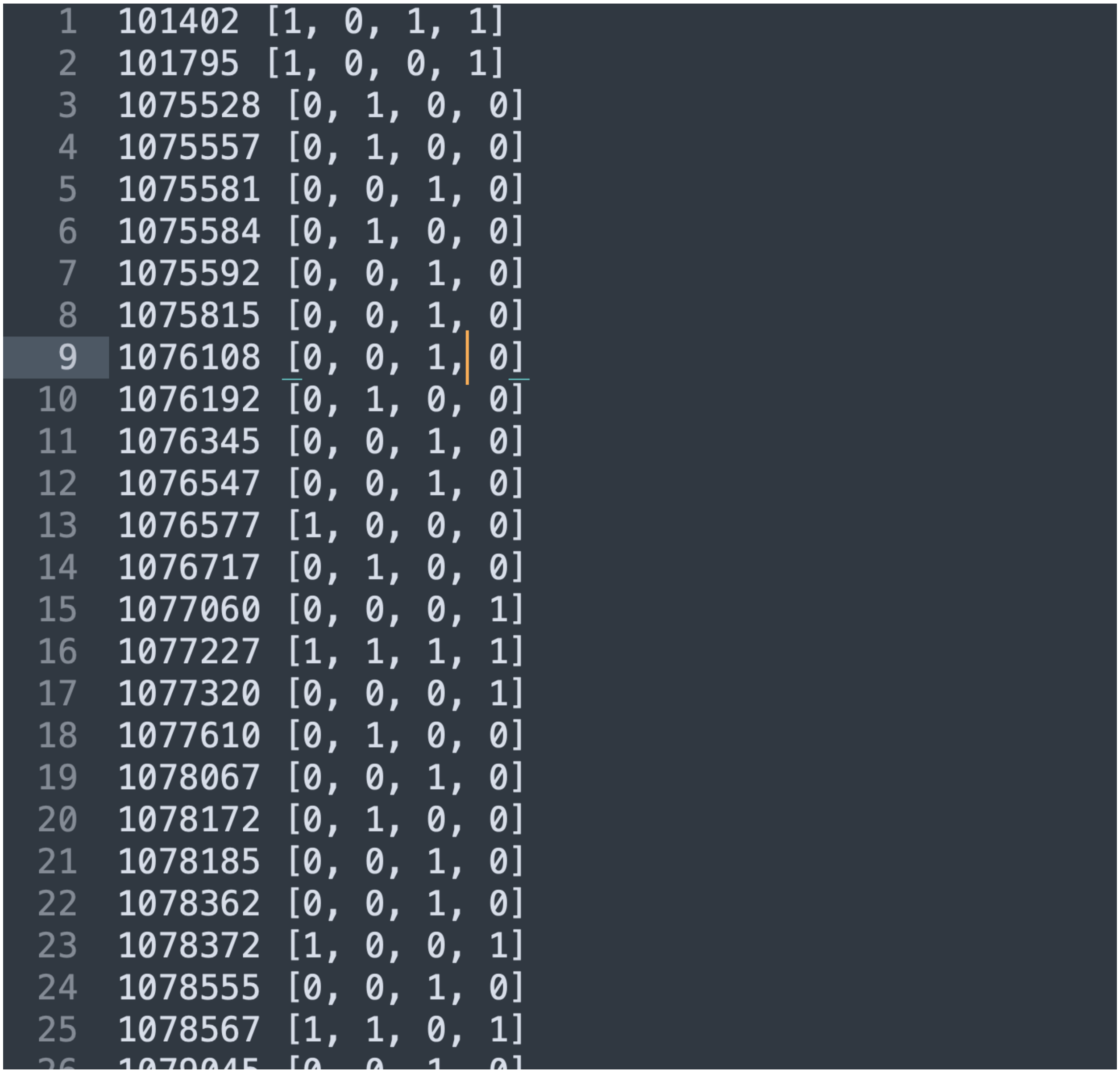


```
TCGA-BH-A0C0    TCGA-AC-A2FM    TCGA-E2-A56Z    TCGA-A7-A26E    TCGA-BH-A0EI    TCGA-EW-A2FV
    TCGA-B6-A40C
ENSG00000242268.2   0.0 0.0 0.0291839658783 0.0174030438813 0.120617924537  0.0
0.018670916974
ENSG00000270112.3   0.00432719095184    0.0 0.0 0.00965891042074    0.0100416720447 0.0
0.00259064902316
ENSG00000167578.15  2.95267308936   3.66370377195   7.50510492587   2.01803698835
2.61462405787   3.86175404208   5.01364153307
ENSG00000273842.1   0.0 0.0 0.0 0.0 0.0 0.0 0.0
ENSG00000078237.5   4.10177049829   1.13553237688   1.96033612678   3.78900476062
2.73092068376   4.36434855619   3.97295866777
ENSG00000146083.10  10.9733166541   14.812269831    10.078419651    10.0290113692
6.26100794932   6.59076186926   14.478933251
ENSG00000225275.4   0.0 0.0 0.0 0.0 0.0 0.0 0.0
ENSG00000158486.12  0.0577237438451 0.522902937932  0.018005882691  0.0107373058171
0.00228980551929    0.016018610161  0.00797507750425
ENSG00000198242.12  215.533968898   138.414959651   158.486296285   94.3951206706
152.942759502   109.782412797   154.499391699
ENSG00000259883.1   0.0574377071351 0.128431396862  0.0 0.0801307467946 0.066644923285
0.155407527891  0.0515811324961
```

A screenshot of output

Code-2: # A code to find a match in multiple files (e.g. 4 files here) and output as matrix of 1 (present) and 0 (absent)

```
f1=open("uniq_IDS.txt").read().split("\n")
f2=open("list").read().split("\n")
array1=f1
array2=f2
dict={}
for i in range(0,len(array2),1):
    # print (i)
    for j in range (0,len(array1),1):
        # print ("j is " , j)
        # dict[array1[j]]=[0]
        if array1[j] in dict:
            # print ("exists")
            dict[array1[j]].append(0)
        else:
            dict[array1[j]]=[0]
        # print (dict)
        # for key in dict:
        #   print (key,dict[key])
for k in range(0,len(array2),1):
    # print ("#######",array2[k])
    f3=open(array2[k]).read().split("\n")
    array3=f3
    # print (array3)
    for l in range(0,len(array3),1):
        # print (array3[l])
        if array3[l] in dict:
            # print "exists"
            fetchvalue = dict.get(array3[l])
            # print ("value is " ,fetchvalue)
            fetchvalue[k]=1
            dict[array3[l]]=fetchvalue
            # print (dict)
# print dict.items()
for key in dict:
    print (key,dict[key])
```

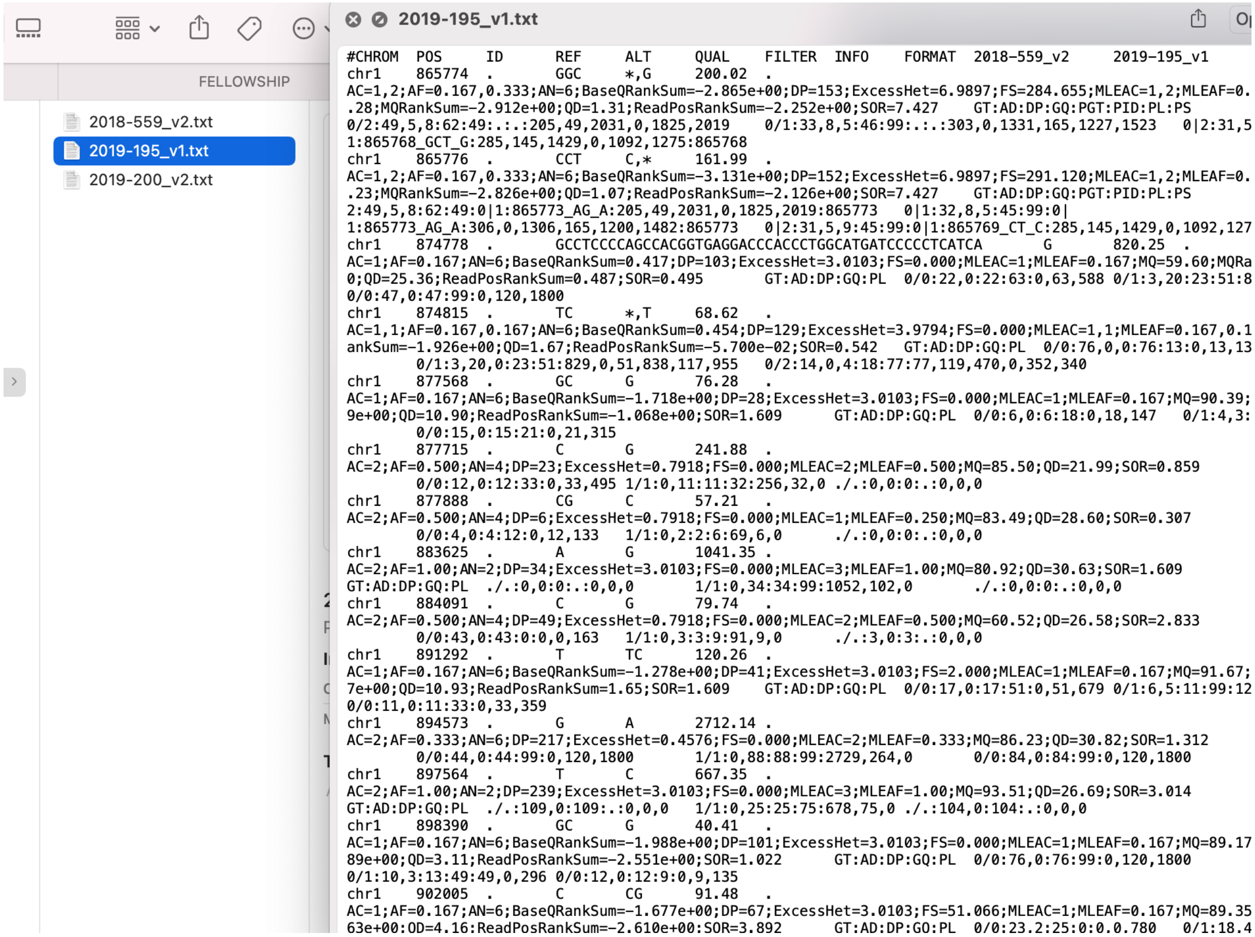


```
101402 [1, 0, 1, 1]
101795 [1, 0, 0, 1]
1075528 [0, 1, 0, 0]
1075557 [0, 1, 0, 0]
1075581 [0, 0, 1, 0]
1075584 [0, 1, 0, 0]
1075592 [0, 0, 1, 0]
1075815 [0, 0, 1, 0]
1076108 [0, 0, 1, 0]
1076192 [0, 1, 0, 0]
1076345 [0, 0, 1, 0]
1076547 [0, 0, 1, 0]
1076577 [1, 0, 0, 0]
1076717 [0, 1, 0, 0]
1077060 [0, 0, 0, 1]
1077227 [1, 1, 1, 1]
1077320 [0, 0, 0, 1]
1077610 [0, 1, 0, 0]
1078067 [0, 0, 1, 0]
1078172 [0, 1, 0, 0]
1078185 [0, 0, 1, 0]
1078362 [0, 0, 1, 0]
1078372 [1, 0, 0, 1]
1078555 [0, 0, 1, 0]
1078567 [1, 1, 0, 1]
1079045 [0, 0, 1, 0]
```

A screenshot of output

Code-3: # A code to print variants specific to a patient sample and variants common to all samples

```
f1=open("variants.vcf","r").read().split("\n")
f2=open("common.txt","w")
List=f1
List2=f2
colindex=[]
header=[]
header.append(List[0])
dict={}
for entity in header:
    splitheader = entity.split("\t")
    head = str("\t".join(header))
    f2.write (str(head))
    f2.write ("\n")
    for i in range(0,len(List),1):
        size=0
        colindex=[]
        CommonBoolean=True
        # print ("common-1 " , CommonBoolean)
        splitline = List[i].split("\t")
        for j in range(9,len(splitline),1):
            # print ("********",j,len(splitline))
            split1=splitline[j].split(":")
            sub = "1"
            if (split1[0].find(sub) == -1):##1 is not present
                CommonBoolean = False
            else:
                colindex.append(j)
                size = (len(colindex))
            if j == len(splitline)-1 and CommonBoolean == True:
                f2.write (List[i])
                f2.write ("\n")
            if size == 1 and j == len(splitline)-1 and CommonBoolean == False:
                # print ("SPECIFIC " , List[i])
                key = splitheader[colindex[0]]
                dict.setdefault(key,[]).append(List[i])
for Key in dict:
    value = str("\n".join(dict[Key]))
    f4=open("{0}.txt".format(Key),"w")
    f4.write(str(head))
    f4.write("\n")
    f4.write((value))
```

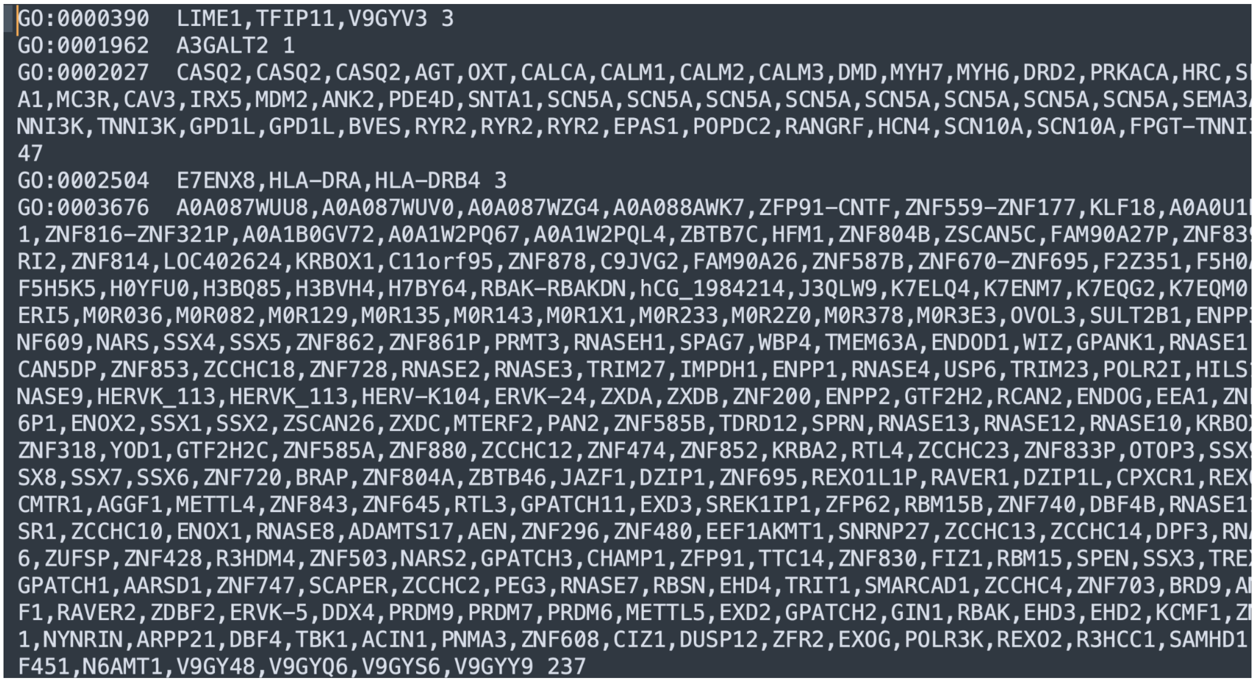


```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  2018-559_v2     2019-195_v1
chr1    865774  .       GGC     *,G     200.02  .
AC=1,2;AF=0.167,0.333;AN=6;BaseQRankSum=-2.865e+00;DP=153;ExcessHet=6.9897;FS=284.655;MLEAC=1,2;MLEAF=0.
.28;MQRankSum=-2.912e+00;QD=1.31;ReadPosRankSum=-2.252e+00;SOR=7.427    GT:AD:DP:GQ:PGT:PID:PL:PS
0/2:49,5,8:62:49:.:.:205,49,2031,0,1825,2019    0/1:33,8,5:46:99:.:.:303,0,1331,165,1227,1523   0|2:31,5
1:865768_GCT_G:285,145,1429,0,1092,1275:865768
chr1    865776  .       CCT     C,*     161.99  .
AC=1,2;AF=0.167,0.333;AN=6;BaseQRankSum=-3.131e+00;DP=152;ExcessHet=6.9897;FS=291.120;MLEAC=1,2;MLEAF=0.
.23;MQRankSum=-2.826e+00;QD=1.07;ReadPosRankSum=-2.126e+00;SOR=7.427    GT:AD:DP:GQ:PGT:PID:PL:PS
2:49,5,8:62:49:0|1:865773_AG_A:205,49,2031,0,1825,2019:865773   0|1:32,8,5:45:99:0|
1:865773_AG_A:306,0,1306,165,1200,1482:865773   0|2:31,5,9:45:99:0|1:865769_CT_C:285,145,1429,0,1092,127
chr1    874778  .       GCCTCCCCAGCCACGGTGAGGACCCACCCTGGCATGATCCCCCTCATCA       G       820.25  .
AC=1;AF=0.167;AN=6;BaseQRankSum=0.417;DP=103;ExcessHet=3.0103;FS=0.000;MLEAC=1;MLEAF=0.167;MQ=59.60;MQRa
0;QD=25.36;ReadPosRankSum=0.487;SOR=0.495       GT:AD:DP:GQ:PL  0/0:22,0:22:63:0,63,588 0/1:3,20:23:51:8
0/0:47,0:47:99:0,120,1800
chr1    874815  .       TC      *,T     68.62   .
AC=1,1;AF=0.167,0.167;AN=6;BaseQRankSum=0.454;DP=129;ExcessHet=3.9794;FS=0.000;MLEAC=1,1;MLEAF=0.167,0.1
ankSum=-1.926e+00;QD=1.67;ReadPosRankSum=-5.700e-02;SOR=0.542   GT:AD:DP:GQ:PL  0/0:76,0,0:76:13:0,13,13
        0/1:3,20,0:23:51:829,0,51,838,117,955   0/2:14,0,4:18:77:77,119,470,0,352,340
chr1    877568  .       GC      G       76.28   .
AC=1;AF=0.167;AN=6;BaseQRankSum=-1.718e+00;DP=28;ExcessHet=3.0103;FS=0.000;MLEAC=1;MLEAF=0.167;MQ=90.39;
9e+00;QD=10.90;ReadPosRankSum=-1.068e+00;SOR=1.609      GT:AD:DP:GQ:PL  0/0:6,0:6:18:0,18,147   0/1:4,3:
        0/0:15,0:15:21:0,21,315
chr1    877715  .       C       G       241.88  .
AC=2;AF=0.500;AN=4;DP=23;ExcessHet=0.7918;FS=0.000;MLEAC=2;MLEAF=0.500;MQ=85.50;QD=21.99;SOR=0.859
        0/0:12,0:12:33:0,33,495 1/1:0,11:11:32:256,32,0 ./.:0,0:0:.:0,0,0
chr1    877888  .       CG      C       57.21   .
AC=2;AF=0.500;AN=4;DP=6;ExcessHet=0.7918;FS=0.000;MLEAC=1;MLEAF=0.250;MQ=83.49;QD=28.60;SOR=0.307
        0/0:4,0:4:12:0,12,133   1/1:0,2:2:6:69,6,0      ./.:0,0:0:.:0,0,0
chr1    883625  .       A       G       1041.35 .
AC=2;AF=1.00;AN=2;DP=34;ExcessHet=3.0103;FS=0.000;MLEAC=3;MLEAF=1.00;MQ=80.92;QD=30.63;SOR=1.609
GT:AD:DP:GQ:PL  ./.:0,0:0:.:0,0,0       1/1:0,34:34:99:1052,102,0       ./.:0,0:0:.:0,0,0
chr1    884091  .       C       G       79.74   .
AC=2;AF=0.500;AN=4;DP=49;ExcessHet=0.7918;FS=0.000;MLEAC=2;MLEAF=0.500;MQ=60.52;QD=26.58;SOR=2.833
        0/0:43,0:43:0:0,0,163   1/1:0,3:3:9:91,9,0      ./.:3,0:3:.:0,0,0
chr1    891292  .       T       TC      120.26  .
AC=1;AF=0.167;AN=6;BaseQRankSum=-1.278e+00;DP=41;ExcessHet=3.0103;FS=2.000;MLEAC=1;MLEAF=0.167;MQ=91.67;
7e+00;QD=10.93;ReadPosRankSum=1.65;SOR=1.609    GT:AD:DP:GQ:PL  0/0:17,0:17:51:0,51,679 0/1:6,5:11:99:12
0/0:11,0:11:33:0,33,359
chr1    894573  .       G       A       2712.14 .
AC=2;AF=0.333;AN=6;DP=217;ExcessHet=0.4576;FS=0.000;MLEAC=2;MLEAF=0.333;MQ=86.23;QD=30.82;SOR=1.312
        0/0:44,0:44:99:0,120,1800       1/1:0,88:88:99:2729,264,0       0/0:84,0:84:99:0,120,1800
chr1    897564  .       T       C       667.35  .
AC=2;AF=1.00;AN=2;DP=239;ExcessHet=3.0103;FS=0.000;MLEAC=3;MLEAF=1.00;MQ=93.51;QD=26.69;SOR=3.014
GT:AD:DP:GQ:PL  ./.:109,0:109:.:0,0,0   1/1:0,25:25:75:678,75,0 ./.:104,0:104:.:0,0,0
chr1    898390  .       GC      G       40.41   .
AC=1;AF=0.167;AN=6;BaseQRankSum=-1.988e+00;DP=101;ExcessHet=3.0103;FS=0.000;MLEAC=1;MLEAF=0.167;MQ=89.17
89e+00;QD=3.11;ReadPosRankSum=-2.551e+00;SOR=1.022      GT:AD:DP:GQ:PL  0/0:76,0:76:99:0,120,1800
0/1:10,3:13:49:49,0,296 0/0:12,0:12:9:0,9,135
chr1    902005  .       C       CG      91.48   .
AC=1;AF=0.167;AN=6;BaseQRankSum=-1.677e+00;DP=67;ExcessHet=3.0103;FS=51.066;MLEAC=1;MLEAF=0.167;MQ=89.35
63e+00;QD=4.16;ReadPosRankSum=-2.610e+00;SOR=3.892      GT:AD:DP:GQ:PL  0/0:23,2:25:0,0,780     0/1:18,4
```

A screenshot of output

Code-4: # A code to convert GO IDS to gene symbols and count number of genes belong to each GO id

```
f1=open("goa_human.gaf").read().split("\n")
f2=open("ids2.txt").read().split("\n")
f4=open("out.txt","w")
array1=f1
array2=f2
dictt = {}
for i in range(0,len(array1),1):
        splitline = array1[i].split("\t")
        key = splitline[4]
        value = splitline[2]
        # print (value)
        if key in dictt:
                fetchvalue = dictt.get(key)
                value = fetchvalue + "," + value
                dictt[key] = value
        else:
                dictt[key] = value
for k in range(0,len(array2),1):
        count = 0
        if array2[k] in dictt:
                f4.write (array2[k])
                f4.write("\t")
                f4.write (dictt[array2[k]])
                f4.write ("\t")
                splitvalue = dictt[array2[k]].split(",")
                for j in range (0,len(splitvalue),1):
                        count += 1
                f4.write (str(count))
                f4.write("\n")
```



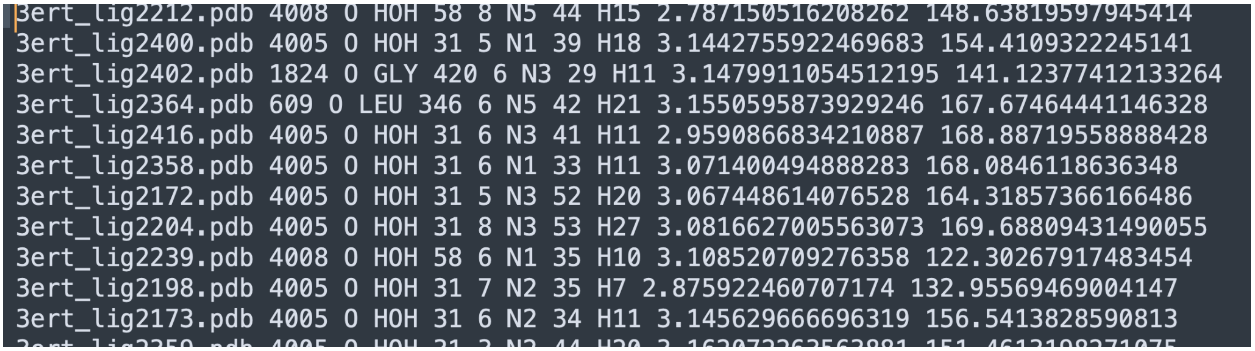
```
GO:0000390  LIME1,TFIP11,V9GYV3 3
GO:0001962  A3GALT2 1
GO:0002027  CASQ2,CASQ2,CASQ2,AGT,OXT,CALCA,CALM1,CALM2,CALM3,DMD,MYH7,MYH6,DRD2,PRKACA,HRC,S
A1,MC3R,CAV3,IRX5,MDM2,ANK2,PDE4D,SNTA1,SCN5A,SCN5A,SCN5A,SCN5A,SCN5A,SCN5A,SCN5A,SCN5A,SEMA3
NNI3K,TNNI3K,GPD1L,GPD1L,BVES,RYR2,RYR2,RYR2,EPAS1,POPDC2,RANGRF,HCN4,SCN10A,SCN10A,FPGT-TNNI
47
GO:0002504  E7ENX8,HLA-DRA,HLA-DRB4 3
GO:0003676  A0A087WUU8,A0A087WUV0,A0A087WZG4,A0A088AWK7,ZFP91-CNTF,ZNF559-ZNF177,KLF18,A0A0U1
1,ZNF816-ZNF321P,A0A1B0GV72,A0A1W2PQ67,A0A1W2PQL4,ZBTB7C,HFM1,ZNF804B,ZSCAN5C,FAM90A27P,ZNF83
RI2,ZNF814,LOC402624,KRBOX1,C11orf95,ZNF878,C9JVG2,FAM90A26,ZNF587B,ZNF670-ZNF695,F2Z351,F5H0
F5H5K5,H0YFU0,H3BQ85,H3BVH4,H7BY64,RBAK-RBAKDN,hCG_1984214,J3QLW9,K7ELQ4,K7ENM7,K7EQG2,K7EQM0
ERI5,M0R036,M0R082,M0R129,M0R135,M0R143,M0R1X1,M0R233,M0R2Z0,M0R378,M0R3E3,OVOL3,SULT2B1,ENPP
NF609,NARS,SSX4,SSX5,ZNF862,ZNF861P,PRMT3,RNASEH1,SPAG7,WBP4,TMEM63A,ENDOD1,WIZ,GPANK1,RNASE1
CAN5DP,ZNF853,ZCCHC18,ZNF728,RNASE2,RNASE3,TRIM27,IMPDH1,ENPP1,RNASE4,USP6,TRIM23,POLR2I,HILS
NASE9,HERVK_113,HERVK_113,HERV-K104,ERVK-24,ZXDA,ZXDB,ZNF200,ENPP2,GTF2H2,RCAN2,ENDOG,EEA1,ZN
6P1,ENOX2,SSX1,SSX2,ZSCAN26,ZXDC,MTERF2,PAN2,ZNF585B,TDRD12,SPRN,RNASE13,RNASE12,RNASE10,KRBO
ZNF318,YOD1,GTF2H2C,ZNF585A,ZNF880,ZCCHC12,ZNF474,ZNF852,KRBA2,RTL4,ZCCHC23,ZNF833P,OTOP3,SSX
SX8,SSX7,SSX6,ZNF720,BRAP,ZNF804A,ZBTB46,JAZF1,DZIP1,ZNF695,REXO1L1P,RAVER1,DZIP1L,CPXCR1,REX
CMTR1,AGGF1,METTL4,ZNF843,ZNF645,RTL3,GPATCH11,EXD3,SREK1IP1,ZFP62,RBM15B,ZNF740,DBF4B,RNASE1
SR1,ZCCHC10,ENOX1,RNASE8,ADAMTS17,AEN,ZNF296,ZNF480,EEF1AKMT1,SNRNP27,ZCCHC13,ZCCHC14,DPF3,RN
6,ZUFSP,ZNF428,R3HDM4,ZNF503,NARS2,GPATCH3,CHAMP1,ZFP91,TTC14,ZNF830,FIZ1,RBM15,SPEN,SSX3,TRE
GPATCH1,AARSD1,ZNF747,SCAPER,ZCCHC2,PEG3,RNASE7,RBSN,EHD4,TRIT1,SMARCAD1,ZCCHC4,ZNF703,BRD9,A
F1,RAVER2,ZDBF2,ERVK-5,DDX4,PRDM9,PRDM7,PRDM6,METTL5,EXD2,GPATCH2,GIN1,RBAK,EHD3,EHD2,KCMF1,Z
1,NYNRIN,ARPP21,DBF4,TBK1,ACIN1,PNMA3,ZNF608,CIZ1,DUSP12,ZFR2,EXOG,POLR3K,REXO2,R3HCC1,SAMHD1
F451,N6AMT1,V9GY48,V9GYQ6,V9GYS6,V9GYY9 237
```

A screenshot of output

Code-5: # A code to calculate H-bond distance and H-bond angle from a directory containing n number of files and a master file in another current working directory

```
import re,glob
import math
f1=open("Receptor_3ert1.pdb").read().split("\n")
path="/Users/apple/Desktop/Python-tasks/Python-Phd-programs/PL/All_ligands_3ert/*.pdb"
List=f1
N=[]
for file in glob.glob(path):
    # print (file)
    splitfilename = file.split("/")
    f2=open("{0}".format(file),"r").read().split("\n")
    List2=f2
    for line in List2:
        # print (line)
        if line.startswith("HET"):
            line = re.sub(" +", " ", line)
            splitline1 = line.split(" ")
            if splitline1[10] == "N":
                N.append(splitline1[5])
                N.append(splitline1[6])
                N.append(splitline1[7])
                N.append(splitline1[1])
                N.append(splitline1[2])
            if splitline1[10] == "H":
                x2=float(splitline1[5])
                y2=float(splitline1[6])
                z2=float(splitline1[7])
                h1=splitline1[1]
                h2=splitline1[2]
                for i in range (0,len(List),1):
                    # print ("i " , i )
                    List[i] = re.sub(" +", " ", List[i])
                    if List[i].startswith("ATOM") or List[i].startswith("HET"):
                        splitline2 = List[i].split(" ")
                        # if splitline2[0] == "ATOM":
                        if splitline2[11] == "O":
                            x3=float(splitline2[6])
                            y3=float(splitline2[7])
                            z3=float(splitline2[8])
                            for j in range(0,len(N),5):
                                x1= float(N[j])
                                y1 =float(N[j+1])
                                z1 = float(N[j+2])
```

```
                                                                        dist5 =
math.sqrt((float(x3)-float(x1))*(float(x3)-float(x1))+(float(y3)-float(y1))*(float(y3)-float(y1))+(float(z
3)-float(z1))*(float(z3)-float(z1)))
                                                                        if (dist5 <= 3.2):
                                                                                # print
(splitfilename[8],splitline2[1],splitline2[2],splitline2[3],splitline2[5],N[j+4],N[j+3],dist5)
                                                                                pass
                                                                                BA= (x1-x2, y1-y2, z1-z2)
                                                                                BC= (x3-x2, y3-y2, z3-z2)
                                                                                BABC=((x1-x2) * (x3-x2) +
(y1-y2) * (y3-y2) + (z1-z2) * (z3-z2))
                                                                                vBA= math.sqrt ((x1-x2) *
(x1-x2) + (y1-y2) * (y1-y2) + (z1-z2) * (z1-z2))
                                                                                vBC= math.sqrt ((x3-x2) *
(x3-x2) + (y3-y2) * (y3-y2) + (z3-z2) * (z3-z2))
                                                                                cosangle = (BABC/(vBA
*vBC))
                                                                                angle = math.acos
(cosangle)
                                                                                deg = math.degrees(angle)
                                                                                        # print (deg)
                                                                                if((deg>=90) and
(deg<=180)):
                                                                                        print
(splitfilename[8],splitline2[1],splitline2[2],splitline2[3],splitline2[5],N[j+3],N[j+4],h1,h2,dist5,deg)
                                                                                        # print
("###########")
        N=[]
```



```
3ert_lig2212.pdb 4008 O HOH 58 8 N5 44 H15 2.787150516208262 148.63819597945414
3ert_lig2400.pdb 4005 O HOH 31 5 N1 39 H18 3.1442755922469683 154.4109322245141
3ert_lig2402.pdb 1824 O GLY 420 6 N3 29 H11 3.1479911054512195 141.12377412133264
3ert_lig2364.pdb 609 O LEU 346 6 N5 42 H21 3.1550595873929246 167.67464441146328
3ert_lig2416.pdb 4005 O HOH 31 6 N3 41 H11 2.9590866834210887 168.88719558888428
3ert_lig2358.pdb 4005 O HOH 31 6 N1 33 H11 3.071400494888283 168.0846118636348
3ert_lig2172.pdb 4005 O HOH 31 5 N3 52 H20 3.067448614076528 164.31857366166486
3ert_lig2204.pdb 4005 O HOH 31 8 N3 53 H27 3.0816627005563073 169.68809431490055
3ert_lig2239.pdb 4008 O HOH 58 6 N1 35 H10 3.108520709276358 122.30267917483454
3ert_lig2198.pdb 4005 O HOH 31 7 N2 35 H7 2.875922460707174 132.95569469004147
3ert_lig2173.pdb 4005 O HOH 31 6 N2 34 H11 3.145629666696319 156.5413828590813
```

A screenshot of output

Code-6: #A code to convert file containing multiple molecules into files of each molecule

```
f1=open("10000.txt").read().split("\n")
array1=f1
for i in range (0,len(array1),1):
    print ("*******************************")
    if array1[i] == "$$$$":
        # print (array1[i])
        a=i+1
        f4 = open( "{0}.txt".format(array1[a]),"w")
        for j in range (a,a+300,1):
            print (j)
            # print (array1[j])
            # if array1[j] == "M  END":
            #       break
            # f4 = open( "{0}.txt".format(array1[a]),"w")
            f4.write(array1[j])
            f4.write("\n")
            if array1[j] == "M  END":
                break
```

Code-7: #A code to replace string (****) in .mol files to the molecule name

```
import glob
path = '/Users/apple/Desktop/Python-tasks/Python-Phd-programs/molfiles/*.mol2'
for filename in glob.glob(path):
    f1 = open("{0}".format(filename),'r').read().split("\n")
    List = f1
    f2 = open( "{0}".format(filename),"w")
    for line in List:
        if line == "****":
            split = filename.split("/")
            split2=split[7].split(".")
            newline = line.replace("****",split2[0])
            f2.write(newline)
            f2.write("\n")
        else:
            f2.write(line)
            f2.write("\n")
```

Code-8: #A code to convert molar mass to molecular formula

```
f1=open("formula1.txt").read().split("\n")
f2=open("mol-mass.txt").read().split("\n")
List=f1;List2=f2
dict={}
H=1;O=16;Fe=55;C=12;Ca=40;Zn=65;He=4
list1=[]
for i in range (0,len(List),1):
    line = list(List[i])
    # print ("*******************", i, List[i])
    for j in range (0,len(line),1):
        value = 1
        # print ("######")
        if line[j].isalpha() == True:
            key = line[j]
            dict[key]=1
            if (line[j].islower()) == True:
                rem_list = key,line[j-1]
                [dict.pop(key) for key in rem_list]
                # print ("updated1 is " , dict)
                key1 =  line[j-1] + key
                key = key.replace(key,key1)
                dict[key] = value
                # print ("updated2 is " , dict)
        if line[j].isnumeric() == True:
            value = line[j]
            fetchvalue = dict.get(key)
            value = line[j]
            dict[key]=value
            # print ("my dict is " , dict)
    for k in range(i,i+1,1):
        list1.append(dict)
        dict={}
for ele in list1:
    # print ("eles " , type(ele), ele)

molweightH=0;molweightHe=0;molweightO=0;molweightC=0;molweightCa=0;molweightZn=0;molweightFe=0
    for key in ele:
        # print (key,ele[key])
        if key == "O":
            molweightO=O*int(ele[key])
        if key == "He":
            molweightHe=He*int(ele[key])
        if key == "H":
            molweightH=H*int(ele[key])
        if key == "Fe":
            molweightFe=Fe*int(ele[key])
        if key == "Ca":
            molweightCa=Ca*int(ele[key])
        if key == "C":
            molweightC=C*int(ele[key])
        if key == "Zn":
            molweightZn=Zn*int(ele[key])

totalwt=molweightH+molweightHe+molweightO+molweightC+molweightCa+molweightZn+molw
eightFe
    print ("totalwt of ", ele, "is " , totalwt)
```

Code-9: #A code to replace three letter amino acid name (vcf file) to one letter code, e.g Tyr222Cys to T222C

```
import re
f1=open("1.txt").read().split("\n")
array=f1
for i in range (0,len(array),1):
	# print (i)
	splitline=array[i].split("\t")
	res = re.split('(\d+)', splitline[1])
	print (splitline[0],end="\t")
	for j in range (0,len(res),1):
		res1 = res[j].replace('Tyr', 'T').replace('Cys', 'C').replace('Gly', 'G').replace('Ala', 'A').replace('Val', 'V').replace('Leu', 'L').replace('Lys', 'K')
		print (res1,end="")
	print ()
```

Code 10: # A code to get average of multiple columns in a file using two methods as:
#1. pandas python package
#2 .using dict without help of any python package

```
##Method 1:
import pandas as pd
data = pd.read_csv('average.csv')
cols = ['A', 'B']
avg = data[cols].mean()
print(avg)

#Method 2:
f1=open("average.txt").read().split("\n")
List=f1
Sum=0
dict={}
for i in range (0,len(List),1):
    split = List[i].split("\t")
    for j in range (0,len(split),1):
        if j in dict:
            value = int(dict[j]) + int(split[j])
            dict[j]=value
        else:
            dict[j]=split[j]
        print (dict)
```