Convert bed to vcf.

Coronavirus_2_isolate_genome.fasta

used for reference in deletions

```
In [9]: fh=open("Coronavirus_2_isolate_genome.fasta", 'r')
fn=fh.readlines()
corona_genome=''
for line in fn:
    if line.startswith(">"):
        continue
    else:
        corona_genome+=line.strip()
#print(corona_genome)
```

Open UK strain .bed file.

(can be any .bed file)

```
import pandas as pd
file_name="P.1.bed"
save_file= file_name.strip(".bed")
bed_file=open(file_name, "r")
bed_file_list=bed_file.readlines()
```

P.1

Create a list called new_file

- 1) gets rid of "\t" Character
- 2) finds deletions and makes reference them in the list
- 3) appends to new_file which is the base for pandas dataframe

```
new file=[]
print("Indels found: ")
for line in bed_file_list:
    df_list=line.split("\t")
    ID=df_list[len(df_list)-1].strip()
    deletion_start=int(df_list[len(df_list)-3].strip())-1
    deletion_end=int(df_list[len(df_list)-2].strip())
    #print(deletion_start,deletion_end)
    if ID.startswith("d") or ID.startswith("i"):
        #Sets pos equal to pos-1
        #to follow the sample vcf file.
        b=line.split()
        b[2] = b[1]
        b='\t'.join(b)
        #print(b)
        original=b.strip()
        original+="\t" +corona_genome[deletion_start:deletion_end]+"\t"+corona_genome
        new_file.append(original)
        print(" •",ID[len(ID)-len(ID):] ,": ",corona_genome[deletion_start:deletion_er
    else:
        original=line.strip()
        original += "\t" + ID[len(ID) - len(ID)] + "\t" + ID[len(ID) - 1]
        new_file.append(original)
```

```
Create Pandas Dataframe
```

GTCTGGTTTT

Indels found:
 del_11288 :

• ins 28263 :

```
df.drop('POS-1', inplace=True, axis=1)
print(df)
                POS
                                        REF ALT
        #CHROM
                             ΙD
0
   NC 045512v2
                  241
                          C241T
                                             Τ
                                         C
   NC
      045512v2
                  733
                          T733C
   NC 045512v2
               2749
2
                         C2749T
                                             Т
   NC 045512v2
3
               3037
                         C3037T
   NC 045512v2 3828
                        C3828T
5
   NC 045512v2 5648
                        A5648C
   NC_045512v2 6319
6
                        A6319G
   NC_045512v2
7
               6613
                         A6613G
                                         Α
   NC_045512v2 11287 del_11288 GTCTGGTTTT
8
                                             G
                      C12778T
9
      045512v2
               12778
10 NC 045512v2
                13860
                        C13860T
                                          C
                                             Т
11 NC 045512v2
               14408
                       C14408T
                                         C
12 NC 045512v2
               17259
                       G17259T
```

df = pd.DataFrame([sub.split("\t") for sub in new file],index=None,columns=["#CHROM",

```
13 NC 045512v2
               21614
                       C21614T
14 NC_045512v2 21621
                       C21621A
15 NC_045512v2 21638
                       C21638T
                                         C
                       G21974T
16
   NC_045512v2
               21974
                                         G
17
      045512v2
                22132
   NC
                        G22132T
      __045512v2 22812
18
   NC
                        A22812C
                                            C
19 NC 045512v2 23012
                       G23012A
                                         G
20 NC 045512v2 23063
                       A23063T
21 NC 045512v2 23403
                       A23403G
22 NC_045512v2 23525
                       C23525T
                                        C
                                            Т
23 NC_045512v2 24642
                       C24642T
                                        C
                                            Т
24
   NC_045512v2
               25088
                        G25088T
                                         G
                                            Τ
25
      045512v2
                        T26149C
                                         Τ
                26149
      _045512v2
   NC
26
                28167
                        G28167A
                                        G
   NC 045512v2
27
               28262 ins_28263
                                        GΑ
                                            G
28 NC 045512v2
               28512
                      C28512G
29 NC 045512v2
               28877
                       A28877T
30 NC 045512v2 28878
                       G28878C
31 NC_045512v2 28881
                       G28881A
32
   NC_045512v2
                28882
                        G28882A
                                        G
   NC 045512v2 28883
                        G28883C
Sample UK VCF
```

##fileformat=VCFv4.2 ##reference=NC_045512v2

#CHROM	POS	ID	REF	ALT	QUAL	FILTER
NC_045512v2	241	C241T	С	Т		
NC_045512v2	913	C913T	С	T		
NC_045512v2	1059	C1059T	С	Т		
NC_045512v2	3037	C3037T	С	T		
NC_045512v2	3267	C3267T	С	T		
NC_045512v2	4345	C4345T	С	Т		
NC_045512v2	5388	C5388A	С	A		
NC_045512v2	5986	C5986T	С	Т		
NC_045512v2	6896	C6896T	С	Т		
NC_045512v2	6954	T6954C	T	С		

```
Write file with Pandas to VCF

df.to csv(save_file+".vcf", sep="\t", header=True, index=False, )
```

Write File from string (NOT USED)

method not used, writes file from string

```
In [84]:
    df_no_indices = df.to_string(index=False)
    #print(df_no_indices)
    test=open("WHAT.vcf", "w+")
    test.write(df_no_indices)
    test.close()
    '''
```

Out[84]: '\ndf_no_indices = df.to_string(index=False)\n#print(df_no_indices)\ntest=open("WHAT.v cf", "w+")\ntest.write(df no indices)\ntest.close()\n'