

1.FASTA Dosya Formatı

FASTA dosya formatı biyoinformatik biliminde yaygın kullanılan dosya türlerinden biridir. Bu dosya formatının içerisinde protein veya nükleotid sekanslarına metin olarak yer verilmektedir.

FASTA dosya formatı aslında bir yazılım paketine ait bir formattır. Fakat biyoinformatik alanlarında yaygın hale gelip python gibi dillerle kolayca işlenebilmektedir.

1.2 FASTA Dosya Formatının Yapısı

FASTA dosyaların yapı olarak iki kısımdan meydana gelir.

İlk kısım “>” sembolü ile başlar ve dosya yapısı gereği sembolün kullanılması bir zorunluluktur. Bu sembol sonrası dosyadaki veriler ile ilgili, isteğe bağlı olarak ve herhangi bir kısıtlama olmadığından istenilen şekilde , açıklama eklenebilir. Burada kısıtlama olmasa da NCBI FASTA kuralları olarak bilinen kurallara uygun açıklama eklemek ve NCBI tanımlama ifadelerini kullanmak, dosyanın çalışacağı uygulamada veya arama için kullanılacağı veri tabanında sorun yaşanmasının önüne geçecektir.

Diğer kısımda ise protein veya nükleotid sekansları yer alır. Bu sekanslar genellikle uzun olduklarından her 60 ila 100 karakterlik sekans yazımı sonrası alt satıra geçilerek devam edilir.

FASTA dosya formatında birden fazla sekansın tek dosya içerisinde yer almasına izin verilir.

```
>JX163924.1 SARS coronavirus isolate Tor2/FP1-10851, complete genome
CGATCTCTTGTAGATCTGTTCTCTAAACGAACTTTAAATCTGTGTAGCTGTCGCTCGGCTGCATGCCTA
GTGCACCTACGCAGTATAACAATAATAATTTTACTGTCGTTGACAAGAAACGAGTAACTCGTCCCTCT
TCTGCAGACTGCTTACGGTTTCGTCCGTGTTGCAGTCGATCATCAGCATACCTAGGTTTCGTCCGGGTGT
GACCGAAAGGTAAGATGGAGAGCCTTGTCTTGGTGTCAACGAGAAAACACACGTCCTCAACTCAGTTTGCC
TGTCTTCAGGTTAGAGACGTGCTAGTGCGTGGCTTCGGGGACTCTGTGGAAGAGGCCCTATCGGAGGCA
CGTGAACACCTCAAAAATGGCACTTGTGGTCTAGTAGAGCTGAAAAAGGCGTACTGCCCCAGCTTGAAC
AGCCCTATGTGTTTCAATAACGTTCTGATGCCCTAAGCACCAATCACGGCCACAAGGTCGTTGAGCTGGT
TGCAGAAATGGACGGCATTTCAGTACGGTCTAGCGGTATAACACTGGGAGTACTCGTGCCACATGTGGGC
GAAACCCCAATTGCATACCGCAATGTTCTTCTTCGTAAGAACGGTAATAAGGGAGCCGGTGGTCATAGCT
ATGGCATCGATCTAAAGTCTTATGACTTAGGTGACGAGCTTGGCACTGATCCCATGAAGATTATGAACA
```

Şekil 1. FASTA dosyası

FASTA dosya formatının uzantısında .fasta, .fna, .ffn, .faa, .frn uzantıları kullanılır.

1.3 FASTA Dosyalarının Kullanım Amaçları

FASTA dosyaları DNA veya protein sekanslarını hizalama yöntemleri ile ortak ve farklı noktaları bulmak adına yapılan karşılaştırma işlemlerinde yaygın olarak kullanılır. Bunun yanı sıra günümüzde pek çok veri tabanında DNA veya protein araması yapılırken kullanılır.

1.4 Projede Kullanılan FASTA Dosyaları

1.4.1 Projenin FASTA Dosyalarının İndirilme Aşaması

1.4.1.1 Canlıya Ait Tüm Sekansın İndirilmesi

GenBank

SARS coronavirus isolate Tor2/FP1-10851, complete genome

GenBank: JX163924.1

[FASTA](#) [Graphics](#)

Go to:

LOCUS JX163924 29646 bp ss-RNA linear VRL 26-SEP-2012

DEFINITION SARS coronavirus isolate Tor2/FP1-10851, complete genome.

ACCESSION JX163924

VERSION JX163924.1

DBLINK BioProject: [PRJNA260063](#)

KEYWORDS .

SOURCE Severe acute respiratory syndrome-related coronavirus

ORGANISM [Severe acute respiratory syndrome-related coronavirus](#)

REFERENCE 1 (bases 1 to 29646)

AUTHORS Town,C.D., Halpin,R.A., Ransier,A., Fedorova,N., Overton,L., Tsitrin,T., Stockwell,T., Amedeo,P., Bishop,B., Chen,H., Edworthy,P., Gupta,N., Katzel,D., Li,K., Schobel,S., Shrivastava,S., Thovarai,V., Wang,S., Powers,N., Grimes,S., Yamamoto,B., Robb,A., Wentworth,D.E. and Noah,D.L.

TITLE Direct Submission

JOURNAL Submitted (25-MAY-2012) J. Craig Venter Institute, 9704 Medical Center Drive, San Diego, CA 92161-5082

Send to:

☒ Complete Record

☐ Coding Sequences

☐ Gene Features

Choose Destination

☒ File

☐ Clipboard

☐ Collections

☐ Analysis Tool

Download 1 item.

Format

GenBank

Summary

GenBank

FASTA

ASN.1

XML

INSDSeq XML

TinySeq XML

Feature Table

Accession List

GI List

GFF3

Remove coronavirus sequence

Related BioProject

Şekil 2. Pubmed veri tabanı aranan canlı sekanslarını veren örnek bir sayfa

Proje ödevinde istenilen SARS coronavirus ve nCovid(2019) virüslerinin nükleotid dosyalarını NCBI sitesindeki Pubmed biyomedikal veri tabanı üzerinde aranıp elde edilmiştir.

İstenilen canlıya ait “complete genom” olan bir doküman seçilir. Şekil 2 ‘deki gibi bir sayfa açılır.

Dosyayı indirmek için sayfada sağ üstte bulunan “Send To” kısmı seçilir. Ardından açılan dosyadan “file” seçilip format bölümünden de FASTA dosyası seçilir ve “create file” butonuna basılarak dosya indirilir.

1.4.1.2 Canlıda Bulunan Özel Bir Proteinin İndirilmesi

Şekil 1’de yer alan sayfa içerisinde canlıya ait proteinler ve bilgilerinin bulunduğu sayfaya ait link bulunmaktadır. Bu proteinlere ait sekansları indirirken

Şekil 2’deki gibi ilgili gene ait linke tıklanır ve Şekil 3’deki proteine ait sayfaya geçilir.

```

gene      225..21445
          /gene="PP1ab"
CDS       join(225..13358,13358..21445)
          /gene="PP1ab"
          /ribosomal_slippage
          /codon_start=1
          /product="polyprotein_orf1ab"
          /protein_id="AFR58685.1"
          /translation="MESLVLGVINETHVQLSLPVLQVRDVLVRGFGDSVEEALSEARE"







```

Şekil 3. Canlıda bulunan protein linki

Bu işlem arkasından açılan sayfada yine 1.4.1.1 başlığı altındaki adımlar takip edilir.

1.5 Projede Kullanılacak FASTA Dosya Listesi

Projede SARS ve nCovid(2019) virüslerine ait virüsün tüm nükleotitlerinin, her iki virüste de bulunan “nucleocapsid” proteinlerinin , “polyprotein” proteinlerinin FASTA dosyaları projede girdi olarak kullanılmıştır.

	COVID.fasta	5.03.2020 14:07	FASTA Dosyası	30 KB
	COVID_Nucleocapsid.fasta	5.03.2020 14:09	FASTA Dosyası	1 KB
	COVID_Polyprotein.fasta	5.03.2020 14:08	FASTA Dosyası	8 KB
	SARS.fasta	5.03.2020 14:06	FASTA Dosyası	30 KB
	SARS_Nucleocapsid.fasta	5.03.2020 14:12	FASTA Dosyası	1 KB
	SARS_Polyprotein.fasta	5.03.2020 14:10	FASTA Dosyası	8 KB

Şekil 4. Girdi olarak kullanılan FASTA dosyaları

1.6 Proje Modülleri , Açıklamaları ve Çıktıları

Projede yapılan temel modüller ; DNA sekanslarına ait amino asit sekansları 1gram, 2gram, 3gram olarak üretmek , ORF bölgelerini nükleotid ve protein bazında çıkarılması , Needleman Wunsch algoritması kullanılarak “nucleocapsid protein” ve “polyprotein” Blosum62 matrisi baz alınarak hizalamalarının yapılması ve iki virüse ait proteinlerin farklı olan kısımlarının başlangıç konumları belirtilerek çıkartılması.

1.6.1 FASTA_DOSYA Sınıfı

FASTA dosyalarını işlemek adına oluşturulan sınıftır. Bu sınıf içerisinde sınıf yapıcısı (constructor) , genomDosyaDonusturucu , proteinDosyaDonusturucu , dosyaOkuma , donusturucu , dosyaYazma, orfBulAminoAsit , orfBulNukleotit fonksiyonları bulunmaktadır.

1.6.1.1 FASTA_DOSYASI Sınıfı Yapıcısı (Constructor)

```
class FASTA_DOSYASI():
    def __init__(self, dosya_adi, dosya_yolu):
        self.dosya_adi = dosya_adi
        self.dosya_yolu = dosya_yolu
```

Şekil 5. FASTA_DOSYASI sınıfının yapıcısı

FASTA dosyasının işlenmesi için oluşturulan sınıftan nesne oluştururken kullanılan fonksiyondur. Bu fonksiyon parametreleri FASTA dosyasının adı , dosyanın yoludur.

1.6.1.2 Program Akışı ve Kullanılan Fonksiyonlar

```
def main():
    fasta_sars_nucleocapsid = FASTA_DOSYASI("SARS_Nucleocapsid", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\SARS_Nucleocapsid.fasta")
    fasta_sars_nucleocapsid.proteinDosyaDonusturucu()

    fasta_sars_polyprotein = FASTA_DOSYASI("SARS_Polyprotein", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\SARS_Polyprotein.fasta")
    fasta_sars_polyprotein.proteinDosyaDonusturucu()

    fasta_covid_nucleocapsid = FASTA_DOSYASI("COVID_Nucleocapsid", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\COVID_Nucleocapsid.fasta")
    fasta_covid_nucleocapsid.proteinDosyaDonusturucu()

    fasta_covid_polyprotein = FASTA_DOSYASI("COVID_Polyprotein", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\COVID_Polyprotein.fasta")
    fasta_covid_polyprotein.proteinDosyaDonusturucu()

    fasta_sars = FASTA_DOSYASI("SARS", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\SARS.fasta")
    fasta_sars.genomDosyaDonusturucu()

    fasta_covid = FASTA_DOSYASI("COVID", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\COVID.fasta")
    fasta_covid.genomDosyaDonusturucu()

    sars = FASTA_DOSYASI("SARS", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\SARS.txt")
    sars.dosyaOkuma()

    covid = FASTA_DOSYASI("COVID", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\COVID.txt")
    covid.dosyaOkuma()

    Needleman_Wunsch.farkBul("C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\COVID_Nucleocapsid.txt", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\SARS_Nucleocapsid.txt", "Nucleocapsid")
    Needleman_Wunsch.farkBul("C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\COVID_Polyprotein.txt", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\SARS_Polyprotein.txt", "Polyprotein")

    Needleman_Wunsch.needlemanWunsch("C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\COVID_Nucleocapsid.txt", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\SARS_Nucleocapsid.txt", "Nucleocapsid")
    Needleman_Wunsch.needlemanWunsch("C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\COVID_Polyprotein.txt", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\SARS_Polyprotein.txt", "Polyprotein")

if __name__ == "__main__":
    main()
```

Şekil 6. Main fonksiyonu

SARS , nCovid virüslerine ait “polyprotein” ve “nucleocapsid” protein FASTA dosyaları proteinDosyaDonusturucu fonksiyonu ile düzenleniyor.

SARS , nCovid virüslerinin tam sekans FASTA dosyaları ile pars edilerek düzenlenmek üzere genomDosyaDonusturucu fonksiyonuna gönderiliyor. Kod parçacıkları aşağıda verilmiştir.

```
def genomDosyaDonusturucu(self):
    fasta_dosyasi = open(self.dosya_yolu, "r")
    data = fasta_dosyasi.read()
    sekans_baslangic = data.find("genome")
    dosya = open('C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\'+
self.dosya_adi + '.txt', "w")
    dosya.write(data[sekans_baslangic+6:])

def proteinDosyaDonusturucu(self):
    fasta_dosyasi = open(self.dosya_yolu, "r")
    data = fasta_dosyasi.read()
    protein_dizilim_baslangic = data.find("[")
    dosya = open('C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\'+
self.dosya_adi + '.txt', "w")
    dosya.write(data[protein_dizilim_baslangic+1:])
```

Şekil 6 ‘ te görüldüğü gibi ilk önce FASTA_DOSYASI isimli sınıftan sars ve covid nesneleri oluşturulur. Ardından sınıf içerisinde tanımlanan Şekil 7 ‘da görünen dosya okuma fonksiyonu çağrılır.

FASTA dosyalarının okunup işlenirken dosya işlemleri yapılırken sorun oluşturacak olan boşlukların kaldırılmasını replace fonksiyonu ile sağlar ve okuma işlemi sonrası düzenlenen veriyi nükleotid bazlı ORF bulan Şekil 8’deki orfBulNükleotit fonksiyonuna gönderir.

```
def dosyaOkuma(self):
    dosya = open(self.dosya_yolu, "r")
    sekanslar = dosya.read()
    sekanslar = sekanslar.replace("\n", "")
    sekanslar = sekanslar.replace("\r", "")
    self.orfBulNukleotit(sekanslar)
    self.donusturucu(sekanslar)
```

Şekil 7. FASTA_DOSYASI sınıfına ait dosya okuma fonksiyonu

orfBulNukleotit fonksiyonunda kodonları tutacak bir dizi oluşturulur. Ardından “ATG” başlangıcı kodonu ile bitiş kodonları arasında yer alan kodonları tutmak adına bir değişken içerisi boş olarak tanımlanır. Bu fonksiyon sonucu nükleotit bazlı ORF çıktı verilerinin kaydedileceği dosya yazmak üzere açılır.

Sonrasında bir for döngüsü ile parametre olarak alınan sekans kodonlara ayrılır ve diziye eklenir.

Diziye eklenen kodonlardan ORF olan yani başlangıç ve bitiş kodonları arasında bulunan sekanslar Şekil 8 ‘deki gibi döngü ve koşullar ile tespit edilerek tanımlanan değişken içerisine atılır.

Her bitiş kodonu görüldükten sonra yeni bir ORF başlayabileceğinden dosyada düzenli görülmesi için değişkene yeni satır kaçış karakteri eklenir.

Tüm bu işlemler sonucunda açılan dosya içerisine değişkende toplanan ORF verileri yazılır.

Projede bu fonksiyon kullanılarak nCovid ve SARS virüslerindeki nükleotit bazlı ORF verileri üretilmiştir.

```
def orfBulNukleotit(self, sekanslar):
    orf_sekans= []
    orf_gen= ""
    dosya = open('C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\'+ self.dosya_adi +'_GEN_ORF.txt', 'w')

    for i in range(0, len(sekanslar)-1, 3):
        if len(sekanslar) > i+3:
            kodon = sekanslar[i:i + 3]
            orf_sekans.append(kodon)

    for i in range(0, len(orf_sekans)-1, 1):
        if orf_sekans[i] == "ATG" and (i+1) < len(orf_sekans):
            i += 1
            while (orf_sekans[i] != "TAA") and (orf_sekans[i] != "TAG") and (orf_sekans[i] != "TGA"):
                orf_gen += orf_sekans[i]
                if (i + 1) < len(orf_sekans):
                    i+=1
                else:
                    break
            orf_gen += "\n\n\n"
    dosya.write(orf_gen)
```

Şekil 8 Nükleotit bazlı ORF bulan fonksiyon

nCovid virüsüne ait nükleotit bazlı ORF çıktı dosyasının resimleri *Şekil 9* ve *Şekil 10* 'da verilmiştir.

```

COVID_GEN_ORF.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
CTTAGTGCACTCACGCAGTATAAT

GAGAGCCTTGTCCCTGGTTTCAACGAGAAAAACACACGTCCTCAACTCAGTTTGCCTGTTTTACAGGTTGCGGACGTGCTCGTACGTGGCTTTGGAGACTCCGTGGAGG
TTTGTGGCACTGAGAATTTGACTAAAGAAAGTGCCACTACTTGTGGTTACTTACCCCAAAATGCTGTTGTTAAAAATTTATTGTCCAGCATGTCACAATTCAGAAGT
ATTTTGGCTTTGTGTGCTGACTCTATCATTATTGGTGGAGCTAAACTTAAAGCCTTGAATTTAGGTGAAACATTTGTACGCACTCAAGGGATTGTACAGAAAC
AATAGTTTGTAGTGGTTATTTAAAACTTACTGACAAATGTATACATTTAAAAATGCAGACATTTGTGGAAGAAGCTAAAAAGGTAACCAACAGTGGTTGTTAATGCAG
TTCTTGGAAGCTGTTTCTTGGAATTTGCGAGAAATGCTTGACATGCAGAAAGAACACGCAAAATTAATGCCGTGCTGTGTTGGAACATAAGCCATAGTTTCAACTAT
TAACTTTTGTGCACTTATCTTAGCCTACTGTAATAAGACAGTAGGTGAGTTAGGTGATGTTAGAGAAACAATGAGTTACTTGTTCACATGCCAATTTAGATTCT
GAAGTAGTGGAAAACTTACCATACAGAAAGACGTTCTTGAGTGTAATGTGAAAACTACCGAAGTTGTAGGAGACATTATACTTAAACCAGCAAAATAATAGTTTAA
ATTATGTATGAAAAAGTTATGTGCATGTTGTAGACGGTTGTAATTCATCAACTTGTATGATGTGTTACAAACGTAATAGAGCAACAAGAGTCGAATGTACAACAT
TGCTAAAAAGAATAACTTACCTTTTAAGTTGACATGTGCAACTACTAGACAAGTTGTTAATGTTGTAACAACAAGATAGCACTTAAGGGTGGTAAAAATTGTTAAT
TACAGTCATGTAGTTGCCTTTAACTTTACTATTCTTATGTCATTCACTGTACTCTGTTTAAACCAGTTTACTCATTCTTACCTGGTGTATTCTGTTATTT
TAGATTATGACTGTGCTCTTTTGTACATGCACCATATGGAATTACCAACTGGAGTTCATGCTGGCACAGACTTAGAAGGTAACCTTTTATGGACCTTTTGTGTA
AGGTATTGTTTTATGTGTGTTGAGTATTGCCCTATTTTCTTCATAACTGGTAATACACTTCAGTGTATAATGCTAGTTTATTGTTTCTTAGGCTATTTTGTACT
ATCCAACAGGTTGTAGATGCAGATAGTAAATTTGTTCAACTTAGTGAATTAGTATGGACAATTACCTAATTTAGCATGGCTCTTATTGTAACAGCTTTAAGG

GTTGAGCTGGTAGCAAACTCGAAGGCATTAGTACGGTCGTAGTGGTGAGACACTTGGTGTCTTGTCCCTCATGTGGGCGAAATACCAGTGGCTTACC GCAAGC
CTAGCGCTAACATAGGTTGTAACCATACAGGTGTTGTTGGAGAAGGTTCCGAAGGCTTAAATGACAACCTTCTTGAATACTC CAAAAAGAGAAAGTCAACATCA
ACCTACTAGTGAAGCTGTTGAAGCTCCATTGGTTGGTACACCAAGTTTGTATTAACGGGCTTATGTTGCTCGAAATCAAGACACAGAAAAAGTACTGTGCCCTTG
CTTGC TAAACACTGTCTTCATGTTGTGCGCCCAAATGTTAACAAAGGTGAAGACATTC AACTTCTTAAGAGTGCTTATGAAAATTTAATCAGCACGAAGTTCTAC
GCTATGTAACACATGGCTTAAATTTGGAAGAAGCTGCTCGGTATATGAGATCTCTCAAAGTGCCAGCTACAGTTTCTGTTTCTTACCTGATGCTGTTACAGCGTA
AGCTACAAAATATCTAGTACAACAGGAGTCACCTTTTGTATGATGTCAGCACCACTGCTCAGTATGAACCTAAGCATGGTACATTTACTTGTGCTAGTGAGTAC
CCTTGGGATACTATAGCTAATTATGCTAAGCCTTTTCTTAACAAAGTTGTTAGTACAACCTAACAATAGTTACACGGTGTTTAAACCGTGTTTGTACTAATTATA
TACAGTTTAAAGACCAATAAATCCTACTGACCAGTCTTCTTACATCGTTGATAGTGTACAGTGAAGAATGGTTCATCCATCTTTACTTTGATAAAGCTGGTCA
CACTCGTGACATAGCATCTACAGATACTTGTGTTGCTAACAAACATGCTGATTTTGACACATGGTTTAGCCAGCGTGGTGGTAGTTATACTAATGACAAAGCTTGC
AGTAATTACCTAAAGAGACGTGTAGTCTTAAATGGTGTTCCTTTAGTACTTTTGAAGAAGCTGCGCTGTGCACCTTTTGTGTTAAATAAAGAAATGTATCAAAGT
AGTACAATTATGAACCTCTAACACAAGACCATGTTGACATACTAGGACCTCTTCTGCTCAAAC TGGAAATGCGCTTTTAGATATGTGTGCTTCATTAAGAAT
CAACATTAATTTGTTGGGTGTTGGTGGCAACCTTGTATCAAAGTAGCCACTGTACAGTCTAAAATGTGAGATGTAAGTGACATCAGTAGCTTACTCTCAGTT
CTTGCACTGTTATCCGATTTACAGGATTTGAAATGGGCTAGATTCCTTAAGAGTGATGGAAC TGGTACTATCTATACAGAAGCTGGAACCACTTGTAGGTTTGTTA

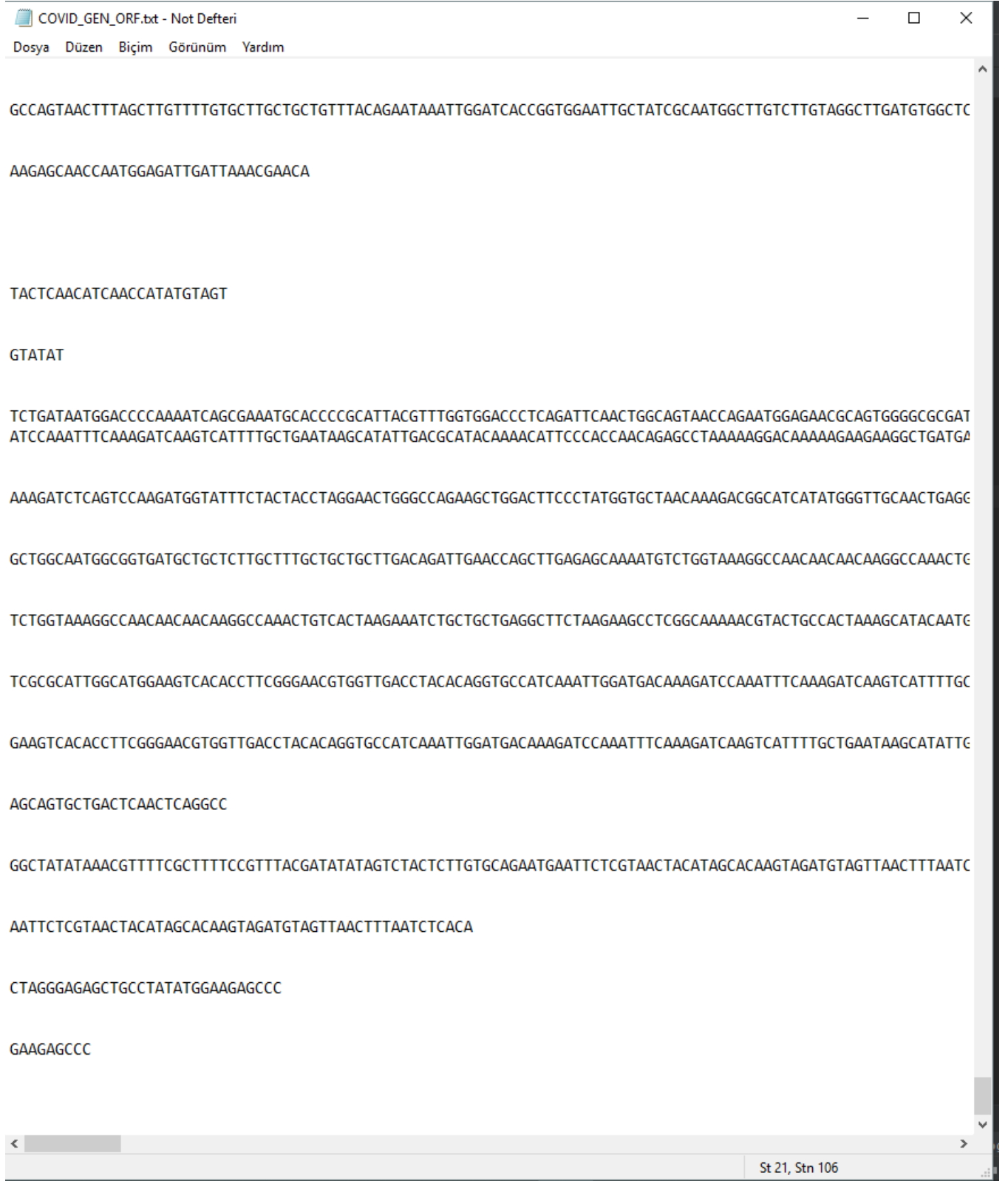
CGTGAGCTTAACGGAGGGGCATACACTCGCTATGTCGATAACAACCTCTGTGGCCCTGATGGCTACCCTCTTGAGTGCATTAAAGACCTCTAGCACGTGCTGGTA
CTAAAAAGGTGCTTGAATATTGGTGAACAGAAATCAATACTGAGTCTCTTTATGCATTTGCATCAGAGGCTGCTCGTGTGTACGATCAATTTTCTCCGCAC
CTCTGCCTATACAGTTGAAC TCGGTACAGAAGTAAATGAGTTGCTGTTGTGGCAGATGCTGTCATAAAAACTTTGCAACCAAGTATCTGAATTACTTACACCA
GTTGAACAAAAGATCGCTGAGATTCCTAAAGAGGAAGTTAAGCCATTTATAACTGAAAGTAAACCTTCAGTTGAACAGAGAAAAACAAGATGATAAGAAAAATCAAA
ATCCTACCACATTCACCTAGATGGTGAAGTTATCACCCTTGACAATCTTAAGACACTTCTTTCTTGGAGAGAAGTGAGGACTATTAAGGTGTTTACAACAGTAGA
TTATAAATGGATGGTGTGTTTGTACAGAAATTGACCCTAAGTTGGACAATTTATTAAGAAGAAAGCAATTCCTATTTCACAGAGCAACCAATTGATCTTGTACCA
TTTTCTAAACTGATAAATATTATAATTGTTTGTACTATTAAAGTGTGCTAGGTTCTTAAATCTACTCAACCGCTGCTTTAGGTGTTTAAATGTCTAATTTAG
TTATGTGTCAACCTATACTGTTACTAGATCAGGCATTAGTGTCTGATGTTGGTGATAGTGCGGAAGTTGCAGTTAAATGTTTGATGCTTACGTTAATACGTTTT
CACTGACTTTGCAACATCAGCTTGTGTTTGGCTGCTGAATGTACAATTTTAAAGATGCTCTGGTAAGCCAGTACCATATTGTTATGATACCAATGTACTAGAA
GATGTTCTTTACCAACCACCACAAACCTCTATCACCTCAGCTGTTTTCAGAGTGGTTTTAGAAAAATGGCATTCCTCATCTGGTAAAGTTGAGGGTTGTATGGTAC
TGACTTCACTTTTAGTTTATGTCAGAGTACTCAATGGCTTTGTTCTTTTTTTGATGAAAAATGCCTTTTTACCTTTTGTATGGGTATTATTGCTATGTCTG
GCTTTGTGAAGAAATGCTGGACAACAGGGCAACCTTACAAGCTATAGCCTCAGAGTTTAGTCCCTTCCATCATATGCAGCTTTTGTACTGCTCAAGAAGCTTAT
GCTTTTGTGCTAGATGCTGCTAAAGCTTACAAGATTATCTAGCTAGTGGGGACAACCAATCAATTTGTGTTAAGATGTTGTGTACACACACTGGTACTGGTC
GGTGAATTCGATCTGTCTATCCAGTTGCGTCACCAAATGAATGCAACCAAATGTGCCCTTCAACTCTCATGAAGTGTGATCATTGTGGTGAAACTTCATGGCAG
TAGAGTTTCTTAGAGACGGTTGGGAAATTTGTTAAATTTATCTCAACCTGTGCTTGTGAAATTTGTCGGTGGACAAATTTGTCACCTGTGCAAGGAAATTAAGGAG
AGATGATGATAGTCAACAACTGTTGGTCAACAAGACGGCAGTGAGGACAATCAGACAACCTACTATTCAACAATTTGTTGAGGTCAACCTCAATTAGAGATGGA
ACAGACAATTATATAACCACTTACCCGGGTGAGGTTTAAATGGTTACACTGTAGAGGAGGCAAGACAGTGTCTAAAAAGTGTAAGGTGCTTTTACATTTCTAC

```

St 21, Strn 106

Şekil 9 nCovid için nükleotit bazlı ORF dosyasının baş kısmı

FASTA SEKANS DOSYALARI



```
COVID_GEN_ORF.txt - Not Defteri
Dosya  Düzen  Biçim  Görünüm  Yardım

GCCAGTAACTTTAGCTTGTTTTGTGCTTGCTGCTGTTTACAGAATAAATTGGATCACCGTGGAATTGCTATCGCAATGGCTTGCTTTGAGGCTTGATGTGGCTC

AAGAGCAACCAATGGAGATTGATTAACGAACA

TACTCAACATCAACCATATGTAGT

GTATAT

TCTGATAATGGACCCCAAATCAGCGAAATGCACCCGCGATTACGTTTGGTGGACCCTCAGATTCAACTGGCAGTAACCAGAATGGAGAACGCAGTGGGGCGCGAT
ATCCAAATTTCAAAGATCAAGTCATTTTGCTGAATAAGCATATTGACGCATACAAAACATTCCCACCAACAGAGCCTAAAAAGGACAAAAAGAAGAAGGCTGATGA

AAAGATCTCAGTCCAAGATGGTATTTCTACTACCTAGGAACCTGGGCCAGAAGCTGGACTTCCCTATGGTGCTAACAAAGACGGCATCATATGGGTTGCAACTGAGG

GCTGGCAATGGCGGTGATGCTGCTCTTGCTTTGCTGCTGCTTGACAGATTGAACCAGCTTGAGAGCAAAATGTCTGGTAAAGGCCAACAACAACAAGGCCAAACTG

TCTGGTAAAGGCCAACAACAACAAGGCCAAACTGTCACTAAGAAATCTGCTGCTGAGGCTTCTAAGAAGCCTCGGCAAAAACGTACTGCCACTAAAGCATACAATG

TCGCGCATTGGCATGGAAGTCACACCTTCGGGAACGTGGTTGACCTACACAGGTGCCATCAAATTGGATGACAAAGATCCAAATTTCAAAGATCAAGTCATTTTGC

GAAGTCACACCTTCGGGAACGTGGTTGACCTACACAGGTGCCATCAAATTGGATGACAAAGATCCAAATTTCAAAGATCAAGTCATTTTGCTGAATAAGCATATTG

AGCAGTGCTGACTCAACTCAGGCC

GGCTATATAAACGTTTTTCGCTTTTCCGTTTACGATATATAGTCTACTCTTGTCAGAATGAATTCTCGTAACTACATAGCACAAAGTAGATGTAGTTAACTTTAATC

AATTCTCGTAACTACATAGCACAAAGTAGATGTAGTTAACTTTAATCTCACA

CTAGGGAGAGCTGCCTATATGGAAGAGCCC

GAAGAGCCC

St 21, Stn 106
```

Şekil 10 nCovid için nükleotid bazlı ORF dosyasının son kısmı

SARS virüsüne ait nükleotit bazlı ORF çıktı dosyasının resimleri *Şekil 11.1* ve *Şekil 11.2*'de verilmiştir.

```

SARS_GEN_ORF.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
CTAGTGCACCTACGCAGTATAACAA

GCACTTGTGGTC

TGTTCAATAACGTTCTGATGCCT

CCT

TGGGCGAAACCCCAATTGCATACCGCAATGTTCTTCTCGTAAGAACGGTAATAAGGGAGCCGGTGGTCATAGCTATGGCATCGATC

TTCTTCTCGTAAGAACGGTAATAAGGGAGCCGGTGGTCATAGCTATGGCATCGATC

GCATCGATC

ACT

AACAAACTGGAACACTAAGCATGGCAGTGGTGCACCTCCGTGAACCTACTCGTGAGCTCAATGGAGGTGCAGTCACTCGCTATGTCGACAACAATTTCTGTGGCCAGATC

GCAGTGGTGCACCTCCGTGAACCTACTCGTGAGCTCAATGGAGGTGCAGTCACTCGCTATGTCGACAACAATTTCTGTGGCCAGATGGGTACCTCTTGATTGCATCAAA

GAGGTGCAGTCACTCGCTATGTCGACAACAATTTCTGTGGCCAGATGGGTACCTCTTGATTGCATCAAGATTTTCTCGCACGCGGGCAAGTCAATGTGCACCTTT

TCGACAACAATTTCTGTGGCCAGATGGGTACCTCTTGATTGCATCAAGATTTTCTCGCACGCGGGCAAGTCAATGTGCACCTTTCCGAACAACCTTGATTACATCG

GGTACCTCTTGATTGCATCAAGATTTTCTCGCACGCGGGCAAGTCAATGTGCACCTTTCCGAACAACCTTGATTACATCGAGTCGAAGAGAGGTGTCTACTGCTGCC

AGCATGAAATTGCCTGGTTCACTGAGCGCTCTGATAAGAGCTACGAGCACCAGACACCTTCGAAATTAAGAGTGCCAAGAAATTTGACACTTCAAAGGGGAATGCCAA

AAATTGCCTGGTTCACTGAGCGCTCTGATAAGAGCTACGAGCACCAGACACCTTCGAAATTAAGAGTGCCAAGAAATTTGACACTTCAAAGGGGAATGCCAAAGTTTC

AAGTTTCATGGCAGACGTGCGACTTTC

CTG

TTGGCTGCTATAATAAGCGTGCCTACTGGGTTCTCGTGCTAGTGCTGATATTGGCTCAGGCCATACTGGCATTACTGGTGACAATGTGGAGACCT

```

St 1, Str 1

Şekil 11.1 SARS için nükleotid bazlı ORF dosyasının baş kısmı

```

SARS_GEN_ORF.txt - Not Defteri
Dosya  Düzen  Biçim  Görünüm  Yardım
GAGTTAGATTATCCA

AATGAGCTCACTTTAATTGACTTCTATTTGTGCTTTTTAGCCTTTCTGCTATTCCTTGTTTTAATAATGCTTATTATTTTGGTTTTCACTCGAAATCCAGGATCTAGAA
CTTATTATATTTTGGTTTTCACTCGAAATCCAGGATCTAGAAGAACCTTGACCAAAGTC

AAACTTCTCATTGTTTTGACTTGATTCTCTATGCAGTTGCATATGCACTGTAGTACAGCGCTGTGCATCTAATAAACCTCATGTGCTTGAAGATCCTTGAAGGTACA
GCACACTATGGTTCAAACATGCACACC

CACACC

GACCCCAATCAAACCAACGTAGTGCCCCCGCATTACATTTGGTGGACCCACAGATTCAACTGACAATAACCAGAATGGAGGACGCAATGGGGCAAGGCCAAAACAGCGCC
GAGGACGCAATGGGGCAAGGCCAAAACAGCGCCGACCCCAAGGTTTACCCAATAATACTGCGTCTTGTTTCACAGCTCTCACTCAGCATGGCAAGGAGGAACCTAGATTCC
GGGCAAGGCCAAAACAGCGCCGACCCCAAGGTTTACCCAATAATACTGCGTCTTGTTTCACAGCTCTCACTCAGCATGGCAAGGAGGAACCTAGATTCCCTCGAGGCCAGC
GCAAGGAGGAACCTAGATTCCCTCGAGGCCAGGGCGTTCCAATCAACACCAATAGTGGTCCAGATGACCAAATTGGCTACTACCGAAGAGCTACCCGACGAGTTCGTGGTG
ACCAAATTGGCTACTACCGAAGAGCTACCCGACGAGTTCGTGGTGGTGACGGCAAAA

CTGCCACCGTGCTACAATTCTCAAGGAACAACATTGCCAAAAGGCTTCTACGCAGAGGGAAGCAGAGGCGGCAGTCAAGCCTCTTCTCGCTCCTCATCAGTAGTCGCC
GAGCCATTAAATTGGATGACAAAGATCCACAATTCAAAGACAACGTCATACTGC

ACAAAGATCCACAATTCAAAGACAACGTCATACTGC

AAGCTCAGCCTTTGCCGCAGAGACAAAAGAAGCAGCCCACTG

ATTTCTCCAGACAACTTCAAAATTCCA

TGTAATAATTTTAGT

```

St 1, Stn 1

Şekil 11.2 SARS için nükleotid bazlı ORF dosyasının son kısmı

Bu çıktılar elde edildikten sonra fonksiyonun çağrıldığı dosyaOkuma fonksiyonuna geri dönlür.

Geri dönlüğünde sıradaki komut DNA sekanslarını aminoaside dönüştüren Şekil 12’ de verilen “donusturucu” isimli fonksiyonun çağrılmasıdır.

```
def donusturucu(self, sekanslar):
    amino_asit_tablosu = {
        'ATA': 'I', 'ATC': 'I', 'ATT': 'I', 'ATG': 'M',
        'ACA': 'T', 'ACC': 'T', 'ACG': 'T', 'ACT': 'T',
        'AAC': 'N', 'AAT': 'N', 'AAA': 'K', 'AAG': 'K',
        'AGC': 'S', 'AGT': 'S', 'AGA': 'R', 'AGG': 'R',
        'CTA': 'L', 'CTC': 'L', 'CTG': 'L', 'CTT': 'L',
        'CCA': 'P', 'CCC': 'P', 'CCG': 'P', 'CCT': 'P',
        'CAC': 'H', 'CAT': 'H', 'CAA': 'Q', 'CAG': 'Q',
        'CGA': 'R', 'CGC': 'R', 'CGG': 'R', 'CGT': 'R',
        'GTA': 'V', 'GTC': 'V', 'GTG': 'V', 'GTT': 'V',
        'GCA': 'A', 'GCC': 'A', 'GCG': 'A', 'GCT': 'A',
        'GAC': 'D', 'GAT': 'D', 'GAA': 'E', 'GAG': 'E',
        'GGA': 'G', 'GGC': 'G', 'GGG': 'G', 'GGT': 'G',
        'TCA': 'S', 'TCC': 'S', 'TCG': 'S', 'TCT': 'S',
        'TTC': 'F', 'TTT': 'F', 'TTA': 'L', 'TTG': 'L',
        'TAC': 'Y', 'TAT': 'Y', 'TAA': '-', 'TAG': '-',
        'TGC': 'C', 'TGT': 'C', 'TGA': '-', 'TGG': 'W',
    }

    protein = ""
    for i in range(0, len(sekanslar), 3):
        if len(sekanslar) > i+3:
            kodon = sekanslar[i:i + 3]
            protein += amino_asit_tablosu[kodon]
    self.orfBulAminoAsit(protein)
    protein += "\n\n\n"
    for j in range(1, len(sekanslar) - 2, 3):
        if len(sekanslar) > (j + 3):
            kodon = sekanslar[j:j + 3]
            protein += amino_asit_tablosu[kodon]
    protein += "\n\n\n"
    for k in range(2, len(sekanslar) - 1, 3):
        if len(sekanslar) > (k + 3):
            kodon = sekanslar[k:k + 3]
            protein += amino_asit_tablosu[kodon]
    self.dosyaYazma(protein)
```

Şekil 12 DNA’dan amino asit üretimini yapan fonksiyon

Bu fonksiyonda dictionary veri türünde kodonlara karşılık gelen amino asit alfabesi tanımlandı. Ardından verilen sekansta yer alan kodonların üzerinden çıkarılan amino asitlerin dosyaya yazılmak üzere birleştirilebilmesi adına bir boş değişken tanımlandı. Ardından 3 for döngüsü ile kodonlar çıkarılıp amino asit alfabesindeki karşılıklarına göre sekans tarandı. 3 for döngüsünden ilki ödevde istenen 1gram , ikincisi 2gram ve üçüncüsü ise 3gram göre amino asit dönüşümünü vermektedir.

Bu fonksiyonda 1grama göre amino asit çıkarıldıktan sonra amino asit bazlı ORF çıkarım fonksiyonu çağrılmaktadır.

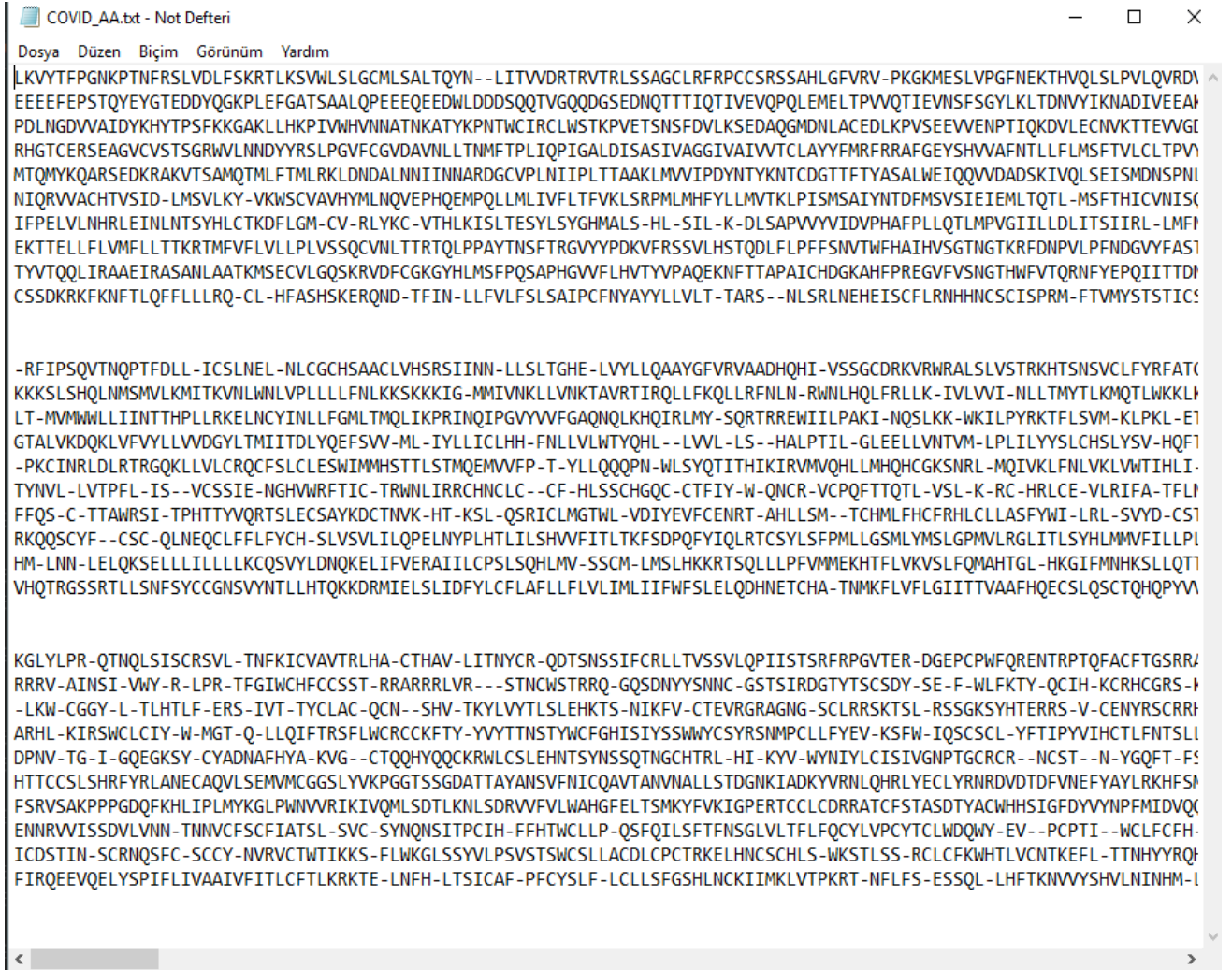
Her for döngüsü sonrası dosyaya yazılırken düzen sağlamak adına yeni satır kaçış karakteri dosyaya yazılacak veriye eklenmektedir.

Son olarak oluşturulan veri Şekil 13’ teki dosyaYazma fonksiyonuna verilerek dosyaya yazılmaktadır.

```
def dosyaYazma(self,amino_asit_sekansi):
    dosya = open('C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\'+ self.dosya_adi +' AA.txt', 'w')
    dosya.write(amino_asit_sekansi)
```

Şekil 13 dosyaYazma isimli fonksiyon

Bu fonksiyon sonucu nCovid virüsüne ait çıktılar Şekil 14 ‘te verilmiştir.



Şekil 14 nCovid için 1gram, 2gram, 3gram için aminoasit çıktı dosyasının baş kısmı

Bu fonksiyonun SARS virüsüne ait amino asit dosya çıktısı *Şekil 15*'te verilmiştir.

```

SARS_AA.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
RSLVDLFSKRTLKSV-LSLGCMPSAPTQYKQ--ILLSLTRNE-LVPLLQTAGFVRVAVDHQHT-VSSGCDRKVRWRALFLVSTRKHTSNSVCLSFRLTC-CVASGTLWKRPA
STVQRMIIKVSLLWNLVPQLKQFELKKKRTGWMILLSNQRLSQNLHLKNQLISLLVI-NLLTMLPLNVLTSRRHKVLILW-L-MLLTYT-NMWWV-QVHSTRQPMVPCKF
LQPRQRSNQTGLVYVFGVQSQ-ILQIHLKFWQ-KTHKEWTILLVKVNNPPLKK-WKILPYRRKS-SVT-KLPKL-AMSYLNHQMVKLK-HKS-VMRILWLLMWKTQALPLRNI
LMR-IS-LTSLLLCNLWVL-MCLLQ-WLVLLPYW-LVLP TTL-NSDVFLVSTTMLLLMHFCF-CLSLYSWVYQLTAFCRESTQSFTCT-HSISPMMFHSLWTFNGLPCFLI
LTTLSTMRVMVVFHSTSYH-LQQPNWLLSLIMVPTRLVMVTPLMHMLHSGSKLLMRIARL FNLVKLTWTIHQIWLGLLLQL-EPTQLLNRYIMN-VQ-HYDRCPVRLVF
RWNIIIR-CYNCLC--CL-HLSSCYSQCKCTSFN-W--DS-QVCPQSTTQAL-VSL-K-GC-S-IRG-VLRPA-TFLHDDSF--CRCVL-Q-LCGSRFSS-H-EL-GSSLLSK-
NAQ-YTERIVRQSRVRPLGAWL-AYINEVLCQDWT-KNVLSV-QTCNLLFYFIRYLCLESFCGF-LCL-PIYD-CSAVGLYG-PSE-P-PTLPGTWKCTCG-L-CYHD-MFSS
VHMF--CSSS-LHSTYFIEYEGGLLS--NF-IRHSLFNSGFISSILF-CYRVSY-SYVWQPCHTF-GWYLFCCHREIKCCPWLGFWFYHEQQVTVGDY-YFY-CCYTSM-L-J
CLPTCHVCAIPGEELHHSASNL-SQSILPS-RCFCV-WHFLVYYTEELLFSTNNYRQYICRLKL-CRYWHH-QHSL-SSAT-A-LIQRRAGQVLQKSYITRC-SWRHFRH-F
LFLVLIIMLIIFWFSLEIQDLEEPCKV-TNMKLLIVLTCISLCSCTVQRCASNKPHVEDPCKVQH-G-YL-HCLALCSRKGFTFS-MAHYGSNMHT-CYQYLSRSSWMC/

DLL-ICSLNEL-NLCSRSAACLVHLRSINNNKFYCR-QETSNSLFCRLLTSSVLQSIISIPRFRPGVTER-DGEPCSWCQRENTPTQFACPSG-RRASAWLRGLCGRGPJ
VRYRG-LSRSPSGIWCLS-NSSS-GRRRGRLAG-YY-AIRD-ARTRTYT-RTS-SVWLFKTY-QCCH-MC-HR-GGTC-K-SYGDKCC-HTTPETWWCSRCTQQGNQWCHAKC
YNQDNVQTKHLVFTLSLEYKASRYFKFI-SSGSRHRTNGQSCL-KSTTHL-RSSGKSYHTEGSHRV-RENYRSCRQCHT-TIR-RC-SNTRVRS-GSYGCLCGKHKHYH-ET-
-CDESHS-HLYSSCATCGFCRCVCFSSGWWYCHIGDLCLLLYEIQTCFW-VQPCCC-CTFVDFVFHYTLSGTSLQLSAGSLSLLLVLDILFHQ-CFILGSPSMCHVFSY
-QHYQCA-WLCSTQHHTIDYSSQTHGCCP-LWYLQEHL-W-HLYICTLGNPASC-CG-QDCST--N-HGQFTKFLGASCYSSKSQLSC-TTE--TESSSTTDLVCGWYF
GGTSSGDATTAYANSVFNICQAVTANVNALLSTDGNKIADKYVRNLQHRLEYECLYRNRDVEHFEVDFYAYLRKHFSMMILSDDAVVCYNSNYAAQGLVASIKNFKAVLYQNI
MLSDTLKGLSDRVFVLWAGFELTSMKYFVKIGPERTCCLCDKRATCFSTSSDYACWNHVSFGFDYVYNPFMIDVQWGF TGNLQSNHDQHCQVHGNAHVASCDAIMTRCLAV
CTTFDDVQAPNYQTHTSSMRGVVYDEIFRSDTLYLTQDLFLPFYSNVTGFHTINHTFGNPVPIPKDGIYFAATEKSNVVRGVFGSTMNKKSQSVIINNSTNVIRACNFEI
VFLHVTYVPSQERNFTTAPAIHEGKAYFPREGVVFNGTSWFITQRNFFSPQIITDNTFVSGNCDVIGIINNTVYDPLQPELDSFKEELDKYFKNHTSPDVLGDISGIN/
YSLF--CLLYFGFHSKSRI-KNLVPKSKRT-NFSLF-LVFLYAVAYAL-YSAVHLINLMCLKILVRYNTRGNTYSTAWL CALGKVL PFHRWHTMVQCTPNVTINCDPAGGAI

ISCRSVL-TNFKICVAVARLHA-CTYAV-TIINFVVDKKRVTRPSSADCLFRPCCSRSSAYLGFVRV-PKGKMESLVLGVNEKTHVQLSLPVLQVRDVLVRGFGDSVEEAL
YGTEDDYQGLPLEFGASAEVVRVEEEEDWLDDETEQSEIEPEPEPTPEEPVNOFTGYLKLTDNVAIKCVDIVKEAQSANPMVIVNAANIHLKHGGGVAGALNKATNGAMQKE
TTKTTFKPNTWCLRCLWSTKPVDTSNSFEVLAVEDTQGMNLAACESQOPTSEEVENPTIQKEVIECDVKTTEVWGNVILKPSDEGVKVTQELGHEDLMAAYVENTSITIKKPI
DAMNLIANIFTPLVQPVGALDVSASVWAGGIIAIVTCAAYYFMKFRRVFGEYNHVAANALLFLMSFTILCLVPAYSFLPGVYSVFYLYLTFTYFNDVSFLAHLQWFAMFSPJ
NNIINNARDGCVPLNIPLTTAAKLMVVPDYGTYNKTCGNTFTYASALWEIQVVDADSKIVQLSEINMDNSPNLAWPLIVTALRANSVAKLQNNELSPVALRQMSCAAGTI
VEHHPVMLQLMLIVSLTFVKLLQPM-MHFFQLMVIR-LTSMIAIYNTGSMVSIEIGMLIMNSWMSFTLTCVNISP--FFLMPLCAITVTMLRKV--LALRTLQRFIIKIN
CSVII-KDCQTESCSSFGRMALSLHQ-STLSRLDLKERVVCVTHVQLAFLHQILMPAGIILWLTMSITHL-LMFSSGALRVTRVMTNIARYMEMHMLVWMLS-LDV-QS
APLLMMFKLLITLNLHL-GGFTILMKFLDQTLFI-LRIYFFHFIILMQGFILLIIRLATLSYLLRMVFILLPQRNQMLSVGVFLVP-TTSHSR-LLLTILLMLLYEHVTLNK
SSYMSRMCHPRRGTSPOQQQFVMKAKHTSLVKVFLCLMALLGLLHRGTSFLHK-LLQTIHLSQEIVMSLLASLTQFMILCNLSLTHSKKSWTSTSKIIHQMLILATFQALTI
IPCFNNAYYILVFTRNPGSRRTLYQSLEHETSHCFDLYFSMQLMHMCSTALCI--TSCA-RSL-GTTLGVILIALLGFVL-ERFYLFDIGTLWFKHAHMLLSTVKIQLVVRIL
    
```

Şekil 15 SARS için 1gram, 2gram, 3gram için aminoasit çıktı dosyasının baş kısmı

Her iki çıktı dosyasında da ilk paragraf 1gram yani 1 numaralı konumdan itibaren üretilen amino asit sekansını , ikinci paragraf 2gram yani 2 numaralı konumdan başladığında üretilen amino asit sekansını ve son paragraf ise 3gram yani 3 numaralı konumdan başladığında üretilen amino asit sekansını göstermektedir.

Bu fonksiyon içerisinde ilk for döngüsü sonrası çağrılan orfBulAminoAsit fonksiyonu Şekil 16’da verilmiştir. Verilen fonksiyon amino asit sekansları içerisinde başlangıç amino asidi ile bitiş amino asitlerine bakarak amino asit bazlı ORF çıkarımı yapmaktadır. Sonuçları dosyaya yazılmaktadır.

```
def orfBulAminoAsit(self, protein):
    orf_protein = ""
    dosya = open('C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\'+ self.dosya_adi + '_AA_ORF.txt', 'w')
    for i in range(0, len(protein) - 1, 1):
        if (protein[i] == 'M'):
            i += 1
            while (protein[i] != '-' and i < len(protein) - 1):
                orf_protein += protein[i]
                i += 1
            if (protein[i] == '-'):
                dosya.write(orf_protein + '\n\n')
                orf_protein = ""
    dosya.write(orf_protein)
```

Şekil 16 orfBulAminoAsit fonksiyonu

Bu fonksiyonda işlenen nCovid virüs amino asitlerinin sonucunda oluşturulan ORF dosyası Şekil 17 ve Şekil 18’de verilmiştir.

Şekil 17 nCovid için amino asit bazlı ORF sekansları dosyasının baş kısmı


```

COVID_AA_ORF.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
ASNFSLFCACCC
LQNKLDHRWNCYRNL
SCRLDVAQLLHCF
FQTVCAFAHVVIQ
SRN

KSNQWRLIKRT

YSTSTICS

VY

SDNGPQNQRNAPRIT
FGGPSDSTGSNQNG
ERSGARSQRRPQGL
PNNTASWFTALTQ
HGKEDLKFPRGQGP
INTNSSPDDQIGY
RRATRRIRGGDGK
MKDLSRWYFYLG
I
KDLSPRWYFYLGTP
EAGLPYGANKDGI
IIVATEGALNTPK
DHIGTRNPANNA
AIVLQLPQGTTL
PKGFYAEGSRGGS
QASSRSSRSRNS
SRNSTPGSSRGTS
PARMAGNGC
AGNGGDAALALL
LDRLNQLESMGKG
QQQGQTVTKKSA
AEASKKPRQKRT
ATKAYNVTQAFG
RRGPEQTQGNFG
DQELIRQGTDYK
HWPQIAQFAPS
SASAFFGMSRIG
MEV
SGKGQQQGQTV
TKKSAEASKKPR
QKRTATKAYNVT
QAFGRRGPEQT
QGNFGDQELIR
QGTDYKHWPQIA
QFAPSASAFFG
MSRIGMEVTPS
GTWLTYTGAIK
LDDKDPNFKI
SRIGMEVTPS
GTWLTYTGAIK
LDDKDPNFKDQ
VILLNKHIDAY
KTFPPTPKDKK
KKKADETQALP
QRQKKQQTVT
LLPAADLDDF
SKQLQQSMSS
ADSTQA
EVTSGTWLT
YTGAIKLDDK
DPNFKDQVILL
NKHIDAYKTF
PPTPKDKKKK
KADETQALP
QRQKKQQTVT
LLPAADLDDF
SKQLQQSMSS
ADSTQA
SSADSTQA
GYINVFAFPFTI
YSLLLCRMNSR
NYIAQVDVNFN
LT
NSRNYIAQVDV
NFNLT
LGRAAYMEEP
EEP
St 1, Str 1

```

Şekil 18 nCovid için amino asit bazlı ORF sekansları dosyasının son kısmı

Bu fonksiyonda işlenen SARS virüs amino asitlerinin sonucunda oluşturulan ORF dosyası *Şekil 19* ve *Şekil 20* ' de verilmiştir.

```

SARS_AA_ORF.txt - Not Defteri
Dosya  Düzen  Biçim  Görünüm  Yardım
PSAPTQYKQ

ALVV

CSLNVLMP

P

WAKPQLHTAMFFVVRTVIREPVVIAMASI

FFFVRTVIREPVVIAMASI

ASI

T

NKTGTLMAVVHVSNSLVSSMEVQSLAMSTTISVAQMGTLIIASKIFSHARASQCALFPNNLITSSRREVSTAAVTMSMKLPGSLSALIRATSTRHPSKLRVPRNLTLKGNAC
AVVHVSNSLVSSMEVQSLAMSTTISVAQMGTLIIASKIFSHARASQCALFPNNLITSSRREVSTAAVTMSMKLPGSLSALIRATSTRHPSKLRVPRNLTLKGNACSLCFLLTQ
EVQSLAMSTTISVAQMGTLIIASKIFSHARASQCALFPNNLITSSRREVSTAAVTMSMKLPGSLSALIRATSTRHPSKLRVPRNLTLKGNACSLCFLLTQKSKSFNHVLRKRKF
STTISVAQMGTLIIASKIFSHARASQCALFPNNLITSSRREVSTAAVTMSMKLPGSLSALIRATSTRHPSKLRVPRNLTLKGNACSLCFLLTQKSKSFNHVLRKRRLRVSWG
GTLIIASKIFSHARASQCALFPNNLITSSRREVSTAAVTMSMKLPGSLSALIRATSTRHPSKLRVPRNLTLKGNACSLCFLLTQKSKSFNHVLRKRRLRVSWGVALCTLLHL
SMKLPGLSALIRATSTRHPSKLRVPRNLTLKGNACSLCFLLTQKSKSFNHVLRKRRLRVSWGVALCTLLHLHRSVTICTCLP
KLPGLSALIRATSTRHPSKLRVPRNLTLKGNACSLCFLLTQKSKSFNHVLRKRRLRVSWGVALCTLLHLHRSVTICTCLP
KFHGRRATF

L

LAAIISVPTGFLVLVILAQAAILALLVTMWRP

WRP

RISLRY

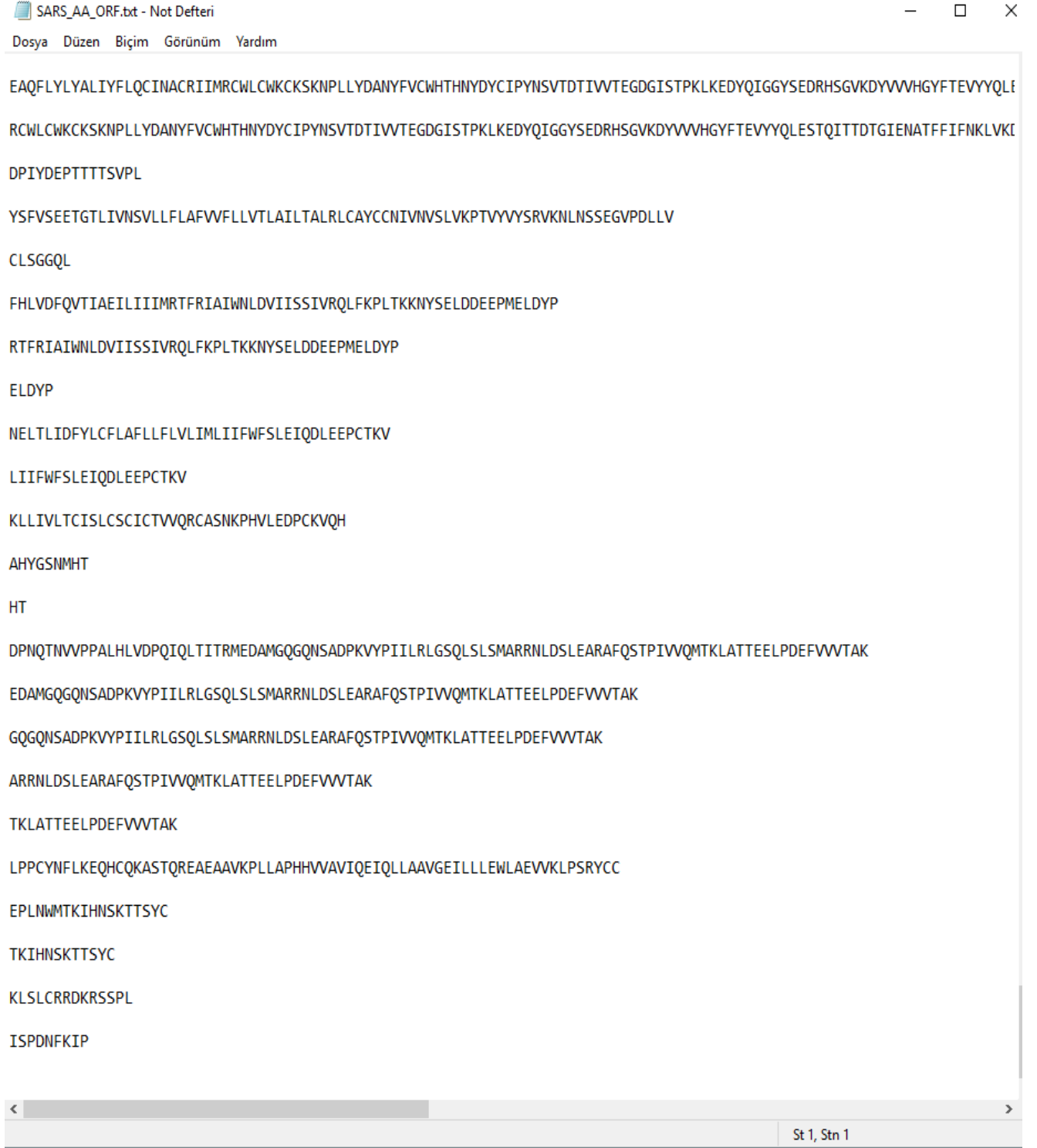
KRLPSFWHLSLLLQVPLLT

QQTTFQLICKEQLSPYLMVFLNSHYVLSTPWFIQTCSPTVSLLWHM

VFLNSHYVLSTPWFIQTCSPTVSLLWHM
  
```

Şekil 19 SARS için amino asit bazlı ORF sekansları dosyasının baş kısmı

FASTA SEKANS DOSYALARI



```
SARS_AA_ORF.txt - Not Defteri
Dosya  Düzen  Biçim  Görünüm  Yardım

EAQFLYLYALIYFLQCINACRIIMRCWLCWKCKSKNPLLYDANYFVCWHTHNYDYCIPYNSVTDTIWVTEGDGISTPKLKEDYQIGGYSEDRHSGVKDYVWHGYFTEVYYQLF
RCWLCWKCKSKNPLLYDANYFVCWHTHNYDYCIPYNSVTDTIWVTEGDGISTPKLKEDYQIGGYSEDRHSGVKDYVWHGYFTEVYYQLESTQITTDGTGIENTFFIFNKLVKI
DPIYDEPTTTTSVPL

YSFVSEETGLIVNSVLLFLAFVWFLVLTALILTALRLCAYCCNIVNVS LVKPTVYVYSRVKNLSSEGVPDLLV

CLSGGQL

FHLVDFQVTIAEILIIIMRTFRIAIWNLDVISSIVRQLFKPLTKKNYSELDDEEPMELDYP
RTFRIAIWNLDVISSIVRQLFKPLTKKNYSELDDEEPMELDYP

ELDYP

NELTLIDFYLCFLAFLFLVLIMLIIFWFSLEIQDLEEPCTKV
LIIFWFSLEIQDLEEPCTKV

KLLIVLTCISLCSCICTVWQRCASNKPHVLEDPCKVQH

AHYGSNMHT

HT

DPNQTNVPPALHLVDPQIQLTITRMEDAMQGQNSADPKVYPIILRLGSQLSLSMARRNLDSEARAFQSTPIVVQMTKLATTEELPDEFVWTAK
EDAMQGQNSADPKVYPIILRLGSQLSLSMARRNLDSEARAFQSTPIVVQMTKLATTEELPDEFVWTAK
QGQNSADPKVYPIILRLGSQLSLSMARRNLDSEARAFQSTPIVVQMTKLATTEELPDEFVWTAK
ARRNLDSEARAFQSTPIVVQMTKLATTEELPDEFVWTAK
TKLATTEELPDEFVWTAK
LPPCYNFLKEQHCKASTQREAAVVKPLLAPHHVAVIQEIQLLAAVGEILLLEWLAEVVKLPSRYCC
EPLNWMTKIHNSKTTSYC
TKIHNSKTTSYC
KLSLCRRDKRSSPL
ISPDNFKIP

St 1, Stn 1
```

Şekil 20 SARS için amino asit bazlı ORF sekansları dosyasının son kısmı

Bu fonksiyonlar çalıştıktan sonra amino asit ve ORF sekansları çıkarılıp dosyaya aktarılmış olup ödevin a ve b kısımları tamamlanır. Hatta a ve b kısmına ek olarak amino asit bazlı ORF sekanslarının çıkarılması da gerçekleştirilmiş olunur.

Fonksiyon çalışmaları bitince main fonksiyonuna geri dönülür ve Needleman Wunsch modülünde yer alan iki sekanstaki farkı bulmaya yarayan farkBul fonksiyonu çağrılır.

Fonksiyon verilen sekanslardaki konum nükleotitlerini karşılaştırır ve farklı nükleotitlei gördüğünde konum bilgisi ile birlikte bir değişkende toplanır . Tüm sekans tarandıktan sonra sonuç dosyaya yazılır.

Fonksiyon parametre olarak virüslere ait proteinlerin dosya yollarını alır ve kayıt esnasında dosya oluşturulurken ad verilmesinde kullanılmak üzere proteinin adını da almaktadır.

Fonksiyon kodu aşağıda verilmiştir.

```
def farkBul(covid , sars , protein_turu):
    covid_protein = proteinDizisi0ku(covid)
    sars_protein = proteinDizisi0ku(sars)

    covid_uzunluk = len(covid_protein)-1
    sars_uzunluk = len(sars_protein)-1

    maX_uzunluk = max(covid_uzunluk , sars_uzunluk)

    dosya_metni = ""

    i = 0
    while i<maX_uzunluk:

        fark_metni=""
        fark_sars =""
        fark_nCovid = ""

        if i<covid_uzunluk and i<sars_uzunluk :
            if covid_protein[i] != sars_protein[i]:
                fark_metni += "Konum : " +str(i)

                while covid_protein[i] != sars_protein[i] and i<covid_uzunluk and
i<sars_uzunluk:
                    fark_sars += sars_protein[i]
                    fark_nCovid += covid_protein[i]
                    i += 1
                fark_metni += "\nnCovid = " + fark_nCovid + "\nSARS    = " + fark_sars +
"\n\n*****\n\n"
            else:
                i += 1

        elif i>=covid_uzunluk and i<=sars_uzunluk:
            fark_metni += "Konum : " +str(i)
            while i<=sars_uzunluk:
                fark_sars += sars_protein[i]
                fark_nCovid += "-"
```

```

        i += 1
        fark_metni += "\nnCovid = " + fark_nCovid + "\nSARS    = "+fark_sars +
"\n\n*****\n\n"

        elif i>=sars_uzunluk and i<=covid_uzunluk :
            fark_metni += "Konum : " +str(i) + "\n"

            while i<covid_uzunluk:
                fark_sars += "-"
                fark_nCovid += covid_protein[i]
                i += 1
            fark_metni += "\nnCovid = " + fark_nCovid + "\nSARS    = " + fark_sars +
"\n\n*****\n\n"
            i+=1

        if fark_metni != "":
            print(fark_metni)
            dosya_metni += fark_metni
        dosya = open("C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\Fark_"+ protein_turu +
".txt" , "w")
        dosya.write(dosya_metni)

```

Bu fonksiyonun çalıştırılması sonucu nCovid ve SARS virüslerine ait nucleocapsid proteinlerinin fark dosya çıktısı Şekil 21 ve Şekil 22’ de verilmiştir.

```

Fark_Nucleocapsid.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
Konum : 8
nCovid = QRNAPRITFG
SARS    = NQRSAPRITF

*****

Konum : 19
nCovid = PSDSTGS
SARS    = GPTDSTD

*****

Konum : 27
nCovid = QNGERSGARSKQR
SARS    = NQNGGRNGARPKQ

*****

Konum : 41
nCovid = PQGLPN
SARS    = RPQGLP

*****

St 1, Stn 1

```

Şekil 21 Nucleocapsid proteinlerinin sekansla farklı kısımları başlangıç konumları ve farklı sekanslarının çıktı dosyasının baş kısmı

```

Fark_Nucleocapsid.txt - Not Defteri
Dosya  Düzen  Biçim  Görünüm  Yardım
*****

Konum : 409
nCovid = SMS
SARS   = NSM

*****

Konum : 413
nCovid = AD
SARS   = GA

*****

Konum : 416
nCovid = TQ
SARS   = AD

*****

Konum : 419
nCovid = --
SARS   = TQ

St 1, Str 1

```

Şekil 22 Nucleocapsid proteinlerinin sekansta farklı kısımları başlangıç konumları ve farklı

Bu fonksiyonun çalıştırılması sonucu nCovid ve SARS virüslerine ait polyprotein proteinlerinin fark dosya çıktısı *Şekil 23* ve *Şekil 24*’te verilmiştir.

```

Fark_Polyprotein.txt - Not Defteri
Dosya  Düzen  Biçim  Görünüm  Yardım
*****

Konum : 76
nCovid = RTAP
SARS   = LSTN

*****

Konum : 83
nCovid = VM
SARS   = KV

*****

Konum : 92
nCovid = E
SARS   = D

*****

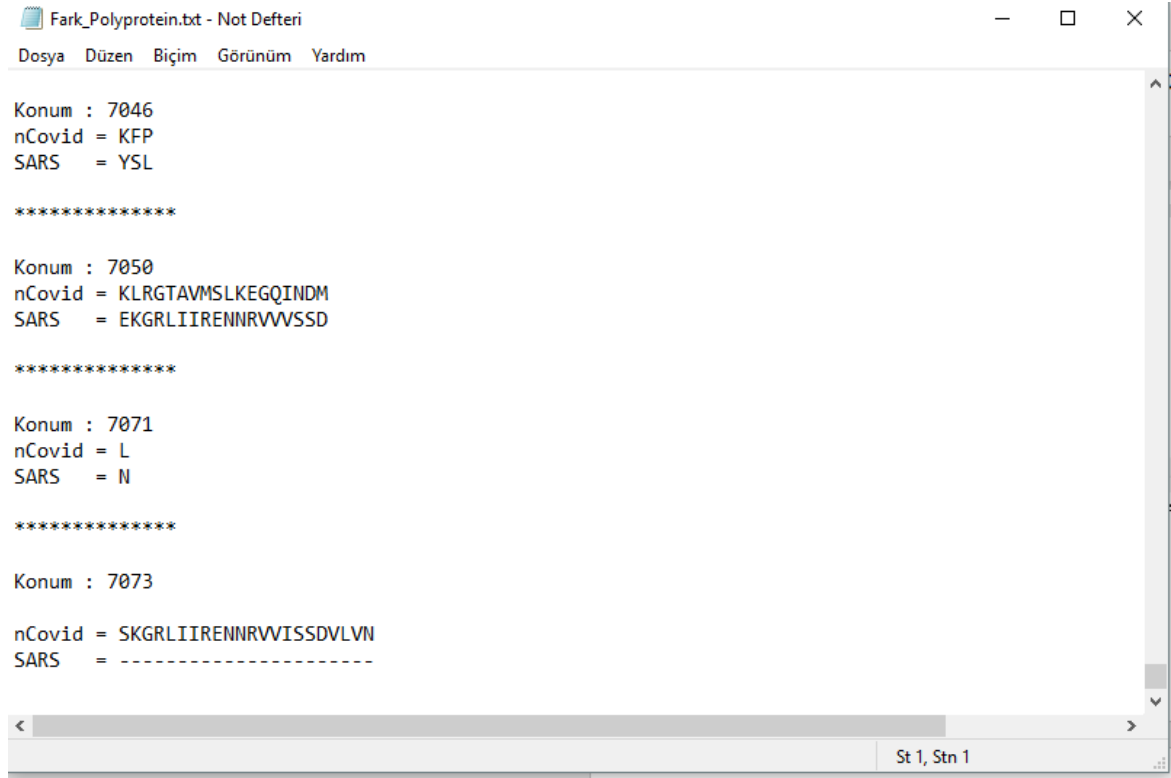
Konum : 142
nCovid = F
SARS   = Y

*****

St 1, Str 1

```

Şekil 23 Polyprotein proteinlerinin sekansta farklı kısımları başlangıç konumları ve farklı sekanslarının çıktı dosyasının baş kısmı



Şekil 24 Polypotein proteinlerinin sekansa farklı kısımları başlangıç konumları ve farklı sekanslarının çıktı dosyasının son kısmı

Dosyaya fark sekansları yazıldıktan sonra main fonksiyona dönülür ve nucleocapsid ve polypotein için Needleman Wunsch algoritması çağrılır.

Bu fonksiyonda kullanılacak BLOSUM62 modülünün içerisinde skorBul fonksiyonu tanımlanmıştır.

BLOSUM62 modülündeki skorBul fonksiyonu aşağıda verilmiştir.

```

def skorBul(satir_degeri , sutun_degeri):

    blosum62 = {
        'WF': 1, 'LR': -2, 'SP': -1, 'VT': 0,
        'QQ': 5, 'NA': -2, 'ZY': -2, 'WR': -3,
        'QA': -1, 'SD': 0, 'HH': 8, 'SH': -1,
        'HD': -1, 'LN': -3, 'WA': -3, 'YM': -1,
        'GR': -2, 'YI': -1, 'YE': -2, 'BY': -3,
        'YA': -2, 'VD': -3, 'BS': 0, 'YY': 7,
        'GN': 0, 'EC': -4, 'YQ': -1, 'ZZ': 4,
        'VA': 0, 'CC': 9, 'MR': -1, 'VE': -2,
        'TN': 0, 'PP': 7, 'VI': 3, 'VS': -2,
        'ZP': -1, 'VM': 1, 'TF': -2, 'VQ': -2,
        'KK': 5, 'PD': -1, 'IH': -3, 'ID': -3,
        'TR': -1, 'PL': -3, 'KG': -2, 'MN': -2,
        'PH': -2, 'FQ': -3, 'ZG': -2, 'XL': -1,
        'TM': -1, 'ZC': -3, 'XH': -1, 'DR': -2,
        'BW': -4, 'XD': -1, 'ZK': 1, 'FA': -2,
        'ZW': -3, 'FE': -3, 'DN': 1, 'BK': 0,
    }

```

FASTA SEKANS DOSYALARI

```
'XX': -1, 'FI': 0, 'BG': -1, 'XT': 0,
'FM': 0, 'BC': -3, 'ZI': -3, 'ZV': -2,
'SS': 4, 'LQ': -2, 'WE': -3, 'QR': 1,
'NN': 6, 'WM': -1, 'QC': -3, 'WI': -3,
'SC': -1, 'LA': -1, 'SG': 0, 'LE': -3,
'WQ': -2, 'HG': -2, 'SK': 0, 'QN': 0,
'NR': 0, 'HC': -3, 'YN': -2, 'GQ': -2,
'YF': 3, 'CA': 0, 'VL': 1, 'GE': -2,
'GA': 0, 'KR': 2, 'ED': 2, 'YR': -2,
'MQ': 0, 'TI': -1, 'CD': -3, 'VF': -1,
'TA': 0, 'TP': -1, 'BP': -2, 'TE': -1,
'VN': -3, 'PG': -2, 'MA': -1, 'KH': -1,
'VR': -3, 'PC': -3, 'ME': -2, 'KL': -2,
'VV': 4, 'MI': 1, 'TQ': -1, 'IG': -4,
'PK': -1, 'MM': 5, 'KD': -1, 'IC': -1,
'ZD': 1, 'FR': -3, 'XK': -1, 'QD': 0,
'XG': -1, 'ZL': -3, 'XC': -2, 'ZH': 0,
'BL': -4, 'BH': 0, 'FF': 6, 'XW': -2,
'BD': 4, 'DA': -2, 'SL': -2, 'XS': 0,
'FN': -3, 'SR': -1, 'WD': -4, 'VY': -1,
'WL': -2, 'HR': 0, 'WH': -2, 'HN': 1,
'WT': -2, 'TT': 5, 'SF': -2, 'WP': -4,
'LD': -4, 'BI': -3, 'LH': -3, 'SN': 1,
'BT': -1, 'LL': 4, 'YK': -2, 'EQ': 2,
'YG': -3, 'ZS': 0, 'YC': -2, 'GD': -1,
'BV': -3, 'EA': -1, 'YW': 2, 'EE': 5,
'YS': -2, 'CN': -3, 'VC': -1, 'TH': -2,
'PR': -2, 'VG': -3, 'TL': -1, 'VK': -2,
'KQ': 1, 'RA': -1, 'IR': -3, 'TD': -1,
'PF': -4, 'IN': -3, 'KI': -3, 'MD': -3,
'VW': -3, 'WW': 11, 'MH': -2, 'PN': -2,
'KA': -1, 'ML': 2, 'KE': 1, 'ZE': 4,
'XN': -1, 'ZA': -1, 'ZM': -1, 'XF': -1,
'KC': -3, 'BQ': 0, 'XB': -1, 'BM': -3,
'FC': -2, 'ZQ': 3, 'XZ': -1, 'FG': -3,
'BE': 1, 'XV': -1, 'FK': -3, 'BA': -2,
'XR': -1, 'DD': 6, 'WG': -2, 'ZF': -3,
'SQ': 0, 'WC': -2, 'WK': -3, 'HQ': 0,
'LC': -1, 'WN': -4, 'SA': 1, 'LG': -4,
'WS': -3, 'SE': 0, 'HE': 0, 'SI': -2,
'HA': -2, 'SM': -1, 'YL': -1, 'YH': 2,
'YD': -3, 'ER': 0, 'XP': -2, 'GG': 6,
'GC': -3, 'EN': 0, 'YT': -2, 'YP': -3,
'TK': -1, 'AA': 4, 'PQ': -1, 'TC': -1,
'VH': -3, 'TG': -2, 'IQ': -3, 'ZT': -1,
'CR': -3, 'VP': -2, 'PE': -1, 'MC': -1,
'KN': 0, 'II': 4, 'PA': -1, 'MG': -3,
'TS': 1, 'IE': -3, 'PM': -2, 'MK': -1,
'IA': -1, 'PI': -3, 'RR': 5, 'XM': -1,
'LI': 2, 'XI': -1, 'ZB': 1, 'XE': -1,
'ZN': 0, 'XA': 0, 'BR': -1, 'BN': 3,
'FD': -3, 'XY': -1, 'ZR': 0, 'FH': -1,
'BF': -3, 'FL': 0, 'XQ': -1, 'BB': 4,
}
try:
    deger = blosum62[satir_degeri+sutun_degeri]
except:
```

```

deger = 100

if deger == 100:
    deger = blosum62[(satir_degeri+sutun_degeri)[:,-1]]

return deger

```

Needleman Wunsch algoritması needlemanWunsch fonksiyonunda tanımlanmıştır. Boşluk cezası (-4) olarak bir değişkene atanmış ve sekanslar matriste satır ve sütun olarak kullanılacağından başlarına boşluk değerini ifade edebilmek adına “-“ karakteri eklenmiştir.

Matrisi oluşturmak için numpy kütüphanesi modüle eklenmiştir.

İki matris tanımlanmış ve biri yol biri skor bilgilerini tutmaktadır.

Tanımlamalar sonrası skor ve yol matrisinin [0 ,..] satırı ve [... , 0] sütunu başlangıç için doldurulmuştur.

Başlangıç atamaları sonrası matrisi dolaşmak adına iç içe for döngüsü oluşturulmuştur.

Döngü içerisinde öncelikli olduğundan ilk olarak nCovid ve SARS sekanslarının mevcut matris konumlarında eşit olup olmadıklarına bakılmış ve eşitler ise skor tablosundaki diagonalde bulunan skora blosum62 modülünün skorBul fonksiyonundan gelen değer eklenip skor matrisine yazılması ve yol matrisine de mevcut konum için 3 yazılması kodlanmıştır. (Yol matrisinde 0 en sol köşeyi , 1 solu , 2 yukarıyı ve 3 diagonali temsil etmektedir.)

Eşit değiller ise diagonale , sola ve yukarıya göre skor hesaplaması yapıp en büyük olanı seçilerek matrislere gerekli verilerin yazılması sağlanmıştır.

Matrisler tamamen dolduktan sonra döngülerden çıkılır ve hizalamalarda kullanılmak üzere nCovid ve SARS değişkenleri tanımlanmıştır. Yol matrisinin sağ en altından başlayacak şekilde indeks tanımlamaları yapılmış ve ardından bir while döngüsü ile yol matrisindeki değer en sol üst köşe olan 0 değerine ulaşılan kadar matris incelenmesi sağlanmıştır.

İncelemede okunan değer 1 ise sola hareketten dolayı sütun değeri ilgili değişkene eklenirken satır değerine boşluk anlamında “-“ karakteri eklenip sütunu ifade eden indeks değeri 1 azaltılır.

İncelemede okunan değer 2 ise yukarı hareketten dolayı satır değeri ilgili değişkene eklenirken sütun değerine boşluk anlamında “-“ karakteri eklenip satırı ifade eden indeks değeri 1 azaltılır.

Son durum ise incelemede okunan değer 3 olmasıdır . Bu değer diagonal hareketi temsil eder. Bundan dolayı satır ve sütun değerleri kendileri ile ilgili olan değişkene eklenip her ikisinin de indeks değerleri 1 azaltılır.

İncelenen değerde 0 değeri görülürse while döngüsünden sonraki adım olan dosya oluşturma ve dosyaya hizalama sonuçlarını yazma işlemi gerçekleştirilir.

Fonksiyon kodu aşağıda verilmiştir.

```
import BLOSUM62
import numpy as np

def needlemanWunsch(covid , sars , protein_turu):
    bosluk_ceza_puani = -4
    covid_protein = "-" + proteinDizisi0ku(covid)
    sars_protein = "-" + proteinDizisi0ku(sars)
    sat = len(covid_protein)
    sut = len(sars_protein)

    skor_matrisi = np.zeros([sat , sut])
    yol_matrisi = np.zeros([sat,sut])

    for i in range(0,sut,1):
        skor_matrisi[0][i] = i * bosluk_ceza_puani
    for j in range(0,sat,1):
        skor_matrisi[j][0] = j * bosluk_ceza_puani

    # Yol Matrisi : 0 bitis , 1 sol , 2 yukarı , 3 diagonal
    yol_matrisi[0][0] = 0

    for i in range(1,sut,1):
        yol_matrisi[0][i] = 1

    for j in range(1,sat,1):
        yol_matrisi[j][0] = 2

    for satir in range(1,sat,1):
        for sutun in range(1,sut,1):
            if covid_protein[satir] == sars_protein[sutun]:
                skor_matrisi[satir][sutun]=skor_matrisi[satir-1][sutun-1]+BLOSUM62.skorBul(covid_protein[satir],sars_protein[sutun])
                yol_matrisi[satir][sutun]= 3
            else:
                skor_diagonal = skor_matrisi[satir - 1][sutun - 1] + BLOSUM62.skorBul(covid_protein[satir], sars_protein[sutun])
                skor_yukari = skor_matrisi[satir-1][sutun] + BLOSUM62.skorBul(covid_protein[satir],sars_protein[sutun])
                skor_sol = skor_matrisi[satir][sutun-1] + BLOSUM62.skorBul(covid_protein[satir],sars_protein[sutun])
                if skor_yukari >= skor_sol and skor_yukari >= skor_diagonal:
                    skor_matrisi[satir][sutun] = skor_sol
                    yol_matrisi[satir][sutun] = 2
                elif skor_sol >= skor_yukari and skor_sol >= skor_diagonal:
                    skor_matrisi[satir][sutun] = skor_yukari
                    yol_matrisi[satir][sutun] = 1
                else:
                    skor_matrisi[satir][sutun] = skor_diagonal
                    yol_matrisi[satir][sutun] = 3
    hizalama_sonucu_covid = ""
    hizalama_sonucu_sars = ""
    yol_sutun = sut-1
```

```

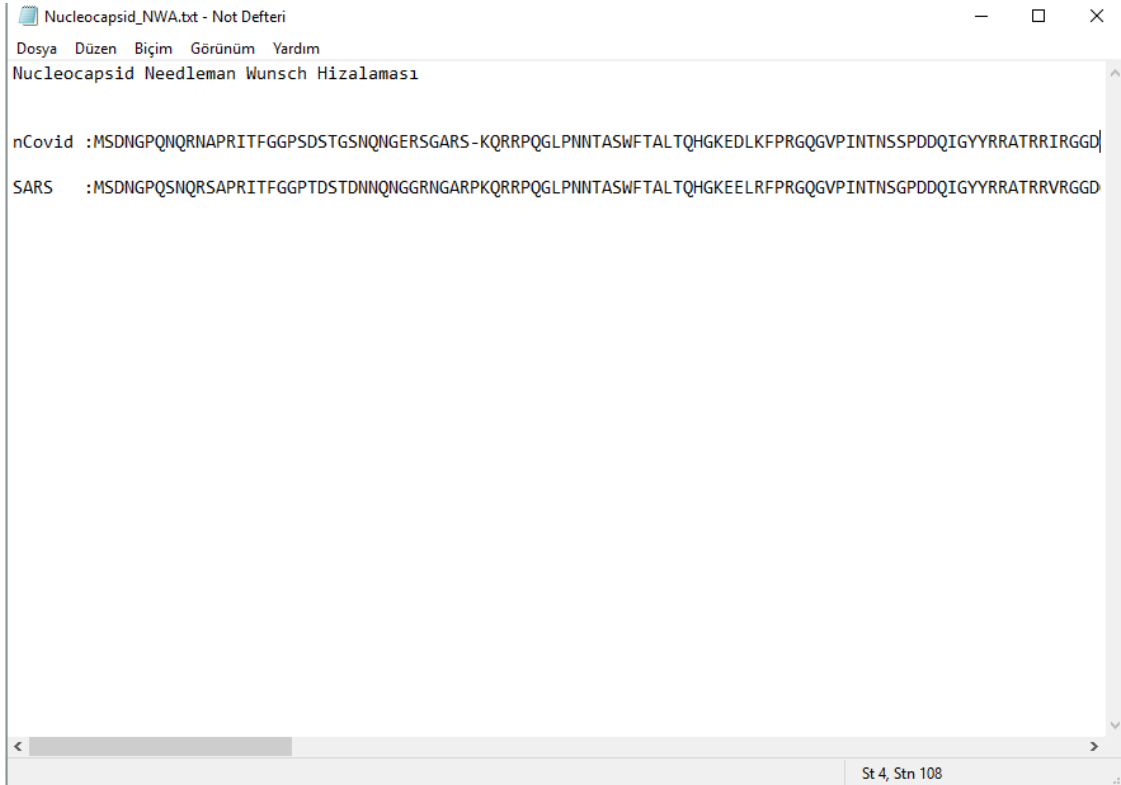
yol_satir = sat-1
adim = yol_matrisi[yol_satir][yol_sutun]

while(adim != 0 ):
    if adim ==1:
        hizalama_sonucu_sars += sars_protein[yol_sutun]
        hizalama_sonucu_covid += "-"
        yol_sutun -= 1
    elif adim == 2:
        hizalama_sonucu_sars += "-"
        hizalama_sonucu_covid += covid_protein[yol_satir]
        yol_satir -= 1
    elif adim == 3:
        hizalama_sonucu_covid += covid_protein[yol_satir]
        hizalama_sonucu_sars += sars_protein[yol_sutun]
        yol_satir -= 1
        yol_sutun -= 1
    adim = yol_matrisi[yol_satir][yol_sutun]

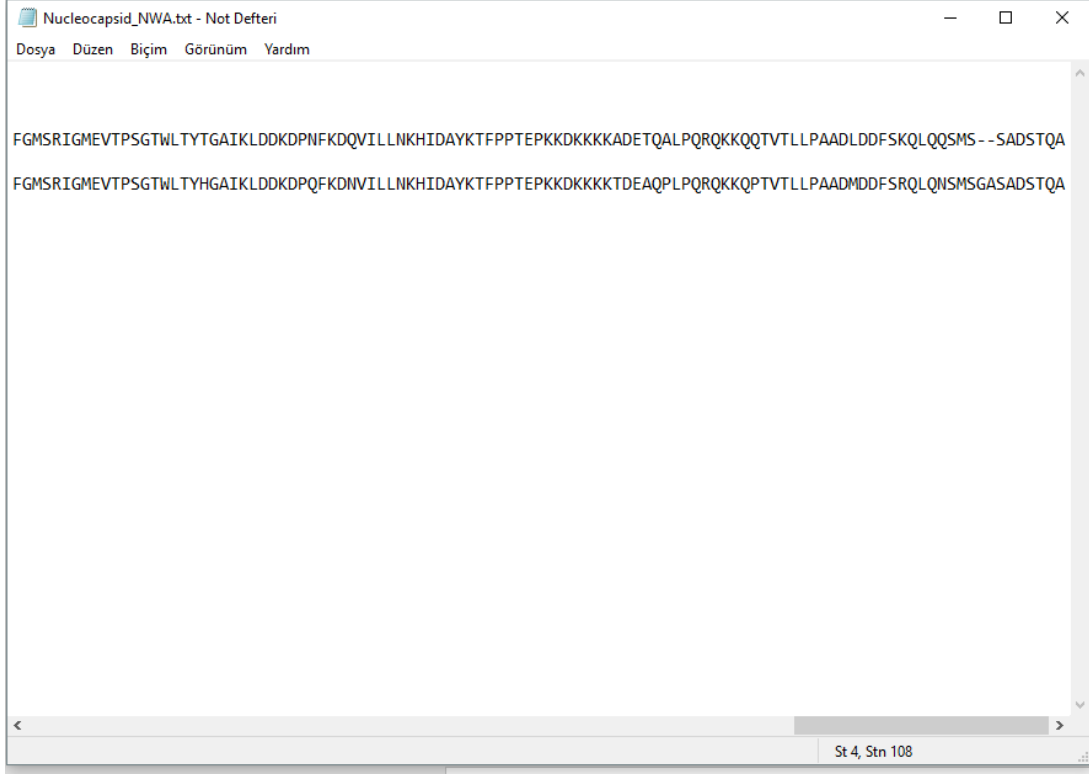
hizalama_dosyasi = open("C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\"+protein_turu+"_NWA.txt" , "w")
hizalama_dosyasi.write(protein_turu+ " Needleman Wunsch Hizalaması
\\n\\n\\n" + "nCovid :" + hizalama_sonucu_covid[::-1] + "\\n\\nSARS  :" +
hizalama_sonucu_sars[::-1] )

```

Bu fonksiyon sonucu nucleocapsid proteinine ait dosya çıktısı *Şekil 25* ve *Şekil 26*’da verilmiştir.



Şekil 25 Nucleocapsid proteinlerinin hizalanmış halinin dosyasını baş kısmı



Şekil 26 Nucleocapsid proteinlerinin hizalanmış halinin dosyasını son kısmı

Dosya içerisindeki nCovid ve SARS nucleocapsid sekanslarının tamamı aşağıdaki gibidir.

nCovid : MSDNGPQNQRNAPRITFGGPSDSTGSNQNGERSGARS-
 KQRRPQGLPNNTASWFTALTQHGKEDLKFPARGQVPINTNSSPDDQIGYYRRA
 TRRIRGGDGKMKDLSPRWYFYLLGTGPEAGLPYGANKDGIWVATEGALNTPK
 DHIGTRNPANNAIIVLQLPQGTTLPKGFYAEGSRGGSQASSRSSSRNNSRNST
 PGSSRGTSARMAGNGGDAALALLLDRLNQLESKMSGKGQQQQGQTVTKKS
 AAEASKKPRQKRTATKAYNVTQAFGRRGPEQTQGNFGDQELIRQGTDYKHWP
 QIAQFAPSASAFFGMSRIGMEVTPSGTWLTYTGAIKLDDKDPNFKDQVILLNKH
 IDAYKTFPPTPKDKKKKKADETQALPQRQKKQQTVTLLPAADLDDFSKQLQQ
 SMS—SADSTQA

SARS: MSDNGPQSNQRSAPRITFGGPTDSTDNNQNGGRNGARPKQRRPQGLPNN
 TASWFTALTQHGKEELRFARGQVPINTNSGPDDQIGYYRRATRRVRGGDGKM
 KELSPRWYFYLLGTGPEASLPYGANKGIVWVATEGALNTPKDHIHIGTRNPNNN
 AATVLQLPQGTTLPKGFYAEGSRGGSQASSRSSSRGNSRNSTPGSSRGNSPA
 RMASGGGETALALLLDRLNQLESKVSGKGQQQQGQTVTKKSAAEASKKPRQ
 KRTATKQYNVTQAFGRRGPEQTQGNFGDQDLIRQGTDYKHWPQIAQFAPSASA
 FFGMSRIGMEVTPSGTWLTYHGAIKLDDKDPQFKDNVILLNKHIDAYKTFPPT
 PKDKKKKKTDEAQPLPQRQKKQPTVTLLPAADMDDFSRQLQNSMSGASADST
 QA

Bu fonksiyon sonucu polyprotein proteinine ait dosya çıktısı Şekil 27 ve Şekil 28’de verilmiştir

```

Polyprotein_NWA.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
Polyprotein Needleman Wunsch Hizalaması

nCovid :MESLVPGFNEKTHVQLSLPVLQVRDVLVRGFGDSVEEVLSEARQHLKDGTCGLVEVEKGVLPQLEQPYVFIKRS DARTAPHGHVMVELV
TPVVQTIEVNSFSGLKLTDMVYIKNADIVEEAKVKPTVVNAANVYLKHGGGVAGALNKATNNAMQVESDDYIATNGPLKVGGSCVLSGHNLAKH
CEDLKPVEEVENPTIQKDVLECNVKTTEVVGDIILKPANNSLKITEEVGHTDLMAAYVDNSSLTIKKPNELSRVLGLKTLATHGLAAVNSVPWDT
MRFRRAFGEYSHVAFNTLLFLMSFTVLC LTPVYSFLPGVYSVIYLYLT FYLTNDVSFLAHLQWFMVFTPLVPFWITAIYIICISTKHFYWFSSNYL
FTYASALWEIQVVDADSKIVQLSEISMDNSPNLAWPLIVTALRANSVAVKLQNNELSPVALRQMSCAAGTTQTACTDDNALAYYNTTKGGRFVLALL
RNLQHLRYECLYRNRDVDTDFVNEFYAYLRKHFSMMILSDDAVVCFNSTYASQGLVASIKNFKSVLYYQNNVFMSEAKCWTETDLTKGPHEFCSQHT
RRATCFSTASDTYACWNHHSIGFDYVYNPFMIDVQQWGTGNLQSNHDLVCQVHGNAHVASCDAIMTRCLAVHECFVKRVDWTIEYPIIGDELKINAA

SARS :-----M---ESLVGVNEKTHVQLSLPVLQVRDVLVRGFGDSVEEALSEAREHLKNGTCGLVEVEKGVLPQLEQF
PEPTPEEPVNFQFTGYLKLTDNVAIKCVDIVKEAQSANPMVIVNAANIHLKHGGGVAGALNKATNGAMQKESDDYIKLNGPLTVGGSCVLSGHNLAKH
ESQQPTSEEVVENPTIQKEVIECDVKTTEVVGNIILKPSDEGVKVTQELGHEDLMAAYVENTSITIKKPNELSLALGLKTIATHGIAAINSPWSKI
KFRRVFGEYNHVVAAANALLFLMSFTILCLVPAYSFLPGVYSVFIYLYLT FYLTNDVSFLAHLQWFMFSPIVPFWITAIYVFCISLKHCHWFFNNYLR
TYASALWEIQVVDADSKIVQLSEINMDNSPNLAWPLIVTALRANSVAVKLQNNELSPVALRQMSCAAGTTQTACTDDNALAYYNNKSGGRFVLALLS
NLQHLRYECLYRNRDVDFEFDVNEFYAYLRKHFSMMILSDDAVVCYNRYAAQGLVASIKNFKAVLYYQNNVFMSEAKCWTETDLTKGPHEFCSQHT
RATCFSTSSDTYACWNHSGVGFYVYNPFMIDVQQWGTGNLQSNHDLVCQVHGNAHVASCDAIMTRCLAVHECFVKRVDWSVEYPIIGDELRVNSAC

```

Şekil 27 Polyprotein proteinlerinin hizalanmış halinin dosyasını baş kısmı

```

Polyprotein_NWA.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım

MYCSFYPPDEDEEEEGDCEEEFEPSTQYEGTEDDYQGKPLEFGATSAALQPEEEQEEEDWLDDSQQTVGQQDGSSEDNQTTTIQTIVEVQPQLEMEL
.TGYKKPASRELKVTFFPDNLNGDVVAIDYKHYPSTFKKGAKLLHKPIVWHVNNATNKATYKPNWCI RCLWSTKPVETSNSFDVLKSEDAQGMNDLA
/LEGSVRVTTFDSEYCRHGTCESEAGVCVSTSGRWLNNNDYRSLPGVFCGVDANLLTNMFTPLIQPIGALDISASIVAGGIVAVVTCLAYYF
JRDAAQMRKLEKMAQAMTQMYKQARSEDKRAKVTSAMQTMLFTMLRKLDNDALNNIINNARDGCVPLNIIPLTAAKLMVVIPDYNTYKNTCDGTT
RAMPNMLRIMASLV LARKHTTCCSLSHRFYRLANCAQVLSEMMCGGSLYVKPGGTSSGDATTAYANSVFNICQAVTANVNALLSTDGKNKIADKYV
/NLVAVPTGYVDTNNNTDFSRVSAKPPPGDQFKHLIPLMYKGLPWNVRIKIVQMLSDTLKGLSDRVVFLWAHGFE LTSMKYFVKIGPERTCCLCD
JINDMILSLLSKGRLIIRENNRVVSSDVLVNN

.DEWSVATFYLFDDAGEENFSSRMYCSFYPPDEEEEDDAECEEEIDECEHEGYTEDDYQGLPLEFGASAETVRVEEEEDWLDDTTEQSEIEPE
TGFTKPPASRELSVTFPDNLNGDVVAIDYRHYSASFKKGAKLLHKPIVWHINQATTKTTFKPNTWCLRCLWSTKPVDTSNSFEVLAVEDTQGMNDLAC
.EGSVRVTTFDSEYCRHGTCESEAGVCVSTSGRWLNNNDYRSLPGVFCGVDAMNLIANIFTPLVQPVGALDVSASVAGGIIAILVTC AAYYFM
JRDAAQMRKLEKMAQAMTQMYKQARSEDKRAKVTSAMQTMLFTMLRKLDNDALNNIINNARDGCVPLNIIPLTAAKLMVVIPDYNTYKNTCDGNTF
AMPNMLRIMASLV LARKHTTCCSLSHRFYRLANCAQVLSEMMCGGSLYVKPGGTSSGDATTAYANSVFNICQAVTANVNALLSTDGKNKIADKYVR
/NLVAVPTGYVDTNNNTDFSRVSAKPPPGDQFKHLIPLMYKGLPWNVRIKIVQMLSDTLKGLSDRVVFLWAHGFE LTSMKYFVKIGPERTCCLCDK
JINDMIYSLLEKGRLLIIRENNRVVSSDI-LVNN

```

Şekil 28 Polyprotein proteinlerinin hizalanmış halinin dosyasını son kısmı

Dosya içerisindeki nCovid ve SARS polypeptit sekanslarının tamamı aşağıdaki gibidir.

nCovid:MESLVPGFNEKTHVQLSLPVLQVRDVLVRGFGDSVEEVLSEARQHLKDGTCGLVEVEK
 GVLPQLEQPYVFIKRS DARTAPHGHVMVELVAELEGIQYGRSGETLGVLPVHVGEIPVAYRKVL
 LRKNGNKGAGGHSYGADLKSFDLGDDELGTDPEYDFQENWNTKHSSGV TRELMRELNGGAYTR
 YVDNNFCGPDGYPLECIKDLLARAGKASCTLSEQLDFIDTKRGVYCCREHEHEIAWYTERSEKSY
 ELQTPFEIKLAKKFDTFNGECPNFVPLNSIIKTIQPRVEKKKLDGFMGRIRSVYPVASEPNECNQM
 CLSTLMKCDHCGETSWQTGDFVKATCEFCGTENLTKEGATTCGYLPQNAVVKIYCPACHNSEV
 GPEHSLAEYHNESGLKTILRKGGRTIAFGGCVFSYVGCHNKCA YWVPRASANIGCNHTGVVGE
 SEGLNDNLLEILQKEKVNINIVGDFKLNIEIAIILASFSASTSAFVETVKGLDYKAFKQIVESC
 GNFKVTGKAKKGAWNIGE QKSILSPLYAFASEAARVVRISFRTLETAQNSVRVLQKAAITIL
 DGISQYSLRLIDAMMFTSDLATNNLVVMAYITGGVVQLTSQWLTNIFGTVYEKLPVLDWLEEK
 FKEGVEFLRDGWEIVKFISTCACEIVGGQIVTCAKEIKESVQTFFKL VNKFLALCADSIIIGGA
 KKLALNLGETFVTHSKGLYRKCVKSREETGLLMPLKAPKEIIFLEGETLPTEVLTEEVVLKT
 GDLPLEQPTS EAVEAPLVGTPVCINGLMLLEIKDTEKYCALAPNMMVTNNTFTLKGGA
 PTKVTFGDDTVIEVQGYKSVNITFELDERIDKVLNEKCSAYTVELGTEVNEFACV VADAVIKTLQ
 PVSELLTPLGIDLDEWSMATYYLFDESGEFKLASHMYCSFYPPDEDEEEGDCEEEEFEPSTQ
 YEYGTEDDYQGKPLEFGATS AALQPEEEQEEDWLDDDSQQT VGGQDGSEDNQTTTIQTIVEV
 QPQLEMELTPVVQTIEVNSFSGY LKLTNDNVYIKNADIVEEAKKVKPTVVVNAANVYLKHG
 GGAVAGALNKATNNAMQVESDDYIAT NGPLKVGGSCVLSGHNLA KHCHLVVGPVNKGE
 DIQLLKSAYENFNQHEVLLAPLLSAGIFGAD PIHSLRVCVDTVRTNVYLAVFDKNLYDKL
 VSSFLEMKSEKQVEQKIAEIPKEEVKPFITESKPSVE QRKQDDKKIKACVEEVTTTLEET
 KFLTENLLL YIDINGNLHPDSATLVSDIDITFLKKDAPYIVGD VVQEGVLTA VVIPTK
 KAGGTTEMLAKALRKVPTDNYITTPGQGLNGYTVEEAKTVLKKCKSA FYILPSIISNEK
 QEILGTVSWNLREMLAHAEETRKLMPVCVETKAIVSTIQRKYKGIKIQEGVVDY GARFY
 FYTSKTTVASLINTLNDLNETLVTMPLGYVTHGLNLEEAARYMRS LKVPATVSVSPDA
 VTA YNGYLTSSSKTPEEHFIETISLAGSYKDWYSYSGQSTQLGIEFLKRGDKSVYYTSN
 PTTFHLDG EVITFDNLKTLTSLREVRTIKVFTTVDNINLHTQVVDMSMTYGGQFGPT
 YLDGADVTIKPHNSH EGKTFYVLPNDDTLRVEAFEYYHTTDP SFLGRYMSALNHTKKW
 KYPQVNGLT SIKWADNNCYL ATALLTLQQIELKFNPPALQDAYYRARAGEAANFCAL
 ILAYCNKTVGELGDVRETMSYLFQHAN LDSCRVLNVVCKTCGQQQTTLKGVEAVMYM
 GTLSYEQFKKGVPCTCGKQATKYL VQQES PFVMMSAPPAQYELKHGTFTCASEYT
 GNYQCGHYKHITSKETLYCIDGALLTKSSEYKGPITDVF YKENSYTTTIKPVTYKLDG
 VVCTEIDPKLDNYYKKDNSYFTEQPIDLVNPQYPYNASFDNFK FVC DNKFADDLNLQ
 LTGYKKPASRELKVTFPPDLNGDVVAIDYKHYTPSFKKGAKLLHKPIVWHVNN ATNKAT
 YKPNTWCIRCLWSTKPVETSNSFDVLKSEDAQGM DN LACEDLKPVSEEVVENPTIQKD
 VLECNVKTTEVVGDIILKPANNLSKITEEVGHTDLMAAYVDNSSLTIKKPNELSRVLGLK
 TLATH GLAAVNSVPWDTIANYAKPFLNKVVSTTTNIVTRCLNRVCTNYMPYFFTL
 LLQLCTFTRSTNSRI KASMPTTIAKNTVKS V GKFCLEASFNYL KSPNFSKLINIIWFL
 LLSVCLGSLIYSTAALGV LMSNL GMPSYCTGYREGYLNSTNVTIATYCTGSIPCSVCL
 SGLDSLDTYPSLETIQITISSFKWDLTAFGLV AEWFLAYILFTRFFYVLGLAAIMQL
 FFSYFAVHFISNSWLMWLIINLVQMAPISAMVRMYIFFASF YYVWKS YVHVVDGCNS
 STCMCYKRN RATRVECTTIVNGVRRSFYVYANGGKGFC LHNWN CVNCDTFCAGSTFIS
 DEVARDSLQFKRPINPTDQSSYIVDSVTVKNGSIHLYFDKAGQKTYERHS LSHFVNLD
 NLRANNTKGS LPINVIVFDGKSKCEESSAKSASVYYSQLMCQPILLDQALVSDVGD
 SAEVAVKMFDAYVNTFSSTFNVPMEKLKTLVATAEAE LAKNVSLDNVLSTFISAARQGF
 VSDSV ETKD VVECLKLSHQSDIEVTGDSCNNYMLTYNKVENMTPRDLGACIDCSARH
 INAQVAKSHNIA LIWNVKDFMSLSEQLRKQIRSAAKKNL PFKLTCAATTRQVVNVVTTK
 IALKGKIVNNWLKQLI KVTLVFLFVAIFYLITPVHVM SKHTDFSSEIIGYKAIDGGVTR
 DIASTDTCFANKHADFDTWFSQ RGGSYTNDKACPLIAAVITREVG FVVPGLPGTILRTN
 GDFLHFLPRVFSAVGNICYTPSKLIEYTD FATSACVLAAECTIFKDASGKPVPCYDTN
 VLEGSVAYESLRPDTRYVLM DGSIIQFPNTYLEGS VRVVTTFDSEYCRHGT CERSEAG
 VCVSTSGRWVLNNDY YRSLPGVFCGVDAVNLLTNMFTPLIQ PIGALDISASIVAGGIV
 AIVVTCLAYYFMRFRRAFGEYSHVVAFN TLLFLMSFTVLCLTPVYSFLP GVYSVIY
 LLYLTFYLTNDVSFLAHIQWMVMFTPLVPFWITIA YIICISTKH FYWFFSNY LKRRV
 VFN GVSFSTFEEAALCTFLLNKEMYLKL RSDVLLPLTQYNNRYLALYNKYKYFSGAM
 DTTSYREAACC HLA KALNDFSNSGSDVLYQPPQTSITSAVLQSGFRKMAFPSGKVEGCMV
 QVTCGTTTTLNLWLD

FASTA SEKANS DOSYALARI

DVVYCPRHVICTSEDMLNPNYEDLLIRKSNHNFLVQAGNVQLRVIGHSMQNCVLKLKVDATANP
KTPKYKFVRIQPGQTFSVLACYNGSPSGVYQCAMRPNFTIKGSFLNGSCGSVGFNIDYDCVSFCY
MHHEMELPTGVHAGTDLEGNFYGPFVDRQTAQAAGTDTTITVNVLAWLAAVINGDRWFLNRF
TTTTLNDFNLVAMKYNIEPLTQDHVDILGPLSAQTGIAVLDMCASLKELLQNGMNGRTILGSALL
EDEFTPFDVVRQCSGVTFQSAVKRTIKGTHHWLLLILTSLVLVQSTQWSLFFFLYENAFPLFA
MGIIAMSAFAMMFVKHKHAFCLCLLPLSATVAYFNMVYMPASWVMRIMTWLDMVDTSLSGF
KLKDCVMYASAVVLLILMTARTVYDDGARRVWTLMNVLTLVYKVYYGNALDQAISMWALIIS
VTSNYSYGVVTTVMFLARGIVFMCVEYCPFITGNTLQCIMLVYCFGLGYFCTCYFGLFCLLNRYFR
LTLGVYDYLSTQEFRYMNSQGLLPKNSIDAFKLNKLLGVGGKPCIKVATVQSKMSDVKCTS
VVLLSVLQQLRVESSSKLWAQCVQLHNDILLAKDTTEAFEKMSVLLSVLLSMQGAVDINKLCEE
MLDNRATLQAIASEFSSPSYAAFATAQEAYEQAVANGDSEVVLKLLKKSLSNVAKSEFDRDAA
MQRKLEKMADQAMTQMYKQARSEDKRAKVTSAMQTMFTMLRKLDNDALNNIINNARDGCV
PLNIPLTTAAKLMVIPDYNTYKNTCDGTTFTYASALWEIQQVVDADSKIVQLSEISMDNSPNLA
WPLIVTALRANSVVKLQNNELSPVALRQMSCAAGTTQACTDDNALAYYNTTKGGRFVLALLS
DLQDLKWARFPKSDGTGTIYTELEPPCRFVTDTPKGPVKYLYFIKGLNNLNRMVGLGSLAATV
RLQAGNATEVPANSTVLSFCAFAVDAAKAYKDYLASGGQPITNCVKMLCTHTGTGQAITVTPEA
NMDQESFGGASCCLYCRCHIDHPNPKGFCDLKGKYVQIPTTCANDPVGFTLKNVTCTVCGMWK
GYGCSCDQLREPMLQSADAQSFLNRVCGVSAARLTPCGTGTSTDVVYRAFDIYNDKVAGFAKF
LKTNCCRFEKDEDDNLIDSYFVVKRHTFSNYQHEETIYNLLKDCPAVAKHDFKFRIDGDMVP
HISRQRLTKYTMADLVYALRHFDEGNCDTLKEILVTYNCCDDDYFNKKDWYDFVENPDILRVY
ANLGERVRQALLKTVQFCDAMRNAGIVGVLTLDNQDLNGNWYDFGDFIQTTPGSGVPVVDSSYY
SLLMPILTLTRALTAESHVDTDLTKPYIKWDLKYDFTEERLKLFDYFYKYWDQTYHPNCVNCL
DDRCILHCANFNVLSTVFPPTSFGPLVRKIFVDGVFPVSTGYHFRELGVVHNQDVNLHSSRSLF
KELLVYAADPAMHAASGNLLLDKRTTCFSAALTNNAFQTVKPGNFNKDFYDFAVSKGFFKE
GSSVELKHFFFAQDGNAAISDYDYRYNLPTMCDIRQLLFVVEVVDKYFDCYDGGCINANQVIV
NNLDKSAGFPFNKWKARLYYDSMSYEDQDALFAYTKRNVITITQMNLYAISAKNRARTVA
GVSICTMTNRQFHQKLLKSIAATRGA TVVIGTSKFYGGWHNMLKTVYSDEVNPHLMGWDYYPK
CDRAMPNMLRIMASLV LARKHTTCCSLSHRFYRLANCAQVLSEVMCMGGS LYVKPGGTSSGD
ATTAYANSVFNICQAVTANVNALLSTDGNKIADKYVRNLQHRLYECLYRNRDVTDFVNEFYA
YLRKHFSMMILSDDAVVCFNSTYASQGLVASIKNFKSVLYYQNNVFMSEAKCWETD LTKGPH
EFCSQHTMLVKQGDDYVYLPYPDPSRILGAGCFVDDIVKTDGTLMIERFVSLAIDAYPLTKHPNQ
EYADV FHLYLQYIRKLHDEL TGHMLDMYSVMLTNDNTSRYWEPEFYEAMYPHTVLQAVGAC
VLCNSQTSRLCGACIRRPFLCCKCCYDHVISTSHKL VLSVNPYVCNAPGCDVTDVTQLYLGMS
YYCKSHKPPISFPLCANGQVFGLYKNTCVGSDNVTDFNAIATCDWTNAGDYILANTCTERLKL
AAETLKATEETFKLSYGIATVREVLSDRELHLSWEVGKPRPPLNRNYVFTGYRVTKNSKVQIGEY
TFEKGDYGDAAVYRGTTTYKLVNGDYFVLTSHTVMPLSAPTLVPQEHYVRITGLYPTLNISDEFS
SNVANYQKVGMQKYSTLQGPPTGKSHFAIGLALYYP SARIVYTACSHA AVDALCEKALKYLPI
DKCSRIIPARARVECFDKFKVNSTLEQYVFCTVNALPETTADIVVFDEISMATNYDLSVVNARLR
AKHYVYIGDPAQLPAPRTLLTKGTLEPEYFNSVCRLMKTIGPDMFLGTCCRCPAEIVDTV SALVY
DNKLKAHKDKSAQCCKMFYKGVITHDVSSAINRPQIGVVREFLTRNPAWRKAVFISPYNSQNAV
ASKILGLPTQTVDSSQGSEYDYVIFTQTETAHSCNVNRFNVAITRAKV GILCIMS DRDLYDKLQF
TSLEIPRRNVATLQAENV TGLFKDCSKVITGLHPTQAPTHLSVDTKFKTEGLCVDIPGIPKDMTYR
RLISM MGFKMNYQVNGYPNMFITREEAIRHVR AWIGFDVEGCHATREAVGTNLPLQLGFSTGV
NLVAVPTGYVDTPNNTDFSRVSAKPPPGDQFKHLIPLMYKGLPWNVVRIVQMLSDTLKNLSD
RVV FVLWAHGFELTSMKYFVKIGPERTCCLCDRRATCFSTASDTYACWHHSIGFDYVYNPFMID
VQQWGFTGNLQSNHDLYCQVHGNAHVASCDAIMTRCLAVHECFVKRVDWTIEYPIIGDELKIN
AACRKVQH MVVKAALLADKFPVLHDIGNPKAIKCPQADVIEWKFYDAQPCSDKAYKIEELFYS
YATHSDKFTDGVCLFWNCNVDRYPANSIVCRFDTRVLSNLNLP GCDGGS LYVNKHAFHTPAFD
KSAFVNLKQLPFFYYS DPCESHGKQVVSIDIDYVPLKSATCITRCNLGGAVCRHHANEYRLYLD
AYNMMISAGFSLWVYKQFDTYNLWNTFTRLQSL ENVAFNVVNKGHFDGQQGEVPVSIINNTVY
TKVDGVDVELFENKTTLPVNVAFELWAKRNIKPVPEVKILNNLGV DIAANTVIWDYKRDAPAH
STIGVCSMTDIAKKPTETICAPLTVFFDGRVDGQVDLFRNARNGLITEGSVKGLQPSVGPKQAS
LNGVTLIGEAVKTQFNYYKKVDGVVQQLPETYFTQSRNLQEFKPRSQMEIDFLELAMDEFIERY
KLEGYAFEHIVYGDFSHSQLGGLHLLIGLAKRFKESPELEDFIPMDSTVKNYFITDAQTGSSKCV

FASTA SEKANS DOSYALARI

CSVIDLLDDFVEIHSQDLSVVSXVVKVTIDYTEISFMLWCKDGHVETFYPKLQSSQAWQPGVA
MPNLYKMQRMLLEKCDLQNYGDSATLPKGIMMNVAKYTQLCQYLNTLTLAVPYNMRVIHFGA
GSDKGVAPGTAVLRQWLPTGTLLVDSDLNDFVSDADSTLIGDCATVHTANKWDLIISDMYDPKT
KNVTKENDSKEGFFTYICGFIQKLALGGSVAIKITEHSWNADLYKLMGHFAWWTAFVTNVNA
SSSEAFLIGCNYLKGPREQIDGYVMHANYIFWRNTNPIQLSSYSLFDMSKFPLKLRGTAVMSLKE
GQINDMILSLLSKGRLLIRENNRVVISSDVLVNN

SARS :-----M---

ESLVLGVNEKTHVQLSLPVLQVRDVLVRGFGDSVEEALSEAREHLKNGTCGLVELEKGVLPQLE
QPYVFIKRS DALSTNHGHKVVELVAEMDGIQYGRSGITLGVLPVHVGETPIAYRNVLLRKNGNK
GAGGHSYGIDLKSYDLGDELGTDPIDYEQNWNTKHGSGALRELTRELNGGAVTRYVDNNFCG
PDGYPLDCIKDFLARAGKSMCTLSEQLDYIESKRGVYCCRDHEHEIAWFTERSDKSYEHQTPFEI
KSAKKFDTFKGECPKFVPLNSKVVKVIQPRVEKKKTEGFMGRIRSVYPVASPQECNNMHLSTLM
KCNHCDEVSWQTCDFLKATCEHCGTENLVIEGPTTCGYLPTNAVVKMPCPACQDPEIGPEHSVA
DYHNHSNIETRLRKGGTRCFGGCVFAYVGCYNKRAYWVPRASADIGSGHTGITGDNVETLNE
DLLEILSRERVNINIVGDFHLNEEVAILASFASSTAFIDTIKSLDYKSFKTIVESCNGYKVTGKGP
VKGAWNIGQQRSVLTPLCGFPSQAAGVIRSIFARTLDAANHSIPDLQRAAVTILDGISEQSLRLVD
AMVYTSDLLTNSVIIMAYVTGGLVQQTSQWLSNLLGTTVEKLRPIFEWIEAKLSAGVEFLKDAW
EILKFLITGVFDIVKGQIQVASDNIKDCVKCFIDVVKALEMCIDQVTIAGAKLRSLNLGEVFIAQS
KGLYRQCIRGKEQLQLLMPLKAPKEVTFLEGDSHDTVLTSEEVLKNGELEALETPVDSFTNGAI
VGTPVCVNGLMLLEIKDKEQYCALSPGLLATNNVFRLLKGGAPIKGVTFGEDTVWEVQGYKNVR
ITFELDERVDKVLNEKCSVYTVESGTEVTEFACVVAEAVVKTLQPVSDLLTNMGIDLDEWSVAT
FYLFDDAGEENFSSRMYSFYPPDEEEEDDAECEEEIIDETCEHEYGTEDDYQGLPLEFGASAET
VRVEEEEEEDWLDDTTEQSEIEPEPEPTPEEPVNQFTGYLKLTDNVAIKCVDIVKEAQSANPMVIV
NAANIHLKHGGGVAGALNKATNGAMQKESDDYIKLNGPLTVGGSCLLSGHNLAKKCLHVVG
NLNAGEDIQLLKAA YENFNSQDILLAPLLSAGIFGAKPLQSLQVCVQTVRTQVYIAVNDKALYEQ
VVM DYLDNLKPRVEAPKQEEPPNTEDSKTEESVQKPVDPVKPKIKACIDEVTTTLEETKFLT
LLLFADINGKLYHDSQNMLRGDDMSFLEKDAPYMGDVITSGDITCVVIPSKKAGGTTEMLSRA
LKKVPVDEYITTPGQGCAGYTLEEAKTALKKCKSAFYVLPSEAPNAKEEILGTVSWNLREMLA
HAEETRLMPICMDVRAIMATIQRKYKGIKIQEGIVDYGVRFFFTYSKEPVASIITKLNSLNEPLVT
MPIGYVTHGFNLEEAA RCMRSLKAPAVVSVSSPDAVTTYNGYLTSSSKTSEEHFVETVSLAGSYR
DWSYSGQRTELGVFLKRGDKIVYHTLESPVEFHLDEVLSDKLKSLSLREVKTIVFTTVDN
TNLHTQLVDMSMTYGGQFGPTYLDGADVTKIKPHVNHEGKTFFVLPSDDTLRSEAFEYYHTLDE
SFLGRYMSALNHTKKWKFPQVGGLTSIKWADNNCYLSSVLLALQQLEVKFNAPALQEAYYRAR
AGDAANFCALILAYS NKTVGELGDVRETMTLLQHANLES AKRVLNVVCKHCGQKTTTTLTGVE
AVMYMGTL SYDNLKTGVSIPCVCGRDATQYL VQQESSFVMMSAPPAEYKLQQTFLCANEYTG
NYQCCHYTHITAKETLYRIDGAHLTKMSEYKGPVTDVFYKETS YTTTTIKPVSYKLDGVTYTEIEP
KLDGYYKKDNAYYTEQPIDLVPTQPLPNASFDNFKLTCSNTKFADDLNQMTGFTKPASRELSVT
FFPDLNGDVVAIDYRHYSASFKKGAKLLHKPIVWHINQATTKTTFKPNTWCLRCLWSTKPVDT
NSFEVLAVEDTQGM DN LACESQQPTSEEVENPTIQKEVIECDVKTTEVVGNVILKPSDEGVKVT
QELGHEDLMAAYVENTSITIKPNELSLALGLKTIATHGIAAINSVPWSKILAYVKPFLGQAAITT
SNCAKRLAQRVFNNYMPYVFTLLFQLCTFTKSTNSRIRASLPTTIKNSVKSVAKLCLDAGINYV
KSPKFSKLFTIAMWLLLLSICLGS LICVTAAGVLLSNFGAPSYCNGVRELYLNSSNVTTMDFCEG
SFPCSICLSGLDSLDSYPALETIQVTISSYKLDLTILGLAAEWVLA YMLFTKFFYLLGLSAIMQVFF
GYFASHFISNSWLMWFIIIVQMAPVSAMVRMYIFFASFYIWKSYVHIMDGCTSSCTMCMCYKR
NRATRVECTTIVNGMKRSFYVYANGGRGFCKTHNWNCLNCDTFCSTGFISDEVARDLSLQFKR
PINPTDQSSYIVDSVAVKNGALHLYFDKAGQKTYERHPLSHFVNLDNLRANNTKGS LPIN VIVFD
GKSKCDESASKSASVYYSQLMCQPILLDDQALVSDVGDSTEVSVKMFDAYVDTF SATFSVPMEK
LKALVATAHSELA KGVALDGVLSTFVSAARQGVVDTDVDTKD VIECLKLSYHSDLEVTGDSCN
NFMLTYNKVENMTPRDLGACIDCNARHINAQVAKSHNVSLIWNVKDYMSLSEQLRKQIRSAAK
KNNIPFRLTCATTRQVVNVITTKISLKGGKIVSTCFKLMLKATLLCVLAALVCYIVMPVHTLSIHD
GYTNEIIGYKAIQDGVTRDIISTDDCFANKHAGFDAWFSQRGGSYKNDKSCPVVAAITREIGFIVP

FASTA SEKANS DOSYALARI

GLPGTVLRAIN GDFLHFLPRVFS AVGNICYTPSKLIEYSDFATSACVLAAECTIFKDAMGKPV PYC
YDTNLL EGSISYSEL RPDTRYV LMDGSIIQFPNTYLEGSVRVVTTFDAEYCRHGT CERSEV GICLS
TSGRWVLNNEHYRALSGVFCGVDAMNLIANIFTPLVQPVGALDVSASV VAGGIIAILVTCAAYY
FMKFRRVFG EYNHVVAANALLFLMSFTILCLVPAYSFLPGVYSVFYL YLTFYFTNDVSFLAHLQ
WFAMFSPIVPFWITAIYVFCISLKHCHWFFNNYL RKRVMFNGVTFSTFEEAALCTFLLNKEMYLK
LRSETLLPLTQYNRYLALYNKYKYFSGALD TTSYREAACCHLAKALNDFSNSGADVLYQPPQTS
ITSAVLQSGFRKMAFPSGKVEGCMVQVTCGTTTLNGLWLD DTVYCPRHVICTAEDMLNP NYED
LLIRKSNH SFLVQAGNVQLRVIGHSMQNC LLRLKVDTSNPKTPKYKFVRIQPGQTFSVLACYN GS
PSGVYQCAMRPNHTIKGSFLNGSCGSVGFNIDYDCVSFCYMHMELPTGVHAGTDLE GKFYGP F
VDRQTAQAAGTDTTITLNVLA WLYAAVINGDRWFLNRFTTTLND FNLVAMKYNYEPLTQDHV
DILGPLSAQTGIAVLDMCAALKELLQNGMNGRTILGSTILEDEFTPF DVVRQCSGVTFQGKF KKI
VKGTHHWMLLTFLTSL LILVQSTQWSLFFFVYENAFLPFTLGIMAI AACAMLLVKHKHAF LCLFL
LPSLATVAYFNMVYMPASWVMRIMTWLELADTSLSGYRLKDCVMYASALVLLILMTARTVYD
DAARRVWTL MNVITLVYKVYYGNALDQAISMWALVISVTSNYS GSVTTIMFLARAI VFVCVEY
YPLLFITGNTLQCIMLVYCF LGYCCCCYFGLFCLLNRYFRLTLG VYDYL VSTQEF RYMNSQGLLP
PKSSIDAFKLN ILLGIGGKPCIKVATVQSKMSDVKCTSVVLLSVLQQLRV ESSSKLWAQCVQLH
NDILLAKDTTEAFEKMSVLLSVLLSMQGAVDINRLCEEMLDNRATLQAIASEFSSLP SYAAYATA
QEAYEQAVANGDSEVVLK LKKSLNVAKSEFDRDAAMQRKLEK MADQAMTQMYKQARSEDK
RAKVTSAMQTM LFTMLRKLDNDALNNIINNARDGCVPLNIPLTTAAKLMVVVPDYGT YKNTC
DGNFTFYASALWEIQQVVDADSKIVQLSEINMDNSPNLAWPLIVTALRANS AVKLQNNELSPVA
LRQMSCAAGTTQTACTDDNALAYYNN SKGGRFVLALLSDHQDLKWARFPKSDGTGTIYTELEP
PCRFTVTDTPKGPKVKYL YFIKGLNNLN RGMVLGSLAATVRLQAGNATEVPANSTVLSFCAFAVD
PAKAYKDYLA SGGQPITNCVKMLCTHTGTGQAITVTP EANMDQESFGGASCCLYCRCHIDHPNP
KGFCDLKGKYVQIPTTCANDPVGFTLRNTVCTVCGMWKG YGCSCDQLREPLMQSADASTFLNR
VCGVSAARLTPCGTGTSTDVVYRAFDIYNEKVAGFAKFLKTNCCRFQEKDEEGNLLDSYFVVKR
HTMSNYQHEETIYNL VKDCPAVA VHDFFKFRVDGDMVPHISRQRLTKYTMADLVYALRH FDEG
NCDTLKEILVTYNCCDDDYFNKKDWYDFVENPDILRVYANLGERVRQSLLKTVQFCDAMRDAG
IVGVLTLDNQDLNGN WYDFGDFVQVAPGCGVPIVDSYYSLLMPILTLTRALAAESHMDADLAK
PLIKWDLLKYDFTEERLCLFD RYFKYWDQTYHPNCINCLDDRCILHCANFNVLSTVF PPTSFGPL
VRKIFVDGVFPFVSTGYHFRELGVVHNQDVNLHSSR LSFKELLVYAADPAMHAASGNLLLDKR
TTCFSVAALTN NVAFQTVKPGNFNKDFYDFAVSKGFFKEGSSVELKHFFFAQDGNAAISDYDYY
RYNLPTMCDIRQLLFVVEVVDKYFDCYDGGCINANQVIVNNLDKSAGFPFNKWKARLYYDSM
SYEDQDALFAYTKRNV IPTITQMNLKYAISAKNRARTVAGVSICSTMTNRQFHQKLLKSIAATRG
ATVVIGTSKFYGGWHNMLKTVYS D VETPHLMGWDYPKCDRAMPNMLRIMASLV LARKHNTC
CNLSHRFYRLAN ECAQVLSEMVMCGGSLYVKPGGTSSGDATTAYANSVFNICQAVTANVNALL
STDGNKIADKYVRNLQHRLYECLYRNRD VDHEFVDEFYAYLRKHFSMMILSDDAVVCYNSNYA
AQGLVASIKNFKA VLYYQNNVFMSEAKCWTE TDLTKGPHEFCSQHTMLVKQGDDYVYLPYPD
PSRILGAGCFVDDIVKTDG TLMIERFVSLAIDAYPLTKHPNQEYADV FHLYLQYIRKLHDEL TGH
MLDMYSVMLTNDNTSRYWEPEFY EAMYTPHTVLQAVGACVLCNSQTS LRCGACIRRPFLCCKC
CYDHVISTSHKL VLSVNPYVCNAPGCDVTDVTQLYLGMSY YCKSHKPPISFPLCANGQVFGLY
KNTCVGSDNVTDFNAIATCDWTNAGDYILANTCTERLKLFAAETL KATEETFKLSYGIATVREV
LSDRELHLSWEVGKPRPPLNRNYVFTGYRVTKNSKVQIG EYTFEKGDYGD AVVYRGTTTYKLN
VG DYFVLTSHTVMPLSAPTLVPQEHYVRITGLYPTL NISDEFSSNVANYQKVG MQKYSTLQGP
GTGKSHFAIGLALY YPSARIVYTACSHAAVDALCEKALKYLPIDKCSRIIPARARVECFDKFKVNS
TLEQYVFCTVNALPETTADIVVFDEIS MATNYDLSVVNARLR AKHYVYIGDPAQLPAPRTLLTKG
TLEPEYFNSVCRLMKTIGPDMFLGTCCRCPAEIVDTVSALVYDNKLKAHKDKSAQCFKMFYKG
VITHDVSSAINRPQIGV VREFLTRNPAWRKAVFISPYNSQNAVASKILGLPTQTVDSSQGSEYDYV
IFTQTTETAHSCNVNRFNVAITRAKIGILCIMS DRDLYDKLQFTSLEIPRRNVATLQAENV TGLFK
DCSKIITGLHPTQAPTHLSVDIKFKTEGLCVDIPGIPKDMTYRR LISMMGFKMNYQVNGYPNMFIT
REEAIRHVRAWIGFDVEGCHATRDAVG TNLPLQLGFSTGVNLVAVPTGYVDTENNT EFTRVNA
KPPPGDQFKHLIPLMYKGLPWNVVR IKIVQMLSDTLKGLSDRVV FVLWAHG FELTSMKYFVKIG
PERTCCLCDKRATCFSTSSD TYACWNH SVGFDYVYNPF MIDVQQWGFTGNLQSNHDQHCQVH
GNAHVASCDAIMTRCLAVHECFVKRVDWSVEYPIIGDEL RVNSACRKVQHMMV VKSALLADKFP
VLHDIGNPKAIKCVPAEVEWKFYDAQPCSDKAYKIEELFYSYATH HDKFTDGVCLFWNCNVD

RYPANAIVCRFDTRVLSNLSNLPDGGSLYVNKHAFHTPAFDKSAFTNLKQLPFFYYSDSPCESH
GKQVVSDDIDYVPLKSATCITRCNLGGAVCRHHANEYRQYLDAYNMMISAGFSLWIYKQFDTYN
LWNTFTRLQSLENAVYNVVKGHFDGHAGEAPVSIINNAVYTKVDGIDVEIFENKTTLPVNVAF
ELWAKRNIKPVEIKILNNLGVDAANTVIWDYKREAPAHVSTIGVCTMTDIAKKPTESACSSLTV
LFDGRVEGQVDLFRNARNGVLITEGSVKGLTPSKGPAQASVNGVTLIGESVKTQFNYFKKVDGII
QQLPETYFTQSRDLEDFKPRSQMETDFLELAMDEFIQRYKLEGYAFEHIVYGDFSHGQLGGLHL
MIGLAKRSQDSPLKLEDFIPMDSTVKNYFITDAQTGSSKCVCSVIDLLDDFVEIISQDLSVISKV
VKVTIDYAEISFMLWCKDGHVETFPYKQLASQAWQPGVAMPNLYKMQRMLLEKCDLQNYGEN
AVIPKGIMMNVAKYTQLCQYLNTLTAVPYNMRVIHFGAGSDKGVAPGTAVLRQWLPTGTLLV
DSDLNDFVSDADSTLIGDCATVHTANKWDLISDMYDPRTKHVTKENDSKEGFFTYLCGFIKQKL
ALGGSIAVKITEHSWNADLYKLMGHFSWWTAFVTNVNASSSEAFBIGANYLGKPKEQIDGYTM
HANYIFWRNTNPIQLSSYSLFDMSKFPLKLRGTAVMSLKENQINDMIYSLLEKGRLIIRENNRVVV
SSDI-LVNN

1.7 Program Kodunun Tamamı

1.7.1 Dosyaİslemleri Modülü

```
import Needleman_Wunsch

class FASTA_DOSYASI():
    def __init__(self, dosya_adi, dosya_yolu):
        self.dosya_adi = dosya_adi
        self.dosya_yolu = dosya_yolu

    def genomDosyaDonusturucu(self):
        fasta_dosyasi = open(self.dosya_yolu, "r")
        data = fasta_dosyasi.read()
        sekans_baslangic = data.find("genome")
        dosya = open('C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\'+
self.dosya_adi + '.txt', "w")
        dosya.write(data[sekans_baslangic+6:])

    def proteinDosyaDonusturucu(self):
        fasta_dosyasi = open(self.dosya_yolu, "r")
        data = fasta_dosyasi.read()
        protein_dizilim_baslangic = data.find("[")
        dosya = open('C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\' +
self.dosya_adi + '.txt', "w")
        dosya.write(data[protein_dizilim_baslangic+1:])

    def dosyaOkuma(self):
        dosya = open(self.dosya_yolu, "r")
        sekanslar = dosya.read()
        sekanslar = sekanslar.replace("\n", "")
        sekanslar = sekanslar.replace("\r", "")
        self.orfBulNukleotit(sekanslar)
        self.donusturucu(sekanslar)

    def donusturucu(self, sekanslar):
        amino_asit_tablosu = {
            'ATA': 'I', 'ATC': 'I', 'ATT': 'I', 'ATG': 'M',
            'ACA': 'T', 'ACC': 'T', 'ACG': 'T', 'ACT': 'T',
            'AAC': 'N', 'AAT': 'N', 'AAA': 'K', 'AAG': 'K',
```



```

        'AGC': 'S', 'AGT': 'S', 'AGA': 'R', 'AGG': 'R',
        'CTA': 'L', 'CTC': 'L', 'CTG': 'L', 'CTT': 'L',
        'CCA': 'P', 'CCC': 'P', 'CCG': 'P', 'CCT': 'P',
        'CAC': 'H', 'CAT': 'H', 'CAA': 'Q', 'CAG': 'Q',
        'CGA': 'R', 'CGC': 'R', 'CGG': 'R', 'CGT': 'R',
        'GTA': 'V', 'GTC': 'V', 'GTG': 'V', 'GTT': 'V',
        'GCA': 'A', 'GCC': 'A', 'GCG': 'A', 'GCT': 'A',
        'GAC': 'D', 'GAT': 'D', 'GAA': 'E', 'GAG': 'E',
        'GGA': 'G', 'GGC': 'G', 'GGG': 'G', 'GGT': 'G',
        'TCA': 'S', 'TCC': 'S', 'TCG': 'S', 'TCT': 'S',
        'TTC': 'F', 'TTT': 'F', 'TTA': 'L', 'TTG': 'L',
        'TAC': 'Y', 'TAT': 'Y', 'TAA': '-', 'TAG': '-',
        'TGC': 'C', 'TGT': 'C', 'TGA': '-', 'TGG': 'W',
    }

    protein = ""
    for i in range(0, len(sekanslar), 3):
        if len(sekanslar) > i+3:
            kodon = sekanslar[i:i + 3]
            protein += amino_asit_tablosu[kodon]
    self.orfBulAminoAsit(protein)
    protein += "\n\n\n"
    for j in range(1, len(sekanslar) - 2, 3):
        if len(sekanslar) > (j + 3):
            kodon = sekanslar[j:j + 3]
            protein += amino_asit_tablosu[kodon]
    protein += "\n\n\n"
    for k in range(2, len(sekanslar) - 1, 3):
        if len(sekanslar) > (k + 3):
            kodon = sekanslar[k:k + 3]
            protein += amino_asit_tablosu[kodon]
    self.dosyaYazma(protein)

    def dosyaYazma(self, amino_asit_sekansi):
        dosya = open('C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\'+
self.dosya_adi + '_AA.txt', 'w')
        dosya.write(amino_asit_sekansi)

    def orfBulAminoAsit(self, protein):
        orf_protein = ""
        dosya = open('C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\'+
self.dosya_adi + '_AA_ORF.txt', 'w')
        for i in range(0, len(protein) - 1, 1):
            if (protein[i] == 'M'):
                i += 1
                while (protein[i] != '-' and i < len(protein) - 1):
                    orf_protein += protein[i]
                    i += 1
                if (protein[i] == '-'):
                    dosya.write(orf_protein + '\n\n')
                    orf_protein = ""
        dosya.write(orf_protein)

    def orfBulNukleotit(self, sekanslar):
        orf_sekans= []
        orf_gen= ""
        dosya = open('C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\'+
self.dosya_adi + '_GEN_ORF.txt', 'w')

```

FASTA SEKANS DOSYALARI

```
for i in range(0, len(sekanslar)-1, 3):
    if len(sekanslar) > i+3:
        kodon = sekanslar[i:i + 3]
        orf_sekans.append(kodon)

for i in range(0, len(orf_sekans)-1, 1) :
    if orf_sekans[i] == "ATG" and (i+1) < len(orf_sekans):
        i += 1
        while (orf_sekans[i] != "TAA") and (orf_sekans[i] != "TAG")
and (orf_sekans[i] != "TGA"):
            orf_gen += orf_sekans[i]
            if (i + 1) < len(orf_sekans):
                i+=1
            else:
                break
            orf_gen += "\n\n\n"
dosya.write(orf_gen)

def main():
    fasta_sars_nucleocapsid = FASTA_DOSYASI("SARS_Nucleocapsid",
"C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\SARS_Nucleocapsid.fasta")
    fasta_sars_nucleocapsid.proteinDosyaDonusturucu()

    fasta_sars_polyprotein =
FASTA_DOSYASI("SARS_Polyprotein", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\SARS_Polyprotein.fasta")
    fasta_sars_polyprotein.proteinDosyaDonusturucu()

    fasta_covid_nucleocapsid =
FASTA_DOSYASI("COVID_Nucleocapsid", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\COVID_Nucleocapsid.fasta")
    fasta_covid_nucleocapsid.proteinDosyaDonusturucu()

    fasta_covid_polyprotein =
FASTA_DOSYASI("COVID_Polyprotein", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\COVID_Polyprotein.fasta")
    fasta_covid_polyprotein.proteinDosyaDonusturucu()

    fasta_sars = FASTA_DOSYASI("SARS",
"C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\SARS.fasta")
    fasta_sars.genomDosyaDonusturucu()

    fasta_covid = FASTA_DOSYASI("COVID",
"C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\COVID.fasta")
    fasta_covid.genomDosyaDonusturucu()

    sars = FASTA_DOSYASI("SARS", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\SARS.txt")
    sars.dosyaOkuma()

    covid = FASTA_DOSYASI("COVID", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\COVID.txt")
    covid.dosyaOkuma()

    Needleman_Wunsch.farkBul("C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\COVID_Nucleocapsid.txt", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
```

```

Hesaplama\SARS_Nucleocapsid.txt", "Nucleocapsid")
    Needleman_Wunsch.farkBul("C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\COVID_Polyprotein.txt", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\SARS_Polyprotein.txt", "Polyprotein")

    Needleman_Wunsch.needlemanWunsch("C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\COVID_Nucleocapsid.txt", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\SARS_Nucleocapsid.txt", "Nucleocapsid")
    Needleman_Wunsch.needlemanWunsch("C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\COVID_Polyprotein.txt", "C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\SARS_Polyprotein.txt", "Polyprotein")

if __name__ == "__main__":
    main()

```

1.7.2 BLOSUM62 Modülü

```

def skorBul(satir_degeri , sutun_degeri):

    blosum62 = {
        'WF': 1, 'LR': -2, 'SP': -1, 'VT': 0,
        'QQ': 5, 'NA': -2, 'ZY': -2, 'WR': -3,
        'QA': -1, 'SD': 0, 'HH': 8, 'SH': -1,
        'HD': -1, 'LN': -3, 'WA': -3, 'YM': -1,
        'GR': -2, 'YI': -1, 'YE': -2, 'BY': -3,
        'YA': -2, 'VD': -3, 'BS': 0, 'YY': 7,
        'GN': 0, 'EC': -4, 'YQ': -1, 'ZZ': 4,
        'VA': 0, 'CC': 9, 'MR': -1, 'VE': -2,
        'TN': 0, 'PP': 7, 'VI': 3, 'VS': -2,
        'ZP': -1, 'VM': 1, 'TF': -2, 'VQ': -2,
        'KK': 5, 'PD': -1, 'IH': -3, 'ID': -3,
        'TR': -1, 'PL': -3, 'KG': -2, 'MN': -2,
        'PH': -2, 'FQ': -3, 'ZG': -2, 'XL': -1,
        'TM': -1, 'ZC': -3, 'XH': -1, 'DR': -2,
        'BW': -4, 'XD': -1, 'ZK': 1, 'FA': -2,
        'ZW': -3, 'FE': -3, 'DN': 1, 'BK': 0,
        'XX': -1, 'FI': 0, 'BG': -1, 'XT': 0,
        'FM': 0, 'BC': -3, 'ZI': -3, 'ZV': -2,
        'SS': 4, 'LQ': -2, 'WE': -3, 'QR': 1,
        'NN': 6, 'WM': -1, 'QC': -3, 'WI': -3,
        'SC': -1, 'LA': -1, 'SG': 0, 'LE': -3,
        'WQ': -2, 'HG': -2, 'SK': 0, 'QN': 0,
        'NR': 0, 'HC': -3, 'YN': -2, 'GQ': -2,
        'YF': 3, 'CA': 0, 'VL': 1, 'GE': -2,
        'GA': 0, 'KR': 2, 'ED': 2, 'YR': -2,
        'MQ': 0, 'TI': -1, 'CD': -3, 'VF': -1,
        'TA': 0, 'TP': -1, 'BP': -2, 'TE': -1,
        'VN': -3, 'PG': -2, 'MA': -1, 'KH': -1,
        'VR': -3, 'PC': -3, 'ME': -2, 'KL': -2,
        'VV': 4, 'MI': 1, 'TQ': -1, 'IG': -4,
        'PK': -1, 'MM': 5, 'KD': -1, 'IC': -1,
        'ZD': 1, 'FR': -3, 'XK': -1, 'QD': 0,
        'XG': -1, 'ZL': -3, 'XC': -2, 'ZH': 0,
        'BL': -4, 'BH': 0, 'FF': 6, 'XW': -2,
        'BD': 4, 'DA': -2, 'SL': -2, 'XS': 0,
        'FN': -3, 'SR': -1, 'WD': -4, 'VY': -1,
    }

```

```

'WL': -2, 'HR': 0, 'WH': -2, 'HN': 1,
'WT': -2, 'TT': 5, 'SF': -2, 'WP': -4,
'LD': -4, 'BI': -3, 'LH': -3, 'SN': 1,
'BT': -1, 'LL': 4, 'YK': -2, 'EQ': 2,
'YG': -3, 'ZS': 0, 'YC': -2, 'GD': -1,
'BV': -3, 'EA': -1, 'YW': 2, 'EE': 5,
'YS': -2, 'CN': -3, 'VC': -1, 'TH': -2,
'PR': -2, 'VG': -3, 'TL': -1, 'VK': -2,
'KQ': 1, 'RA': -1, 'IR': -3, 'TD': -1,
'PF': -4, 'IN': -3, 'KI': -3, 'MD': -3,
'VW': -3, 'WW': 11, 'MH': -2, 'PN': -2,
'KA': -1, 'ML': 2, 'KE': 1, 'ZE': 4,
'XN': -1, 'ZA': -1, 'ZM': -1, 'XF': -1,
'KC': -3, 'BQ': 0, 'XB': -1, 'BM': -3,
'FC': -2, 'ZQ': 3, 'XZ': -1, 'FG': -3,
'BE': 1, 'XV': -1, 'FK': -3, 'BA': -2,
'XR': -1, 'DD': 6, 'WG': -2, 'ZF': -3,
'SQ': 0, 'WC': -2, 'WK': -3, 'HQ': 0,
'LC': -1, 'WN': -4, 'SA': 1, 'LG': -4,
'WS': -3, 'SE': 0, 'HE': 0, 'SI': -2,
'HA': -2, 'SM': -1, 'YL': -1, 'YH': 2,
'YD': -3, 'ER': 0, 'XP': -2, 'GG': 6,
'GC': -3, 'EN': 0, 'YT': -2, 'YP': -3,
'TK': -1, 'AA': 4, 'PQ': -1, 'TC': -1,
'VH': -3, 'TG': -2, 'IQ': -3, 'ZT': -1,
'CR': -3, 'VP': -2, 'PE': -1, 'MC': -1,
'KN': 0, 'II': 4, 'PA': -1, 'MG': -3,
'TS': 1, 'IE': -3, 'PM': -2, 'MK': -1,
'IA': -1, 'PI': -3, 'RR': 5, 'XM': -1,
'LI': 2, 'XI': -1, 'ZB': 1, 'XE': -1,
'ZN': 0, 'XA': 0, 'BR': -1, 'BN': 3,
'FD': -3, 'XY': -1, 'ZR': 0, 'FH': -1,
'BF': -3, 'FL': 0, 'XQ': -1, 'BB': 4,
}
try:
    deger = blosum62[satir_degeri+sutun_degeri]
except:
    deger = 100

if deger == 100:
    deger = blosum62[(satir_degeri+sutun_degeri)[::-1]]

return deger

```

1.7.3 Needleman_Wunsch Modülü

```

import BLOSUM62
import numpy as np

def needlemanWunsch(covid , sars , protein_turu):
    bosluk_ceza_puani = -4
    covid_protein = "-" + proteinDizisi0ku(covid)
    sars_protein = "-" + proteinDizisi0ku(sars)
    sat = len(covid_protein)
    sut = len(sars_protein)

```

FASTA SEKANS DOSYALARI

```
skor_matrisi = np.zeros([sat , sut])
yol_matrisi = np.zeros([sat,sut])

for i in range(0,sut,1):
    skor_matrisi[0][i] = i * bosluk_ceza_puani
for j in range(0,sat,1):
    skor_matrisi[j][0] = j * bosluk_ceza_puani

# Yol Matrisi : 0 bitis , 1 sol , 2 yukarı , 3 diagonal
yol_matrisi[0][0] = 0

for i in range(1,sut,1):
    yol_matrisi[0][i] = 1

for j in range(1,sat,1):
    yol_matrisi[j][0] = 2

for satir in range(1,sat,1):
    for sutun in range(1,sut,1):
        if covid_protein[satir] == sars_protein[sutun]:
            skor_matrisi[satir][sutun]=skor_matrisi[satir-1][sutun-1]+BLOSUM62.skorBul(covid_protein[satir],sars_protein[sutun])
            yol_matrisi[satir][sutun]= 3
        else:
            skor_diagonal = skor_matrisi[satir - 1][sutun - 1] + BLOSUM62.skorBul(covid_protein[satir], sars_protein[sutun])
            skor_yukari = skor_matrisi[satir-1][sutun] + BLOSUM62.skorBul(covid_protein[satir],sars_protein[sutun])
            skor_sol = skor_matrisi[satir][sutun-1] + BLOSUM62.skorBul(covid_protein[satir],sars_protein[sutun])
            if skor_yukari >= skor_sol and skor_yukari >= skor_diagonal:
                skor_matrisi[satir][sutun] = skor_sol
                yol_matrisi[satir][sutun] = 2
            elif skor_sol >= skor_yukari and skor_sol >= skor_diagonal:
                skor_matrisi[satir][sutun] = skor_yukari
                yol_matrisi[satir][sutun] = 1
            else:
                skor_matrisi[satir][sutun] = skor_diagonal
                yol_matrisi[satir][sutun] = 3
hizalama_sonucu_covid = ""
hizalama_sonucu_sars = ""
yol_sutun = sut-1
yol_satir = sat-1
adim = yol_matrisi[yol_satir][yol_sutun]

while(adim != 0 ):
    if adim ==1:
        hizalama_sonucu_sars += sars_protein[yol_sutun]
        hizalama_sonucu_covid += "-"
        yol_sutun -= 1
    elif adim == 2:
        hizalama_sonucu_sars += "-"
        hizalama_sonucu_covid += covid_protein[yol_satir]
        yol_satir -= 1
    elif adim == 3:
        hizalama_sonucu_covid += covid_protein[yol_satir]
        hizalama_sonucu_sars += sars_protein[yol_sutun]
```

FASTA SEKANS DOSYALARI

```
        yol_satir -= 1
        yol_sutun -= 1
        adim = yol_matrisi[yol_satir][yol_sutun]

        hizalama_dosyasi = open("C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik
Hesaplama\\"+protein_turu+"_NWA.txt" , "w")
        hizalama_dosyasi.write(protein_turu+ " Needleman Wunsch Hizalaması
\\n\\n\\n" + "nCovid :" + hizalama_sonucu_covid[:: -1] + "\\n\\nSARS  :" +
hizalama_sonucu_sars[:: -1] )

def proteinDizisiOku(dosya_yolu):
    dosya = open(dosya_yolu, "r")
    protein = dosya.read()
    protein = protein.replace("\\n", "")
    protein = protein.replace("\\r", "")
    return protein

def farkBul(covid , sars , protein_turu):
    covid_protein = proteinDizisiOku(covid)
    sars_protein = proteinDizisiOku(sars)

    covid_uzunluk = len(covid_protein)-1
    sars_uzunluk = len(sars_protein)-1

    maX_uzunluk = max(covid_uzunluk , sars_uzunluk)

    dosya_metni = ""

    i = 0
    while i<maX_uzunluk:

        fark_metni=""
        fark_sars =""
        fark_nCovid = ""

        if i<covid_uzunluk and i<sars_uzunluk :
            if covid_protein[i] != sars_protein[i]:
                fark_metni += "Konum : " +str(i)

                while covid_protein[i] != sars_protein[i] and i<covid_uzunluk
and i<sars_uzunluk:
                    fark_sars += sars_protein[i]
                    fark_nCovid += covid_protein[i]
                    i += 1
                fark_metni += "\\nnCovid = " + fark_nCovid + "\\nSARS  = " +
fark_sars + "\\n\\n*****\\n\\n"
            else:
                i += 1

        elif i>=covid_uzunluk and i<=sars_uzunluk:
            fark_metni += "Konum : " +str(i)
            while i<sars_uzunluk:
                fark_sars += sars_protein[i]
                fark_nCovid += "-"
                i += 1
            fark_metni += "\\nnCovid = " + fark_nCovid + "\\nSARS  = " +
fark_sars + "\\n\\n*****\\n\\n"
```



```























elif i>=sars_uzunluk and i<=covid_uzunluk :
    fark_metni += "Konum : " +str(i) + "\n"

    while i<covid_uzunluk:
        fark_sars += "-"
        fark_nCovid += covid_protein[i]
        i += 1
    fark_metni += "\nnCovid = " + fark_nCovid + "\nSARS = " +
fark_sars + "\n\n*****\n\n"
    i+=1

if fark_metni != "":
    print(fark_metni)
    dosya_metni += fark_metni
    dosya = open("C:\\Users\\ASUS\\Desktop\\8.Dönem\\Genomik Hesaplama\\Fark_"+
protein_turu + ".txt" , "w")
    dosya.write(dosya_metni)

```

1.8 Girdi ve Çıktı Dosyalarının Toplandığı Klasörün Görünümü

 SARS_Polyprotein.txt	6.03.2020 16:06	Metin Belgesi	8 KB
 SARS_Nucleocapsid.txt	6.03.2020 16:06	Metin Belgesi	1 KB
 SARS_GEN_ORF.txt	6.03.2020 16:06	Metin Belgesi	22 KB
 SARS_AA_ORF.txt	6.03.2020 16:06	Metin Belgesi	8 KB
 SARS_AA.txt	6.03.2020 16:06	Metin Belgesi	29 KB
 SARS.txt	6.03.2020 16:06	Metin Belgesi	30 KB
 Polyprotein_NWA.txt	6.03.2020 13:38	Metin Belgesi	14 KB
 Nucleocapsid_NWA.txt	6.03.2020 13:14	Metin Belgesi	1 KB
 Fark_Polyprotein.txt	6.03.2020 16:06	Metin Belgesi	38 KB
 Fark_Nucleocapsid.txt	6.03.2020 16:06	Metin Belgesi	3 KB
 COVID_Polyprotein.txt	6.03.2020 16:06	Metin Belgesi	8 KB
 COVID_Nucleocapsid.txt	6.03.2020 16:06	Metin Belgesi	1 KB
 COVID_GEN_ORF.txt	6.03.2020 16:06	Metin Belgesi	574 KB
 COVID_AA_ORF.txt	6.03.2020 16:06	Metin Belgesi	192 KB
 COVID_AA.txt	6.03.2020 16:06	Metin Belgesi	30 KB
 COVID.txt	6.03.2020 16:06	Metin Belgesi	31 KB
 SARS_Polyprotein.fasta	5.03.2020 14:10	FASTA Dosyası	8 KB
 SARS_Nucleocapsid.fasta	5.03.2020 14:12	FASTA Dosyası	1 KB
 SARS.fasta	5.03.2020 14:06	FASTA Dosyası	30 KB
 COVID_Polyprotein.fasta	5.03.2020 14:08	FASTA Dosyası	8 KB
 COVID_Nucleocapsid.fasta	5.03.2020 14:09	FASTA Dosyası	1 KB
 COVID.fasta	5.03.2020 14:07	FASTA Dosyası	30 KB

1.9 Sonuç

Verilen ödevde istenilen tüm aşamalar gerçekleştirilmiştir. Ayrıca ek olarak; ORF tespitinde istenen nükleotit bazlı sonucu üretmenin yanı sıra amino asit bazlı ORF tespitini yapan fonksiyonda yazılmıştır.