Bioinformatics Project:

I-Introduction:
This project is about comparing two virals genome, the SARS-COV2 and the SARS-TOR2 by using K-mer Indexing, Jaccard Index and Mash Distance.

1. K-mer Indexing: First of all we need to extract the k-mers of varying lengths to quantify similarities, base on this formula :

$$L - k + 1 \text{ (max)}$$

We had to index all the k-mers of our two genomes in a Python dictionary. For that, we wrote a Python function with three parameters: **the sequence we wanted to index, the k-mer size and the step.**
We obtain the following script:

```python
import math
def index(seq, k, step):
    '''seq=input("Seq?")
    k=int(input("k?"))
    step=int(input("Steps?"))'''
    k_mer_dict={}
    for nuc in range(0,len(seq)-k+1,step):
        k_mer=seq[nuc:nuc+k]
        k_mer_dict[k_mer] = k_mer_dict.get(k_mer, 0) + 1
    return k_mer_dict
```

The function `index(seq, k, step)` generates a dictionary of k-mers from a genome sequence.

2.Jaccard Index: Measuring the similarity between k-mer sets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \approx \frac{|S(A \cup B) \cap S(A) \cap S(B)|}{|S(A \cup B)|}$$

3. Mash Distance: Here we calculate the distance based on the Jaccard index, which is useful for rapid sequence comparison:

$$D = -\frac{1}{k} \ln \frac{2j}{1+j}$$

The function Jaccard_mash(seq1, seq2, k) calculates the Jaccard index and Mash distance:

```python
def Jaccard_mash(seq1,seq2,k):
    seq1=set(seq1)
    seq2=set(seq2)
    intersection = len(seq1.intersection(seq2))
    union = len(seq1.union(seq2))
    Jaccard=(intersection/union)
    mash_distance = - (1 / k) * math.log(Jaccard) if Jaccard > 0 else float('inf')
    return Jaccard, mash_distance
```

The analyses are conducted with different k-mer sizes and step values, and the results are compared with global alignment results obtained using the Needleman-Wunsch algorithm:

```
########################################
# Program: needle
# Rundate: Thu 19 Dec 2024 16:47:26
# Commandline: needle
#    -auto
#    -stdout
#    -asequence emboss_needle-I20241219-164625-0600-14538843-p1m.asequence
#    -bsequence emboss_needle-I20241219-164625-0600-14538843-p1m.bsequence
#    -datafile EDNAFULL
#    -gapopen 10.0
#    -gapextend 0.5
#    -endweight
#    -endopen 10.0
#    -endextend 0.5
#    -aformat3 pair
#    -snucleotide1
#    -snucleotide2
# Align_format: pair
# Report_file: stdout
########################################

#=======================================
#
# Aligned_sequences: 2
# 1: NC_045512.2
# 2: NC_004718.3
# Matrix: EDNAFULL
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 30475
# Identity:   24151/30475 (79.2%)
# Similarity: 24151/30475 (79.2%)
# Gaps:        1296/30475 ( 4.3%)
# Score: 95872.0
#
#
#=======================================
```

Table1: Global Alignment Results (Needleman-Wunsch on JobDispatcher)

4. We then apply our Python functions to compute that mash distance between the two virus genomes with different values of k: [7, 11, 21, 31, 41, 51, 61, 71, 81, 91, 101, 111, 121] and with two step values: [1, 2] and compare the results:

```
from Bio import SeqIO
file = "SARS_COV2.fna"
for record in SeqIO.parse(file, "fasta"):
    SARS_COV2=record.seq


file = "SARS_TOR2.fna"
for record in SeqIO.parse(file, "fasta"):
    SARS_TOR2=record.seq



result=[]
k_values=[7, 11, 21, 31, 41, 51, 61, 71, 81, 91, 101, 111, 121]
STEP_VALUE=[1,2]
for k in k_values:
    for step in STEP_VALUE:
        sars_cov2_kmers = index(SARS_COV2, k, step)
        sars_tor2_kmers = index(SARS_TOR2, k, step)
        Jaccard, mash_distance = Jaccard_mash(sars_cov2_kmers, sars_tor2_kmers, k)


        '''print(f"Jaccard Index: {jaccard_index:.4f}")'''
        '''print(f"Mash Distance: {mash_distance:.4f}")'''
        result.append({
            "k": k,
            "step": step,
            "Jaccard": Jaccard,
            "Mash Distance": mash_distance
        })
print(result)
import pandas as pd
import matplotlib.pyplot as plt
df = pd.DataFrame(result)
print(df)
```

II-Results:

1. After running our Jaccard_mash function on our 2 genome, we obtain those result as shown on this table:

k	Step	Jaccard	Mash Distance
7	1	0.693175	0.052353
7	2	0.515191	0.094745
11	1	0.090263	0.218639
11	2	0.058272	0.258422
21	1	0.025059	0.175548
21	2	0.014088	0.202973
111	2	0.000000	∞

Table2: Jaccard Index and Mash Distance for Different K-mer Sizes and Step Values

2. To better understand them and how our different parameters infer on them, we do a graphe thanks to this script:

```
plt.figure(figsize=(20,6))
# Jaccard Index
plt.plot(df[df["step"]==1]["k"], df[df["step"]==1]["Jaccard"], marker='o', label="Jaccard Index (Step=1)")
plt.plot(df[df["step"]==2]["k"], df[df["step"]==2]["Jaccard"], marker='s', label="Jaccard Index (Step=2)")
# Mash Distance
plt.plot(df[df["step"]==1]["k"], df[df["step"]==1]["Mash Distance"], marker='o', linestyle='--', label="Mash Distance (Step=1)")
plt.plot(df[df["step"]==2]["k"], df[df["step"]==2]["Mash Distance"], marker='s', linestyle='--', label="Mash Distance (Step=2)")
plt.xlabel("k-mer Size (k)")
plt.ylabel("value")
plt.title("Jaccard Index and Mash Distance in relation to k-mer Size")
plt.legend()
plt.grid(True)
plt.show()
```

Her is the result of the script:

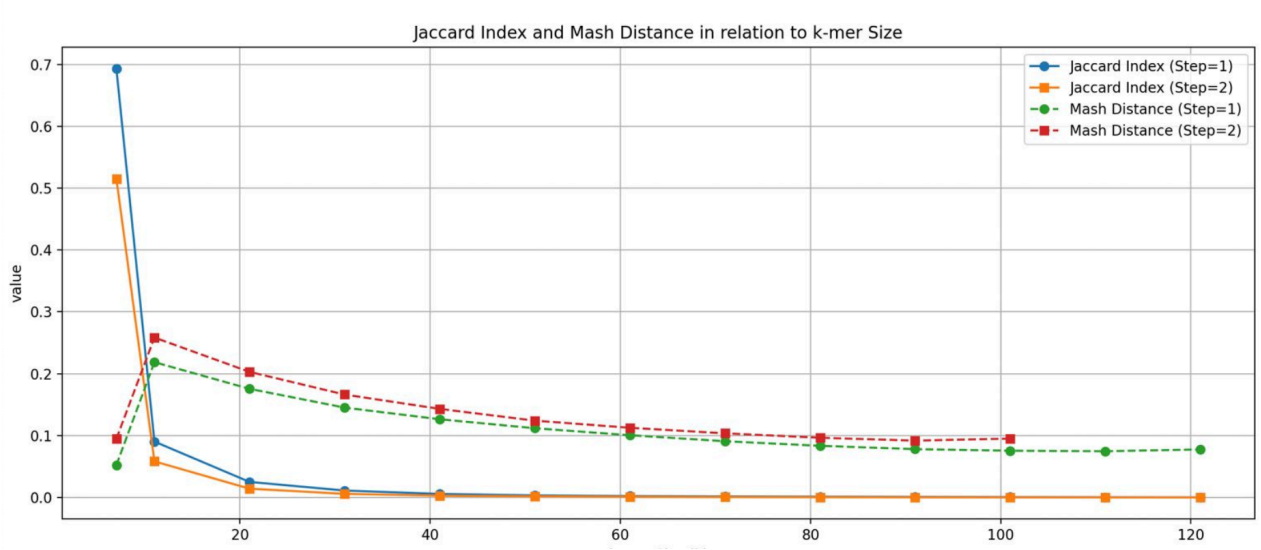


Table1: Jaccard Index and Mash Distance in relation to k-mer Size

III-Observation and interpretation:

1. **Mash Distance and Jaccard Index Results** (from k-mer analysis): As we can see on tables 1 and 2, as the k-mer size increases, the Mash Distance also gradually increases, while the Jaccard Index decreases. The smallest k-mer sizes, specifically **k = 7** and **k = 11**, exhibit the highest Jaccard index values and the lowest Mash distances, indicating a higher degree of similarity between the sequences. In contrast, for larger k-mers, such as k = 111 and k = 121, the Jaccard index drops significantly. In some cases, particularly for k = 111 and k = 121, both with a step size of 2, the Jaccard index reaches 0.0, **which results in an infinite Mash distance (inf)**. This suggests that while the sequences are globally similar, they diverge considerably when comparing longer k-mer subsequences. **Likely due to mutations or structural variations?**

2. **Global Alignment Results** (Needleman-Wunsch on JobDispatcher): Cf. Table 1

    - **Identity**: 79.2% (24,151 out of 30,475 positions).
    - **Similarity**: 79.2% (24,151 out of 30,475 positions).
    - **Gaps**: 4.3% (1,296 gaps).
    - **Score**: 95,872.0.

    High Identity and Similarity (79.2%):

    - The Needleman-Wunsch alignment shows that the genomes are **79.2% identical and similar**.This high percentage aligns well with the **low Mash distances** and **high Jaccard index** for smaller k-mers (k = 7 and k = 11), where the sequences share many short k-mer subsequences, **which align with our precedent interpretation of the k-mers and may suggest that those two viral sequence share a commun origin/ancestor.**

3. **Mash Distance Trends**:

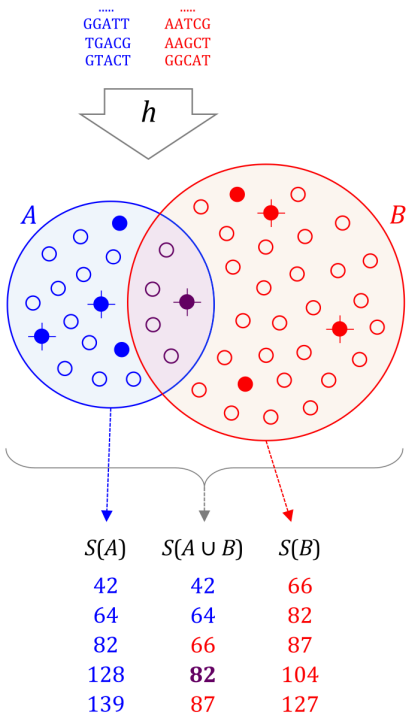
    - As said earlier the **k-mer size increases**, the Mash distance increases because longer k-mers are less likely to match exactly between two sequences, even if they are globally similar.
    - This is consistent with the alignment results because **small sequence variations** (substitutions, gaps) impact longer k-mers more significantly than shorter ones. **This is the kind of thing that can be observe between two sequence of two organism sharing a community origin/ancestor, in this context : SARS-COV2 and SARS-TOR2.**

4. **Infinite Mash Distance (inf)**:

    - For the largest k-mer sizes (k = 111 and k = 121) with step = 2, the Mash distance becomes infinite (inf). This occurs when the Jaccard index is 0.0, meaning there is **no common k-mers** of that length.
    - This suggests that while the sequences are globally similar, they diverge significantly when comparing very long subsequences. This could be due to **mutations** or **rearrangements** that prevent exact matches for longer k-mers.

Bonus: This analogy provides a compelling way to conceptualize the relationship between the Jaccard index and Mash distancewhen comparing two sequences. Imagine two circles, A and B, where circle A is smaller and we want to determine if it fits within circle B. To do this, we align both circles and check if the small internal circles (subcomponents) within circle A align with those in circle B.

In this context, the **Jaccard index** measures how accurately the smaller circles (k-mers) in **circle A** overlap with those in **circle B**. It tells us the proportion of shared coordinates or matching patterns. On the other hand, the **Mash distance**represents a measure of scale or magnitude. As we increase the size of the circles (or the scale of comparison), we also increase the number of smaller circles (k-mers) required for comparison.



When the circles grow larger (i.e., increasing the k-mer size), the number of matching coordinates between **circle A** and **circle B** becomes less significant, reducing the **Jaccard index**. Consequently, the **Mash distance** increases, reflecting the difficulty of maintaining alignment at larger scales. Therefore, the larger the circles (or k-mers) become, the harder it is to determine if **circle A** is truly within **circle B** based solely on matching smaller subcomponents, as the shared coordinates become fewer and less meaningful.

This analogy captures the essence of why the Jaccard index decreases and the Mash distance increases with larger k-mer sizes. It highlights how the resolution of the comparison diminishes as the scale (k-mer size) increases, making it harder to identify precise overlaps between the two structures.

5. **Impact of Step Size**:
    - **Step Size = 1** results in a more detailed k-mer analysis, capturing more overlapping k-mers.
    - **Step Size = 2** skips more positions, which can lead to fewer matching k-mers and higher Mash distances, especially for larger k-mers.

Metric	Needleman-Wunsch Global Alignment	K-mer Analysis (Small k)	K-mer Analysis (Large k)
Identity/Similarity	79.2%	High Jaccard Index	Low Jaccard Index
Gaps	4.3%	Low Mash Distance	High Mash Distance
Mash Distance	N/A	Low (e.g., 0.05 - 0.25)	High (e.g., 0.1 - inf)

Table4. Summary of the results

Conclusion:

The results from our k-mer analysis and Mash distance computations are consistent with the global alignment results obtained using the Needleman-Wunsch algorithm. The Jaccard index and Mash distance illustrate expected behavior when comparing genomic sequences. Specifically, **smaller k-mers** (e.g., **k = 7** and **k = 11**) capture more sequence similarity due to shorter subsequence matches, resulting in a **higher Jaccard index** and a **lower Mash distance**. These findings align with the Needleman-Wunsch alignment, which shows a **high identity and similarity of 79.2%** between the SARS-COV2 and SARS-TOR2 genomes. This high degree of similarity supports the idea that these two viruses may share a **common origin or ancestor**, a phenomenon commonly observed in viral genomes, such as during the COVID-19 pandemic.

As the **k-mer size increases** (e.g., **k = 111** and **k = 121**), the Jaccard index decreases significantly, and in some cases, it reaches **0.0** (particularly with a step size of 2). This results in an **infinite Mash distance** (`inf`). The inability to find matching k-mers at these lengths suggests that while the genomes are globally similar, they diverge at the level of longer subsequences due to **mutations** or **structural rearrangements**. This observation reflects the fact that larger k-mers are more sensitive to exact matches, and small sequence variations such as **substitutions** or **gaps** can prevent these matches.

The impact of **step size** on the analysis is also noteworthy. A **step size of 1** captures a more detailed view by including more overlapping k-mers, leading to higher Jaccard index values and lower Mash distances. In contrast, a **step size of 2**skips more positions, reducing the number of overlapping k-mers and increasing the Mash distance, particularly for larger k-mers.

This behavior can be conceptualized through the analogy of **two circles, A and B**, where circle A is smaller and we want to determine if it fits within circle B. The **Jaccard index** measures how accurately the smaller internal circles (k-mers) in circle A align with those in circle B. The **Mash distance** reflects the scale of comparison; as the circles increase in size (larger k-mers), the number of matching internal circles decreases, making it harder to determine alignment, thus reducing the Jaccard index and increasing the Mash distance.

In summary, while **k-mer-based methods** provide a rapid estimation of sequence similarity, they are sensitive to the choice of k-mer size and step size. **Global alignment methods** like Needleman-Wunsch offer a more detailed and accurate comparison, especially for closely related sequences. The consistency of these results supports the hypothesis that **SARS-COV2** and **SARS-TOR2** are closely related, potentially as **variants of one another**. This highlights the importance of combining multiple methods for comprehensive genomic analysis and understanding viral evolution.

Perspective:

Apply the analysis to other viral genomes to study evolutionary relationships.
To push the analyse of our two genome even further we could use different distance metrics for additional insights.