Team Members:

Ashwin Agesh Somanathan	A59025915
Kumar Divij	A69028007
Suraj Sathya Prakash	A59026390

Proposal: Phylogenetic Tree Generation Using UPGMA

OVERVIEW

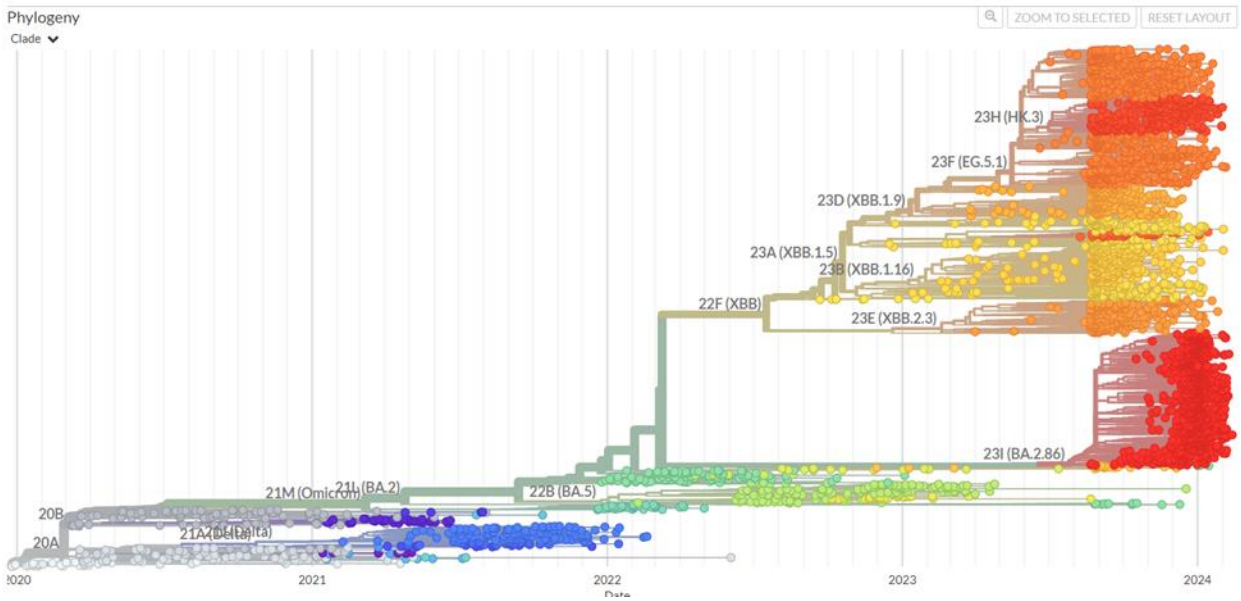
Phylogenetic trees are graphical representations of evolutionary relationships among different species, organisms, or genes. They are often depicted as a branching pattern from a common ancestor. Nodes within a tree are called taxonomic units.

Phylogenetic trees are a powerful tool in the field of bioinformatics for deciphering the complexity of biological systems and addressing fundamental questions in biology and medicine. They provide insights into the spread of diseases, the evolution of drug resistance in pathogens, and the design of effective treatments and vaccines. They also aid in taxonomic classification, biodiversity assessment, and conservation efforts.

A recent example is that of the Covid-19 virus. Phylogeny of the SARS-CoV-2 virus has been used to estimate the rates of spread across regions.

Source: Nature - Phylogenetic and phylodynamic approaches to understanding and combating the early SARS-CoV-2 pandemic

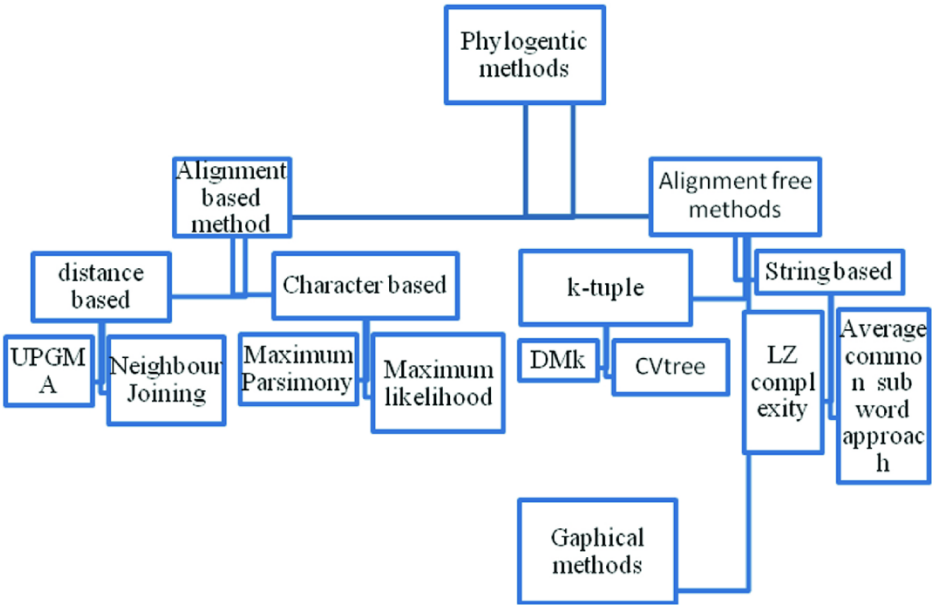
Genomic epidemiology of SARS-CoV-2



Source: https://nextstrain.org/ncov/gisaid/global/6m

LITERATURE REVIEW

Of these many solutions, we have opted to study UPGMA, which stands for unweighted pair group method with arithmetic mean for reasons that will be discussed in the next section. In their work, (Chen, et al., 2012) presented GPU-UPGMA, an implementation of the algorithm on an NVIDIA Tesla C2050 GPU using CUDA for software development, achieving a 95x improvement in speed compared to a 2.13 GHz CPU.



Source: Phylogenetics Algorithms and Applications

(Hung, Lin, Wu, & Chan) extended this to multiple GPUs, using 4 NVIDIA GTX 980s. (Hua, et al., 2017) took this even further, using a DGX-1 server with 8 NVIDIA P100 graphic cards in tandem through the use of NVIDIA Collective Communications Library. They achieved a linear increase in speed compared to a single GPU and CPU.

While matching and surpassing the results mentioned above may be beyond the scope of this project given the time and resource constraints, we plan on deriving inspiration from these works to implement and optimize the UPGMA algorithm on a single NVIDIA 2080-Ti GPU, taking the work done by (Chen, et al., 2012) as reference.

METHODOLOGY

As mentioned in the previous section, we will be attempting to implement the UPGMA algorithm on a GPU and efficiently distribute the workload on the CPU. We chose to study this algorithm because it is an alignment-based method that uses distance matrices. Matrix based algorithms are highly efficient on GPUs due to their ability to handle large number of computations in parallel, specialized hardware blocks like Streaming Processors and a large memory bandwidth.

The UPGMA algorithm broadly involves the following steps:

1. Compute the distance matrix between all the objects (species, genes, etc). The options for this include:

a) Similarity (Levenshtein)
b) Dissimilarity (Hamming)
c) Jukes Cantor Distance

2. Find and merge the pair with the shortest separation

3. Update the distance matrix, considering the merged objects as a single entity

a) This can be done through arithmetic means, and does not need the whole matrix to be recalculated

4. Repeat from step 2 until all objects have been grouped (or has been reduced to a predefined threshold)

We will attempt to build CUDA kernels to parallelize each step on a GPU. Furthermore, weighing the relative gain, we will pipeline the wrapper code that will manage files, memory transfers and cleanup on the CPU using TBB.

(Wu, 2021) discusses various matrix operations that can be performed efficiently on GPUs. These should prove useful as we progress through the implementation of this project.

VERIFICATION AND PROFILING

Dataset

- SARS-CoV-2 genomes
- Raw genetic data: https://greengenes2.ucsd.edu/

Base line: We will develop a serial implementation of UPGMA in C++, as a baseline, despite the presence of Decenttree's CPU-parallelized OpenMP version

Algorithm: UPGMA for Phylogenetic Tree generation, Distance Matrix Computation - (Jukes Cantor genetic distance/Hamming)

REFERENCES

[1] "Parallel UPGMA Algorithm on Graphics Processing Units Using CUDA | IEEE Conference Publication | IEEE Xplore," *ieeexplore.ieee.org*. https://ieeexplore.ieee.org/document/6332258 (accessed Feb. 24, 2024).

[2] G.-J. Hua, C.-L. Hung, C.-Y. Lin, F.-C. Wu, Y.-W. Chan, and C. Y. Tang, "MGUPGMA: A Fast UPGMA Algorithm With Multiple Graphics Processing Units Using NCCL," *Evolutionary Bioinformatics*, vol. 13, p. 117693431773422, Jan. 2017, doi: https://doi.org/10.1177/1176934317734220.

[3] "Efficient parallel UPGMA algorithm based on multiple GPUs | IEEE Conference Publication | IEEE Xplore," *ieeexplore.ieee.org*. https://ieeexplore.ieee.org/abstract/document/7822640 (accessed Feb. 24, 2024).

[4] L. Wu, "Accelerating Matrix Power Operations with GPUs," Apr. 2021, doi: https://doi.org/10.1145/3462676.3462680.