Gorilla Report

Alexander Berg, Thomas Hoffmann Kilbak, Kristoffer Bruun Højelse, Emil Jäpelt, Adam Negaard

October 3, 2022

Results

Our implementation produces the expected results on all test cases.

Implementation details

We chose an iterative implementation.
The implementation uses a two-dimensional array, M, of size $(n+1)(m+1)$, where n and m are the lengths of the strings s_0 and s_1, to store the solutions to subproblems. When constructing the alignments of a solution, the algorithm navigates through M starting at index (n, m) (indices starting at 0). For each iteration it looks at costs $\delta + M[x-1, y]$, $\delta + M[x, y-1]$, and $c_x y + M[x-1, y-1]$, where $c_x y$ is the cost of elements at indices x and y in s_0 and s_1. Depending on which of these three costs is equal to $c_x y$ it decrements one or both of x and y and appends one of $s_0[x-1]$, $s_1[y-1]$, or $-$ to each alignment.
For two strings of length n and m, respectively, our implementation uses $O(nm)$ time and $O(mn)$ space for two strings of length m and n.
The program can be run using .NET 6.0 by navigating to the folder containing "Program.cs" and executing the command:

```
dotnet run
```

When the program starts it will read the file BLOSUM62.txt and every input file in the data folder. The program writes the solutions to standard output.