These results show that the model *CBI-RWS-OPC-SBM-FSR* outperforms the other models. This result is unsurprising, as the baselines do not actively search for an optimal solution. Furthermore, we see that most evolutionary models surpass their baselines by a wide margin.

The difference in computation time is most likely due to the similarity and sparsity measures. The computation of the Damerau-Levenshstein distance is quadratic. As we also apply an additional custom cost function, the longer the sequence, these computation times increase. The evolutionary algorithm, as described in **??** is a sequential operation that also increases with the sequence length. However, we can deduce that the time difference between the *CBI-ES-UC3-SBM-RR* stems from either the *Ranking-Recombination* or the *Uniform-Crossing* operation. As those two are the only discernible operators.

In contrast, the baselines have been implemented in ways that vectorise most operations using NumPy. Meaning they can vastly decrease their computation time. On the other hand, the evolutionary algorithms are subject to python's notorious slow-looping procedures. However, this is not a vital issue for two reasons. First, it is possible to run evolutionary algorithms in a parallel manner. Second, we have not explored more optimised implementations of either the Semi-strucured Damerau-Levenshtein distance (SSDLD) or the evolutionary algorithm. However, we are confident that there are better and fast implementations available.

Knowing these results, a couple of questions remain, namely, whether the results remain consistent for longer sequences and other datasets. Furthermore, how does this procedure compare to other methods in the literature? The remaining experiments will address these questions.