We introduced most of the operators in **??**. In this section, describe the operators in detail and select a subset that we want to explore further. **[FOR XIXI: Are implementation details important?]**

Operators

We implemented a number of different evolutionary operators. Each one belongs to one of five categories. The categories are inititiation, selection, crossing, mutation and recombination.

Inititation

DI: The Default-Initiator generates an initial population entirely randomly.

SBI: The *Sampling-Based-Initiation* generates an initial population using a distribution estimated from the data.

CBI: *Case-Based-Initiation* uses examples of the data as initial population.

Selection

RWI: *Roulette-Wheel-Selection* Selects individuals randomly, but proportionate to their fitness score.

TS: *Tournament-Selection* Compares two or more individuals and selects a winner among them.

ES: *Elitism-Selection* selects each individual solely on their fitness

Crossing

OPC: *One-Point-Crossing* Chooses on point in the sequence and creates offspring by taking everything from or after that point from another individual.

TPC: *Two-Point-Crossing* Chooses two points in the sequence and creates offspring by taking everything between or outside these points from another individual.

UCx: *Uniform-Crossing* Uniformly selects positions in the sequence to take from another individual. The amount of selected positions is determined by a crossing-rate between 0 and 1.

Mutation

RM: *Random-Mutation* creates entirely random features for inserts and substitution.

SBM: *Sampling-Based-Mutation* creates sampled features based on data distribution for inserts and substitution.

Recombination

FSR: *Fittest-Survivor-Recombination* Determines the survivor among the mutated offsprings and the population.

BBR: *Best-of-Breed-Recombination* Determines better than average survivors among the mutated offsprings and adds them to the population.

RR: *Ranked-Recombination* Determines survivors based on ranking

We use abbreviations to refer to them in figure, tables and so one. For instance, *CBI-RWI-OPC-RM-PR* refers to an evolutionary operator configuration, that samples its initial population from the data, probablistically samples parents based on their fitness, crosses them on one point and so on. For the *Uniform-Crossing* operater we additionally indicate its crossing rate using a number. For instance, *CBI-RWI-UC3-RM-PR* is a model using the *Uniform-Crossing* with a child receiving roughly 30% of the genom of one parent and 70% of another parent.

Hyper Parameters

As with all models, the evoulutionary approach comes with a number of hyper parameters. Generally, it is hard to determine athe best set of hyperparameters as they interact and depend on the task setting. Nevertheless, it is important to discuss them.

We first discuss the model configuration. As show in this section, there are a number of ways to combine different operators. Depending on each individual operator, we might see very specific behaviours. For instance, it is obvious, that initiating the population with a random set of values can hardly converge at the same speed as a model which leverages case examples. Similarly, the selection of only the fittest individuals is heavily prone to local optima issues. The decision of the appropriate set of operators is by far the most important in terms of convergence speed and result quality.

The next hyperparamter is the *termination point*. Eventually, most correctly implemented evolutionary algorithms will converge to a local optimum.

Especially if only the best individuals are allowed to cross over. If the termination point was chosen to early, then the generated individual will most likely underperform. In constrast, choosing a termination point too far in the furture might yield optimal result at the cost of time performance. Furthermore, the existence of local optima may result in very similar solutions in the end. Optimally, we find a termination point, which finds a reasonable middle point.

The *mutation rate* is another important hyperparamter. It signifies how much a child can differ from its parent. Again, choosing a rate that is too low does not explore the space as much as it could. In turn, a mutation rate that is too high significantly reduces the chance to congverge. The optimal muitation rate allows for exploring novel solutions without immediately pursuing suboptimal solution spaces. Our case is special, as we have four different mutation rates to consider. The change rate, the insertion rate, the deletion rate and the transposition rate. Naturally, these strongly interact. For instance, if the deletion rate is higher than the insertion rate there's a high chance that the sequence will be shorter, if not 0, at the end of its iterative cycles. Mainly, because we remove more events, than we introduce. However, we cannot assume this behaviour across the board as other hyperparamters interplay. Most prominently, the fitness function. Say, we have a high insertion rate but the fitness function rewards shorter sequences. Subsequently, both factors cancel each other out. Hence, the only way to determine the best set of mutation-rates requires an extensive search.