Protein optimization is currently dominated by two opposing paradigms. One relies on a mechanistic understanding of a protein's function based on structure. Using this knowledge, changes to the protein are made that are expected to modify a specific function. This approach is often too hard because structure prediction and function prediction from structure are inherently error prone and incomplete. The second method is based on the creation of large libraries of variants, the empirical testing of these variants for the desired properties and the selection of the best members of the library during repeated cycles of variation and selection. The variants are produced with DNA and family shuffling (rather "blind" search methods that require thousands of variants to be tested via inaccurate high throughput screening).

We use a new third method which blends the previous two. We start with a protein that partially achieves the desired function. We then determine all substitutions that occur in related proteins and restrict our search space to proteins reachable from the original via substitutions. We test the desired function on an initial set of proteins and build a model of the sequence function relationship using machine learning algorithms. The algorithms score all proteins in the search space. In each iteration we select a batch of high scoring and diverse proteins to be tested and refine our model based on the new measurements. Our method has the advantage of requiring a much smaller number of proteins to be tested (< 100). More accurate testing becomes feasible and the whole optimization can be done with minimal equipment.

In our prototype experiment, we optimized proteinase K as our target protein because we are generally interested in the ability of this enzyme to modify a variety of polymers. A set of 19 amino acid substitutions was selected for testing. These 19 substitutions are found in close homologues of proteinase K, as well as in more distantly related serine proteases. We designed three batches of variants. The initial batch of 34 synthetic proteinase K variants was designed by making random combinations of substitutions. They contain between 1 and 6 substitutions. We then synthesized these genes individually, expressed and purified protein from each of them and tested for the activity. The test values became the initial training data for the machine learning algorithms. The algorithms proposed the second batch of 24 variants of proteinase K. These variants were tested and then the final third set of 36 variants was chosen.

Overall, we made significantly improvement in optimizing the activity of proteinase K. In 3 batches and with less than 100 variants tested in total, we were able design a variant of proteinase K with a 20- fold higher activity than the protein we started with.