Re-engineering Enzyme Catalysis Using Computer Modeling and Combinatorial Libraries

Kathryn A. Armstrong
Bruce Tidor

MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge MA, 02139 USA

1. Introduction

Enzymes are proteins that act as molecular machines to efficiently catalyze chemical reactions. An enzyme is made of a chain of amino acids whose sequence is determined by the information in DNA/RNA. The enzymes existing in nature catalyze only a subset of the possible chemical reactions and so the development of new enzymes is of interest to those doing chemical synthesis. New enzymes have been created in the past by making random mutations to the protein sequence of a known enzyme and then screening for new function (Farinas et al., 2001; Glieder et al., 2002). However, the number of sequences available to an enzyme is phenomenally large, and such searches can only examine a tiny fraction of them. Efforts to computationally model and predict new catalytic sequences directly are underway but have yet to create new catalytic activity. Our strategy is more broad; we computationally analyze the sequences available to an enzyme’s structure and eliminate those that are incompatible. A catalytically active enzyme must first fold correctly, and so screening this reduced list of sequences for function should be more effective than screening random mutants.

2. Reducing the Search Space

It is thought that mutations in an enzyme active site (the enzyme section most directly involved with catalysis) are most often detrimental but have a large impact on catalysis (Heering et al., 2002). Conversely, mutations far from the active site have subtle effects but are more easily tolerated (Morawski & Arnold, 2001). We would like to substantially modify catalytic activity, so we have decided to analyze the active site of the enzyme Horseradish peroxidase. The three-dimensional structure of this enzyme (produced by X-ray crystallography, shown in Figure 1) was used to start our search of the allowed active site amino acid sequences. The dead-end elimination and A* algorithms allow us to search all possible sequences of amino acids and their structures in the enzyme active site. By fixing the enzyme’s backbone conformation and allowing only discrete conformations of each amino acid, dead-end elimination guarantees that we will find the global minimum energy sequence and structure in this discrete space (Desmet et al., 1992). Then A*, a branch-and-bound search algorithm, is used to create a list of sequences and structures ranked by energy (Leach & Lemon, 1998). We assume that any sequence more than 15 kcal/mol worse in energy than the native sequence will not fold into the correct enzyme structure, and so we eliminate all sequences with energies above this cutoff. The A* algorithm efficiently generates these feasible sequences by pruning the branch-and-bound tree of sequences using our energy cutoff along with estimation of the lowest possible energy required to make each sequence and structure. For the five positions we chose in the enzyme active site, this restricts the feasible list of sequences from $20^5$ (3,200) to about 50000. This substantially reduced list of sequences can now be searched exhaustively using high-throughput experimental techniques.
3. Creating the Feasible Sequences

While 50000 sequences is a reasonable number of new enzymes to screen once they exist, it is beyond the limit of the number of enzymes a person would be willing to experimentally build one by one. Therefore, we have developed two techniques to create all of the feasible enzymes at once. The first method uses the list of 50000 active site sequences to create a larger combinatorial library of sequences that includes most of the sequences we desire. The most popular amino acids at each of the five active site positions in our original list are made experimentally in all possible combinations. While this technique creates some unwanted sequences, the resulting combinatorial library is still much smaller than the original search space and about 70% of the library sequences are from the original list.

The second method for experimentally creating the feasible sequences uses an evolutionary map of the feasible sequences to suggest better starting points for the random mutation experiments mentioned in the Introduction. The evolutionary map of all 50000 sequences is too large to print, but to give an idea of the graph structure we show an evolutionary map in Figure 2 of the 49 compatible sequences for only 2 positions in the active site. In this case, we can see that some sequences have many neighbors, and therefore could easily mutate to other feasible sequences and might be good starting points for mutation experiments. The natural sequence, on the other hand, has no neighbors, indicating that a random mutation experiment starting from this sequence might be less successful.

4. Results and Future Work

Our computational analysis merges with experimental efforts underway in the Wittrup and Klibanov laboratories at MIT. The Wittrup lab uses yeast-display and high-throughput screening to completely search our combinatorial libraries, and the Klibanov laboratory uses advanced enzymology to thoroughly study the properties of variant enzymes. The first experimental screen of our Horseradish peroxidase sequence library was for specific catalysis of the L over the D conformation of tyrosine, both natural substrates of this enzyme. One mutant enzyme with increased specificity was found in our designed library, though the full enzymological characterization of the new enzyme has not been completed. That only one new sequence showed this modified catalytic activity indicates the rarity of catalytic function in sequence space. Further computational work must also be done to better characterize our evolutionary maps. For example, clusters in these graphs could yield other suggestions for mutation experiments. We hope that this joint theoretical and experimental approach will produce understanding of the inner workings of this enzyme through iterative analysis, design, and experimental testing. The techniques developed for this problem may also be generally applicable to other hard protein design problems, involving other functions, in the future.

References


