

1 Optimal antibiotic cyclic

We are considering 14 antibiotics, labeled AM, AMC, AMP, CAZ, CEC, CPD, CPR, CRO, CTT, CTX, CXM, FEP, SAM, TZP, and ZOX. For each of these 14 antibiotics, we select exactly one TEM *fitness landscape*. Such a landscape is a real $2 \times 2 \times 2 \times 2$ tensor $\mathbf{f} = (f_{ijkl})$. The indices i, j, k, l are 0 or 1. We can identify f with a vector whose coordinates are indexed by $\{0, 1\}^4$.

A *mutation model* is a function $M : \mathbb{R}^{16} \rightarrow \mathbb{R}^{16 \times 16}$ that assigns a transition matrix to each fitness landscape. Recall that a *transition matrix* has nonnegative entries and its rows sum to 1. The rows and columns of $M(f)$ are labeled by $\{0, 1\}^4$, in some order that is fixed throughout. We require that our transition matrices respect the adjacency structure of the 4-cube, that is, $M(f)_{a,b} = 0$ unless a and b are vectors in $\{0, 1\}^4$ that differ in at most one coordinate. Thus each row of $M(f)$ has at most 5 non-zero entries.

Two mutation models are described in Section 3.2 of `projectdraft0204.pdf`. The second model is obtained by simply considering the directed graph on $\{0, 1\}^4$ where $a \rightarrow b$ means that $f_a < f_b$. The non-zero diagonal entries of $M(f)$ are $M(f)_{a,a} = 1$ if a has no outgoing edges, and the non-zero off-diagonal entries are $M_{a,b} = 1/\text{outdegree}(a)$ for every directed edge $a \rightarrow b$.

Let f_1, f_2, \dots, f_{14} denote our 14 given fitness landscape, with derived transition matrices $M(f_1), M(f_2), \dots, M(f_{14})$. Let \mathcal{W} denote a finite set of words on the alphabet $\{0, 1, \dots, 14\}$. These words represent the feasible *treatment plans* we are considering. Every word $w = w_1 w_2 \dots w_k$ represents a new transition matrix, namely the corresponding product of 16×16 -matrices

$$M[w] = M(f_{w_1}) \cdot M(f_{w_2}) \cdot \dots \cdot M(f_{w_k}).$$

Our task is to solve the following discrete optimization problem:

Maximize the entry upper left entry $M[w]_{0000,0000}$ over all words w in \mathcal{W} .

The methodology used to solve this problem will depend on the choice of \mathcal{W} . For instance, it would be natural to take \mathcal{W} as the set of all words of length exactly k , for some small positive integer k . Then \mathcal{W} has 14^k elements. For $k \leq 5$ we can solve our problem by brute-force enumeration, but for $k \leq 6$ something more clever will be needed. At this point, I do not know whether a polynomial-time algorithm exists. The problem is reminiscent of the MAP inference problem for Hidden Markov Models, which can be solved efficiently by a dynamic programming approach known as the *Baum-Welch algorithm*. Our problem seems to be more difficult. We will need to do some literature search and talk to some experts to find more efficient algorithms.

But, for starters, let's run the bruce-force computation for some small sets \mathcal{W} of treatment plans that make sense from a biomedical perspective.