

Biological Computation – Final Project

why there are only 18 monotonic functions despite there are 2^9 possible functions?

The reduction from 512 to 18 monotonic functions is due to two key constraints:

1. More activators can't turn the gene OFF if it was ON.
2. Fewer repressors can't turn the gene ON if it was OFF.

These rules significantly limit valid state transitions, ensuring they occur only in one direction (OFF to ON) as activators increase or repressors decrease. This directional constraint eliminates most of the 512 possible functions, leaving only 18 that adhere to the monotonic rules while still representing different gene activation patterns. Each of these 18 functions specifies when the gene turns ON, consistently following the rule that more activators or fewer repressors can't cause the gene to turn OFF.

Explanation on our code:

The program operates through a series of steps to identify and verify monotonic Boolean functions.

It begins by generating all possible combinations of activator and repressor states, considering three states for each: none, some, or all. This creates a total of 9 distinct states.

Next, the program generates all 512 possible Boolean functions for these states, representing each function as a sequence of 9 binary digits (0 or 1), where 0 represents the gene being OFF and 1 represents it being ON for each corresponding state.

The core of the program lies in its monotonicity check. This is implemented in the ``is_monotonic`` function, which examines each function by comparing adjacent states. It ensures that as the number of activators increases (with repressor states unchanged), the gene doesn't switch from ON to OFF (1 to 0), and as the number of repressors decreases (with activator states unchanged), the gene doesn't switch from OFF to ON (0 to 1). This check is performed by iterating through the binary representation and comparing values at specific indices that correspond to these adjacent states. Functions that pass this check are retained as monotonic.

The program then verifies its results by comparing the generated monotonic functions with a predefined list of expected functions from Table D. Finally, it outputs the identified monotonic functions, describes the conditions under which the gene is ON or OFF for each function, provides a count of the total

