**Complex Generative Model:**

The Complex Generative Model (CGM) is based on the paper “A characterization of the DNA data storage channel”. In the paper, the process of synthesizing and sequencing (reading) DNA strands in current technologies is described, and the common errors that occur in each part of the process. the rates of errors is estimated using datasets collected by 3 different research groups. We built our CGM based on the errors described in the paper. The stages in the common errors in each one of them is:

1. Synthesis – this is the process of converting the digital sequence of symbols to physical DNA strands. The current technology does that by positioning nucleotides on a chip, one by one. Here error like deletion (when a nucleotide is missed) insertion (when a nucleotide reaches a location he does not belong to) and substitution can occur. It is noted in the paper that also entire strands can be terminated during synthesis, i.e. it is only partially synthesized. It is also noted that the error rates are not statistically independent, but are strand dependent and are much more likely on strands containing specific patterns, and on location on the board.
2. Amplification – after synthesizing the DNA pool, the strands are amplified with several polymerase chain reaction (PCR) rounds are performed. In each cycle each strand that has the correct primers on both ends is amplified at a rate close to 2 (almost duplicated) using engineered enzymes. So after ~15 rounds each strand can be duplicated times. But the amplification rate at each step is a little less then 2 and it depends on properties of the strand, like high CG contents (large proportions of C, G nucleotides) has a negative effect on that value. Because of the exponential nature of the process, small difference in the rate can cause significant difference in the number of representatives each strand has in the DNA pool.
3. Storage – during storage, a natural decay or the strands occur, with some nucleotides more likely to decay then others. This effects the Amplification and Sequencing stages, where damaged strands might not be duplicated in the Amplification stage, and in the sequencing, stage might be read incorrectly resulting in substitution, or the entire strand might be ignored, depending on the exact technology used for sequencing.
4. Sequencing – sequencing is the process of reading the DNA strands. The most likely errors here are substitution errors, and deletions and insertions are much less common, in the order of . Here also the error rates are not uniform, but are more frequent at the ends of the strand and strand properties like high CG-contents and long homopolymer stretches (e.g., GGGGGG) increase the error rates.

We modeled some of the aspects described in the paper into our CGM, to try to evaluate how the performance of the different methods changes under a different and more realistic Generative model. A short description of the generative model:

1. Sample random sequences uniformly
2. Simulate the Synthesis process, by introducing insertions, deletions, substitutions, and early terminations. The probability of each error is a parameter, and the probability for early termination is:

This selection is to get a probability in , and the probability that will be constant, because it is approximately (because we need identical symbols, and we can start on positions)

1. Simulate the PCR process, where the probability for duplication in each cycle is:

Where is the frequency of CG in the strand.

1. Simulate the storage decay process, where each nucleotide has some (a parameter) probability to decay.
2. Then a sample from the pool is taken – some of the original strands might not have any representative at this point.
3. Simulate the sequencing, where there is some probability for substitution errors, that are higher on the first 10 and last 10 symbols of the strand, by a constant factor.

**Bibliography:**

Heckel, Reinhard, Gediminas Mikutis, and Robert N. Grass. "A characterization of the DNA data storage channel." *Scientific reports* 9.1 (2019): 1-12.