When Second Best Is Good Enough: Another Probabilistic Look at Saturation Mutagenesis

Yuval Nov
Department of Statistics, University of Haifa, Haifa, Israel

We developed new criteria for determining the library size in a saturation mutagenesis experiment. When the number of all possible distinct variants is large, any of the top-performing variants (e.g., any of the top five) is likely to meet the design requirements, so the probability that the library contains at least one of them is a sensible criterion for determining the library size. By using a criterion of this type, one may significantly reduce the library size and thus save costs and labor while minimally compromising the quality of the best variant discovered. We present the probabilistic tools underlying these criteria and use them to compare the efficiencies of four randomization schemes: NNN, which uses all 64 codons; NNB, which uses 48 codons; NNK, which uses 32 codons; and MAX, which assigns equal probabilities to each of the 20 amino acids. MAX was found to be the most efficient randomization scheme and NNN the least efficient. TopLib, a computer program for carrying out the related calculations, is available through a user-friendly Web server.

Saturation mutagenesis (also called oligonucleotide-directed randomization) is a protein-engineering technique that has been used widely and successfully to improve protein properties such as catalytic activity, enantioselectivity, thermostability, and binding affinity (3, 12, 14, 16). We use the term “activity” for the protein’s property under optimization, but the methodology developed below is aimed at any desirable protein feature that may be influenced by mutation.

In saturation mutagenesis, one or more positions along the protein sequence are identified as likely to accommodate beneficial mutations and are then randomized, i.e., the amino acids at these positions are replaced by random ones. The randomization originates at the DNA level, typically via degenerate primers containing a mixture of sequences at the chosen codons. To decrease the chances of introducing a premature stop codon, reduced codon sets are often used: NNB, NNS, and NNK codons (where N = A/C/G/T, B = C/G/T, S = C/G, and K = G/T) are popular choices that still encode all 20 amino acids, but the use of codon sets encoding fewer amino acids has been also advocated (9, 15). More sophisticated randomization schemes, such as MAX (6, 7), result in equal probabilities for all 20 amino acids (or for some predetermined subset thereof) without encoding stop codons. Either way, a large number of random variants, which together constitute a library, are produced and then screened in an attempt to discover a highly active variant among them. Clearly, the larger the library, the higher the probability of exploring more distinct variants. We shall use the term “variant space” to denote the set of all possible distinct variants in a given experiment; this space is determined by the number of positions randomized and the randomization scheme.

The probabilistic literature on saturation mutagenesis (1, 5, 8, 11) focuses mainly on two mathematical quantities: the first is the expected percentage of variant space that is represented in the library, and the second is the probability of “full coverage,” i.e., the probability that the library contains all of variant space. Which quantity is more meaningful to the practitioner? The ideal outcome of the experiment, from the practitioner’s point of view, is to discover the best variant in variant space; if a hypothetical oracle could foretell in advance which variant that is, the practitioner would gladly produce this variant alone, as there is no inherent benefit in screening a large number of variants. Thus, the event of interest is “the library contains the best variant in variant space.” Under very mild assumptions, the probability of this event coincides with the first quantity mentioned above, namely, the expected percentage of variant space represented in the library. We shall see that trying to achieve full coverage is an overly conservative engineering strategy.

However, is discovering the single top-performing variant the only desirable outcome? It may well happen, especially when variant space is large, that the second-best variant, or even the third best or fourth best, is still perfectly adequate. As an analogy, consider a large-scale athletic competition, such as the New York City Marathon with its tens of thousands of participants: the times of (say) all top five participants are usually very good on any scale, and the differences among them are small relative to the difference between the time of the winner and that of a typical participant. Returning to the protein world, if one could save considerable laboratory resources while sacrificing relatively little in the activity of the best variant discovered, by agreeing to also consider, e.g., the second- and third-best variants, it might be an attractive tradeoff.

In this work, we developed a methodology for determining the library size required to ensure a high probability of discovering a top-performing variant that is not necessarily the best one. We then used this methodology to compare the efficiencies of four randomization schemes: NNN, NNB, NNK (which is equivalent to NNS), and MAX.
MATERIALS AND METHODS

Probability of discovering top variants. In saturation mutagenesis, the randomization at the DNA level at each position is typically uniform across all codons compatible with the randomization scheme: under NNN randomization, each of the 64 possible codons has a 1/64 probability; under NNB randomization, each of the 48 possible codons has a 1/48 probability; etc. Since the events of interest are more directly related to the resulting amino acid sequences, one needs to convert probabilities at the DNA level to probabilities at the amino acid level; given the randomization scheme, the amino acid probabilities are induced by the genetic code, or are determined directly when using a protocol such as MAX, as follows.

Let $M$ be the number of (protein) randomized positions and $Q_i$, the probability that a random variant will have amino acid $i$ at the $i$th randomized position. For example, when the codon corresponding to position $i$ is subject to NNN randomization, we have $Q_{A_{i-1}A_iA_{i+1}} = 4/64$ (as 4 out of the 64 codons in the genetic code encode alanine), $Q_{A_{i-1}A_iA_{i+1}} = 6/64$, etc. Under MAX randomization, with equal probabilities for each of the 20 amino acids, one would have $Q_{A_i} = Q_{B_i} = \cdots = Q_{V_i} = 1/20$. Note that if the randomization at position $i$ may result in a stop codon, then $\sum Q_i < 1$. Let $A_i$ be the number of amino acids having a nonzero probability to appear at position $i$, so that the size of variant space is $n = A_1 \cdot A_2 \cdots A_M$. For example, when randomizing two NNN positions (so that $M = 2$), the size of variant space is $n = A_1 \cdot A_2 = 20 \cdot 20 = 400$. Assuming independent randomization across the $M$ positions, the probability that a random variant will have amino acid $a_1$ at position 1, amino acid $a_2$ at position 2, etc., is $Q_{a_1} \cdot Q_{a_2} \cdots Q_{a_M}$. Continuing the last example, the probability that a random variant will have, e.g., alanine in the first randomized position and arginine in the second is $Q_{A_1} \cdot Q_{A_2} = (4/64)(6/64) = 0.005859$. Let $P_1, P_2, \ldots, P_n$ be an enumeration of these product probabilities, corresponding to the $n$ variants in variant space. We assume that the genes resulting from the randomization are stable, i.e., that the reversion rate is zero.

Let $L$ be the library size. To derive the probability of full coverage, define $C_i$ to be the event "the library contains at least one copy of variant $i$" for $i \in \{1, 2, \ldots, n\}$. Then, $P($full coverage$) = P(C_1 \cap C_2 \cap \cdots \cap C_n)$. Calculating exactly this probability requires the inclusion-exclusion principle (4), whose use becomes exponentially more burdensome as $n$ grows. However, because of the asymptotic independence of the events $C_1, \ldots, C_n$, a close (and known [5]) approximation for large $n$ is $P($full coverage$) \approx P(C_1) P(C_2) \cdots P(C_n) = \prod_{i=1}^{n} (1 - (1 - p_i)^L)$.

Let $T_k$ be the event "at least one of the top $k$ variants in variant space was discovered (i.e., was included in the library)." Assuming a continuous activity measurement scale, so that there are no ties, and assuming that no variant in variant space may be considered $a$ priori to be better than another, the distribution of the activity ranking (from best to worst) across variant space is uniform over all $n!$ permutations of its elements. Consequently, by conditioning on events of the type $B_v = \"the best variant in variant space is $v\"$, we get

$$P(T_k) = \sum_{v=1}^{n} P(T_k | B_v) P(B_v) = \frac{1}{n} \sum_{v=1}^{n} \left( 1 - (1 - p_v)^L \right)$$

Similarly, by conditioning on the events $B_{v_1v_2} = \"the two best variants in variant space are $v_1$ and $v_2\"$, for $v_1, v_2 \in \{1, 2, \ldots, n\}$, we get

$$P(T_k) = \sum_{v_1 < v_2} P(T_k | B_{v_1v_2}) P(B_{v_1v_2}) = \frac{1}{n(n-1)} \sum_{v_1 < v_2} \left( 1 - (1 - p_{v_1} - p_{v_2} - \cdots - p_n)^L \right)$$

and, more generally,

$$P(T_k) = \frac{1}{k!} \sum_{v_1 < v_2 < \cdots < v_k} \left[ 1 - (1 - p_{v_1} - p_{v_2} - \cdots - p_n)^L \right]$$

The number of summands in the last expression grows exponentially with both $k$ and $M$. When all 20 amino acids are considered at each of the randomized positions (as is most often the case), there are $20^M$ summands, an expression that exceeds $8 \times 10^{10}$ already when $k = M = 3$. To overcome the ensuing computational difficulties, we define the events $D_i = \"the $i$th-best variant in variant space was discovered\" and use the negative dependence among their complements (see the supplemental material) to bound the probabilities from below:

$$P(T_k) = P(D_1 \cup D_2 \cup \cdots \cup D_k) = 1 - P(D_1^c \cap D_2^c \cap \cdots \cap D_k^c) > 1 - P(D_1) P(D_2) \cdots P(D_k) = 1 - \left[ \left( 1 - p(L) \right)^k \right] \frac{1}{k!} \sum_{v_1 < v_2 < \cdots < v_k} \left[ 1 - (1 - p_{v_1} - p_{v_2} - \cdots - p_n)^L \right]$$

Because of the asymptotic independence of the $D_i$'s, this bound is very tight even for small values of $k$ and thus provides an excellent approximation for $P(T_k)$. For example, for $k = 3$, under an NNN randomization of $M = 3$ positions and when using a library of $L = 10,000$ variants, the exact value of $P(T_k)$ is 0.92037, whereas the approximation is 0.92035. The approximation improves as either $k$ or $M$ becomes larger.

When the probabilities $p_1, \ldots, p_n$ are all equal—as is the case when using the MAX protocol, for example—the problem becomes similar to the "occupancy problem" from probability theory (4), and some of the analysis is simplified.

**RESULTS**

In a saturation mutagenesis experiment, one needs to distinguish between two types of "best variant": the first is the best variant in variant space (i.e., the best among all distinct variants that could possibly be generated in the experiment), and the second is the best variant discovered in the experiment (i.e., the best among those that happened to constitute the library). Clearly, the activity of the latter can never exceed that of the former; if the best variant in variant space is included in the library, then the two types of "best variant" coincide. Similarly, there are two types of "second-best variant," "third-best variant," etc.

We will be interested in events of the type "at least one of the top $k$ variants in variant space was discovered," which are denoted by $T_k$. Note that $T_k$ does not mean "the best variant discovered is the $k$th-best variant in variant space": if $T_k$ has occurred, for example, it may well be that the best variant in variant space was discovered and not (or not only) the third best.

**Probability of discovering top variants.** Figure 1 shows how $P(T_k)$, the probability that at least one of the top $k$ variants in variant space was discovered, grows as a function of the library size, for various values of $k$, and when randomizing 1, 2, 3, and 4 positions. The probabilities of full coverage are shown as well, and all values correspond to NNN randomization (the results for NNN, NNB, and MAX randomization are qualitatively similar). Notice the logarithmic scale of the horizontal axis.

Most evident in the figure is the immense difference between the library size required to achieve a certain probability of full coverage and the size required to achieve that same probability, but of discovering the single best variant in variant space (the event $T_1$). For example, when randomizing two positions (so that
there are \( n = 20 \cdot 20 = 400 \) variants in variant space, 8,128 variants are required to achieve a 0.95 probability of full coverage; with this library size, the probability of discovering the best variant in variant space is \( P(T_1) = 0.99987 \), whereas only 2,130 variants are required to make \( P(T_1) = 0.95 \), a reduction of 74% in library size. This difference in the required library size grows as the number of randomized positions, and thus also the size of the variant space, increases.

Importantly, one can further reduce the library size by agreeing to also consider the second-best variant, the third-best, etc. For example, when randomizing two positions, only 875 variants are required to ensure a 0.95 probability of discovering any of the top two variants in variant space \([i.e., \text{to ensure that } P(T_2) = 0.95]\), and only 533 variants are required when considering any of the top three variants \([P(T_3) = 0.95]\). These numbers correspond, respectively, to 89% and 93% reduction in library size compared to the 8,128 variants required for a 0.95 probability of full coverage and to 59% and 75% reduction compared to the 2,130 variants required to ensure that \( P(T_1) = 0.95 \). Again, the savings grow as the number of randomized positions increases.

Comparing randomization schemes. Next, we compared four randomization schemes, all of which encode all 20 amino acids. The first is NNN randomization, which spans all 64 codons; the second is NNB randomization, which spans 48 codons, only one of which is a stop codon; the third is NNK randomization, which spans 32 codons, only one of which is a stop codon; and the fourth is MAX randomization, in which each of the 20 amino acids has a 1/20 probability to appear at each position.

Table 1 lists the library sizes required to achieve 0.95 and 0.99 probabilities of four events: full coverage, discovering the best variant in variant space \( (T_1) \), discovering any of the top two variants \( (T_2) \), and discovering any of the top three variants \( (T_3) \). Uniformly, MAX is the most efficient randomization scheme (it can be shown mathematically that this is true in general, and not only for the cases considered in Table 1), and NNN is the least efficient. Among the two remaining in-between schemes, NNK is at least as efficient as NNB in all cases considered and is strictly better in most cases, sometimes by a large margin. However, when the number of randomized positions is very small relative to \( k \) (a somewhat unrealistic case, which was therefore not considered in Table 1), the conclusion might be reversed; for example, when randomizing a single NNB position, a library of 15 variants is needed to ensure a 0.95 probability of discovering at least one of the top \( k = 4 \) variants, whereas under NNK randomization, the required library size is 16.

The differences between the required library sizes under the various schemes are very large in some cases. This is true especially when the probability of full coverage is the criterion (for example, there is more than an order of magnitude between the \( \sim 1.76 \times 10^6 \) variants required to attain a 0.99 probability when randomiz-
The idea at the basis of this work is that in a saturation mutagenesis experiment there is often more than one variant whose activity satisfies the design requirements. By being willing to consider any of the top-performing variants, and not only the single best one, one may substantially reduce the number of variants that need to be screened, and hence also the related cost and labor.

The main contribution of this work is in establishing novel criteria for determining the library size. Much of the existing probabilistic literature on saturation mutagenesis highlights the probability of full coverage as a criterion (1, 5, 11). Indeed, this criterion is used in practice (7). We have shown for the first time, to our knowledge, that using this criterion leads to needlessly large libraries. For example, when randomizing a single NNK (or NNS) position, one needs 130 variants to attain a 0.99 probability of discovering the best variant in variant space (i.e., to ensure that $P(T_k) = 0.99$); with this library size, the probability of full coverage is 0.82. When repeating the calculations for three randomized positions, 102,478 variants are needed to ensure the same highly conservative criterion $P(T_k) = 0.99$; with this library size, the probability of full coverage drops to 8.5%. Thus, the probability of full coverage is an overly conservative criterion for determining the library size.

Choosing the library size in an informed manner can save considerable resources. Champion et al. (2), for example, report screening about 8,000 variants when randomizing two NNK positions. When screening 8,000 variants, the probability of discovering the best variant in variant space, $P(T_1)$, is 0.997; to make this probability “only” 0.99, about 5,500 variants are needed. Both probabilities may be considered “practically equal to 1,” but the difference between the corresponding library sizes is substantial. A less stringent criterion, but one that is still reasonable, e.g., ensuring that $P(T_1)$ is equal to 0.95, results in even greater savings.

Equipped with the new criteria, we compared four randomization schemes: NNN, NNB, NNK (which is equivalent to NNS), and MAX. Patrick and Firth (10) had already compared several

### DISCUSSION

---

**Sensitivity to vector-only background.** All the results reported above assumed 100% yield (i.e., zero probability of getting vector-only background in the randomization; see Materials and Methods). Table 2 lists the library sizes required to achieve 0.95 and 0.99 probabilities of three events: discovering the best variant in variant space ($T_1$), discovering either of the top 2 variants ($T_{1,2}$), and discovering any of the top 3 variants ($T_{1,2,3}$) when randomizing two NNK positions and under various yield levels.

Perhaps surprisingly, the required increase in library size due to incomplete yield is not very large, even when the true yield is as low as 80%. Equivalently, ignoring the yield issue does not significantly distort the probabilities of the events $T_k$; if the library size is determined so that $P(T_k) = p$ for some $p$ close to 1 (say, $p \approx 0.95$) while falsely assuming 100% yield, then the true value of $P(T_k)$ will be at most 2 or 3% lower than $p$.

---

### TABLE 1 Library size required to achieve 0.95 and 0.99 probabilities of full coverage and of discovering any of the top $k$ variants when randomizing 1, 2, and 3 positions under NNN, NNB, NNK, and MAX randomizations

<table>
<thead>
<tr>
<th>No. of positions</th>
<th>Randomization</th>
<th>Library size required for:</th>
<th>0.95 probability</th>
<th>0.99 probability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Full coverage $k = 1$</td>
<td>$k = 2$</td>
<td>$k = 3$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$k = 1$</td>
<td>$k = 2$</td>
<td>$k = 3$</td>
</tr>
<tr>
<td></td>
<td>NNN</td>
<td>240</td>
<td>92</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>NNB</td>
<td>219</td>
<td>87</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>NNK</td>
<td>172</td>
<td>80</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>MAX</td>
<td>117</td>
<td>59</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>NNN</td>
<td>18,269</td>
<td>2,738</td>
<td>995</td>
</tr>
<tr>
<td></td>
<td>NNB</td>
<td>14,272</td>
<td>2,462</td>
<td>899</td>
</tr>
<tr>
<td></td>
<td>NNK</td>
<td>8,128</td>
<td>2,130</td>
<td>875</td>
</tr>
<tr>
<td></td>
<td>MAX</td>
<td>3,581</td>
<td>1,197</td>
<td>598</td>
</tr>
<tr>
<td></td>
<td>NNN</td>
<td>1,344,623</td>
<td>79,041</td>
<td>25,585</td>
</tr>
<tr>
<td></td>
<td>NNB</td>
<td>862,662</td>
<td>67,394</td>
<td>22,070</td>
</tr>
<tr>
<td></td>
<td>NNK</td>
<td>341,601</td>
<td>55,485</td>
<td>21,051</td>
</tr>
<tr>
<td></td>
<td>MAX</td>
<td>95,654</td>
<td>23,965</td>
<td>11,983</td>
</tr>
</tbody>
</table>

---

### TABLE 2 Library size required to achieve 0.95 and 0.99 probabilities of discovering any of the top $k$ variants for various yields and $k$ values when randomizing two NNK positions

<table>
<thead>
<tr>
<th>Yield (%)</th>
<th>0.95 probability</th>
<th>0.99 probability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k = 1$</td>
<td>$k = 2$</td>
</tr>
<tr>
<td>100</td>
<td>2,130</td>
<td>875</td>
</tr>
<tr>
<td>95</td>
<td>2,242</td>
<td>921</td>
</tr>
<tr>
<td>90</td>
<td>2,366</td>
<td>972</td>
</tr>
<tr>
<td>80</td>
<td>2,662</td>
<td>1,093</td>
</tr>
</tbody>
</table>
characteristics of these four schemes, but not those at the center of this work, namely, the library size required to ensure a high probability of discovering a top-performing variant or of full coverage. It is intuitively clear why NNN emerged as the least efficient scheme in this sense, as it induces both the highest per-position probability for a premature stop codon (namely, 3/64) and the most uneven collection of per-position probabilities for the 20 amino acids (ranging from 1/64 for the rarest amino acid to 6/64 for the most common). Similarly, MAX is clearly the most efficient scheme, with a zero probability for a stop codon and perfectly even probabilities of 1/20 for each of the 20 amino acids. However, it is not immediately clear which of the two intermediate schemes, NNB and NNK, is better, as NNB induces a lower probability for a stop codon (1/48 in NNB versus 1/32 in NNK), but NNK enjoys a lower ratio between the most common and the rarest amino acids (5/48 to 1/48 in NNB versus 3/32 to 1/32 in NNK). The comparison between these two schemes is indeed more delicate: in all the cases considered in Table 1, NNK is at least as efficient as NNB, but as mentioned in Results, in other (more extreme) cases, NNB performs better. Thus, there is no categorical rule for choosing between NNB and NNK, and practitioners who are facing the choice are advised to calculate the required library sizes for both schemes under their specific experimental conditions (the number of randomized positions, desired probability, and choice of k) and to choose the scheme that is found to be more efficient.

The methodology presented in this work is general enough that it can be applied to any randomization scheme, including schemes that do not encode all 20 amino acids, such as the NDT approach of Reetz et al. (15). Furthermore, the methodology can be used in conjunction with more elaborate design approaches that involve a sequential combination of saturation mutagenesis steps, such as iterative saturation mutagenesis (ISM) (14) and combinatorial active-site saturation test (CAST) (13).

To validate empirically the predictions of the calculations presented in this paper, one needs to choose a protein, identify in its sequence positions suitable for saturation mutagenesis, systematically produce all variants in the corresponding variant space (possibly with the aid of site-directed mutagenesis), measure the activities of all of these variants to find with certainty all top-performing ones, and then run a large number of saturation mutagenesis experiments (with various library sizes and with several repetitions for each library size). The results of these experiments need to be analyzed statistically and compared with the predictions outlined above. We plan to run this experiment as a sequel to this work.

ACKNOWLEDGMENTS

I thank Ayelet Fishman for introducing me to the problem of saturation mutagenesis, for illuminating discussions, and for providing feedback on a draft of this paper. I also thank three anonymous reviewers for their constructive comments.

REFERENCES