Probabilistic Methods in Directed Evolution: Library Size, Mutation Rate, and Diversity

Yuval Nov

Abstract

Directed evolution has emerged as an important tool for engineering proteins with improved or novel properties. Because of their inherent reliance on randomness, directed evolution protocols are amenable to probabilistic modeling and analysis. This chapter summarizes and reviews in a nonmathematical way some of the probabilistic works related to directed evolution, with particular focus on three of the most widely used methods: saturation mutagenesis, error-prone PCR, and in vitro recombination. The ultimate aim is to provide the reader with practical information to guide the planning and design of directed evolution studies. Importantly, the applications and locations of freely available computational resources to assist with this process are described in detail.

Key words Directed evolution, Probabilistic modeling, Library generation, Computational tools, Saturation mutagenesis, Error-prone PCR, DNA shuffling, Staggered extension process

1 Introduction

Over the past two decades, directed evolution has proven to be an invaluable tool in the hands of protein engineers. It has been successfully used in thousands of experiments to improve various desirable protein properties, including catalytic activity, thermostability, enantioselectivity, and binding affinity [1–4].

In a directed evolution experiment, researchers “breed” in vitro a population of protein variants via successive rounds of mutagenesis and screening. In each mutagenesis stage, the DNA sequences of one or several template genes are randomly altered, yielding a large number of slightly mutated genes. In the following screening stage, proteins expressed from a random sample of these genes are screened (or selected), and improved variants, if present, are identified; the set of variants sampled for screening is called *a library*. This process may continue by taking one or more of the best performing variants to serve as templates for the next round.

Because the very essence of directed evolution hinges on randomness, the process is amenable to probabilistic analysis. The goal of this chapter is to summarize and review in a nonmathematical way some of the works and tools related to the probabilistic analysis of directed evolution. Such an analysis allows the practitioner both to design better experiments and to interpret correctly the results of an experiment already undertaken. This chapter will focus on three common directed evolution methods: saturation mutagenesis, error-prone PCR, and in vitro recombination.

Regardless of the method used, screening multiple copies of the same protein variant is undesirable, as it expends laboratory resources without adding genetic diversity. The same is true for variants expressed from genes mutated to include a frameshift or a premature stop codon, which are essentially guaranteed to lack function. The total number of protein variants screened, here termed *the library size*, should therefore be distinguished from the library's *effective size*: the number of distinct, full-length protein variants in the library. The library's effective size is a central concept in this chapter. A related important concept is that of *sequence space*—the set of all protein sequences that *could possibly* be generated and included in the library; depending on experimental design, the sequence space may be hyper-astronomically large, well beyond that which could be plausibly generated or screened in a laboratory. Probability theory allows us to study how these three concepts are interrelated and to calculate, for example, what percentage of sequence space is expected to be represented in a library of given size, or by how much the effective library size is expected to increase if the library size is increased by, say, a thousand variants.

“All models are wrong, but some are useful,” wrote the great statistician George E. P. Box. His words are apt. Like all mathematical models of real-world phenomena, the predictions of probabilistic models for directed evolution are only as valid as their assumptions. But these assumptions are never fully satisfied: different oligonucleotides may have different annealing probabilities; mutations, even at distant positions, do not occur completely independently; etc. Indeed, as Patrick and Firth [5] note, there is still a gap between the perfect library on paper and the perfect library in the lab. Fortunately, this gap is closing, and the analyses and results presented below may provide at least a good approximation for the experimental results.

2 Saturation Mutagenesis

In saturation mutagenesis, several protein positions are first identified as having a high potential to accommodate favorable mutations. The chosen positions are then randomized at the DNA level using degenerate oligonucleotides, so that each randomized position

in each of the translated protein variants contains a random amino acid. The resulting library is then screened, in the hope of discovering among its members highly improved variants.

Mathematically, saturation mutagenesis is the most tractable of the methods discussed in this chapter. Let M denote the number of randomized positions. When all 20 amino acids are allowed at each of the randomized positions, as is commonly the case, the size of the protein sequence space is 20^M . This expression grows very rapidly with M , so that when randomizing more than 10 positions or so, even the most efficient screens or selections are unable to interrogate more than a minuscule fraction of the corresponding sequence space. In practice, however, saturation mutagenesis experiments usually involve $M \leq 4$ randomized positions, so that a manageable library may cover all or almost all of sequence space—hence the word “saturation” in the method’s name.

2.1 Randomization Schemes

The simplest form of randomization in saturation mutagenesis is termed NNN: here, when synthesizing the degenerate oligonucleotides, each of the four bases is equally likely to appear at each of the three nucleotide positions of the randomized codon, so that in the resulting gene, all $4^3 = 64$ codons are equally likely to appear. In contrast, because of the redundancy of the genetic code, the resulting distribution across the 20 amino acids at the protein level is far from uniform, with probabilities ranging from $1/64$ (for Met and Trp) to $6/64$ (for Arg, Leu, and Ser); importantly, there is a $3/64$ probability that NNN randomization will produce a premature stop codon at any randomized position.

Both these phenomena—uneven amino acid distribution and nonzero probability for a stop codon—can be shown mathematically to reduce the effective library size, and their negative effects amplify as more positions are randomized. Figure 1 shows how the probability of having at least one stop codon in the randomized gene increases as a function of the number of randomized positions.

To mitigate these problems, reduced codon sets are often used, i.e., only some of the four bases are allowed to appear (in equimolar proportions) at one or more nucleotide positions of the randomized codons. The most common such randomization schemes are NNK, NNS, and NNB; see Table 1 for the meaning of these abbreviations and the right panel of Fig. 1 for the resulting amino acid distributions. NNK randomization is in principle identical to NNS in terms of the translated sequences but is more efficient in practice when used in *E. coli* and *S. cerevisiae*, as it conforms better to the codon usage of these organisms [5].

An ideal randomization scheme generates a perfectly uniform distribution over the 20 amino acids while assigning zero probability for a stop codon. Such schemes, which we term UNIF, may be achieved either through the special MAX oligonucleotides [6] or through a proper mixture of several standard

Fig. 1 *Left:* The probability of introducing at least one premature stop codon in a gene, as a function of the number of randomized positions, for six saturation mutagenesis randomization schemes. *Right:* The probability distribution across the 20 amino acids and the stop codon (denoted by an *asterisk*) for six randomization schemes

Table 1
Degenerate base abbreviations

N	V	H	D	B	M	R	W	S	Y	K
A/C/G/T	A/C/G	A/C/T	A/G/T	C/G/T	A/C	A/G	A/T	C/G	C/T	G/T

degenerate oligonucleotides (e.g., a mixture of NDT, VMA, ATG, and TGG at 12:6:1:1 molar ratio [7]). Using the latter approach, one can reduce the number of required degenerate codons to three, at the expense of a slight deviation from a perfectly uniform distribution, by using the “22c-trick” of Kille et al. [8].

Yet another way to influence the amino acid distribution is via “doped” (also known as “spiked”) oligonucleotides. Here, during oligonucleotide synthesis, non-equimolar proportions of the four bases are used at some (or all) of the three nucleotide positions of the randomized codon. This approach may be used in conjunction with the mixing approach mentioned above, to produce extra flexibility in shaping the resulting amino acid distribution. The probabilistic and algorithmic aspects of such randomization schemes have received attention in the literature [9–11], but perhaps because of the special equipment and expertise they require, they are seldom used [12].

Rather than allowing all 20 amino acids at each of the randomized positions, some authors advocate using a smaller, yet chemically balanced, subset of the amino acids. Sequence space can be reduced significantly in this way, thus making it easier to explore. See, for example, the NDT randomization approach of Reetz et al. [13].

2.2 Library Metrics

Once the randomized positions and the randomization scheme are determined, the main decision left to be made is of the library size. Clearly, the more variants screened, the higher the probability of discovering improved variants, but one needs to balance the benefits of this increased probability with the additional laboratory work and expense large libraries entail.

To choose judiciously the library size, the experimenter needs a metric to quantify how exhaustive is a library, *a priori*, given its size. Several such metrics have been suggested in the literature. The first is the probability of “full coverage,” i.e., the probability that the library contains *all* variants in sequence space. The rationale behind this definition is that in saturation mutagenesis, the experimenter typically wishes to consider all possible variants, so the probability that all are indeed represented in the library appears initially a reasonable metric. The mathematical analysis of this metric is in essence the same as that of the “coupon collector’s problem” from probability theory, which is also known as the “occupancy problem” [14]. The second metric is “expected coverage” (sometimes called “completeness” or “fractional completeness”), which is the fraction of sequence space that is expected to be represented in the library. The expected coverage is related to the expected effective library size via the simple formula:

expected effective library size = expected coverage $\times$ size of sequence space.

To illustrate these concepts, consider a hypothetical randomization scheme for a single position, such that under a certain library size, there is a 0.8 probability for having all 20 amino acids in the library, a 0.15 probability for 19, and a 0.05 probability for only 18. Then, the probability of full coverage is 0.8, the expected coverage is $0.8 \cdot (20/20) + 0.15 \cdot (19/20) + 0.05 \cdot (18/20) = 0.9875$, and the expected effective library size is $0.9875 \cdot 20 = 19.75$. Exact and approximate formulae for computing the probability of full coverage and the expected coverage can be found in the pioneering works of Patrick et al. [15], Bosley and Ostermeier [16], and Firth and Patrick [17].

Recently, Nov [18] proposed a third metric, which is in fact a family of metrics: the probability that the library contains at least one of the best k variants. When $k=1$, this metric is the probability that the library contains the single best variant in sequence space, which is clearly a desirable outcome of the experiment; this probability can be shown to always equal the expected coverage.

Fig. 2 Saturation mutagenesis library metrics as a function of the library size, when randomizing a single NNK position (*left*) and two NNK positions (*right*). Note the logarithmic scale of the horizontal axis

The rationale behind this metric for $k \geq 2$ is that often, especially when sequence space is large (say, is at least in the hundreds), there are several variants that satisfy the design requirements of the protein at hand, so discovering any of them will be considered a success. By being willing to not restrict oneself to solely the best variant, the library size may be reduced significantly, and the experimenter may thus reduce labor and expenses.

Figure 2 shows the abovementioned metrics as a function of the library size, when randomizing one or two NNK positions. As the figure shows, the differences between the metrics at a given library size are substantial. For example, when screening 500 variants after randomizing two NNK positions, the probability of full coverage is essentially zero, the probability of discovering the single best variant is 0.61 (this is also the expected coverage), and the probability of discovering at least one of the top 3 variants is 0.94. More importantly, the differences are substantial also when reading the graph in inverted axis order: to ensure a 0.95 probability of full coverage, 8,128 variants are required; for a 0.95 probability of discovering the best variant, the number drops to 2,130; and for a 0.95 probability of discovering any of the top 3 variants, the number is 534.

Which metric should be used, then? The probability of full coverage is not recommended, as it leads to needlessly large libraries. The reason for this is that screening all—or almost all—of sequence space is not a goal in its own right, but rather just a means to an end: discovering the best variants. The search for the best variants, in turn, obeys the law of diminishing marginal utility—after, say, 90 % of sequence space has been explored, it

Fig. 3 The probability of discovering the best variant as a function of the library size for six saturation mutagenesis randomization schemes

becomes increasingly rare for subsequent screened variants to turn out to be new ones, so more and more variants need to be screened to increase the coverage by one percent; at the same time, the probability that the remaining 10 % of sequence space will contain variants surpassing all those previously screened is greatly reduced.

In a high-stakes protein design process, the designer may not wish to settle for less than the best possible variant. The recommended strategy in this case is to choose a library size such that the probability of discovering the best variant (which is the same as the expected coverage) is very close to 1—say, equal to 0.99. As the design requirements become less stringent, the designer may lower this probability or change the metric to the probability of discovering any of the top k variants, for larger and larger k . In situations where there are likely to be many satisfactory solutions for the design problem—say, when trying to improve thermostability moderately—the designer may choose a rather large k .

Figure 3 is concerned with the role of the randomization scheme in a saturation mutagenesis experiment. The curves show the probability of discovering the best variant when randomizing two protein positions with various randomization schemes. It can be seen that to ensure a 0.9 probability of discovering the best variant, the most efficient method is UNIF, which requires a library of 920 variants; next best, by a small margin, is the “22c-trick” scheme (1,025 variants), followed by NNK or NNS (1,513), NNB (1,651), and NNN (1,834). The results are qualitatively similar for all other metrics and for any number of randomized positions.

2.3 Computational Resources

Several online computational resources concerning saturation mutagenesis are listed below, ordered by their publication date.

GLUE [15, 19] (<http://guinevere.otago.ac.nz/cgi-bin/aef/glue.pl>) is a web program through which users can compute three quantities: (1) the expected coverage corresponding to a given library size, (2) the library size corresponding to a given expected coverage (the inverse calculation), and (3) the library size corresponding to a given probability of full coverage. Importantly, all possible variants are assumed in *GLUE* to be equally probable, so the program applies, for example, to randomization at the DNA level and to MAX randomization at the protein level, but not to randomization at the protein level under nonuniform schemes such as NNN or NNK.

CASTER [20] (http://www.kofo.mpg.de/media/2/D1108347/0987095526/ISM_tools.zip) is an Excel worksheet that computes the library size required to ensure a specified expected coverage of DNA (not protein) sequence space, under arbitrary codon-based randomization.

GLUE-IT [17] (<http://guinevere.otago.ac.nz/cgi-bin/aef/glue-IT.pl>), where IT stands for “Including Translation,” is an improved version of *GLUE* which computes library statistics at the protein level. On specifying a library size and any codon-based randomization scheme involving up to six codons, the program computes the expected effective library size, the expected coverage, the probability of full coverage, and the number of possible sequences both at the DNA and at the protein levels.

Kong [21] developed a web server (http://graphics.med.yale.edu/cgi-bin/lib_comp.pl) that allows users to compute the mean and variance of the effective library size. The reported results concern sequence diversity at the DNA (not protein) level. Of the servers reviewed in this chapter, this is the only one that computes also the variance and also the only one that allows arbitrary doping via direct specification of the molar ratio between the four bases. The number of randomized (DNA) positions is not restricted, but when this number is large and the doping varies across positions, the computation of the variance may take more than several minutes.

TopLib [18] (<http://stat.haifa.ac.il/~yuval/toplib>) is a web server that handles computations of the probability of discovering at least one of the top k variants (recall that the case $k=1$ gives the expected coverage), as well as the probability of full coverage. The server can compute either the probability corresponding to a given library size or the library size required to ensure a given probability (the inverse calculation). Any randomization scheme of up to 12 positions is allowed, either by specifying degenerate codons or by specifying directly the probabilities for the 20 amino acids at each position. The latter option is useful for MAX randomization, as well as for randomization via doped oligonucleotides

(because each doping scheme induces a distribution over the 20 amino acids). Users can also specify the yield factor—the per-variant probability that the randomization succeeds.

DC-Analyzer [7] (http://cobi.uestc.edu.cn/resource/dc_analyzer/view) is a downloadable Java program that computes the optimal oligonucleotide mixtures (*see* Subheading 2.1) required to generate specified subsets of the 20 amino acids.

3 Error-Prone PCR

Error-prone PCR (epPCR) is a modification of the standard PCR protocol that deliberately introduces mutations during the elongation step. The mutation rate can be tuned by the experimenter via the choice of the polymerase used and by adjusting the reaction conditions. To avoid too many inactivated variants in the resulting library, the mutation rate is typically set to produce an average of no more than 5 mutations per variant at the protein level.

Unlike saturation mutagenesis, epPCR is therefore a whole-protein mutagenesis method, which targets the entire DNA sequence. Consequently, sequence space consists in principle of all 20^M protein variants (where M now denotes the entire protein length); even for a short protein of length $M=100$, this number dwarfs the estimated number of atoms in the universe. But again unlike saturation mutagenesis, there are vast differences between the probabilities corresponding to the variants: only those that differ from the template protein in relatively few positions have a non-negligible probability to occur, whereas the vast majority of the theoretical sequence space is practically inaccessible.

3.1 Probabilistic Analysis

From the probabilistic point of view, epPCR is closely related to the well-studied theory of branching processes [22]. Unfortunately, certain characteristics of epPCR differentiate it from a pure branching process and thus complicate its analysis: parent sequences do not “die,” but rather remain in the reaction tube; the mutation process is modulated on the amplification process, so that different mutations on the same sequence may have different ages; and the resulting DNA sequences need to be translated to protein sequences. In the words of Piau [23], epPCR, “although easy to describe, is not analytically tractable.” Indeed, researchers invariably rely on various mathematical approximations and simplifying assumptions in its analysis.

Unlike in the case of saturation mutagenesis, the probabilistic analysis of epPCR depends crucially on parameters that need to be estimated from experimental data. These are the reaction’s efficiency, which is the per-cycle probability of successfully replicating a sequence, and the mutational spectrum, which is a set of parameters describing the mutation probabilities from each base to each

of the other three during elongation, as well as the probabilities of insertions and deletions. The need to estimate these parameters, typically through a small, preliminary experiment, complicates the analysis further and weakens the prediction ability.

Early probabilistic works on epPCR analyzed the process only at the DNA level and tended to address questions having more of a theoretical flavor [24–27]. Examples of the mathematical results derived include the distribution of the number of DNA mutations in a randomly chosen sequence, the distribution of the distance (measured in the number of different bases) between two randomly chosen sequences, the distribution of the number of PCR replications separating two randomly chosen sequences, and the probability that a randomly chosen DNA molecule will follow a specified sequence. Piau used later mean field theory to provide a mathematically rigorous justification for some of the approximations used in the above studies [23].

Later works tended to focus on analysis that is more directly relevant to epPCR practitioners. Bosley and Ostermeier [16] appear to be the first to address the issue of DNA–protein translation in their analysis of epPCR. Following their earlier work on DNA diversity in epPCR libraries [15], Firth and Patrick [17] derived a method to estimate the expected effective library size at the protein level and to partition the expected library composition into sub-libraries according to the number of mutations: single-mutant variants, double mutants, etc. (Note that the term “effective library size” used here is called “number of unique, full-length variants” in their terminology, as they reserve the term “effective library size” to the total number of full-length variants, distinct or not.)

The left panel of Fig. 4 shows the expected effective library size, broken down into sub-libraries according to the number of mutations, as a function of the library size, as well as the expected number of distinct DNA sequences. The gene used in this example is *vsI* (GenBank accession number EU627691), an 891-bp-long gene encoding the tyrosinase VS1 of *Bacillus megaterium* [28]. The first important feature of this graph is that DNA diversity is significantly greater than protein diversity; since only the latter is relevant to practitioners, one is advised to not design epPCR experiments based on DNA diversity analysis. The second important feature is that the curves grow only slightly sublinearly, so that doubling the library size almost doubles the expected effective library size and also almost doubles the numbers corresponding to each of the sub-libraries. The deviations from perfect linearity are seen more clearly in the right panel of Fig. 4, which shows the expected fraction of each of the effective sub-libraries from the total expected effective library size. The number of distinct single mutants, for example, can never exceed $19 \cdot 297 = 5,643$ (19 substitutions for each of the 297 protein positions); as the library size increases, more and more single mutants are recurring variants

Fig. 4 *Left*: Expected effective epPCR library size, broken into sub-libraries according to the number of mutations, as a function of the library size. *Right*: Expected fractions (in percent) of the effective sub-library sizes. The underlying gene is *vs1* [28], and the graphs were drawn based on the output of the PEDEL-AA program [17], using the PCR distribution and the server's default parameters

Fig. 5 Expected effective epPCR library size for the *vs1* gene, broken into sub-libraries according to the number of mutations, as a function of the mutation rate. Library size is 50,000 DNA sequences, and other parameters are as in Fig. 4

already included in the library, and because this phenomenon happens more slowly to multiple-point mutants, the fraction of single-point mutants decreases as a function of the library size.

Figure 5 shows the influence of the mutation rate on epPCR library diversity, for a fixed-size library of 50,000 variants. DNA diversity is again much greater than protein diversity, and the

expected number of distinct DNA sequences attains its maximal value (which is 50,000, the library size) rather quickly. In contrast, the expected effective protein library size not only never reaches this value but starts decreasing at a certain point. The reason for this is that a high mutation rate increases the fraction of variants with premature stop codons and thus lowers the effective library size.

To maximize the expected effective library size, the experimenter needs to set the mutation rate to 6.1 (indicated by the right vertical line). However, if the experimenter wishes to consider only variants having up to, say, four mutations (because further mutations are believed to inactivate even full-length, in-frame variants), then the optimal mutation rate is 3.7 (left vertical line). These numbers apply to the *ysl* gene under certain specific experimental conditions and will vary as either the gene or the conditions are changed.

The interrelations between mutation rate, library diversity, and retention of function are at the center of the work of Drummond et al. [29]. The authors model epPCR somewhat differently from Firth and Patrick [17]: instead of basing the analysis on the precise template DNA sequence and on the entire estimated mutational spectrum, they use one parameter for the probability that a mutation (regardless of its position) will result in a non-synonymous change and a second parameter for the probability that a mutation will result in a truncation (due to a premature stop codon, insertion, or deletion). Another important difference is that whereas Firth and Patrick consider all full-length, in-frame variants, Drummond et al. recognize that many of these variants may be devoid of function and model this event via a third parameter. They use their model to study the number of distinct and functional variants that are expected to appear in the library and to determine the optimal mutation rate that maximizes this number. Unlike PEDEL-AA (see below), there is currently no publicly available computer program for calculations under this model.

Although using different models, both Drummond et al. and Patrick et al. conclude that epPCR practitioners should generally use a higher mutation rate than they normally do, to increase library diversity and the probability of discovering improved variants.

3.2 Computational Resources

PEDEL and PEDEL-AA. PEDEL [15, 19] (<http://guinevere.otago.ac.nz/cgi-bin/acf/pedel.pl>) is a web server through which users can calculate the expected number of distinct DNA sequences in an epPCR library. PEDEL-AA [17] (<http://guinevere.otago.ac.nz/cgi-bin/acf/pedel-AA.pl>), where AA stands for “Amino Acid,” is a much improved version of the program, which computes various statistics of epPCR libraries at the protein level. The most important of the many statistics reported are the effective library size and its breakdown into sub-libraries according to the number of mutations.

Volles and Lansbury [30] developed a C program (downloadable from http://lansbury.bwh.harvard.edu/Michael_Volles.htm) that computes various epPCR library statistics. The program can calculate some of these statistics analytically, but other statistics, among them the effective library size, are estimated only via simulation (the program simulates only a single realization of the process). The user needs to prepare two input files—one for the template DNA sequence and one for the mutational spectrum—and to answer, via a terminal interface, about 20 queries regarding the program's execution.

MAP^{2.0}3D [31] (<http://map.jacobs-university.de/map3d.html>), where MAP stands for “Mutagenesis Assistant Program,” is a web server that compares the performance of 19 mutagenesis methods having different mutational spectra. Given a template DNA sequence, the server computes various protein-level statistics for each of the mutagenesis methods, e.g., the fraction of variants with a premature stop codon, the average number of amino acid substitution per residue, and a codon diversity coefficient. All statistics are calculated assuming a single nucleotide substitution in the mutagenized gene and thus do not incorporate parameters such as the library size and the reaction's efficiency. The server can also incorporate structural information about the target protein, when available, to aid in selecting the mutagenesis method. The reader is referred to chapter 19 by Verma et al. in this volume for further information.

4 In Vitro DNA Recombination

In in vitro DNA recombination, fragments from two or more homologous parent genes are assembled randomly into genes with novel DNA sequences. Common protocols for DNA recombination include DNA shuffling [32] and the staggered extension process (StEP) [33].

Variable positions that are nearby along the recombined sequence are likely to belong to the same fragment and hence to originate from the same parent gene. Consequently, the mutations in a recombined gene are in general not independent; the dependence between any two such mutations decreases as the distance between them increases and vanishes completely if this distance is greater than the maximal possible fragment length.

4.1 Probabilistic Analysis

DNA recombination experiments are more difficult to model and analyze than saturation mutagenesis and epPCR experiments, as they involve more parameters set by the experimenter (e.g., the number and composition of the parent genes and the experimental conditions that determine the distribution of the fragment lengths),

Fig. 6 Expected effective library size as a function of the library size in a DNA recombination experiment with two genes, based on the model of Patrick et al. [15]. All genes are 600 bp long, and their variable positions are either at 100, 150, 200, 250, 300, 350, 400, 450, and 500 (“distant”) or at 280, 285, 290, 295, 300, 305, 310, 315, and 320 (“near”). The mean number of crossovers per sequence is indicated by λ . The *dashed curve* corresponds to the diversity in a saturation mutagenesis library with nine randomized positions, each with two equiprobable possible amino acids

and these parameters influence the composition of the resulting genes in an intricate way. Further difficulties arise from the need to estimate some other parameters. As a result, the probabilistic modeling approaches in the literature vary greatly, and as in the case of epPCR, the analysis relies on various approximations and simplifying assumptions.

Early mathematical works on DNA recombination addressed questions such as the distribution of the number of mutations in a randomly chosen reassembled gene [34], the distribution of the fragment number and fragment length in the reassembled gene, the recombination probability between two mutations, and the probability of assembling a given target sequence [27, 35–37].

Patrick et al. [15] studied the expected effective library size resulting from a DNA recombination between two homologous genes, assuming a Poisson distribution for the number of crossovers between any two consecutive variable positions. Figure 6 is based on the model of Patrick et al., and shows the expected effective library size as a function of the library size. There are 9 variable positions in this example, so sequence space includes $2^9 = 512$ variants, a number indicated by the horizontal dotted line. The distance between adjacent variable positions is either 5 bp (“near”) or

50 bp (“distant”), and the mean number of crossovers per sequence was either $\lambda=2$ or $\lambda=10$. It can be seen that diversity increases as the variable positions are further apart and as the crossovers are more frequent.

As Patrick et al. observe, in the extreme case of remote variable positions and highly frequent crossovers, the mutations are essentially independent of each other, and all possible variants become equally likely. This situation is similar to a saturation mutagenesis experiment in which there are two equiprobable possible amino acids at each randomized position. The dashed curve in Fig. 6 corresponds to the expected effective library size in such a saturation mutagenesis experiment; indeed, the curve corresponding to the high-crossover distant mutation follows it closely.

4.2 Computational Resources

DRIVeR [15, 19] (<http://guinevere.otago.ac.nz/cgi-bin/aef/driver.pl>), whose name stands for “Diversity Resulting from In Vitro Recombination,” is to date the only publicly available computer program for calculations related to DNA recombination. This program uses the model and analysis of Patrick et al. [15] and thus applies only to experiments with two recombined genes. Given the library size, the gene length, the mean number of crossovers per gene, and the location of the variable positions, the program computes the expected effective library size. The program can also estimate the true mean number of crossovers from the observed mean number.

5 Concluding Remarks

In this chapter, we have reviewed the probabilistic models and analyses behind three widely used directed evolution methods—saturation mutagenesis, error-prone PCR, and DNA recombination. Of the many other existing methods, some fall into the same mathematical analysis category as one of the abovementioned methods; e.g., SISDC [38] is a shuffling method that achieves complete independence between the mutations at the variable positions and thus should be analyzed using the probabilistic methodology of saturation mutagenesis. Other methods require moderate modification of existing analyses, such as ISOR [39], which is mathematically similar—but not identical—to saturation mutagenesis. Still other methods, such as SeSaM [40], call for completely new modeling and analysis.

The existing probabilistic methods can certainly be improved. For example, the chemistry underlying saturation mutagenesis dictates that not all oligonucleotides designated to a given position are equally likely to anneal. The resulting bias can be mitigated by adjusting the reaction conditions [41] or by using a proper combination of polymerases [42] but should also be accounted for in the

probabilistic analysis; Denault and Pelletier [43] suggest to use a chi-squared test to detect such a bias. Another possible improvement is *interval prediction*: rather than reporting only the expected value of the predicted quantity, it might be illuminating to account for sampling variability by reporting, e.g., “when screening 5,000 variants, there is a 95 % probability that the effective library size will be between 4,150 and 4,320.” Kong [21], who analyzed saturation mutagenesis with doping, made an important step in this direction by deriving the variance of the expected effective library size but stopped short of deriving the prediction interval.

Probability theory can aid in protein design also beyond its role in directed evolution. Recent works demonstrate that given a moderate amount of sequence-activity data, probabilistic reasoning can help in deciding how to optimally combine existing mutations and point at few, highly targeted variants with a high predicted activity [44–47]. The mathematics underlying these methods is more involved than that behind the methods discussed in this chapter.

References

1. Moore JC, Arnold FH (1996) Directed evolution of a para-nitrobenzyl esterase for aqueous-organic solvents. *Nat Biotechnol* 14(4):458
2. Giver L, Gershenson A, Freskgard PO, Arnold FH (1998) Directed evolution of a thermostable esterase. *Proc Natl Acad Sci U S A* 95(22):12809–12813
3. Reetz MT, Zonta A, Schimossek K, Jaeger K-E, Liebeton K (1997) Creation of enantioselective biocatalysts for organic chemistry by in vitro evolution. *Angew Chem Int Ed* 36(24):2830–2832
4. Boder ET, Midelfort KS, Wittrup KD (2000) Directed evolution of antibody fragments with monovalent femtomolar antigen-binding affinity. *Proc Natl Acad Sci U S A* 97(20):10701–10705
5. Patrick WM, Firth AE (2005) Strategies and computational tools for improving randomized protein libraries. *Biomol Eng* 22(4):105–112
6. Hughes MD, Nagel DA, Santos AF, Sutherland AJ, Hine AV (2003) Removing the redundancy from randomised gene libraries. *J Mol Biol* 331(5):973–979
7. Tang L, Gao H, Zhu X, Wang X, Zhou M, Jiang R (2012) Construction of “small-intelligent” focused mutagenesis libraries using well-designed combinatorial degenerate primers. *Biotechniques* 52(3):149–158
8. Kille S, Acevedo-Rocha CG, Parra LP, Zhang Z-G, Opperman DJ, Reetz MT, Acevedo JP (2013) Reducing codon redundancy and screening effort of combinatorial protein libraries created by saturation mutagenesis. *ACS Synth Biol* 2(2):83–92
9. Tomandl D, Schober A, Schwienhorst A (1997) Optimizing doped libraries by using genetic algorithms. *J Comput Aided Mol Des* 11(1):29–38
10. Jensen LJ, Andersen KV, Svendsen A, Kretzschmar T (1998) Scoring functions for computational algorithms applicable to the design of spiked oligonucleotides. *Nucleic Acids Res* 26(3):697–702
11. Wolf E, Kim PS (1999) Combinatorial codons: a computer program to approximate amino acid probabilities with biased nucleotide usage. *Protein Sci* 8(3):680–688
12. Neylon C (2004) Chemical and biochemical strategies for the randomization of protein encoding DNA sequences: library construction methods for directed evolution. *Nucleic Acids Res* 32(4):1448–1459
13. Reetz MT, Kahakeaw D, Lohmer R (2008) Addressing the numbers problem in directed evolution. *Chembiochem* 9(11):1797–1804
14. Feller W (1971) An introduction to probability theory and its applications, vol 2. Wiley, New York
15. Patrick WM, Firth AE, Blackburn JM (2003) User-friendly algorithms for estimating completeness and diversity in randomized protein-encoding libraries. *Protein Eng* 16(6):451–457

16. Bosley AD, Ostermeier M (2005) Mathematical expressions useful in the construction, description and evaluation of protein libraries. *Biomol Eng* 22(1–3):57–61
17. Firth AE, Patrick WM (2008) GLUE-IT and PEDEL-AA: new programmes for analyzing protein diversity in randomized libraries. *Nucleic Acids Res* 36(suppl 2):W281–W285
18. Nov Y (2012) When second best is good enough: another probabilistic look at saturation mutagenesis. *Appl Environ Microbiol* 78(1):258–262
19. Firth AE, Patrick WM (2005) Statistics of protein library construction. *Bioinformatics* 21(15):3314–3315
20. Reetz MT, Carballeira JD (2007) Iterative saturation mutagenesis (ISM) for rapid directed evolution of functional enzymes. *Nat Protoc* 2(4):891–903
21. Kong Y (2009) Calculating complexity of large randomized libraries. *J Theor Biol* 259(3):641–645
22. Grimmett GR, Stirzaker DR (2001) Probability and random processes. Oxford University Press, Oxford
23. Piau D (2004) Mutation–replication statistics of polymerase chain reactions. *J Comput Biol* 9(6):831–847
24. Sun F (1995) The polymerase chain reaction and branching processes. *J Comput Biol* 2(1):63–86
25. Wang D, Zhao C, Cheng R, Sun F (2000) Estimation of the mutation rate during error-prone polymerase chain reaction. *J Comput Biol* 7(1–2):143–158
26. Weiss G, von Haeseler A (1995) Modeling the polymerase chain reaction. *J Comput Biol* 2(1):49–61
27. Moore GL, Maranas CD (2000) Modeling DNA mutation and recombination for directed evolution experiments. *J Theor Biol* 205(3):483–503
28. Shuster V, Fishman A (2009) Isolation, cloning and characterization of a tyrosinase with improved activity in organic solvents from *Bacillus megaterium*. *J Mol Microbiol Biotechnol* 17(4):188–200
29. Drummond DA, Iverson BL, Georgiou G, Arnold FH (2005) Why high-error-rate random mutagenesis libraries are enriched in functional and improved proteins. *J Mol Biol* 350(4):806–816
30. Volles MJ, Lansbury PT (2005) A computer program for the estimation of protein and nucleic acid sequence diversity in random point mutagenesis libraries. *Nucleic Acids Res* 33(11):3667–3677
31. Verma R, Schwaneberg U, Roccato D (2012) MAP^{2.0}3D: a sequence/structure based server for protein engineering. *ACS Synth Biol* 1(4):139–150
32. Stemmer WP (1994) DNA shuffling by random fragmentation and reassembly: in vitro recombination for molecular evolution. *Proc Natl Acad Sci U S A* 91(22):10747–10751
33. Zhao H, Giver L, Shao Z, Affholter JA, Arnold FH (1998) Molecular evolution by staggered extension process (StEP) in vitro recombination. *Nat Biotechnol* 16(3):258–261
34. Moore JC, Jin H-M, Kuchner O, Arnold FH (1997) Strategies for the in vitro evolution of protein function: enzyme evolution by random recombination of improved sequences. *J Mol Biol* 272(3):336–347
35. Sun F (1999) Modeling DNA shuffling. *J Comput Biol* 6(1):77–90
36. Moore GL, Maranas CD, Gutshall KR, Brenchley JE (2000) Modeling and optimization of DNA recombination. *Comput Chem Eng* 24(2–7):693–699
37. Moore GL, Maranas CD, Lutz S, Benkovic SJ (2001) Predicting crossover generation in DNA shuffling. *Proc Natl Acad Sci U S A* 98(6):3226–3231
38. Hiraga K, Arnold FH (2003) General method for sequence-independent site-directed chimeraesis. *J Mol Biol* 330(2):287–296
39. Herman A, Tawfik DS (2007) Incorporating Synthetic Oligonucleotides via Gene Reassembly (ISOR): a versatile tool for generating targeted libraries. *Protein Eng Des Sel* 20(5):219–226
40. Wong TS, Tee KL, Hauer B, Schwaneberg U (2004) Sequence saturation mutagenesis (SeSaM): a novel method for directed evolution. *Nucleic Acids Res* 32(3):e26
41. Airaksinen A, Hovi T (1998) Modified base compositions at degenerate positions of a mutagenic oligonucleotide enhance randomness in site-saturation mutagenesis. *Nucleic Acids Res* 26(2):576–581
42. Vanhercke T, Ampe C, Tirry L, Denolf P (2005) Reducing mutational bias in random protein libraries. *Anal Biochem* 339(1):9–14
43. Denault M, Pelletier JN (2007) Protein library design and screening: working out the probabilities. In: Arndt KM, Müller KM (eds) *Protein engineering protocols*, vol 352, *Methods in molecular biology*. Humana Press Inc., Totowa, NJ
44. Fox RJ, Davis SC, Mundorff EC, Newman LM, Gavrilovic V, Ma SK, Chung LM, Ching C, Tam S, Muley S, Grate J, Gruber J, Whitman JC, Sheldon RA, Huisman GW (2007) Improving

- catalytic function by ProSAR-driven enzyme evolution. *Nat Biotechnol* 25(3):338–344
45. Nov Y, Wein LM (2005) Modeling and analysis of protein design under resource constraints. *J Comput Biol* 12(2):247–282
46. Barak Y, Nov Y, Ackerley DF, Matin A (2007) Enzyme improvement in the absence of structural knowledge: a novel statistical approach. *ISME J* 2(2):171–179
47. Brouk M, Nov Y, Fishman A (2010) Improving biocatalyst performance by integrating statistical methods into protein engineering. *Appl Environ Microbiol* 76(19):6397–6403