A computer program for the estimation of protein and nucleic acid sequence diversity in random point mutagenesis libraries

Michael J. Volles* and Peter T. Lansbury, Jr

Center for Neurologic Diseases, Brigham and Women’s Hospital and Department of Neurology, Harvard Medical School, 65 Landsdowne Street, Cambridge, MA 02139, USA

Received March 31, 2005; Revised May 11, 2005; Accepted June 6, 2005

ABSTRACT

A computer program for the generation and analysis of in silico random point mutagenesis libraries is described. The program operates by mutagenizing an input nucleic acid sequence according to mutation parameters specified by the user for each sequence position and type of point mutation. The program can mimic almost any type of random mutagenesis library, including those produced via error-prone PCR (ep-PCR), mutator Escherichia coli strains, chemical mutagenesis, and doped or random oligonucleotide synthesis. The program analyzes the generated nucleic acid sequences and/or the associated protein library to produce several estimates of library diversity (number of unique sequences, point mutations, and single point mutants) and the rate of saturation of these diversities during experimental screening or selection of clones. This information allows one to select the optimal screen size for a given mutagenesis library, necessary to efficiently obtain a certain coverage of the sequence-space. The program also reports the abundance of each specific protein mutation at each sequence position, which is useful as a measure of the level and type of mutation bias in the library. Alternatively, one can use the program to evaluate the relative merits of preexisting libraries, or to examine various hypothetical mutation schemes to determine the optimal method for creating a library that serves the screen/selection of interest. Simulated libraries of at least $10^9$ sequences are accessible by the numerical algorithm with currently available personal computers; an analytical algorithm is also available which can rapidly calculate a subset of the numerical statistics in libraries of arbitrarily large size. A multi-type double-strand stochastic model of ep-PCR is developed in an appendix to demonstrate the applicability of the algorithm to amplifying mutagenesis procedures. Estimators of DNA polymerase mutation-type-specific error rates are derived using the model. Analyses of an alpha-synuclein ep-PCR library and NNS synthetic oligonucleotide libraries are given as examples.

INTRODUCTION

Random point mutagenesis of a nucleic acid sequence is a useful technique for probing structure and function and for directed evolution of proteins, peptides and nucleic acids. Mutagenesis libraries can be created by several methods (1), including error-prone PCR (ep-PCR) (2–4), passage through mutator Escherichia coli strains (5), chemical mutagenesis (6) (e.g. sodium bisulfite, nitrous acid), and oligonucleotide synthesis (7) (NNN, NNS, or arbitrary doping; N = A, G, C, or T; S = C or G). A mutagenesis library is typically characterized by its size (number of independent clones) and by how heavily it is mutated. However, statistics such as library diversity (8) (e.g. number of unique sequences, point mutations, and single point mutants), amino acid mutation bias and the distribution of the number of mutations per sequence would in some cases be of more direct interest to the experimentalist, if they were available. The reverse issue of optimizing amino acid mutation bias in synthetic oligonucleotide encoded libraries has been explored previously (7,9–11).

Prior to screening, estimates of these parameters could be used to evaluate the relative merits of preexisting libraries. They can also be used to evaluate a variety of potential mutagenesis schemes and levels, in order to determine the most

*To whom correspondence should be addressed. Tel: +1 617 768 8617; Fax: +1 617 768 8606; Email: mvolles@rics.bwh.harvard.edu

© The Author 2005. Published by Oxford University Press. All rights reserved.

The online version of this article has been published under an open access model. Users are entitled to use, reproduce, disseminate, or display the open access version of this article for non-commercial purposes provided that: the original authorship is properly and fully attributed; the Journal and Oxford University Press are attributed as the original place of publication with the correct citation details given; if an article is subsequently reproduced or disseminated not in its entirety but only in part or as a derivative work this must be clearly indicated. For commercial re-use, please contact journals.permissions@oupjournals.org
appropriate type of library to construct, for the given objective. Subsequent to selecting or preparing the most appropriate library, one would ideally screen all of its available diversity. However, if the screening process is costly in terms of time or expense, knowledge of the amount of diversity covered as a function of the number of clones screened allows the investigator to determine the optimal screen size. This is the endpoint that allows sufficient and efficient exploration of a defined portion of the library diversity while avoiding inefficient oversampling. Finally, when the screen has been completed, one would like to know how much of sequence space was actually covered, and with what amino acid mutation bias.

A number of library statistics are amenable to an analytical treatment. For example, amino acid mutation frequencies, the fraction of sequences that are wild-type, the length distribution of truncated proteins, and some simpler diversity statistics (the number of unique sequence-position-specific point mutations and single point mutants; Supplementary Appendix H) can be calculated analytically. In contrast, the distribution of the number of mutations per sequence and the overall sequence diversity (number of unique sequences) are in general not analytically tractable.

Previous analytical work has shown that probability theory can be used to estimate the overall diversity of a nucleic acid library when the equations are simplified by requiring that all sequences are equiprobable, or that all mutations occur in the early 1990’s when computer power was much more limiting). To our knowledge, estimates of protein/nucleic acid library diversity cannot be practically obtained with any currently available methods, analytical or numerical. Limiting). To our knowledge, estimates of protein/nucleic acid library diversity cannot be practically obtained with any currently available methods, analytical or numerical.

**Materials and Methods**

**Construction of a library of randomly mutated α-synuclein cDNA molecules**

The ep-PCR method of Cadwell and Joyce was used (3,16). Template for the ep-PCR was generated by standard PCR (non-error generating; Pfu polymerase; forward primer: cgagctctcaatggtagttcgtgaaagac; reverse primer: cgagctctcaagcttggatggaacatctgtcagc) of the α-synuclein cDNA. The template extends from 12 bp upstream of the α-synuclein start codon through ~60 bp downstream of the stop codon, and was purified using agarose gel-electrophoresis. Approximately 30 ng (100 fmol) of template was used in a 100 μl ep-PCR [10 mM Tris, pH 8.3, 50 mM KCl, 0.01% gelatin, 0.2 mM each dATP and dGTP, 1 mM each dCTP and dTTP, 7 mM MgCl₂, 0.5 mM MnCl₂, 0.3 μM forward and reverse primers (same sequences as above), 5 U Taq polymerase]. Thirty reaction cycles [94°C 1 min, 66°C 1 min, 72°C 75 s; this number of cycles is probably excessive, given the maximum amplification under these conditions of 300-fold, but see (3) and Supplementary AppendixD] were performed followed by product purification (Qiagen PCR purification kit).

The insert was digested with NdeI and HindIII, the product was ligated (Takara ligation kit) into a digested and phosphatased pT7-7 E.coli expression vector (17). The ligated DNA was purified into a low-salt buffer for electroporation (Qiagen PCR purification kit). Ligated DNA (10 μl) was added to 30 μl of electrocompetent E.coli strain DH10B prepared by the Dower method (18) on ice, and two 20 μl electroporations were carried out at 4°C. The electroporator and cuvette were constructed in the laboratory, and provide equivalent efficiency to commercially available devices. The electroporator individually charges each of a set of 12 serially connected capacitors to several hundred volts using an electrophoresis power supply; this provides a total end-to-end voltage in the kilovolt range. The cuvette is Plexiglas with stainless steel electrodes (1.4 mm gap). An initial electric field of ~14 kv/cm with an exponential decay (time constant of 5 ms) was measured with an oscilloscope. Immediately following electroporation, the cells were diluted into 2 ml SOC medium and incubated with shaking for 1 h at 37°C. Aliquots were then plated on LB-ampicillin media to determine the number of transformants, and the remainder was grown in several hundred milliliters of LB-ampicillin media overnight for a mini-prep.

Two additional libraries with similar properties were generated during optimization of this mutagenesis procedure (dNTP, Mn²⁺ and Mg²⁺ concentrations were never varied). Total overall mutation frequencies of these three individual libraries were similar (range of 0.006–0.009 mutations per base pair). The specific overall DNA mutation frequencies from each of these three libraries also appear to be similar [significant differences between the libraries were only detected in the mutation pair T→A, A→T (χ² significance level <0.05)]. Therefore, sequence data from all three sub-libraries were combined.

**The computational algorithm**

**Inputs.** First, the initial DNA sequence is read in from a text file, as are a set of mutation parameters. In the case of ep-PCR,
these parameters are 12 probabilities \( p_{xy} \) which describe the frequency that the polymerase creates a mutation by misincorporating base \( y \) across from base \( x \). Note the distinction between these incorporation probabilities and the actual mutation which results (e.g. \( p_{AT} \) creates an A\(\rightarrow\)T mutation; mutation of base \( x \) to base \( y \) is denoted by \( x\rightarrow y \)). The estimation of these probabilities from DNA sequencing data is discussed in Appendix B. Also, the number of PCR cycles \( n \) and the PCR efficiency \( \lambda \) are specified (rough estimates of these parameters are sufficient, see Supplementary Appendix C.II). In non-amplifying mutagenesis methods such as oligonucleotide synthesis, the input mutagenic parameters are the direct mutation frequencies, for example, the frequency that wild-type base A is mutated to a G. These frequencies may be different at each position in the sequence and for each of the 12 mutation types (i.e. A\(\rightarrow\)G, A\(\rightarrow\)C, A\(\rightarrow\)T, etc.). In both cases, the number of sequences to generate is also specified, and the option is given to proceed with a numerical (see sections below) or analytical algorithm (Supplementary Appendix H). The analytical algorithm has the advantage of speed and can work with an arbitrarily large library, but cannot calculate the total library diversity, the distribution of the number of mutations per sequence, or the distribution of the number of times sequences in the library were repeatedly generated.

**Numerical production of a sequence.** The program generates a random number with the Mersenne Twister algorithm (19). This number is used to decide whether to accept one of the three possible mutations or to leave the base as wild-type. The decision to accept a mutation occurs with a specified probability, discussed in the next section. The program then repeats this procedure, using a new random number and the applicable acceptance frequencies, for the second base, the third base, and so on, to the end of the sequence. This single-pass mutagenesis is in contrast with the multi-pass, amplifying process of PCR. One object of Appendix A is to define a method by which the result of the latter process is simulated using the former.

**How the acceptance frequencies are determined from the inputs.** The method of determining the acceptance frequencies differs for non-amplifying methods (e.g. oligonucleotide synthesis, chemical mutagenesis) and amplifying methods (ep-PCR, *E.coli* mutator strain). For the case of non-amplifying methods, the acceptance frequencies are simply the inputs directly specified by the user, as described above. In the case of ep-PCR, the acceptance frequencies are determined for each new sequence as follows: First, the generation number of the sequence and the strand (top or bottom) of its zeroth generation ancestor are chosen randomly, according to their probability distributions (these terms and their probability distributions are defined in Appendix A). Inclusion of the zeroth generation (initial templates) in the probability distribution used to choose generation is optional. In some experimental procedures, these molecules are not incorporated into the library, for example, if they are lacking restriction sites (for subsequent cloning) introduced using the PCR primers. Having decided these two values, the appropriate mutation acceptance rates are derived by a series of matrix transformations on the input polymerase incorporation frequencies (see Appendix A for details). These frequencies are then held constant throughout production of a single sequence, because all bases of a sequence share the same generation and ancestor strand.

**Protein translation.** If protein diversity is being examined, the mutated DNA is translated. The program uses the standard genetic code by default, but by altering a single line of the program code one can specify an alternative codon translation, for example, as necessary with amber stop codon suppression or mitochondrial protein synthesis.

**Further iterations.** This process for mutating a wild-type sequence is then repeated exactly as above, the number of times (number of sequences) specified by the user. Each iteration begins anew with a wild-type sequence and is independent of any previous iteration.

**Library storage in memory.** Every mutated sequence is stored in a binary search tree (20) as it is generated. Each mutation requires two bytes of memory, the low bits (5 for protein, 2 for DNA) of which store the mutation type, while the remaining high bits are used to store the sequence position. This scheme limits protein sequences to \( \sim 2000 \) amino acids \([2^{16–5}]\), but this should be adequate for almost all cases. DNA alone can be examined to lengths of \( \sim 16 \) kb \([2^{16–2}]\). Sequences are unambiguously ordered in the tree using a scheme based on the number, position, and types of mutation they contain. The number of times each sequence has been generated is also recorded. Memory is allocated dynamically for all critical parts of the algorithm, so that the problem size accessible to the program is only limited by machine hardware. For simulations which have memory requirements beyond the physical memory of the computer, an efficient disk caching routine was written which significantly extends the upper size limit of practical simulations. Reliance on operating system disk paging for this purpose would have been unacceptably slow; the non-locality of reference of the binary tree would cause continuous seeking on the hard disk.

**Library analysis.** Statistics of the nucleic acid and/or protein libraries are collected during and following the sequence iterations. For diversity estimation, these are the number of unique sequences, mutations, and single point mutants as a function of the total number of sequences generated, the distribution of the number of mutations per sequence, the distribution of the number of times sequences in the library were repeatedly generated, and a listing of the most often generated sequences. Additional statistics include the specific mutation frequencies generated at each position, the total mutation frequencies summed over all positions, the number of extended sequences (protein stop to sense mutation), and the distribution and number of truncations. When truncated proteins are generated, the occurrence and sequence position is recorded, but the sequence is not used in other statistics. The user has the option of discarding or keeping sequences in which a stop codon has been mutated to a sense codon.

**Library size limitations.** With current personal computers, physical memory is the most likely factor to set the upper practical limit on numerical simulation size, which will typically be on the order of \( 10^9–10^{10} \) sequences for average problems. Although the scratch disk function extends the upper limit well beyond that which would be possible with physical
RESULTS AND DISCUSSION

Example 1: ep-PCR of the α-synuclein gene

An α-synuclein ep-PCR random mutagenesis library was created, and will be analyzed below in order to demonstrate the type of results which can be expected from the algorithm and the types of conclusions that can be drawn from them. The library contains $\sim 3.8 \times 10^6$ clones with full length, potentially expressible, inserts. A number of these were sequenced (Table 1), and the data together with the estimator of Appendix B were used to derive polymerase incorporation frequencies (Table 3).

Using these incorporation frequencies, we estimated the characteristics of the protein library (Table 2) by generating and analyzing $\sim 3.8 \times 10^6$ sequences with the computer algorithm. A larger library containing $10^9$ sequences was also generated with the program (not experimentally), for the purpose of comparison. The DNA diversity and statistics will not be discussed, since we are primarily interested in the protein translation of this library. The simulation of the library required 2–3 min and 100 MB of memory for $\sim 3.8 \times 10^6$ sequences (both protein and DNA; the size of our actual library) and 10 h with 1 GB of RAM and the disk cache function enabled for $10^9$ sequences (protein only), using a 1.8 GHz Pentium IV computer.

The standard deviations of the average values are quite low (Table 2); with simulations of this size or greater, random fluctuations in the output of the algorithm can be considered negligible, relative to errors from other sources. The width of the 99% confidence intervals on the estimated properties leads us to conclude that a modest amount of sequencing (e.g. Table 1) is sufficient for the algorithm to produce very useful estimates.

Figure 1A shows the number of unique sequences (black diamonds), unique point mutations (red squares), and unique single point mutants (green triangles) which were produced during the simulation, as a function of the number of
new point mutations (red curve), because multiple new mutations can occur in a sequence during the process of DNA replication. These curves are the derivatives of the theoretical maximum number of unique point mutations generated; a unique point mutant is defined as a sequence with only a single amino-acid mutation, occurring at a specified sequence position. The general term 'unique elements' is used on the y-axis and encompasses all three of the above terms. The theoretical maximum number of unique point mutations and unique single point mutants is 19 times the sequence length; both are referred to the right-hand, linear, y-axis). Proteins with truncations or extensions (mutated stop codon) are not included on either x-axis. This mimics an actual screen: significantly truncated or extended proteins are often not effective candidates, yet they must still be screened because they cannot be easily removed from the library. Points with abscissa values below 40,000 are averages of values from three independent simulations. Sampling without replacement is assumed (see Supplementary Appendix E). (B) The efficiency of sequence space coverage as a function of number of sequences screened. The efficiency is the expected number of unique sequences generated. The slopes of the lines in Figure 1A are shown with an identical color scheme in Figure 1B. As discussed in the introduction, this information is important for judging the appropriate screen size to use, and the efficiency of the screening procedure in examining sequence space. For example, if one were interested in looking at as many point mutations as possible, without regard for whether multiple point mutations existed per sequence, the red curves in Figure 1 would be appropriate. In the first several thousand clones examined, the efficiency ranges from >1.6 new mutations per clone, down to ~0.05, and a total of ~700 different mutations can be expected after 2500 clones. Screening for new point mutations beyond this first several thousand sequences is very inefficient; only infrequently will a new point mutation be looked at. In this system, there are 2527 possible single point mutations (133 mutatable amino acids times 19). Therefore, if the screening procedure is not trivial in terms of time and expense, and if one wished to look beyond the ~25% of those mutations which can be efficiently examined in this library, a different experimental approach to creating the library might be necessary. However, almost all of the possible point mutations would be accessible in a full screen of a library with the mutation frequencies shown in Table 3, and which contained 10^7 independent clones. This is relevant if the screen/selection size is not limiting (e.g. phage display). If one wishes to consider only unique single point mutants, the situation is similar (Figure 1, green), but somewhat higher numbers of clones must be screened relative to the unrestricted case (Figure 1, red) in order to get equal coverage. A similar analysis of the number of unique sequence variants (Figure 1, black), shows that a library of 10^7 clones can be mined for new variants with comparative efficiency throughout its entirety (at least one in five clones examined are new sequences). The relative frequencies of each type of protein mutation from each wild-type amino acid are shown in Figure 2. As expected, some mutations are much more probable than others. One reason for this is that many protein mutations require two or three DNA base changes in a single codon, which is an infrequent event in ep-PCR. Other contributing factors are likely to be mutations that are more likely to occur due to the polymerase incorporation frequencies, which is an infrequent event in ep-PCR. Other contributing factors are likely to be mutations that are more likely to occur due to the polymerase incorporation frequencies, which are shown in Table 2.

**Table 3.** Estimated polymerase paired incorporation frequencies for the α-synuclein ep-PCR DNA library

<table>
<thead>
<tr>
<th>Specific event</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{ac} )</td>
<td>0.00161</td>
</tr>
<tr>
<td>( p_{cg} )</td>
<td>0.00037</td>
</tr>
<tr>
<td>( p_{at} )</td>
<td>0.00121</td>
</tr>
<tr>
<td>( p_{ag} )</td>
<td>0.00026</td>
</tr>
<tr>
<td>( p_{ct} )</td>
<td>0.00005</td>
</tr>
<tr>
<td>( p_{ca} )</td>
<td>0.00043</td>
</tr>
</tbody>
</table>

*This set of six paired mutagenic incorporation frequencies was derived using the estimator of Appendix B with the sequencing data of Table 1, and \( n = 9 \) and \( \lambda = 0.88 \). The minimum value of \( \lambda \) in the estimator was taken as zero, because this library can potentially incorporate initial template sequences (see Appendix B). The values of \( n \) and \( \lambda \) were chosen as reasonable estimates that also coincide with the experimental amplification factor of ~300. The accuracy of this set of probabilities, as measures of the paired incorporation frequencies of a real polymerase, depend strongly on the accuracy of the estimated values of \( n \) and \( \lambda \). In contrast, the results of the simulation/analysis are relatively insensitive to these two parameters (Supplementary Appendix C.II). Statistical uncertainty in these values is reflected in the confidence intervals of Table 2.*
factors are that certain DNA mutations are much more frequent than others (Table 3) (3,13), and that amino acids have varying degrees of degeneracy with respect to the genetic code. This mutation bias underlies, in part, the limitations on screening efficiency shown in Figure 1. The bias is further underscored by the output of the program on the most frequently generated protein sequences, which are a pool of several hundred single point mutants. The most common, single point mutant F87L, exists more than $10^6$ times in a library of size $10^9$, and the top 100 sequences make up 7% of all $10^9$ generated sequences. The computer program also reports the number of occurrences of each mutation type at each position of the sequence (data not shown). This allows for the examination of region or codon specific bias.

Another way to examine the data in Figure 2 is by grouping amino acids with similar properties. Specifically, the $x$- and $y$-axes in Figure 2 are arranged in order of increasing hydropathy index (21). Consider Figure 2 divided into four quadrants: hydrophobic → hydrophobic, hydrophobic → hydrophilic, hydrophilic → hydrophobic, hydrophilic → hydrophilic. All four are reasonably well populated; by this measure, the bias is not nearly as great as in the above analysis. In certain cases, e.g. optimization of structural stability, this latter approach may be the most appropriate bias indicator. In others, such as optimization of a catalytic site, the former may be more relevant.

**Example 2: synthetic NNS oligonucleotides**

Simulations were performed of synthetic NNS oligonucleotides of various lengths, in multiples of three, and after

![Figure 3. Peptide diversity and screening efficiency of NNS (S = G or C) oligonucleotide libraries. (A) Number of unique peptide sequences generated (logarithmically spaced $y$-axis) as a function of the number of NNS DNA sequences produced (log-spaced $x$-axis). Sequences which contained a stop codon were discarded during the simulation. Results with varying numbers of NNS triplets are labeled with amino acid length. Data for all 1 through 10mer peptides are shown, but 8–10, which have a much higher diversity than the maximum number of simulated sequences, are not separately discernible. Data points below 1000 represent average values from three separate runs, to reduce the noise inherent in very small simulations. Sampling without replacement is assumed (see Supplementary Appendix E). (B) Efficiency ($y$-axis; see Figure 1 legend for description) as a function of number of sequences screened (log-spaced $x$-axis). Colors are used to distinguish some of the overlapping curves (6mer, light green; 7mer, dark green; 8mer, blue; 9mer, pink; 10mer, red). The colors in (A) follow an identical scheme. The abrupt decrease in noise level at 100 000 in (B) is due to an increase in the $D_x$ used to calculate slope from the points in (A), and is not inherent in the data. Note the decrease in initial efficiency [visible in (B), especially in the less noisy, colored, curves], as the size of the peptide increases: longer sequences have a greater chance of being eliminated due to incorporation of a nonsense codon.]

![Figure 2. Mutation frequencies and bias as a function of wild-type amino acid in an $\alpha$-synuclein ep-PCR library. A grey-scale is used to show the number of times (labeled every 10 000 units) a particular wild-type amino acid ($x$-axis) was observed to be changed to a particular mutated amino acid ($y$-axis) after numerical simulation of 3 770 580 sequences (see Table 2 legend for details), normalized (divided) by the number of times the wild-type amino acid appears in the sequence. Sense mutations are recorded only in untruncated, unextended sequences. Positions in the graph corresponding to wild-type (no mutation) have values of zero. Both the $y$- and $x$-axes are arranged in order of increasing hydropathy index (21). Mutation to a stop codon is represented by an asterisk. $\alpha$-Synuclein does not contain cysteine, arginine, or tryptophan, therefore, these amino acids do not appear on the $x$-axis. The graph was created using HeatMap Builder (http://mozart.stanford.edu/heatmap.htm).]
translation the peptide libraries were examined. The wild-type DNA sequence was specified as the appropriately sized poly-A, with 25% mutation frequencies to each of the three other bases (for N), or a 50% chance each of mutating to G or C (for S). Protein diversity, in terms of number of unique sequences, as a function of library size, is shown in Figure 3A. We simulated the lesser of 10 times the theoretical number of unique sequences or $10^9$ sequences, per peptide length; the latter typically required 8 h on the machine described above. These statistics are useful for estimating the number of peptides which need to be screened in order to approach a certain coverage of the available diversity. The slopes of the curves in Figure 3A are shown in Figure 3B, describing the efficiency of continued screening, as a function of number of clones screened. Note that because the NNS library is well-distributed, with relatively little mutation bias, the curves of Figure 3A are regular and evenly spaced (compare Figure 1A; this suggests that an analytic fit of this empirical data may be usefully extrapolated in order to solve larger, otherwise inaccessible problems). Still, in most cases the point at which efficiency drops off severely (Figure 3B) would not be entirely predictable without some sort of diversity sampling calculation. On the other hand, with a very large unbiased library, such as the NNS 10mer library, diversity predictions are superfluous; we know that essentially every additional clone examined will be unique.

SUPPLEMENTARY MATERIAL
Supplementary Material is available at NAR Online.

ACKNOWLEDGEMENTS
We thank Alina M. Vrabioiu and Allegra A. Petti for critical reading of the manuscript and helpful discussions. This work was supported by a Morris K. Udall Parkinson’s Disease Research Center of Excellence Grant (NS38375). M.J.V. is supported by a postdoctoral fellowship from the American Parkinson Disease Association. Funding to pay the Open Access publication charges for this article was provided by NS38375.

Conflict of interest statement. None declared.

REFERENCES

APPENDICES
Appendix A: a multi-type double-strand mathematical model of independent sequences from an ep-PCR
The algorithm described here stochastically mutates the bases of a wild type-sequence, stores the finished sequence, and
then repeats the process, beginning again with a new wild type
sequence. No replication of previously mutated sequences is
involved. In contrast, ep-PCR entails multiple rounds of
mutagenesis with amplification. We here define the
method by which our single-pass algorithm accurately models
experimental ep-PCR libraries. The treatment should also
apply to E.coli mutator strain mutagenesis, whose mechan-
ism similarly involves uncorrected errors of replication, with
amplification (22).

The type of model. Previous analytical work (23–28) has
modeled the mutagenic PCR process using a generic single
DNA base type, a single mutation type, and a single DNA
strand. An additional report considered a multi-type PCR
model, but did not address the issue of DNA double-
strandedness (29). We, therefore, developed for our purposes a
mathematical model of ep-PCR that is multi-type and double-
stranded; it considers the distinctness of the 4 nucleotides, and
explicitly treats the double-stranded nature of DNA. We assume
that replication of a strand occurs with constant probabil-
ity (efficiency) λ. Note that this number expresses the prob-
ability that a replication is completed, the alternative being
that a replication is not begun; partially completed replications
are not considered, nor are insertion and deletion mutations.
The appropriateness and goodness-of-fit to the data of this type
of model is considered in Supplementary Appendix C.III.

DNA polymerase in the model. The polymerase activity is
parameterized by 16 base incorporation probabilities (e.g. the
probability the polymerase will install a T across from a G,
causing a G→A mutation) which are conveniently expressed
as \( p_{ij} \) = the probability that the polymerase pairs base \( j \) with
template base \( i \). These probabilities are one of the inputs spe-
cified by the user, and are considered constant throughout
the PCR. They are also assumed to be sequence-context inde-
dent [for experimental justification, see (2,13); estimators
for these parameters are derived in Appendix B]. In theory one
could account for sequence context dependence, but this will
not be considered here. It is currently impractical to gather
enough sequencing data to routinely determine position
dependent mutation frequencies. Since in our model a poly-
merase always pairs a base \( i \) with exactly one other base,

\[
\sum_{j=1}^{4} p_{ij} = 1
\]

Twelve of the sixteen possible incorporations are muta-
genic, and the other four are the standard Watson–Crick
base pairs. Using the above equation, we are free to set each
Watson–Crick base pair equal to 1 minus the sum of the other
three mutagenic pairings; there are 12 degrees of freedom.

We now introduce a \( 4 \times 4 \) random matrix \( N \), which may
intuitively be thought of as representing the stochastic oper-
ation of a polymerase incorporating a single base. The members
of the set of matrices which \( N \) can realize (the individual
members of this set we designate \( n_k \), where \( x \) ranges from
1 to \( 4^4 \) each have the following three properties: (i) Every
row contains a single element 1, and three 0 elements. A 1 at
some position \( ij \) of a particular realization of \( N \) indicates that
given a base \( i \) as template, the polymerase will copy a base
\( j \) across from it. (ii) The probability that element \( N_{ij} \) is a 1 is
given by the probability \( p_{ij} \). (iii) Each row is completely inde-
dependent 

DNA sequences in the model. We can specify the sequence
of a single-stranded DNA molecule of length \( L \) bases using an
\( L \times 4 \) matrix \( Z \). The four elements in any row of \( Z \) consist
of three zeros and a single 1. The coordinate that contains
the 1 defines the base type at that position according to the
order (A,G,C,T; the same arbitrary order must also be used
in the rows and columns of matrix \( N \)). For example, a base
G is represented by \([0,1,0,0]\).

The notation \( Z_{kr} \) will be used to refer specifically to a
sequence which is a generation \( k \) PCR product whose initial
strand ancestor in the PCR was a ‘top’ strand \( (r = 0) \), or ‘bot-
tom’ strand \( (r = 1) \). Generation number is defined as the num-er of polymerase copying events separating the molecule from
a zeroth generation (initial template) sequence. Of course, all
bases of a single strand of DNA are the same generation \( k \)
and have the same value of \( r \).

For convenience, we will deal exclusively with \( Z \) matrices
expressed in terms of an arbitrarily designated ‘top’ strand
composition (as opposed to its bottom-strand complement); we
transform bottom-strand composition to top by right
operating with the exchange matrix \( T \):

\[
T = \begin{bmatrix}
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix}
\]

Note that for \( x \geq 0 \):

\[
T^x = \begin{cases} T & x \text{ odd} \\
I & x \text{ even}
\end{cases}
\]

where \( I \) is the identity matrix.

A PCR typically begins not only with top strand templates
(designated \( Z_{00} \)), but also with their bottom strand com-
plements (\( Z_{00}T \)). \( T \) is also applied to expressions as necessary,
in order that final product sequences are always referenced
to top strand composition. This matches the reality of
ep-PCR: reaction products are cloned in E.coli prior to DNA
sequencing, thereby converting (we assume without error) all
bottom strand PCR products to top plus bottom strand (a related
issue is discussed in Supplementary Appendix D). One can
think of matrix \( T \) as the particular realization of \( N \) that produces
mutation-free copies.

First generation products. We operate with \( N \) on the right to
transform a zeroth generation top strand template sequence
(\( Z_{00} \)) into a first generation, random, bottom strand copy (\( Z_{10} \)):

\[
Z_{10} = (Z_{00}N)T.
\]

As discussed above, \( T \) is used in order that the first generation
product, random matrix \( Z_{10} \) (which exists as a bottom strand),
is given in terms of top strand composition.
Similarly, beginning with a bottom strand template, we generate a top strand:
\[ Z_{11} = (Z_{00} T) N. \]

**Higher generation products.** To generate a random second generation sequence, we operate with \( N \) on a first generation sequence. This process may be repeated indefinitely to generate higher generation sequences. In general, random matrix \( Z_{1r} \), which is a generation \( k (=1) \) sequence with \( r \)-strand zeroth generation ancestor, will result from operating on the right of \( Z_{00} T \) with \( k \) independent random matrices \( N_j \):
\[
Z_{kr} = Z_{00} T^r \left( \prod_{j=1}^{k} N_j \right) T^{k+r}.
\]

The final \( T^{k+r} \) in the equation converts sequences to top strand composition, where necessary. A matrix to the zeroth power is to be interpreted as the identity matrix. Note that the possible realizations of the product of \( k \) independent random \( N \) matrices is the set of \( n_s \). The subscript \( i \) is used to emphasize that each of the matrices \( N \) (total of \( k \)) are independent and identically distributed; each row of \( Z_{1r} \) is also generated with a completely independent set of \( N_j \). In fact, every \( N \) of every equation in this article is independent and is constant for only a single base; a polymerase has no history in our model. Consequently, only a single row from any given \( \Pi(N_l) \) is ever used when generating sequences, and in practice it is sufficient to generate this random vector alone.

**Overall distribution of ep-PCR products.** We now consider the total products of a complete ep-PCR, and how one should computationally model an ep-PCR library. We can use the above equation to simulate a random sequence chosen from the library, once we have specified a realization of \( K \) and \( R \) according to their probability distributions. Sun (24) derived the probability distribution of \( K \) after \( n \) PCR cycles, assuming a large number of initial templates relative to the numerical value of the PCR amplification factor squared. This is always satisfied in typical ep-PCR, though the assumption is not stringent (24,28).

\[
P(K = k) = \binom{n}{k} \lambda^k \left( \frac{1}{1 + \lambda} \right)^n.
\]

The sequence will also have originated from either a top strand or a bottom strand, described by random variable \( R = 0 \) or \( 1 \); the probability distribution of \( R \) will be divided equally between the strands.

All the bases of a single strand share the same generation (value of \( K = k \)) and initial template strand (value of \( R = r \)), but are mutated (the \( N_j \)) independently. Mathematically, the random variables \( R \) and \( K \) are chosen once, and kept constant throughout a single sequence.

When computationally producing a base in a sequence, as stated above, we only need to generate a realization of the single relevant row vector of \( \Pi(N_l) \). Since the vector will contain three zeros and a 1, its probability distribution is completely defined if one knows the probability of a one being in each of the four positions. Given that \( K = k \),
\[
P \left( \prod_{j=1}^{k} N_j \right)_{ij} = 1 = \left( \sum_{i=1}^{k} \prod_{j=1}^{k} N_j \right)_{ij}.
\]

\[
E \left( \prod_{j=1}^{k} N_j \right) = \sum_{i=1}^{k} \left( \prod_{j=1}^{k} n_i P(N = n_i) \right) = E(N)^k.
\]

Note that:
\[
E(N) = \left[ \begin{array}{cccc}
p_{aa} & p_{at} & p_{ac} & 1 - (p_{aa} + p_{at} + p_{ac}) \\
p_{ta} & p_{tt} & p_{tc} & 1 - (p_{ta} + p_{tt} + p_{tc}) \\
p_{ca} & p_{ct} & p_{cc} & 1 - (p_{ca} + p_{ct} + p_{cc}) \\
p_{na} & p_{nt} & p_{nc} & p_{nn} \end{array} \right].
\]

We have defined how to computationally model one sequence from the total PCR products, by specifying the appropriate mutation acceptance frequencies for a single-pass Monte Carlo mutagenesis. In general, subsequent sequences from a library cannot be considered independent from the first, since they may share a common ancestor. However, the number of initial templates will almost always be significantly greater than the number of individuals screened or selected against. For example, in the \( \alpha \)-synuclein ep-PCR, \( 6 \times 10^{10} \) initial template molecules were used, and the final library contained \(<6 \times 10^6 \) clones (the bottleneck being ligation and transformation). Under these conditions, all sampled sequences can be considered to be independent, and the equations above can be used repeatedly in order to simulate all of the sequences in the library. For this discussion to hold in mutator strain mutagenesis, an appropriate bottleneck would need to be artificially imposed. Alternatively, a brute force realistic simulation of the entire branching process may be feasible, but only if the number of initial templates were very low, and the number of generations not too large [also note that some work with finite population models has been reported (27,30)].

**Appendix B: an estimator of the DNA polymerase incorporation frequencies.**

Our goal is to computationally produce a set of PCR products with a distribution of mutated sequences as near as possible to those in an experimental PCR library of interest. However, our only insight into that experimental library is limited DNA sequencing information, estimates of \( n, \lambda \), and library size, and the model presented above. To be able to generate sequences computationally, using the method of Appendix A, we must first derive an estimate of \( E(N)_{\text{real}} \), the true state of nature. We use the method of moments to derive an estimator.

The expectation of a sequence drawn at random from the library is given by:
\[
E(Z) = \sum_{K=0}^{\infty} \sum_{R=0}^{\infty} \left( \sum_{\text{All possible sequences } i} \left( \sum_{K=0}^{\infty} \sum_{R=0}^{\infty} \left( \sum_{\text{All possible sequences } j} Z_i Z_j P(Z_i, Z_j) \right) P(K=k) P(R=r) \right) \right) P(K=k) P(R=r).
\]
E(Z) = \sum_{k=0}^{n} \sum_{r=0}^{1} (E(Z|K=k, R=r)P(K=k)P(R=r)),

E(Z) = \sum_{k=0}^{n} \left( Z_{00}^{T} \left( \frac{E(N)^{k}}{2} T^{k} + TE(N)^{k} T^{k}\right) \right) \left( \frac{n}{k} \right) \frac{\lambda^{k}}{(1+\lambda)^{n}}.

Note that E(Z), unlike Z, may have non-zero elements in every position, corresponding to the expected level of that mutation type at that position. Additionally, complementary mutations (e.g. A→C and T→G) are expected to occur in the library with equal probability. This is intuitively clear from the double-stranded complementary nature of DNA, and is embodied in the equations by the centrosymmetry of E(N)^k T^k + TE(N)^k T^k.

In our model, all bases of a single type in a single sequence are equivalent. We may, therefore, contract all L × 4 matrices to 4 × 4 matrices, by setting each row in the 4 × 4 matrix (there is one for each, A,G,C,T) equal to the sum of all rows in the L × 4 matrix that originate with that wild-type base. Mathematically, this is equivalent to multiplying the matrix on the left with the transpose of matrix Z_{00}:

E(Z^{T}_{00} Z) = \sum_{k=0}^{n} \left( Z_{00}^{T} Z_{00} \right) \left( \frac{E(N)^{k}}{2} T^{k} + TE(N)^{k} T^{k}\right) \left( \frac{n}{k} \right) \frac{\lambda^{k}}{(1+\lambda)^{n}}.

Our experimental estimate of this quantity, based on DNA sequencing of s sequences, will be a matrix with element ij equal to:

\frac{(b+d)}{s} \frac{f}{(f+h)}

where b = number of observed i → j changes in s sequences
d = number of observed i → f changes in s sequences
f = number of i bases in one wild type sequence
h = number of f bases in one wild type sequence

Numbers of bases in the above are to be taken from a single arbitrarily chosen DNA strand. A barred variable indicates Watson–Crick complement, A = T, G = C, C = G, T = A. The term b + d divided by s gives our estimate of the average number of mutations of this pair type per sequence and the second term takes the appropriate fraction of this value, based on wild-type sequence composition. This formula also removes any differences in frequency between the two complementary members of a mutation pair, since these are due solely to random sampling, as discussed above.

To formulate the estimate in matrix terms, we begin with matrix X of the sequencing data. Specifically, from s = 89 sequences of the α-synuclein library (Table 1):

X = \begin{bmatrix}
11233 & 64 & 19 & 73 \\
20 & 11979 & 2 & 12 \\
8 & 2 & 6296 & 13 \\
44 & 8 & 24 & 5975
\end{bmatrix}.

Then the experimental estimate equals,

\frac{1}{s} \frac{1}{Z_{00}^{T} Z_{00}} \left( \frac{1}{Z_{00}^{T} Z_{00}} + T Z_{00}^{T} Z_{00} T \right)^{-1} (X + TXT).

Setting the experimental estimate equal to the expected value given above, and canceling terms:

\frac{1}{s} \left( Z_{00}^{T} Z_{00} + T Z_{00}^{T} Z_{00} T \right)^{-1} (X + TXT)

= \sum_{k=0}^{n} \left( \frac{E(N)^{k} T^{k} + TE(N)^{k} T^{k}}{2} \right) \left( \frac{n}{k} \right) \frac{\lambda^{k}}{(1+\lambda)^{n}}.

There is not a unique solution to this equation. We choose to solve for an E(N) that is centrosymmetric, designated E(N)_{cen}. This reduction in degrees of freedom is motivated by the analysis in Supplementary Appendix F, and allows us to proceed algebraically towards a unique solution. The resulting estimator is statistically analyzed in Supplementary Appendix G. The equation above simplifies to,

\frac{1}{s} \left( Z_{00}^{T} Z_{00} + T Z_{00}^{T} Z_{00} T \right)^{-1} (X + TXT)

= \sum_{k=0}^{n} \left( E(N)^{k} T^{k} \right) \left( \frac{n}{k} \right) \frac{\lambda^{k}}{(1+\lambda)^{n}}.

Furthermore, E(N)_{cen} = TE(N)_{cen} T, and therefore E(N)_{cen} T^{k} = [E(N)_{cen} T]^{k}.

Assume that the left hand side of the above equation is real-diagonalizable (to matrix D_{s}) by a matrix C. We multiply both sides on the left by C^{-1}, and both sides on the right by C.

D_{s} = C^{-1} \left( \frac{1}{s} \left( Z_{00}^{T} Z_{00} + T Z_{00}^{T} Z_{00} T \right)^{-1} (X + TXT) \right) C

= \sum_{k=0}^{n} \left( C^{-1} (E(N)_{cen} T)^{k} C \right) \left( \frac{n}{k} \right) \frac{\lambda^{k}}{(1+\lambda)^{n}}.

A matrix shares identical eigenvectors with all of its powers. Following from this, the overall eigenvectors of a power series of a matrix must equal the eigenvectors of its terms. Therefore, the matrix C must diagonalize E(N)_{cen} T to matrix D_{s} = C^{-1} E(N)_{cen} T C, and,

D_{s} = \sum_{k=0}^{n} \left( D_{s}^{k} \right) \left( \frac{n}{k} \right) \frac{\lambda^{k}}{(1+\lambda)^{n}}.

Each diagonal entry of D_{s} can be found independently by solving an nth-degree polynomial in the field of real numbers. There may be multiple real solutions for each entry in D_{s}. In our data, real solutions existed for each entry at ~1 and ~3. To choose the appropriate values, we note that E(N)_{cen} T and D_{s} are similar matrices, and therefore share identical eigenvalues. Because E(N)_{cen} T will always be nearly the identity matrix in ep-PCR, its eigenvalues will all be near unity and, therefore, the appropriate solutions are close to 1.
Finally, the unique physically sensible solution for matrix \( E(N)_{cen} \) can be found using the relationship,

\[
E(N)_{cen} = CD_q C^{-1} T.
\]

The quality of \( E(N)_{cen} \) so derived, in terms of the accuracy and precision of the resulting estimated library properties, is analyzed in Supplementary Appendix G and summarized in the confidence intervals (see Supplementary Appendix C) of Table 2. A Mathematica notebook is available from the authors as a template for computation of the matrix estimator. As a technical point, if any of the elements of \( X \) is zero, the solution \( E(N)_{cen} \) may contain very small negative values. This reflects the discreteness and limited size of our actual observations. Such negative values should be changed to the most acceptable substitute, zero, before proceeding. When the experimental library excludes the template sequences (see above), the summation in this set of equations should begin at \( k = 1 \), and the right hand side should be divided by \( 1 - P(K = 0) \), to normalize the summation of the probabilities \( P(K = k) \) to a total of one.

Note the close approximation between the preliminary incorporation frequencies obtained with the Sun estimator under hypothesis II (Supplementary Appendix F) and those in \( E(N)_{cen} \) (Table 3). The reason that these initial estimates are so good is that the mutation rate during ep-PCR is low enough that multiple mutations at a single site are very rare. We use the terms ‘low’ and ‘high’ in this respect, throughout this article. It has previously been implied that the error rate of ep-PCR is quite high in this respect [‘\( \sim 0.7\% \)’ per base per PCR cycle (26), citing Cadwell and Joyce (2)]. However, the rate in the Cadwell and Joyce paper is over the entire reaction, and is not given per cycle: ‘we used this method to mutagenize the gene... with a mutation rate of \( 0.66\% \pm 0.13\% \) (95% C.I.) per position per PCR’ (2).