Shuffled: a software suite that assists the analysis of recombinant products resulting from DNA shuffling

Enrique Morett and Alejandro G. Garciarrubio

Universidad Nacional Autónoma de México, Cuernavaca, Morelos, Mexico

BioTechniques 37:354-358 (September 2004)

In vitro recombination of gene variants is a powerful and efficient strategy to generate new useful variability. Unlike random mutagenesis, which is mainly detrimental, recombination can bring together adaptations present in the parent genes. This strategy has been implemented under different protocols (broadly known as DNA shuffling methods) and used for a variety of purposes, such as improving enzymes or giving them novel activities (1–11). Usually, after the random recombination of two (or more) parent sequences, some offspring variants are chosen for analysis (perhaps after a functional selection step). Because recombination is generally a random process, the question arises of which parts of an offspring variant are derived from which parent. Furthermore, as a result of being based on PCR, these methods introduce *de novo* mutations (something that is frequently desirable), so it becomes important to know where the new mutations lie. In one strategy particularly suitable for high-throughput experiments, both these questions are answered by the computational analysis of microarray hybridization patterns (12).

In another common strategy, the offspring variants are sequenced, and the resulting sequences are aligned with the parent variants, using a multiple sequence alignment program such as Clustal or PileUp. With a proper multiple alignment, one can look for regions that are more similar to one or another parent and for sites where the offspring differ from both parents. If a region is found where the offspring variant is much similar to one parent,

but not to any other, one can be sure that the former parent was the donor of that region. If a region is found further downstream where the similarity is closer to another parent, this suggests that a recombination took place somewhere between the first and second region. Single differences between a parent and an offspring can arise for two reasons: either that parent is not the donor of that region or the recombination methodology introduced mutations that made them different. Deciding between these possibilities is simple if at that position all parents differ from the offspring sequence. In that case, the most parsimonious explanation is that a mutation has occurred. However, if at that position some parent has the same nucleotide, but its local similarity to the offspring sequence is limited to a relatively small region, a dilemma arises: did a recombination make that parent the donor of that small region or is a mutation a better explanation? The decision has to be made by considering the probabilities of recombination and mutation events (13).

In our experience, if done by hand, this analysis can be tedious and prone to error. Thus, we created a computer program, *shuffled.pl*, which automatically finds the optimal, most parsimonious path for the recombination of the parent sequences to explain the offspring sequence produced. Here “optimality” is defined by the costs associated with recombination and two types of mutation events, nucleotide substitutions and indel openings.

To find the optimal recombination paths, *shuffled.pl* analyzes a multiple

alignment containing the parents and offspring sequences. In Shuffled terminology, a path is a hypothetical series of recombination and mutation events that explain how an offspring sequence was derived from the parent sequences. An optimal path is that which minimizes the accumulated cost of mutations and recombinations. If there are 6 parents and the multiple alignment of those parents with the offspring variant is 100 symbols long, there are 6 to the power of 100 possible paths. Instead of scoring all possible paths, *shuffled.pl* uses a dynamic programming strategy that is certain to find, with much less effort, at least one optimal path. *Shuffled.pl* does not find all optimal paths. For instance, among a series of equally correct recombination points, *shuffled.pl* chooses the position closest to the 5' end.

The program does not create multiple alignments itself, but can call *clustalw* to make them. You can provide your premade alignment, but multiple alignments containing parents plus several offspring sequences are rarely satisfactory because aligning the offspring to one another introduces unnecessary gaps. Instead, you can provide an alignment with the parents, and, in a separate file, the unaligned sequences of the offspring. This is preferable because a proper parent alignment (verified by human inspection) is a good starting point for producing good parents plus one offspring alignment.

To change the default values, the cost of recombination, substitution, and indel openings can be provided through a configuration file. These values should relate inversely to the expected probabilities of these events. These depend on how the experiment was performed; that is, the protocol, the polymerases, and the number and similarity of the parent sequences. As a rule of thumb, if there are many parents and they are very dissimilar, the cost of recombination should be increased in comparison to the costs of mutations. If, on the other hand, the experiment used mutation-prone conditions, the reverse should be applied.

A Real Life Example

MonoTIM is a monomeric vari-

ant of the *Trypanosoma brucei* triose phosphate isomerase (14). Here variants of MonoTIM generated by error-prone PCR were selected for their ability to catalyze a new biochemical reaction (on a vitamin synthesis pathway). In order to achieve higher enzymatic activity, MonoTIM and 12 selected variants were used as parents in an in vitro recombination experiment using StEP (2). The offspring variants with the most prominent phenotype were sequenced. The 24 variants remaining after eliminating identical clones were analyzed with Shuffled. For instance, shuffled.pl found 2 optimal paths to explain how variant m7-step could have arisen by recombination. Both paths were almost identical but differed in who was the donor of the first 142 nucleotides at the 5' end. In one path, the donor is parent Par3-6890, and in the other, it is parent Par5-9857. However, both parents are identical in that region. Figure 1 shows the summary lines for these paths, extracted from the output of shuffled.pl (the anota file). These lines tell that both paths include 3 recombination points (sw:3), 14 nonidentical substitutions (st:14), one insertion (in:1), 4 deletions (de:4), and 4 donors parents (dn:4). The anota file provides plenty more information about the paths.

Because anota files are plain text, they can be read by humans and machines. They are convenient for automatic evaluation and post-processing. Two associated programs, anota2rtf.pl and anota2html.pl, put anota files into visually enhanced forms. Anota2rtf.pl creates a colored rich-text format document that can be read and edited with most word processors (such as Microsoft® Word). The colors used, page size, and alignment numbering (on/off) can be configured. Figure 2 shows the first page of such document. Anota2html.pl creates a graphic, clickable html-based viewer that can be used with a Javascript-enabled web browser, such as Microsoft Internet Explorer and Netscape (Figure 3). The Shuffled package also includes a Common Gateway Interface (CGI) front end interface that allows the user to run all programs from a web browser. Although designed under Linux, the complete package is written in Perl, so it can be used in

most platforms. The package and documentation are available from shuffled's home page (http://www.ibt.unam.mx/shuf/shuffled_home.html), where the functionality of these programs can also be tried.

Our approach makes no assumption about the recombination protocol but that the parents are homologous and that parents and offspring can be aligned. This makes it very general, and it would probably even be useful to analyze in vivo and natural recombination and recombination by methods yet to be developed.

ACKNOWLEDGMENTS

The authors thank Leticia Olvera for providing experimental data, and Shirley Ainsworth, Juan Manuel Hurtado, Alma Martinez, and Nelly Mellado for technical assistance.

COMPETING INTERESTS STATEMENT

The authors declare that they have no competing interests.

REFERENCES

1. Stemmer, W.P. 1994. Rapid evolution of a protein *in vitro* by DNA shuffling. *Nature* 370:389-391.
2. Zhao, H., L. Giver, Z. Shao, J.A. Affholter, and F.H. Arnold. 1998. Molecular evolution by staggered extension process (StEP) *in vitro* recombination. *Nat. Biotechnol.* 16:258-261.
3. Cramer, A., S.A. Raillard, E. Bermudez, and W.P. Stemmer. 1998. DNA shuffling of a family of genes from diverse species accelerates directed evolution. *Nature* 391:288-291.
4. Shao, Z., H. Zhao, L. Giver, and F.H. Arnold. 1998. Random-priming *in vitro* recombination: an effective tool for directed evolution. *Nucleic Acids Res.* 26:681-683.
5. Kikuchi, M., K. Ohnishi, and S. Harayama. 2000. An effective family shuffling method using single-stranded DNA. *Gene*. 243:133-137.
6. Stoop, A.A., L. Jespers, I. Lasters, E. Eldering, and H. Pannkoek. 2000. High-density mutagenesis by combined D.N. shuffling and phage display to assign essential amino acid residues in protein-protein interactions: application to study structure-function of plasminogen activation inhibitor 1 (P.A.-I). *J. Mol. Biol.* 301:1135-1147.
7. Lutz, S., M. Ostermeier, G.L. Moore, C.D. Maranas, and S.J. Benkovic. 2001. Creating multiple-crossover DNA libraries independent of sequence identity. *Proc. Natl. Acad. Sci. USA* 98:11248-11253.
8. Coco, W.M., W.E. Levinson, M.J. Crist, H.J. Hektor, A. Darzins, P.T. Pienkos, C.H. Squires, and D.J. Monticello. 2001. DNA shuffling method for generating highly recombined genes and evolved enzymes. *Nat. Biotechnol.* 19:354-359.
9. Suenaga, H., M. Mitsuoka, Y. Ura, T. Watanabe, and K. Furukawa. 2001. Directed evolution of biphenyl dioxygenase: emergence of enhanced degradation capacity for benzene, toluene, and alkylbenzenes. *J. Bacteriol.* 183:5441-5444.
10. Osuna, J., A. Perez-Blancas, and X. Söberon. 2002. Improving a circularly permuted T.E.M.-1 beta-lactamase by directed evolution. *Protein Eng.* 15:463-470.
11. Hiraga, K., and F.H. Arnold. 2003. General method for sequence-independent site-directed chimeragenesis. *J. Mol. Biol.* 330:287-296.
12. Abecassis, V., L. Jaffredo, D. Rickman, L. Aggerbeck, C. Herbert, G. Truan, and D. Pompon. 2003. Microarray-based method for combinatorial library sequence mapping and characterization. *BioTechniques* 34:1272-1279.
13. Moore, G.L. and C.D. Maranas. 2000. Modeling DNA mutation and recombination for directed evolution experiments. *J. Theor. Biol.* 205:483-503.
14. Borchert, T.V., R. Abagyan, R. Jaenicke, and R.K. Wierenga. 1994. Design, creation, and characterization of a stable, monomeric triosephosphate isomerase. *Proc. Natl. Acad. Sci. USA* 91:1515-1518.

Received 16 October 2003; accepted 17 May 2004.

Address correspondence to Alejandro Garcíarrubio, Instituto de Biotecnología, UNAM, Av Universidad #2001, Col. Chamilpa, Cuernavaca, Morelos, Mexico 62210. e-mail: alejandro@ibt.unam.mx