indel-Seq-Gen: A New Protein Family Simulator Incorporating Domains, Motifs, and Indels

Cory L. Strope,* Stephen D. Scott,* and Etsuko N. Moriyama†

*Department of Computer Science and Engineering, University of Nebraska–Lincoln; and †School of Biological Sciences and Plant Science Initiative, University of Nebraska–Lincoln

Reconstructing the evolutionary history of protein sequences will provide a better understanding of divergence mechanisms of protein superfamilies and their functions. Long-term protein evolution often includes dynamic changes such as insertion, deletion, and domain shuffling. Such dynamic changes make reconstructing protein sequence evolution difficult and affect the accuracy of molecular evolutionary methods, such as multiple alignments and phylogenetic methods. Unfortunately, currently available simulation methods are not sufficiently flexible and do not allow biologically realistic dynamic protein sequence evolution. We introduce a new method, indel-Seq-Gen (iSG), that can simulate realistic evolutionary processes of protein sequences with insertions and deletions (indels). Unlike other simulation methods, iSG allows the user to simulate multiple subsequeces according to different evolutionary parameters, which is necessary for generating realistic protein families with multiple domains. iSG tracks all evolutionary events including indels and outputs the “true” multiple alignment of the simulated sequences. iSG can also generate a larger sequence space by allowing the use of multiple related root sequences. With all these functions, iSG can be used to test the accuracy of, for example, multiple alignment methods, phylogenetic methods, evolutionary hypotheses, ancestral protein reconstruction methods, and protein family classification methods. We empirically evaluated the performance of iSG against currently available methods by simulating the evolution of the G protein–coupled receptor and lipocalin protein families. We examined their true multiple alignments, reconstruction of the transmembrane regions and beta-strands, and the results of similarity search against a protein database using the simulated sequences. We also presented an example of using iSG for examining how phylogenetic reconstruction is affected by high indel rates.

Introduction

Generating multiple alignments and reconstructing phylogenetic relationships from protein sequences are frequently the most important first steps for various bioinformatics and molecular evolutionary analyses. Reconstructing reliable phylogenetic relationships, for example, depends on the quality of multiple alignments. Generating reliable multiple alignments, however, becomes extremely difficult when one has to deal with distantly related, remotely similar sequences. During such long sequence evolution, dynamic evolutionary events such as duplication, recombination, insertion, and deletion are frequently found. Such dynamic changes are important when we consider the evolution of protein functions.

Rigorous evaluation of multiple alignment and phylogenetic methods, especially for highly diverged sequences with large numbers of amino acid insertions or deletions (indels), is possible if we have a tool that can simulate realistic sequence evolution incorporating indel events based on a valid biological model derived from empirical data. However, current simulation methods are mostly focused on reconstructing “substitution” events (for both nucleotides and amino acids) over entire protein sequences, for example, evolver (Yang 1997) and the simulator in Molphy (Adachi and Hasegawa 1996). They cannot accommodate the numerous evolutionary forces that act upon functional domains found in families of proteins. These domains frequently, but not always, coincide with structural domains and are connected together by amino acid sequences that have no rigid structure (e.g., random coils). Conservation of these domain regions often imposes different evolutionary constraints, including different evolutionary models (e.g., amino acid frequencies and substitution rates) and indel parameterizations (e.g., maximum indel size, length distribution, and probability of opening an indel). Smaller functional units, or motifs, embedded within domain regions may require other specific models, such as invariability of amino acid positions and prevention of indel events within the motif. Seq-Gen (Rambaut and Grassly 1997) introduced substructures that are allowed to have different lengths and evolve along different evolutionary trees. However, one cannot change evolutionary pressures acting upon the substructures, such as differing substitution models and motif conservation. Simulating sequence evolution with site-specific interactions (SISSI) (Gesell and von Haeseler 2006) accounts site-specific interactions in its nucleotide sequence simulation, enabling a process such as RNA evolution with secondary-structure constraints. To apply SISSI for protein sequence evolution, however, we need information on residue interactions in greater detail than we currently have.

Although indels are a necessary component for portraying sequence evolution, their addition in sequence simulation poses a number of problems. One such set of problems relates to the indel creation itself, that is, where to and where not to place indels, the distribution of indel lengths, and how to generate inserted sequences. The pioneering application for the addition of indel events during simulation is ROSE (Stoye et al. 1998). ROSE implements indel events using a simple, strict model: Indels occur linearly with respect to evolutionary distance and with a user-defined length distribution. The sequence inserted is a random sequence based on the given amino acid frequencies. Empirical indel models whose indel probabilities are nonlinear with respect to evolutionary distance, such as Benner et al. (1993) and Chang and Benner (2004), cannot be utilized in ROSE. User input parameters are also global across the entire sequence, not allowing for domain-specific interactions.
Recently, several other sequence simulators including indels have been developed: SIMPROT (Pang et al. 2005), MySSP (Rosenberg 2005), EvolveAGene (Hall 2004), and DAWG (Cartwright 2005). SIMPROT can generate protein sequences and, like Seq-Gen, allows the user to create subsequences evolving using different parameters, in the end concatenating these subsequences into a single sequence. However, SIMPROT does not include the options for the input of a root sequence, the preservation of sequence motifs, or changing amino acid frequencies between subsequences.

indel-Seq-Gen (iSG) is our new method of generating realistic protein families, accomplished through the introduction of multiple models of indel evolution and the ability to parameterize and simulate heterogeneous domains. It provides a simple and biologically realistic method of modeling a protein family. It allows the use of multiple related root sequences, instead of a single static root sequence or multiple random sequences, to generate a realistic large sequence space. This is useful for evaluating, for example, protein classification methods. We showed that when compared with other methods, our method could generate divergent protein sequences without destroying important functional properties, whereas providing a clean and concise method of model creation.

Methods
Protein Sequence Evolution Engine without Indels

As the name implies, we use Seq-Gen (Version 1.3.2) from Rambaut and Grassly (1997) as the substitution evolution engine. The current Seq-Gen incorporates both nucleotide and amino acid sequence simulation and succeeded the original protein sequence generator PSeq-Gen (Grassly et al. 1997). The reason for using Seq-Gen as the core engine is threefold: 1) We found no reason to reinvent a method that evolves sequences with substitutions as many methods already exist (Adachi and Hasegawa 1996; Rambaut and Grassly 1997; Yang 1997; Stoye et al. 1998; Pang et al. 2005). 2) Seq-Gen has been widely used in simulation studies, and 3) Seq-Gen’s setup is a good fit to iSG’s family-modeling system. Our approach adds new capabilities to Seq-Gen as described below.

Protein Family Creation

Protein families are often characterized by their structural and functional components, domains, and motifs. The heterogeneity in evolutionary rates caused by domains and motifs is often modeled with continuous or discrete gamma rate distributions (e.g., Yang 1993, 1997; Adachi and Hasegawa 1996). However, if the functional regions are well known and the user wishes to simulate a sequence family that reflects such characteristics, it is better to provide region- or site-specific constraints and evolutionary models, rather than to allow the simulator to randomly select conserved sites. For example, a transmembrane region should not be guided by the same substitution model as a hydrophilic coil region. Neither should a sequence motif occurring in a loop region be changed at the same rate as the surrounding region. Highly diverged families such as the G protein–coupled receptors (GPCRs) also accumulate indel events throughout their evolution without destroying their functional units. Our strategy for simulating protein family evolution is through 1) the introduction of “domain” units and 2) the introduction of invariable sites, “motifs.”

Motif Conservation

The conservation of essential motifs in sequence generation can be accomplished by disallowing substitution events within the motifs. Invariable sites are introduced for this purpose. However, when indels are incorporated, the conservation of length-sensitive motifs becomes a problem. For example, ROSE allows for the preservation of motifs by joining the gamma-distribution rates and invariable sites into a mutation probability vector, which we call the I + γ array. Each site in the I + γ array specifies the substitution rate for the corresponding amino acid site. If the rate is 0.0, then the site is invariable. If the rate is less than 1.0, the site is not allowed to have any indel. Tying indels to the relative substitution frequencies is a drawback in this representation because it cannot represent low-frequency indels in regions that have high substitution rates, such as transmembrane regions. Neither can it represent highly variable sites in length-based motifs that depend on the distance between particular amino acids, such as the “CXXC” motif. When shorter or longer, this motif loses its propensity for creating the disulfide bridges that are essential for thioredoxin-fold proteins (Chivers et al. 1996).

To rectify these issues, we created a new representation of invariable arrays with 4 classes of invariable sites: 0, no constraint; 1, invariable, 2, no indel; and 3, invariable and no indel. “Invariable” sites can have no substitution, but insertions are allowed between consecutive invariable sites. “No-indel” sites refer to the position in which indels are not allowed but substitutions are. “Invariable and no-indel” sites allow neither substitution nor indel to occur. Neither an insertion nor a deletion can exist between 2 consecutive “no-indel” positions (positions represented as “22,” “23,” “32,” and “33” in the array). For example, the thioredoxin-fold CXXC motif (where X is any amino acid) can be represented as “3223” in the invariable array. This will hold the 2 C’s invariable, allow the 2 X’s to be substituted independently, and disallow indels from occurring within the motif, preserving the length dependence.

Heterogeneous Evolution among Domains

Seq-Gen (Rambaut and Grassly 1997) introduced the idea of “partitions” to allow variations in evolutionary rates and patterns among domains. Each partition represents a subsequence of a protein, and each partition evolves independently as specified by the evolutionary tree. However, more variations in evolutionary patterns need to be incorporated to represent, for example, different indel rates in secondary-structure regions compared with unstructured coil regions (Thorne 2000). In iSG, the percentage of invariable sites, branch lengths (representing substitution rates), amino acid frequencies, substitution models, and indel rates can all be varied between partitions. Such flexible options allow us to generate realistically complex protein families.
Indel Event Handling

Indel events are governed by 4 parameters: the probability of an indel occurring, the placement of indels, the length distribution, and the maximum indel length. For insertion events, 1 more parameter is required for generating the amino acids for an inserted sequence.

Probability of Indels

In iSG, the following 2 indel models are used:

- Linear model, which assumes that the number of indel events is linearly related to the substitution rate: \( P(\text{indel}) = kd \), where \( k \) is a user-defined constant and \( d \) is the substitution rate specified by the branch length between the ancestral and descendent sequences.

- The Chang and Benner (2004) model, which specifies the probability of an indel based on the following exponential equation:

\[
P(\text{indel}) = 0.224 - 0.0219e^{-0.0116kd}.
\] (1)

Placement of Indels

The placement of indels is chosen randomly. For insertions, invariable and no-indel sites are excluded from consideration. Deletion can happen only at “no constraint” positions (“0” in the invariable array). For a deletion of size \( n \), any position that is within \( n - 1 \) positions from a nonzero position in the invariable array is excluded.

Length Distribution of Indels

The distribution of gap lengths in protein sequences has been studied through the use of pairwise alignments (Chang and Benner 2004) as well as by examining structural databases (Qian and Goldstein 2001; Gooneskere and Lee 2004). By default, iSG will create the normalized Zipfian distribution for indel lengths (Chang and Benner 2004):

\[
N = 2628(X^{-1.821}),
\] (2)

where \( N \) is the number of proteins of length \( X \). Other models can also be input by the users as precalculated indel distributions (e.g., following a structural indel distribution developed by Qian and Goldstein [2001] or Gooneskere and Lee [2004]).

Insertion Sequence Options

iSG incorporates 2 methods of amino acid sequence generation. The random model randomly chooses amino acids based on the given frequencies. The second method is unique because it chooses amino acids in the insertion sequence by incorporating the “neighbor preference” (Xia and Xie 2002) along with the amino acid frequencies. Neighbor preference was derived by studying neighboring amino acid positions in functional proteins in order to empirically determine the effect of the side chains in the acceptance of functionally beneficial amino acid changes.

We use their \( 20 \times 20 \) matrix to generate insertion sequences in a Bayesian fashion:

\[
P(j|i) = \frac{P(j|i)P(j)}{P(i)}
\] (3)

where \( P(j|i) \) is the probability that amino acid \( j \) follows amino acid \( i \) and \( P(j) \) and \( P(i) \) are the frequencies of amino acids \( j \) and \( i \) in the sequence, respectively. Given an amino acid, we choose the next amino acid based on the probability \( P(j|i) \). When the first amino acid is inserted, we use the amino acid preceding the insertion point as amino acid \( i \). If the insertions are longer than 1 amino acid, we use the newly generated amino acid as the predecessor to find the next amino acid.

Originally, neighbor preferences were built on the protein sequences derived from the Escherichia coli K-12 genome. We also provide neighbor preferences calculated over all of the protein sequences contained in the SwissProt database (Bairoch et al. 2005). The neighbor preferences are global over the entire simulation run of a family of sequences.

Root Sequence Options

iSG provides 2 options to incorporate a root sequence provided by the user: a single root sequence or a set of sequences. For the single root sequence option, a sequence and its invariable array are read in and used verbatim for all simulation runs. However, for protein families that may contain highly variable regions, varying the root sequence in order to explore a larger sequence space may be advantageous. To facilitate this, iSG incorporates the option of inputting a set \( S \) of sequences in the form of a multiple alignment. For each simulation run, a single root sequence, which may vary between simulation runs, is constructed from \( S \) in the following manner. For each position (column) in the multiple alignment, where half or more of the sequences are nongap, we choose a representative amino acid using either of the following 2 methods: 1) the consensus method, which chooses the representative using majority rule or 2) the random method, which probabilistically chooses the representative based on the amino acids that exist in the column of the multiple alignment. If any alignment position has a gap in more than half of the sequences, the position is considered to be an insertion position and is ignored in the construction of the root sequence. The following 3 parameters can be set for building the root sequence from a set of sequences:

- The range of columns in the multiple alignments to use.
- The number of sequences used. All (the default) or a given number of sequences can be selected from the multiple alignment using the bootstrap-sampling method (selecting randomly with replacement). The bootstrap-sampling option is useful, for example, when using highly divergent sequences with a large number of gap sites. The simple consensus from such divergent sequences may include mainly gaps and very few amino acids. On the other hand, when there are many equally probable root sequences in \( S \), the user may opt to randomly select 1 sequence as the root for each simulation run.
The root sequence generation method. The default consensus method uses the majority rule to choose the amino acid for the column, using a coin toss to break ties. The random method randomly chooses the amino acid representing the column based on the position-specific amino acid frequencies, with the exception of invariable columns. For invariable columns, the representative amino acid is chosen using the consensus method.

Implementation

iSG is freely available at http://bioinfolab.unl.edu/~cstrop/e/IG. iSG has been tested on the RedHat Linux, SuSE Linux, IRIX, and Macintosh OSX operating systems with the PERL and gcc compilers.

Comparison of Simulation Methods

Transmembrane Region Prediction

For the GPCR family simulation, we used HMMTOP 2.0 (Tusnady and Simon 2001) to predict the number of transmembrane regions as well as the position of the N-terminal region (intra- or extracellular). Realistically simulated sequences should have 7 transmembrane regions and extracellular N-terminal regions.

Beta-Strand Prediction

For the lipocalin family simulation, we used PSIPRED (Jones 1999) to predict the secondary structures. Realistically simulated sequences should have 8 beta-strands that correspond to the beta-barrel structure found in the lipocalin family proteins.

Blast Similarity Search

In order to see how the simulated sequences preserved the similarities against the template proteins, we performed BlastP (Altschul et al. 1997) protein similarity searches against the protein database UniProt (Bairoch et al. 2004), using the simulated sequences as the queries. We did not use the option to filter out the low-complexity sequences. The default E-value threshold (10) was used to cut off the search results.

Pfam Search

Simulated sequences were used to search the profile hidden Markov model database, Pfam (Bateman et al. 2004), using the program hmmsearch of the HMMER package (Eddy 1998) with the Pfam_ls database to find the global alignments. The scores against the models “7tm_1” (PF00001; for GPCRs) and “Lipocalin” (PF00061; for lipocalins) were recorded. The default E-value threshold (10) was used to cut off the search results.

Parametric Bootstrap and Phylogenetic Analysis

Following the phylogenetic tree reconstructed from the template alignment (see Results and Discussion), 1,000 simulated data sets were generated, each including the same number of protein sequences as in the template set. Phylogenetic trees were reconstructed from these data sets. The maximum parsimony and Neighbor-Joining (Saitou and Nei 1987) with JTT distance (Jones et al. 1992) estimation methods, consensus trees, and bootstrap values were calculated using PHYLIP programs (version 3.65; Felsenstein 2005).

Results and Discussion

Simulation Setup

Template Multiple Alignments

We chose to model protein sequences based on 2 protein family data sets: 1) the vertebrate olfactory receptor family and 2) the lipocalin protein family.

GPCR–Coupled Receptors. The vertebrate olfactory receptor family belongs to the GPCR superfamily, Class A (rhodopsin-like). GPCR proteins have 7 transmembrane regions, and their sequences are known to be highly diverged including many indels. Among them, vertebrate olfactory receptors are relatively conserved and generating a template multiple alignment is not too difficult. We chose 2 sequences from each of the 14 subfamilies, yielding 28 protein sequences. We also included 1 outgroup sequence (OPSD_BOVIN) from the GPCR rhodopsin class (Class A), as was done previously (Gilad et al. 2005). Thus, the total number of sequences in our template alignment is 29.

The process of building the template multiple alignment is illustrated in figure 1, steps 1–3. Based on the sequence characterization in UniProt, we first split each sequence into 15 parts: the 7 transmembrane regions (TM1–TM7), 4 extracellular regions (EC1–EC4), and 4 cytosolic regions (CY1–CY4). We manually adjusted region boundaries where the UniProt characterizations were conflicting.

Multiple alignments of each region were done using T-Coffee (Notredame et al. 2000) and adjusted manually. These regional alignments were concatenated together obtaining the template multiple alignment for the entire protein sequences (see fig. 2a; the actual alignment is shown in supplementary fig. 1, Supplementary Material online). In Step 4, we reconstructed the maximum parsimony phylogeny using PAUP* version 4.0b10 (Swofford 2003). Using the topology obtained from the template multiple alignment, the branch lengths were calculated for each region. The average number of changes per site for each segment was obtained by dividing each tree length by the number of sites in the segment. Subsequence parameters are set as shown in step 6 of figure 1 (full specifications are found in supplementary fig. 2a, Supplementary Material online). They include amino acid frequencies, relative evolutionary rates, root sequence templates, indel parameters, and phylogenies. We obtained the amino acid frequencies specific to the 3 regions (EC, CY, and TM) using 1594 GPCR Class A sequences found in UniProt.

Lipocalins. Lipocalin proteins are a family of small globular proteins belonging to the calycin superfamly. They are often implicated in allergic reactions. Members of the lipocalin family are highly diverged, but all share a conserved beta-barrel conformation consisting of 8 beta-strands (Flower et al. 1993).

The template multiple alignment and phylogeny for the lipocalin family including 23 sequences were obtained...
from an evolutionary study done by Sánchez et al. (2003). The sequences were split into 12 regions including 5 beta-strand regions as shown in figure 3c. Note that the region including 4 short beta-strand regions (B5678 in fig. 3c) is treated as a single evolution unit. Branch lengths were estimated from each segment as described before. Four coil regions (indicated by ‘ ‘C’ ‘ in fig. 3c) were very short (each with 4–6 amino acids in length). Therefore, their branch lengths were obtained based on their concatenated sequences. The amino acid frequencies specific to the beta-strand, coil, or alpha-helix regions were calculated using those obtained from the template multiple alignment (75% weighting) combined with pseudocounts (25% weighting). The pseudocounts were obtained either from the conformational parameters by Chou and Fasman (1974) for beta-strands and alpha-helices or from Jones et al. (1992) for the N terminus, C terminus, and coil regions.

Setting Parameters

The setting of the parameters was done so that the simulated sequences were close to the template alignment, but still general enough to allow for variations. For example, in the GPCR template alignment (fig. 2a and supplementary fig. 1a, Supplementary Material online):

1. TM1, 2, 4, and 6 contain more gaps than TM3, 5, and 7 and
2. The loop regions between TM4 and TM7 (EC4, CY3, and EC5) are more diverged than other regions.

Parameters such as indel rates for each region are set to closely follow these features. Supplementary figure 1 (Supplementary Material online) shows the invariable array used with iSG that holds the conserved motifs found in the vertebrate olfactory receptors by Fuchs et al. (2001). For the lipocalin template alignment, we allowed no indel in the beta-strand or alpha-helix regions. For the other lipocalin regions, we set the deletion rate to be 1 per 100 substitutions and the insertion rate to be twice higher than the deletion rate. Supplementary table 1 (Supplementary Material online) shows a list of the parameters used in iSG, ROSE, and SIMPROT. In this study, a single global substitution matrix (JTT for iSG, SIMPROT, and Seq-Gen and PAM for ROSE) was used. However, with iSG we can specify different substitution matrices for different subsequences (e.g., a TM-specific substitution matrix). Using these parameters a data set of 29 GPCR-like sequences or 23 lipocalin-like sequences were simulated following their template phylogenies.
Comparison of Simulated Sequences between iSG and Other Methods

We compared iSG with ROSE, Seq-Gen, and SIMPROT for their protein family simulation performance. As described before, ROSE uses the $I + \gamma$ array for simulating protein family sequences, and SIMPROT and Seq-Gen both allow subsequence generation. The major disadvantage of using ROSE was that we could not allow low-frequency indels although allowing amino acid substitutions as observed in TM regions in GPCRs. Because indel rates are linearly correlated with substitution rates in ROSE, in low-frequency indel regions, we were required to set low substitution rates. Because parameters (e.g., indel rates and amino acid frequencies) could not be varied among subsequences in ROSE, we were forced to set parameters with the “average case” values across the entire sequence. Note also that setting these parameters in ROSE was very tedious and time consuming. For example, over 300 real-valued numbers had to be assigned to the $I + \gamma$ array in the case of GPCRs, and the array does not visually align with the sequence because the former is a real-valued array and the latter is a character array. iSG, on the other hand, uses integer values in the invariable array, which can be easily aligned with a character array (see supplementary fig. 1, Supplementary Material online).

Setting the parameters for SIMPROT and Seq-Gen were more straightforward. The Seq-Gen root sequence was constructed using iSG’s root sequence construction method, and all parameters available to Seq-Gen were set the same as iSG. For SIMPROT, we partitioned the sequence as we did in iSG, and many parameters could be set as done with iSG. The root sequence is randomly generated in SIMPROT. Additionally, SIMPROT uses an equal rate for insertions and deletions.

One thousand data sets were simulated using each method. The “true” multiple alignments showing the actual indel positions were collected. We compared the true alignments generated by iSG and ROSE against the profile HMM built from the template GPCR alignment (supplementary fig. 3, Supplementary Material online; Edgar and Sjolander 2004). No significant differences were shown between the normal score distributions obtained from iSG (mean $= 75.03$ and variance $= 17.20$) and ROSE (mean $= 72.30$ and variance $= 16.72$), indicating that the sequences generated by the 2 methods using the set of the parameters we chose were approximately equivalent with respect to the template multiple alignment.

Simulated Sequences

In general, functional domains are under strong selective constraints and very few indels are tolerated. TM regions are one such example. On the other hand, changes (both substitutions and indels) within loop regions that simply connect functional or structural domains often have much smaller negative consequences and thus are under very weak constraints. In figure 2, TM regions are illustrated with a gray color and indels are shown as white gaps. As figure 2a shows, vertebrate olfactory receptor proteins are more conserved in the regions between TM2 and TM4 than other regions (as found in Fuchs et al. 2001). Note that some TM regions (TM1 and TM4) can have a few small indels. In the simulated sequences, we attempted to recreate
these heterogeneous sequence features: fewer indels between TM2 and TM4 and, conversely, more indels in N- and C-terminal regions and those between TM4 and TM7.

As shown in figure 2h, the simulated sequences by iSG reproduced the intended sequence features well. Because the equivalent parameters could be set, SIMPROT produced the true alignments similar to those with iSG (data not shown). ROSE, however, spread the indels throughout the entire length of the proteins (fig. 2c). When the simulated sequences were aligned using T-Coffee multiple alignment method, reconstructed alignments based on iSG’s data sets placed indels in a manner that is much closer to the template multiple alignment than those based on ROSE’s data sets.

iSG allowed low-frequency indels within TM regions. From figure 2, it appears as if iSG allows far too many indels in the TM regions compared with the template alignment. However, in this experiment we intentionally chose parameters to introduce a large number of indels to our simulated data sets. In ROSE, on the other hand, it was not possible to allow indels within TM regions. In order to introduce any indels within TM regions, the mutation array values used in ROSE need to be set at 1.0 or higher. Because indel rates are linearly correlated to substitution rates, using such high mutation array values would perforate each region with so many indels that the resulting loss of protein function would be inevitable. To avoid such unreasonable simulation conditions, we used conservative substitution rates with ROSE (0.7 in the $I$ + $Y$ array). Consequently, in the true alignments produced by ROSE (fig. 2c) there are gray “islands” (TM regions) where no indel was found. The side effect of this was an easier multiple alignment reconstruction for T-Coffee, which was nearly perfect in the TM region reconstructions. Supplementary figure 4 (Supplementary Material online) shows examples of the actual true alignments given by different simulation methods.

**Conservation of Transmembrane Regions**

To check if the TM regions were retained in the reconstructed data sets, we predicted TM regions from simulated GPCR sequences. The numbers of predicted TM regions were $7.03 \pm 0.30$ by iSG, $5.94 \pm 1.25$ by ROSE, $0.20 \pm 0.37$ by SIMPROT, and $6.84 \pm 0.91$ by Seq-Gen. Clearly, iSG has the greatest accuracy in reproducing the 7 TM regions. The N terminus (EC1) was correctly predicted to be extracellular 94% of the time in simulated data sets of iSG. Conversely, with ROSE and Seq-Gen, only 44% and 55% of simulated sequences, respectively, were predicted to have the extracellular N terminus correctly. These results clearly show that iSG’s ability to allow different amino acid frequencies between domains is effective in preserving important features of transmembrane proteins more accurately than other methods. This is important because results from Panchenko and Madej (2005) suggested patterns found in nondomain regions, such as loop or N/C-terminal regions, were more important in recognizing superfamilies than considering only the conserved core regions.

**Beta-Region Prediction**

To examine the conservation of beta-strand regions, secondary structures were predicted from simulated lipocalin sequences. As shown in figure 3, comparing with the reference sequences, all simulated sequences have lower proportions of beta-strands indicated by lower “mean-beta” values and lower peaks of beta-strand plots corresponding...
to the 8 beta-strand regions (gray regions in fig. 3c). However, iSG performed better than the other methods with just over 5% better accuracies. Among the simulation methods compared, SIMPROT performed most poorly. Eight beta-strand regions were indistinguishable in its simulated sequences, indicated also by the lowest mean-beta and highest “mean-coil” values. The poor performance by SIMPROT is explained by its inability to use a specific root sequence.

In this study, beta-strand regions were characterized only by the amino acid composition derived from small samples of lipocalin sequences. Therefore, the performance by iSG to reconstruct beta-strand structures was surprisingly good considering the use of such simple parameters. Better performance can be expected by optimizing pseudo-count methods and using secondary structure-specific amino acid composition and substitution matrices derived from larger samples.

**Blast and PFAM Search Results**

In order to further examine how simulated sequences preserved functionally important sequence properties, we used the simulated sequences as queries for BlastP protein similarity search. Table 1 summarizes the search results. The top 250 hits obtained by all methods except SIMPROT were correctly from various members of vertebrate olfactory receptors. Seq-Gen performed the best of the 3 methods indicated by the highest average scores. The absence of indels in Seq-Gen generated sequences must have caused a low percentage of gaps in the alignments. iSG performed nearly as well as Seq-Gen and outperformed ROSE with higher scores and lower percentages of gaps. The lower percentage of gaps found in Blast alignments with iSG sequences than those with ROSE sequences can be explained by the use of amino acid frequencies specific to subsequences (e.g., TM, extracellular, and cytosolic regions) in iSG. It must have produced amino acids that are most likely to appear in each region of GPCRs. As a result, fewer gaps were required in the iSG alignment making better hit scores than in ROSE. Contrary to iSG, ROSE, and Seq-Gen, simulated sequences by SIMPROT did not find any vertebrate olfactory receptor protein under the E-value threshold used with the BlastP search. Within the threshold, SIMPROT sequences averaged between 3 and 5 unrelated hits, with the average highest score = 30 (E value = 1.5).

PFAM search results showed similar conclusions as BlastP. Except for SIMPROT sequences, the search against PFAM profile HMM database using simulated sequences returned 7tm_1, the profile HMM entry for GPCR Class A, as the most common hit. Table 1 (in the bottom half) summarizes the scores against the 7tm_1 profile HMM by the 4 simulation methods. iSG clearly outperformed ROSE. The difference between iSG and Seq-Gen was again very small, with slight advantage to iSG-generated sequences. No significant hit by GPCR-derived profile HMMs were found using SIMPROT-simulated sequences as queries.

For simulated lipocalin proteins, BlastP and PFAM search did not produce as many significant hits as we observed with GPCR sequences, which implies the difficulty in simulating beta-strand proteins. As shown in table 2, the BlastP results were comparable among the simulation methods except for SIMPROT. Only about 2 lipocalins were found per simulated sequence query. Interestingly, slightly more different lipocalin sequences were found among simulation sets produced by iSG than among those produced by ROSE and Seq-Gen. This is probably a result of the multiple alignment root sequence option in iSG changing the root sequences between data sets. Against the PFAM Lipocalin profile HMM entry, iSG had the lowest scores. However, iSG produced the highest number of sequences (33.62%) that showed a hit to the lipocalin profile, compared with ROSE and Seq-Gen (25.68% and 22.41%)

### Table 1

The Results of BlastP and PFAM Search When Using Simulated GPCR Sequences as the Query

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of Top Hits</th>
<th>Avg a</th>
<th>Min a</th>
<th>Max a</th>
<th>% ID b</th>
<th>% SIM b</th>
<th>% GAP b</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlastP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iSG</td>
<td>25</td>
<td>174.0</td>
<td>115.0</td>
<td>303.2</td>
<td>34.3</td>
<td>55.9</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>164.7</td>
<td>106.0</td>
<td>303.2</td>
<td>33.1</td>
<td>54.6</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>155.9</td>
<td>95.8</td>
<td>303.2</td>
<td>32.1</td>
<td>53.5</td>
<td>3.4</td>
</tr>
<tr>
<td>ROSE</td>
<td>25</td>
<td>141.1</td>
<td>93.6</td>
<td>258.8</td>
<td>35.3</td>
<td>53.1</td>
<td>7.5</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>132.7</td>
<td>85.4</td>
<td>258.8</td>
<td>34.1</td>
<td>52.1</td>
<td>7.6</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>124.4</td>
<td>74.4</td>
<td>258.8</td>
<td>33.0</td>
<td>51.0</td>
<td>7.6</td>
</tr>
<tr>
<td>Seq-Gen</td>
<td>25</td>
<td>196.7</td>
<td>132.2</td>
<td>324.8</td>
<td>34.5</td>
<td>55.9</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>187.3</td>
<td>122.2</td>
<td>324.8</td>
<td>33.5</td>
<td>54.8</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>177.8</td>
<td>110.2</td>
<td>324.8</td>
<td>32.5</td>
<td>53.7</td>
<td>0.5</td>
</tr>
<tr>
<td>SIMPROT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iSG</td>
<td>7tm_1</td>
<td>5.09</td>
<td>−99.5</td>
<td>158.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROSE</td>
<td>7tm_1</td>
<td>31.47</td>
<td>−114.9</td>
<td>97.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq-Gen</td>
<td>7tm_1</td>
<td>7.18</td>
<td>−129.2</td>
<td>160.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIMPROT</td>
<td>7tm_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a The average (avg), minimum (min), and maximum (max) bit scores from each set of the top hits.

b The average percentage identity (ID), similarity (SIM), and gaps (GAP) obtained from each set of the top hit alignments.

c SIMPROT queries found only 3–5 unrelated hits by BlastP. PFAM profile HMM search did not find 7tm_1.

d Only scores for the most commonly found PFAM model, 7tm_1, are shown.
Table 2
The Results of BlastP and PFAM Search When Using Simulated Lipocalin Sequences as the Query

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of BlastP Hits</th>
<th>Number of Different Lipocalin Hits</th>
<th>Average PFAM Score</th>
<th>% Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>iSG</td>
<td>2.22 (43.36)</td>
<td>60</td>
<td>−30.12</td>
<td>33.62</td>
</tr>
<tr>
<td>ROSE</td>
<td>2.25 (42.16)</td>
<td>56</td>
<td>−18.34</td>
<td>25.86</td>
</tr>
<tr>
<td>Seq-Gen</td>
<td>2.31 (45.87)</td>
<td>56</td>
<td>−14.14</td>
<td>22.41</td>
</tr>
<tr>
<td>SIMPROT</td>
<td>0 (—)</td>
<td>0</td>
<td>—</td>
<td>0</td>
</tr>
</tbody>
</table>

- The average number of BlastP hits to lipocalins for each simulated sequence.
- The average bit score for these lipocalin hits is shown in parentheses.
- The total number of different lipocalin sequences found by BlastP using all simulated sequence queries.
- The average score against the Lipocalin profile HMM among queries. SIMPROT queries did not obtain lipocalin hits.
- The proportion of simulated sequences that are matches to the PFAM Lipocalin model.

Table 3
Phylogenetic Analysis Using Simulated GPCR and Lipocalin Sequences with Various Substitution and Indel Rates

<table>
<thead>
<tr>
<th>% Supporta</th>
<th>Substitution Rate × 1b</th>
<th>Substitution Rate × 2b</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lipocalinc</td>
<td>GPCRd</td>
</tr>
<tr>
<td>x = 100</td>
<td>14 (14)</td>
<td>25 (25)</td>
</tr>
<tr>
<td>90 &lt; x &lt; 100</td>
<td>6 (6)</td>
<td>1 (1)</td>
</tr>
<tr>
<td>80 &lt; x &lt; 90</td>
<td>0 (0)</td>
<td>0 (0)</td>
</tr>
<tr>
<td>70 &lt; x &lt; 80</td>
<td>0 (0)</td>
<td>0 (0)</td>
</tr>
<tr>
<td>x &lt; 70</td>
<td>0 (0)</td>
<td>0 (0)</td>
</tr>
</tbody>
</table>

- The average percentage of supportd obtained from all internal branches.

Phylogenetic Analysis with Parametric Bootstrap

Neighbor-Joining phylogenies were reconstructed based on the T-Coffee as well as the true multiple alignments of simulated sequences. For both GPCR and lipocalin simulations using the parameters described earlier, the topologies of the consensus trees obtained from all 4 methods were identical to the original phylogenies reconstructed from the template multiple alignments, and all nodes were supported with 99% or better bootstrap values. To explore the effect of indels on phylogenetic reconstruction, we simulated another 100 data sets of GPCRs and lipocalins using iSG, with twice higher substitution rates and indel rates used in iSG, ROSE, and Seq-Gen. Neither ROSE nor Seq-Gen in their original methods included this capability.

Conclusion

iSG is a new method for simulating protein sequence evolution and generating realistic protein families. It has a unique ability to simulate heterogeneous evolution of multidomain protein families in addition to the incorporation of indel evolution. iSG is highly flexible and various sequence features can be easily included in the simulation. For example, iSG allows highly variable regions in the sequence although still maintaining low indel rates. This is useful in simulating motifs such as CXXC in thiooxidin fold proteins or zinc fingers. This is not possible in other simulation methods. The use of a template multiple alignment is also a unique strength of iSG. With these new functions, iSG can be used effectively for testing the
performance of various molecular evolution and bioinformatics methods when they are applied to extremely divergent heterogeneous protein analysis. Because each subsequence can be simulated with a different evolutionary tree, lateral gene transfer and recombination detection methodologies can also be examined. For protein families with well-known features, iSG can be used to generate synthetic model sequences. Such model sequences can be used to search their candidate members from databases.

Supplementary Material

Supplementary figures 1–4 and table 1 are available at Molecular Biology and Evolution online (http://www.mbe.oxfordjournals.org/).

Acknowledgments

We thank Drs Nick Grassly (Imperial College) and Andrew Rambaut (Department of Zoology, Oxford University) whose program, Seq-Gen, is the simulation engine on which iSG is built. We also thank Dr Dave Posada and 3 anonymous reviewers for their helpful comments.

Literature Cited


Dan Graur, Associate Editor

Accepted November 29, 2006