Biological Sequence Simulation for Testing Complex Evolutionary Hypotheses: indel-Seq-Gen Version 2.0

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

*Department of Computer Science and Engineering, University of Nebraska; †School of Biological Sciences, University of Nebraska; and ‡Center for Plant Science Innovation, University of Nebraska

Sequence simulation is an important tool in validating biological hypotheses as well as testing various bioinformatics and molecular evolutionary methods. Hypothesis testing relies on the representational ability of the sequence simulation method. Simple hypotheses are testable through simulation of random, homogeneously evolving sequence sets. However, testing complex hypotheses, for example, local similarities, requires simulation of sequence evolution under heterogeneous models. To this end, we previously introduced indel-Seq-Gen version 1.0 (iSGv1.0; indel, insertion/deletion). iSGv1.0 allowed heterogeneous protein evolution and motif conservation as well as insertion and deletion constraints in subsequences. Despite these advances, for complex hypothesis testing, neither iSGv1.0 nor other currently available sequence simulation methods is sufficient. indel-Seq-Gen version 2.0 (iSGv2.0) aims at simulating evolution of highly divergent DNA sequences and protein superfamilies. iSGv2.0 improves upon iSGv1.0 through the addition of lineage-specific evolution, motif conservation using PROSITE-like regular expressions, indel tracking, subsequence-length constraints, as well as coding and noncoding DNA evolution. Furthermore, we formalize the sequence representation used for iSGv2.0 and uncover a flaw in the modeling of indels used in current state of the art methods, which biases simulation results for hypotheses involving indels. We fix this flaw in iSGv2.0 by using a novel discrete stepping procedure. Finally, we present an example simulation of the calycin-superfamily sequences and compare the performance of iSGv2.0 with iSGv1.0 and random model of sequence evolution.

Introduction

The goal of simulating sequence evolution is to realistically portray the evolutionary wrestling match between 1) processes that change a biological sequence through point mutations, insertion/deletion (indel), as well as more dynamic chromosomal rearrangement, and 2) functional constraints of a sequence that restrict such changes. Such processes are intertwined during the course of evolution, forming the patterns that we see in extant homologous biological sequences.

When sequence simulation was initiated, simulation methods mainly dealt with substitution processes, incorporating information on substitution patterns, relative substitution rates across sites based on the Gamma distribution, and sites that are invariable throughout the evolutionary history of a group of sequences (Yang 1994; Rambaut and Grassly 1997). These simulation methods, however, did not incorporate processes of insertion and deletion of sequence positions, that is, indels.

The pioneering sequence simulation application to introduce indel events is random model of sequence evolution (ROSE; Stoye et al. 1998). ROSE extended the Gamma distribution to encompass indel constraints as well, that is, if the Gamma substitution rate is below a certain threshold, it forbids an indel to occur in the region. ROSE also provides the "true" multiple alignment that reflects the true evolutionary path of the sequences. True multiple alignments can be used to test the accuracy of multiple sequence alignment methods and various hypotheses (Stoye et al. 1997; Lassmann and Sonnhammer 2002; Subramanian et al. 2005). However, due to the limitations in biological realism with simulated sequences, benchmark data sets are generated based on mainly hand-curated or structure-based alignments of protein sequences (Raghava et al. 2003; Edgar 2004; Subramanian et al. 2005; Thompson et al. 2005; van Walle et al. 2005). These data sets also have limitations, such as their small data set size and ambiguous positional homology among others (Notredame 2007). Furthermore, the benchmark alignments cannot be used for testing phylogenetic methods, because the evolutionary history among the sequences is unknown. The primary advantage of simulated data sets is that the true evolutionary history of the sequences is known.

To improve the realism of simulated sequences, two areas must be addressed: sequence conservation and indel processes. Table 1 compares the functions of various simulation methods. Lineage- and site-specific conservation as well as heterogeneous evolution is needed to improve the realism of sequence evolution simulators. Homogeneous sequence evolution, as illustrated in figure 1*B*, is found in many simulation methods, including EvolveAGene3 (Hall 2008) and DNA assembly with gaps (DAWG; Cartwright 2005). Richer representations allow heterogeneous evolution among sequence "partitions," where each partition can be defined by a different set of substitution and indel parameters (e.g., fig. 1*C*, gray lineage). SIMPROT (Pang et al. 2006) and iSGv1.0 (Strope et al. 2007) include such "partition"wise simulation. For site-specific conservation, the current state of the art is found in iSGv1.0 and ROSE (Stoye et al. 1997), implemented by disallowing sites and subsequences from accepting indels. Site-specific substitution processes, however, are either constrained to be either completely invariable or mutable to any other character. Thus, functional constraints on substitution patterns within the conserved region cannot be simulated, although functional regions often depend on the properties of their residues to maintain their functions. This inability to conserve residue sets in the sequences affects the ability to simulate highly diverged superfamily-level evolution. Lineage-specific evolution is represented only by MySSP (Rosenberg 2005), which allows users to set substitution and indel parameters on each branch of the input guide tree.

Key words: protein superfamily, sequence simulation, domains, motifs, indels.

E-mail: emoriyama2@unl.edu.

Mol. Biol. Evol. 26(11):2581–2593. 2009
doi:10.1093/molbev/msp174
Advance Access publication August 3, 2009

Table 1
A Comparison of Sequence Simulation Methods

	ROSE	DAWG	MySSP	SIMPROT v1.03	iSGv1.0	EvolveAGene3	iSGv2.0
Data simulated							
Noncoding DNA	Yes	Yes	Yes	No	No	Yes	Yes
Coding DNA	No	No	No	No	No	Yes	Yes
Protein	Yes	No	No	Yes	Yes	No[a]	Yes
Indel treatment							
Continuous	Yes	Yes	Yes	Yes	Yes	Yes	No
Dynamic length adjustment	No	Yes	No	No	No	No	Yes
Event tracking	No	No	No	No	No	No	Yes
P_{ins}, P_{del} independent	Yes	Yes	Yes	No	Yes	Yes	Yes
Indel placement treated differently	No	No	No	No	No	No	Yes
Empirical length distribution[b]	No	CB04	No	CB04,QG01	CB04	*Escherichia coli*	CB04
Overlapping indels	Yes	Yes	Yes	Yes	Yes	No	Yes
Heterogeneous evolution (partitions)							
Gamma distribution	No	Yes	No	No	No	No	Yes
Invariable site proportion	No	No	No	Yes	Yes	No	Yes
Indel probabilities	No	No	No	Yes	Yes	No	Yes
Lineage treatment & functional constraints							
Lineage options							
Parameter changing	No	No	Yes	No	No	No	Yes
Pseudogene simulation	No	No	No	No	No	No	Yes
Indel modeling							
Indel depends on N_{ins}, N_{del}	No	No	No	No	No	No	Yes
Motif conservation							
Lineage-specific	No	No	No	No	No	No	Yes
Length-specific	Yes	No	No	No	Yes	No	Yes
Site-specific	No	No	No	No	No	No	Yes

[a] EvolveAGene3 can output amino acid sequences. However, simulation is done only at the DNA level. The amino acid sequence outputs are translations of the resulting DNA sequences.

[b] CB04: Zipfian distribution (Chang and Benner 2004), QG01: Qian–Goldstein distribution (Qian and Goldstein 2001).

For ROSE, indels are simulated based only upon user-input probabilities and length distributions. Other sequence evolution simulators follow similar schemes but add novel functionalities that fit the developers' purposes. DAWG (Cartwright 2005), which simulates noncoding DNA evolution, introduced indels based on an exponential time distribution that determines the waiting time until the next indel event. The waiting time is calculated based on the indel probability as a function of sequence size. DAWG adjusts the sequence length after each indel event, the effect of which is described in more detail in Results. DAWG also introduced an indel length distribution that follows the empirically derived power law distribution of Chang and Benner (2004). MySSP simulates noncoding DNA and chooses indel lengths in a normally distributed fashion centered around a user-input mean length (Rosenberg 2005). SIMPROT introduced a parameterized model of another empirically determined indel length model, the Qian–Goldstein

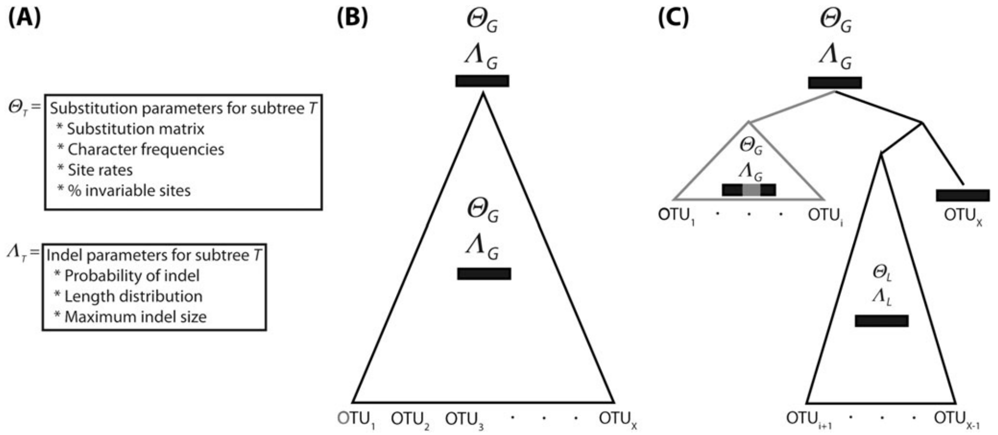


FIG. 1.—A comparison of the basic simulation paradigm and more realistic biological sequence evolution. (*A*) Substitution (Θ) and indel (Λ) parameters used for simulation methods. (*B*) The basic simulation paradigm. Given a root sequence and a global set of substitution and indel parameters (Θ_G and Λ_G, respectively), simulation proceeds by applying changes in a Monte Carlo manner over all sequence positions, following the guide tree and ending with a set of operational taxonomic units (OTUs). (*C*) More realistic biological sequence evolution. Starting with a root sequence and an initial level of substitution and indel parameters (Θ_G and Λ_G, respectively), evolution at sites may become constrained by gaining a functional motif (the gray box shown in the gray lineage), and the substitution and indel parameters may be changed (Θ_L and Λ_L) or the initial parameters are maintained without gaining any functional motif as shown with OTU_X.

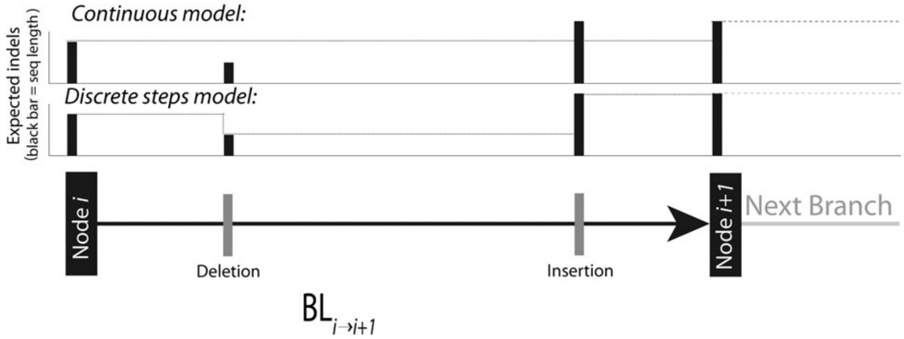


FIG. 2.—The continuous and discrete-step models of indel events. The continuous model calculates the expected number of indel events based on sequence length at node i and uses this same value throughout the branch length, $BL_{i\rightarrow i+1}$. This causes either over or underestimating the number of indels along the branch until recalculating the expected number of indel events at node $i + 1$. The discrete-step model reduces the impact of this by recalculating the expected number of indel events based on the sequence length after each such event.

distribution (Qian and Goldstein 2001), and simulates a continuous indel model by correcting for multiple indels in the same position based on the starting branch length (Pang et al. 2006). EvolveAGene3 simulates coding sequences and indel frequencies as empirically observed in *Escherichia coli* evolution (Hall 2008). Recently introduced GSIMULATOR directly estimates parameters by training transducers on a set of pairwise alignments and uses these transducers to perform the simulations (Bradley and Holmes 2007; Varadarajan et al. 2008). Consequently, users cannot set indel parameters in GSIMULATOR. indel-Seq-Gen version 1.0 (iSGv1.0) (Strope et al. 2007), which used Seq-Gen (Rambaut and Grassly 1997) as the substitution engine, specifically addressed functional subsequence conservation for indels using a novel quaternary invariable array and also allowed for heterogeneous subsequence parameters. In this study, we upgrade iSGv1.0 to indel-Seq-Gen version 2.0 (iSGv2.0) by 1) further improving the realism of biological sequence evolution through the introduction of motif conservation using PROSITE-like regular expressions and lineage-specific evolution and 2) incorporating the DNA substitution engine of Seq-Gen to add both coding and noncoding DNA-sequence simulations. We introduce novel functional constraint enforcement in sequence simulation and formalize how these constraints change the modeling of substitutions, insertions, and deletions (their probabilities of occurrence and placement). We demonstrate a fundamental flaw in simulation of indel processes in many of the current simulation algorithms and perform a comparative analysis of the indel schemes among these methods. In iSGv2.0, we introduce our solution to this problem by incorporating indel simulation in discrete evolutionary steps. The output of iSGv2.0 includes true multiple alignments and information on each indel event including the relative timing and location on the branch (event tracking). iSGv2.0 allows restrictions on minimum and maximum lengths of subsequences by constraining indel events, as is often the case for protein regions with secondary structures. Conservation of folds, as well as motif conservation/gain along different lineages, will be useful to simulate protein superfamily evolution. In addition to the ability to conserve subsequence lengths in DNA sequences, exon–intron structure can also be incorporated to coding-sequence simulation.

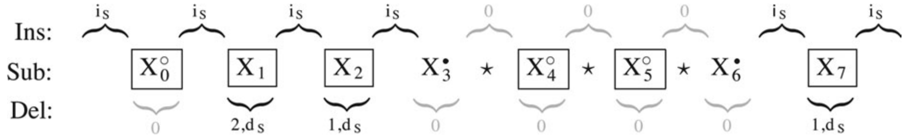
iSGv2.0 is the first tool that is capable of simulating complex substitution and indel processes in constrained evolutionary scenarios. iSGv2.0 incorporates coding DNA, noncoding DNA, and protein simulation. It allows for testing problems such as phylogenetic reconstruction, functional-site inference, joint estimation of alignment and phylogeny, and multiple sequence alignments. As an example of a complex evolutionary scenario, we present a simulation of calycin protein superfamily evolution.

Materials and Methods

We first describe the discrete evolution paradigm introduced in iSGv2.0, along with the implications for substitution and indel evolution. We formalize the sequence representation for simulating evolutionary events such as substitutions and indels for a functionally constrained sequence. We then describe iSGv2.0′s other novel mechanisms: 1) lineage-specific models, 2) site-specific functional constraints, 3) coding DNA-sequence simulation, and 4) indel-event tracking.

Discrete Evolution

The most fundamental structure needed for sequence simulation is the guide tree, which specifies the branching order and the expected number of substitutions that will occur from an ancestral sequence to its descendant. Substitution processes are generally modeled over continuous time, allowing multiple substitutions at the same site. No established model exists for insertions and deletions. Current sequence simulation methods introduce indels in a continuous fashion (Stoye et al. 1997; Rosenberg 2005; Pang et al. 2006; Strope et al. 2007; Hall 2008), even though indels alter the sequence length, as shown in figure 2. In order to keep the indel rate constant along the branch, the number of expected indels needs to be recalculated based on the sequence length after each event. The current continuous model uses the same rate no matter how many positions are inserted or deleted along the branch. Consequently, the number of events can be under or overestimated, which in turn incorrectly decreases or increases the indel rate after insertion or deletion events, respectively, for the remainder of the branch until the probability of an indel event is



Symbols and their constraints imposed on sequence sites.

Symbol	Event Constrained	Description
X°	Deletion	Position *X* cannot be deleted
$X^{\bullet}$	Deletion, Substitution	Position *X* can neither be chaged nor removed
$X_i \star X_{i+1}$	Deletion, Insertion	Length-dependent constrained positions; events that change the subsequence length cannot occur between positions linked by ⋆'s

FIG. 3.—A sequence model that includes substitutions (Sub), insertions (Ins), and deletions (Del) for a length-constrained subsequence $S(X_0,X_1...X_7$, where X_i is the *i*th residue of the sequence). The description of the symbols used and their effects are listed in the table below. For example, positions $X_3...X_7$ are conserved in a way akin to the CXXC motif of the Thioredoxin sequence motif (Chivers et al. 1996). The maximum lengths of insertions are shown above the sequence ranging from 0 (no insertion), 1, 2,. . . to i_S (upper bound of the subsequence length). The maximum lengths of deletions are determined by either 1) the number of sequence positions to the first deletion-constrained position (2 for X_1 in this figure) or d_S, defined as the number of positions that can be deleted before reaching the minimum subsequence length.

recalculated based on the length of the descendant sequence at the next node. The order and time of events within the branch are also unknown in the continuous model of sequence evolution, because the expected numbers of substitutions and indels are estimated based on the entire branch length.

To minimize the effects of sequence-length changes and to allow for event tracking (described later in this section), iSGv2.0 simulates sequence evolution with discrete steps using a sufficiently small step size. The step size ε is defined as

$$\varepsilon = \frac{\text{max\_path}}{2^{10+c}}, \tag{1}$$

where max_path is the maximum length of the root-to-tip paths in the guide tree. The constant *c* is set such that ε is less than 0.01 substitutions per site or smaller than the minimum branch length. If the minimum branch length is less than 0.00001, which is the minimum value ε can take, it is considered to be a zero branch length. We call this simulation method the "discrete evolutionary steps" (DES) model.

Substitution and Indel Models

The branch lengths given in the guide tree determine the rates of evolution, which are expressed as the number of substitutions per site for the branch and are introduced based on models of continuous substitution evolution processes, for example, Jones–Taylor–Thornton (Jones et al. 1992), PAM (Dayhoff et al. 1978), or BLOSUM (Henikoff and Henikoff 1992) for proteins, and Hasegawa–Kishino–Yano (Hasegawa et al. 1985) or general time reversible model (Yang 1994) for nucleotides. In the current simulation methods, indels are also introduced in a continuous fashion. As mentioned earlier, this continuous method assumes that the length of the sequence will remain the same during the evolution along the branch (fig. 2). This is clearly incompatible with insertion and deletion processes and is the primary motivation for adopting the DES model. In the following section, we formalize the model of insertions and deletions with respect to the sequence and functional constraints, which will make clear the flaw in the indel representation used in current methods.

Formalization of Substitution and Indel Processes

Note that although we refer to our model as "discrete," our substitution processes are simulated using the continuous evolutionary models described above, the difference being that substitutions are simulated in multiple ε-sized steps. With respect to insertion and deletion processes, most simulation methods treat them similarly. However, insertions, deletions, and substitutions all work differently with respect to the sequence and functional constraints placed on them. One major difference between insertion and deletion processes is that insertions occur "between" sites, whereas deletions occur "on" sites. This fundamental difference not only affects the number of sites that can accept a deletion versus an insertion but also introduces maximum and minimum length requirements to subsequences in order to enforce possible selective constraints on such subsequence length. This in turn restricts the number of acceptable deletion and insertion lengths in subsequences. Figure 3 specifies the model of these constraints for realistic sequence evolution, where the positions X_3L to X_6, for example, approximate the "CXXC" motif of Thioredoxin-fold proteins (Chivers et al. 1996). Although amino acids X_3 and X_6 can be neither changed nor deleted, amino acids X_4 and X_5 can be substituted but cannot be deleted. Furthermore, the length from X_3 to X_6 (four residues) is constrained and no indel is allowed in this region.

Novel Indel Characterization

Indel modeling requires four parameters: $\Lambda = \{P_{\text{ins}}, P_{\text{del}}, \lambda_{\text{ins}}, \lambda_{\text{del}}\}$, where P_{ins} and P_{del} are the probabilities of an insertion and a deletion, respectively, and λ_{ins} and λ_{del} are the length probability distributions defined as

$$\lambda = \begin{cases} f(x) & x \in \{1, 2, \ldots, x_{\max}\}, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where *x* is the number of residues and $x_{\max}$ is the maximum insertion or deletion size. λ_{del} is defined similarly as λ_{ins}.

For convenience, we assume $\lambda_{\text{ins}} = \lambda_{\text{del}}$ for the remainder of this section, although iSGv2.0 does allow for λ_{ins} and λ_{del} to be different. $f(x)$ is the probability density function of indel lengths.

Simulating Indel Occurrence

As shown in figure 3, the maximum number of sites that potentially accept insertions is equal to the number of positions plus 1. Therefore, for an unconstrained sequence at node i with $N(i)$ residues, the number of sites that accept insertions, $N_{\text{ins}}(i)$, is $N(i) + 1$, whereas the maximum number of sites that potentially accept deletions, $N_{\text{del}}(i)$, is $N(i)$.

The expected number of residues at node $N(i + 1)$ is calculated as

$$E[N(i + 1)] = N(i) + BL_{i\rightarrow i+1} \times (N_{\text{ins}}(i) \times P_{\text{ins}} - N_{\text{del}}(i) \times P_{\text{del}}), \qquad (3)$$

where $BL_{i\rightarrow i+1}$ is the branch length from node i to node $i + 1$ and $N_{\text{ins}}(i) = N(i) + 1$, $N_{\text{del}}(i)=N(i)$. $BL_{i\rightarrow i+1} \times N_{\text{ins}}(i) \times P_{\text{ins}}$ and $BL_{i\rightarrow i+1} \times N_{\text{del}}(i) \times P_{\text{del}}$ are the expected numbers of insertions and deletions on the branch $i \rightarrow i + 1$, respectively. Note that in equation (3), each time an indel event occurs, $N_{\text{ins}}(i)$ and $N_{\text{del}}(i)$ fluctuate, in turn changing the expectation of future indel events for the remainder of $BL_{i\rightarrow i+1}$. For brevity, hereafter, we avoid using the node index (i) unless it is necessary. Thus, for example, we use N_{ins} instead of $N_{\text{ins}}(i)$.

With constraints as shown in figure 3, the number of sites available for insertion ($N^{'}_{\text{ins}}$) and deletion ($N^{'}_{\text{del}}$) is subject to the constraints $N^{'}_{\text{del}} \leq N_{\text{del}}=N$ and $N^{'}_{\text{ins}} \leq N_{\text{ins}}=N + 1$. Therefore, under these constraints, equation (3) becomes

$$E[N(i + 1)] = N(i) + BL_{i\rightarrow i+1} \times (N^{'}_{\text{ins}} \times P_{\text{ins}} - N^{'}_{\text{del}} \times P_{\text{del}}). \qquad (4)$$

Most current simulation methods do not correct for sequence-length fluctuation during the evolution with indels, which causes either the underestimation or overestimation of the number of events that will occur for the remainder of the branch as illustrated in figure 2. In Results and Discussion, we examine the consequences of these oversights.

Lineage-Specific Evolution

iSGv2.0 accepts guide trees in Newick format with clade labels. Specifying clades allows lineage-specific parameters to be set. The sequence parameters (character frequency, proportion of invariable sites, site rates, and substitution matrix) and indel parameters (maximum indel size, P_{ins}, P_{del}, λ_{ins}, and λ_{del}) can be changed among subtrees (clades). iSGv2.0 also provides a lineage-specific flag for a lineage evolving as a pseudogene. With this flag, all constraints to the sequence positions, that is, invariable array, positional γ parameters, and codon rates are removed. It causes the lineage to evolve with a uniform rate and unconstrained for indel events across all sites.

Functional Constraint Modeling

Site-Specific Constraints

iSGv2.0 introduces site-specific conservation using regular expression patterns found in PROSITE (Sigrist et al. 2002). The quaternary invariable array introduced in iSGv1.0 (Strope et al. 2007) is also retained because of its simplicity of representation. Because motifs are preserved along lineages, sites that correspond with the potential motif in the ancestral sequences still carry the length constraints of the motif, that is, motifs cannot be gained from insertions and potential motif sites cannot be deleted. Although indels are constrained on these sites, substitutions are freely accepted until the sites are accepted as the motif. When the site becomes a part of the motif, which occurs when the site is mutated into a motif-satisfying residue, the site becomes constrained based on the patterns specified in the motif. These constraints on a motif, by definition, cause a slower evolutionary rate within the motif region. Thus, iSGv2.0 compensates for the slower evolutionary rate by increasing the substitution rate in the partition that includes the motifs so that the resultant sequences will evolve at the expected rate, on average, based on the input branch length. For a motif with k characters, $m = m_0 \ldots m_{k-1}$, we calculate the average rate of substitution rejection at each motif site, $\hat{g}_n$, as follows:

$$\hat{g}_n = \frac{\sum_{i\in a_n}(1 - \sum_{j\in a_n} s_{ij})}{|a_n|}, \qquad (5)$$

where a_n is the set of acceptable residues for the motif position n, $|a_n|$ is the number of acceptable residues in the set, and s_{ij} is the probability of substitution of residue i to residue j (for the chosen substitution matrix and character frequencies) over the DES size ε. The term $(1 - \sum_{j\in a_n} s_{ij})$ is the probability of rejected substitutions from residue i. We then define $g_n = \sum_{i=0}^{k-1} \hat{g}_n$, the amount of reduction in evolutionary rate in the motif. Because the expected number of substitutions in an unconstrained sequence with N characters along the branch length BL is $N \times BL$, adjusting for the reduced evolutionary rate in the motif, we calculate BL' as follows:

$$\text{BL}^{'} = \frac{g_n + N \times BL}{N} = \frac{g_n}{N} + BL. \qquad (6)$$

Subsequence-Length Constraints

Protein family sequences are often composed of a set of domains that define the folding pattern of member sequences. Functional domains are often under length constraints whose violation could be detrimental to protein function, such as the destabilization of tertiary structure, improper folding, or removal of functionally important regions. iSGv2.0 represents these constraints through the introduction of a sequence “template.” The template specifies both the minimum and maximum number of residues or nucleotides that can occupy a region. Such constraints limit the number of insertion or deletion events. An example of sequence templates is given in supplementary figure S1, Supplementary Material online.

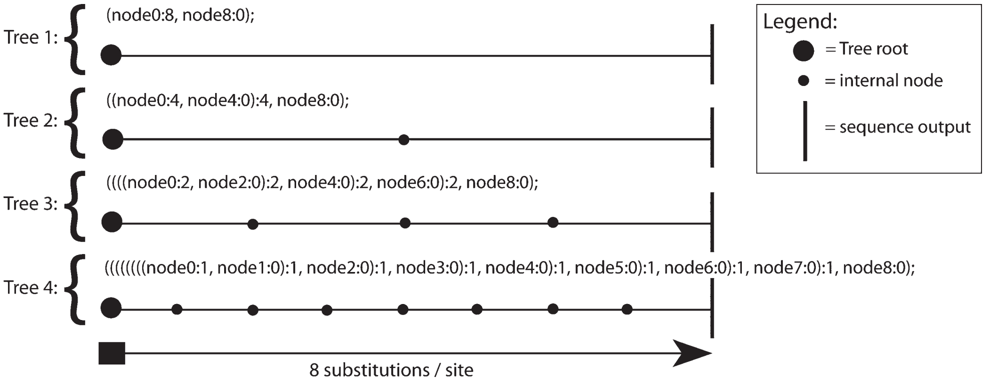

Tree 1:
(node0:8, node8:0);
Tree 2:
((node0:4, node4:0):4, node8:0);
Tree 3:
((((node0:2, node2:0):2, node4:0):2, node6:0):2, node8:0);
Tree 4:
((((((((node0:1, node1:0):1, node2:0):1, node3:0):1, node4:0):1, node5:0):1, node6:0):1, node7:0):1, node8:0);
Legend:
● = Tree root
• = internal node
| = sequence output
8 substitutions / site

FIG. 4.—The four simple guide trees and their corresponding Newick formats used to test indel simulation schemes. These have 0, 1, 3, and 7 branching points for Trees 1, 2, 3, and 4, respectively. Note that at each branching point (or node), one branch is given a zero length branch, as shown in the Newick format. The total length of the guide tree is set to eight substitutions per site. Branching points are named from "node0" (at the root) to "node8." During the simulation, sequences are saved at each internal node as well as terminal nodes and used for indel analysis.

Nucleotide Sequence Simulation

Coding and noncoding nucleotide sequence simulation has been added to iSGv2.0 using the substitution engine Seq-Gen (Rambaut and Grassly 1997). In the coding-sequence simulation, exons and introns can be specified as partitions. Exons are influenced by different rates for codon positions and restriction of stop codon formation. Introns can split codons (i.e., phase 1 and phase 2 introns). In introns, elements (such as the lariat formation sites) can be controlled by the quaternary invariable array and through motifs. Indel length distributions for exons can be set to zero for all lengths not divisible by 3 to avoid introducing frameshifting indel mutations. Although Indels in exons can be restricted to be between codons, such restrictions are not mandatory.

Event Tracking

One benefit introduced with the DES model is the ability to track indel events by the time of the events, affected taxa, event type (insertion or deletion), and indel length. The positions of indels can be reported in the true alignment. Along with the DES model, iSGv2.0 has added a new presentation method, "time-relative steps" (TRS). With the TRS presentation, the tree is rescaled relative to the time. The resultant tree is "ultrametric-like," having equal height (time) from the root to tips, which allows the mapping of all indel events with respect to the relative time of occurrence. Supplementary figure S2, Supplementary Material online, illustrates this TRS presentation. With the TRS presentation, events are reported as an ordered list based on the relative time of occurrence as shown in supplementary figure S2C, Supplementary Material online. When the TRS presentation is not used, events are listed by partition.

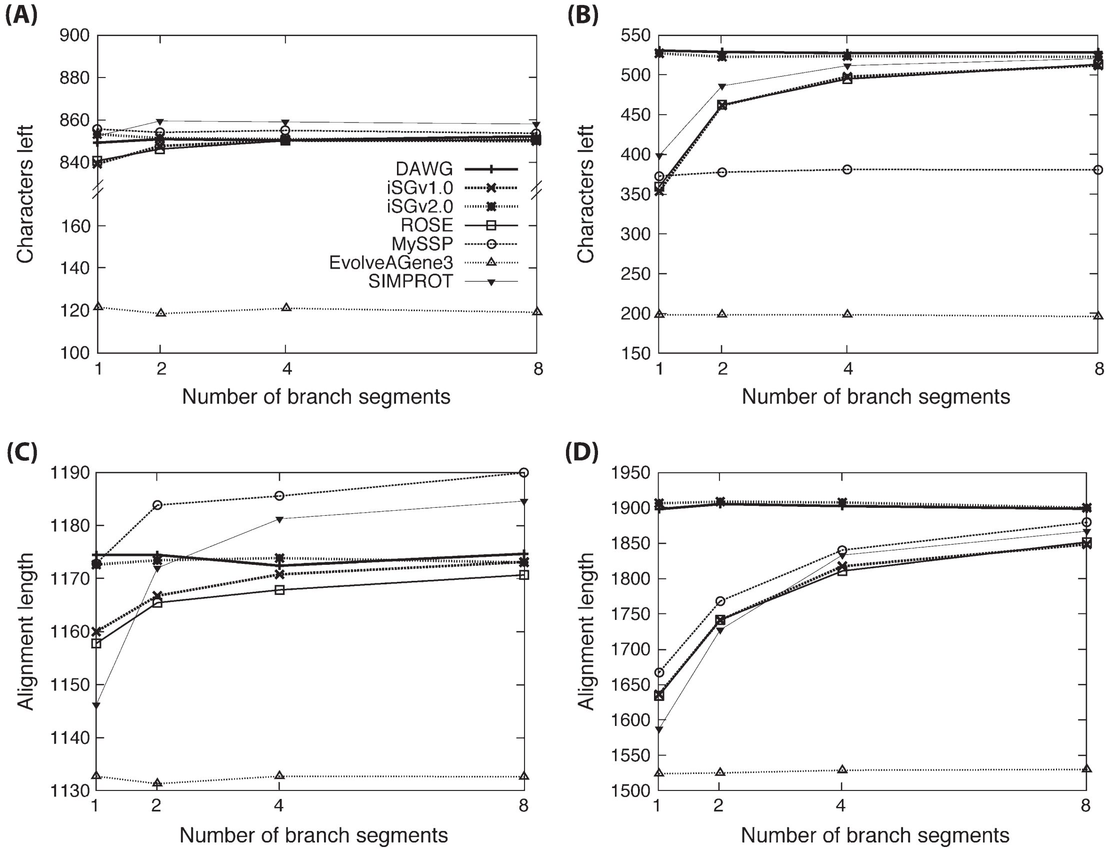
Implementation

iSGv.2.0 is written in ANSI C++. iSGv2.0 calculates substitution probabilities using the Seq-Gen (Rambaut and Grassly 1997) formulation. Only rooted trees can be used as guide trees. It has been tested on Linux, Mac OS X 10.4–10.5, and also on Windows XP running MinGW (http://www.mingw.org/), MSYS, and GNU gzip and tar. It is packaged using GNU autotools and should compile on most systems with a standard C++ compiler. The output can be in PHYLIP, Nexus, and FASTA formats. iSGv2.0 (executables and source codes) and its user manual describing the functionalities are freely available at http://bioinfolab.unl.edu/~cstrope/iSG/.

Indel Simulation Comparison

We compared seven indel-capable simulation methods, including iSGv1.0 and iSGv2.0. These methods are listed in table 1. Two tests were performed to examine the indel formulation of each method. In the first test, we analyzed the over and underestimation of insertions and deletions individually for each simulation method using guide trees with varying numbers of internal nodes. In the second test, a similar analysis was done, varying the relative rates of insertions versus deletions.

Note that the indel scheme implemented in EvolveAGene3 (Hall 2008) is very different from other methods. EvolveAGene3 simulates codon evolution based on empirical models obtained from *E. coli*. EvolveAGene3 calculates indel probabilities with two spectra: The first spectrum determines the event to take place, with probabilities to be 0.6284, 0.0744, or 0.2972 for a substitution, insertion, or deletion, respectively. In the second spectrum, EvolveAGene3 determines the indel length, rejecting any event that is not a factor of 3. For the second spectrum, we summed up the probabilities of all acceptable lengths to obtain single accepting probabilities for insertions and deletions: 0.144 and 0.261, respectively, and used them for each type of event regardless of the length. "Selection against deletions and insertions" were both set to 1 (no selection). Thus, with EvolveAGene3, the insertion and deletion probabilities are $0.0107 = 0.0744 \times 0.144 \times 1$ and $0.0776 = 0.2972 \times 0.261 \times 1$, respectively ("event probability" × "accepting probability" × selection against insertions or deletions). We also modified the EvolveAGene3 code to allow frameshifting indel mutations, in order



Standard deviations for each data point

	MySSP		EvolveAGene3		SIMPROT		ROSE		iSGv1.0		DAWG		iSGv2.0	
deletion-only:														
# branch segments	(A)	(B)	(A)	(B)	(A)	(B)	(A)	(B)	(A)	(B)	(A)	(B)	(A)	(B)
1	13.00	53.77	28.19	24.96	12.80	42.92	11.82	49.87	10.68	44.73	11.55	33.88	11.72	30.03
2	12.06	51.73	30.12	23.53	10.64	34.46	12.03	35.85	10.46	38.43	11.45	31.08	13.00	28.95
4	14.15	55.09	30.68	24.84	12.39	27.91	11.04	34.38	11.52	34.19	11.36	31.78	10.44	31.77
8	12.37	59.10	31.21	25.21	10.47	37.51	10.29	37.26	10.56	35.89	11.27	30.59	12.30	32.66
insertion:														
# branch segments	(C)	(D)	(C)	(D)	(C)	(D)	(C)	(D)	(C)	(D)	(C)	(D)	(C)	(D)
1	13.71	54.27	12.43	45.46	12.28	59.06	12.03	46.99	10.40	45.52	14.70	83.93	13.96	79.31
2	13.96	72.40	12.23	41.94	17.58	58.49	12.09	59.69	11.76	59.44	14.46	89.25	12.23	84.11
4	14.16	82.19	11.60	47.86	15.77	68.70	13.54	61.85	12.81	65.68	14.23	75.62	14.05	75.36
8	16.93	80.22	11.68	48.60	17.77	77.90	13.30	78.15	15.28	83.02	13.51	82.34	13.55	91.90

FIG. 5.—Comparison of indel simulation performance (Test 1) among seven methods. Correct simulations are expected to produce a plot with a horizontal line. Indel sizes used are (*A*) size-1 deletions, (*B*) size-4 deletions, (*C*) size-1 insertions, and (*D*) size-4 insertions. The *y*-axes show the number of characters left in the leaf sequence (*A*,*B*) and the true alignment length (*C*, *D*). The average values obtained from 100 simulations are plotted. The table below summarizes the standard deviations for each data point. Supplementary figure S4, Supplementary Material online, shows all test results in color.

to create similar indel-generation conditions with other methods.

Experimental Setup

For our tests, we set the total tree length to be 8 substitutions per site, with an indel rate of 1 insertion or deletion per 50 substitutions. For EvolveAGene3, insertion and deletion rates were 0.535 and 3.88 per 50 substitutions, respectively, as explained above. We simulated with random root sequences of 1,000 characters. DNA sequences were generated by DAWG, EvolveAGene3, and MySSP, and protein sequences were generated by ROSE, SIMPROT, iSGv1.0, and iSGv2.0. The difference in character sets (DNA or protein) is of no consequence in our tests for two reasons: 1) We used the same indel length distributions for both sets and 2) we set the probability of insertion and deletion occurrence equally regardless of the character set. Figure 4 illustrates the four simple guide trees with varied numbers of internal nodes. At each node, the external branch was set to zero length, as shown in the Newick format in figure 4, effectively making it a leaf node, so that the direct effect of the different number of nodes can be examined. We performed two tests: 1) simulating insertions alone or deletions alone and 2) simulating both insertions and deletions with varying relative rates. Test 1 is intended to show the effect of the indel placement paradigms of each sequence simulation

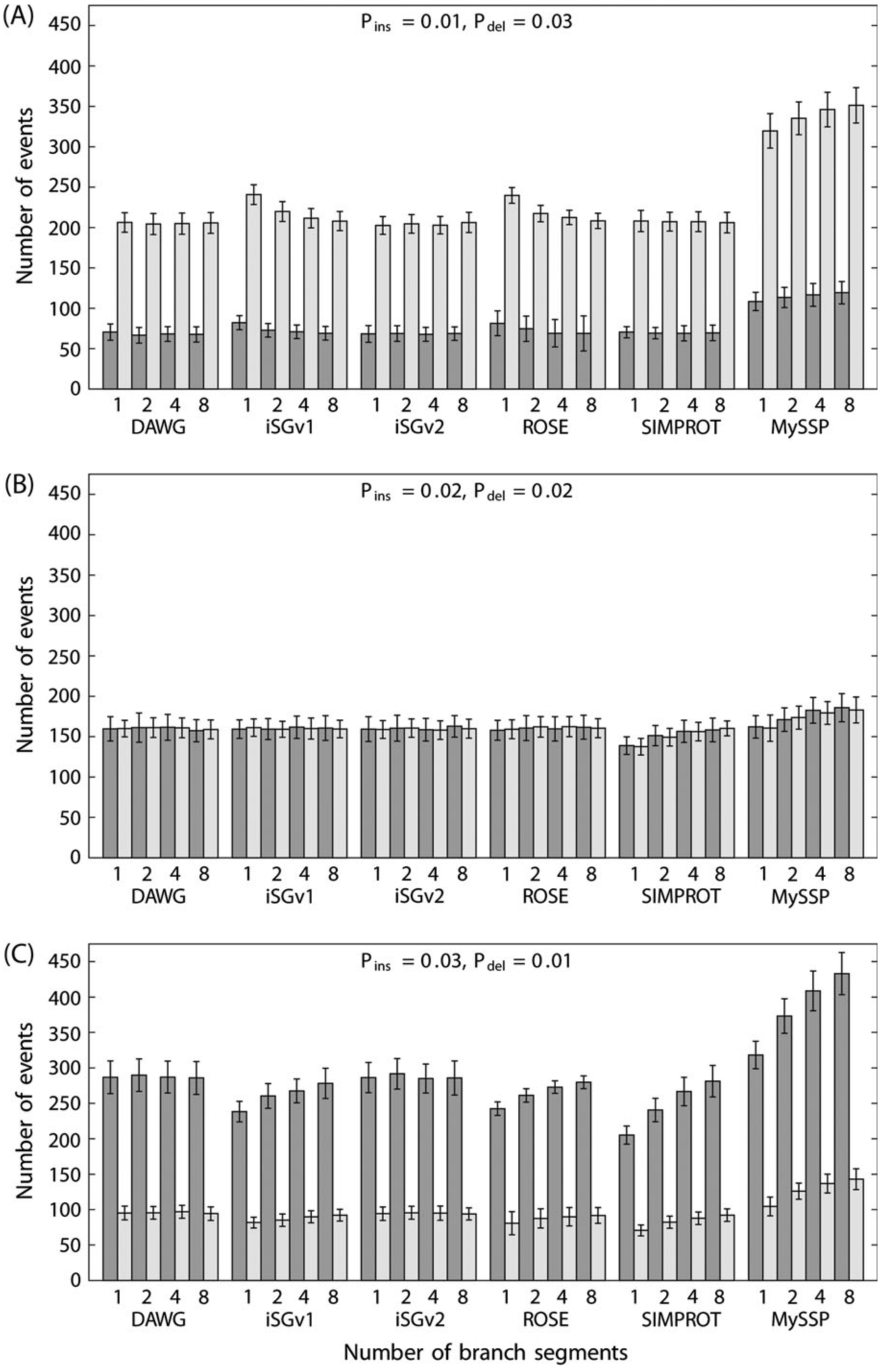


Fig. 6.—Test 2 results with different indel probability ratios. For each method, the total numbers of insertions (dark bars) and deletions (light bars) generated are shown for simulation experiments using the guide trees with different numbers of segments (see fig. 4). Note that for MySSP, we used the average expected value of the Zipfian distribution to obtain the results. For all methods, the average values obtained from 100 simulations are used.

method. Test 2 is intended to show the effect of the indel methods when both insertions and deletions are generated. If insertions and deletions are simulated properly, all simulation runs are expected to return similar numbers of insertions or deletions regardless of the number of internal nodes, because all four guide trees have identical lengths. Differing results between guide trees implies that adding branching points (nodes) affects the remainder of evolution, which is clearly undesirable.

Protein Superfamily Comparison

To present the sequence simulation capabilities of iSGv2.0, we simulated the calycin-superfamily proteins. We performed the simulation using iSGv2.0, ROSE, and iSGv1.0, and compared their simulation results.

Overview of the Calycin Superfamily

In the Structural Classification of Proteins database (Lo Conte et al. 2000), they belong to the "all-beta proteins" class.

Table 2
The Effects of Internal-Node Numbers with Varying Insertion and Deletion Rates among Methods[a]

	σ^2/μ											
	MySSP		SIMPROT		DAWG		iSGv1.0		iSGv2.0		ROSE	
	ins	del	ins	del	ins	del	ins	del	ins	del	ins	del
(A) $P_{\text{ins}} = 0.01$, $P_{\text{del}} = 0.03$	0.145	0.433	0.003	0.012	0.015	0.004	0.216	0.726	0.021	0.002	0.226	0.734
(B) $P_{\text{ins}} = 0.02$, $P_{\text{del}} = 0.02$	0.507	0.417	0.380	0.435	0.014	0.002	0.004	0.001	0.022	0.007	0.003	0.000
(C) $P_{\text{ins}} = 0.03$, $P_{\text{del}} = 0.01$	4.871	1.675	2.736	0.757	0.006	0.003	0.734	0.295	0.005	0.031	0.816	0.265

[a] ins: insertion, del: deletion.

The calycin superfamily consists of a number of beta-barrel protein families, such as the lipocalins, avidins, and metalloproteinase inhibitors (MPIs). As illustrated in figure 7, the lipocalin family proteins have three structurally conserved regions (SCRs 1, 2, and 3; Flower et al. 2000). Lipocalins are divided into two groups: kernel lipocalins, which contain all three SCRs, and outlier lipocalins, which contain at least one SCR. Each group of lipocalins is further divided into subfamilies. The avidin family contains a motif different from the lipocalins, whereas the MPIs have no motif recorded.

Experimental Setup

We sampled two sequences each from the avidins, MPIs, two subfamilies of outlier lipocalins, and four subfamilies of kernel lipocalins. The base alignment of these sequences was obtained using PROMALS3D (Pei et al. 2008) with manual adjustment. The alignment, along with the annotated beta-strands and motifs, is given in supplementary figure S1, Supplementary Material online. Figure 7 shows the phylogeny reconstructed using proml from PHYLIP version 3.68 (Felsenstein 2008). Based on the phylogeny, we determined the most likely scenario of motif gain and loss as follows: 1) SCR1 was gained before the divergence of the lipocalin family and subsequently lost in the alpha-1-acid glycoprotein lineage; 2) SCR2 and SCR3 were gained before the divergence of the kernel lipocalin family; and 3) the avidin motif was gained after the avidin family was diverged from the MPI family. SCR1 and the avidin motifs are obtained from PROSITE regular expressions PS00213 and PS00577, respectively (Sigrist et al. 2002). For SCR2 and SCR3, we gathered the motif alignments in the PRINTS database (the lipocalin family motifs 2 and 3; PR00179: Attwood et al. 1994) and calculated the percentage of each amino acid in each column of the alignment. The regular expression patterns were generated using amino acids with at least 5% representation as the acceptable residue set for each alignment position. Figure 7 lists these regular expressions. Using the guide tree shown in figure 7, we simulated the calycin-superfamily evolution. We specified the template for the input sequences based on the secondary structures of the sequence, and specified four motifs: SCR1, SCR2, SCR3, and the avidin motif. Because iSGv1.0 and ROSE do not have the ability to conserve specific lineages, we chose to conserve the motifs in the global invariable and the I + γ arrays, respectively. We conserved fixed-length regions using the invariable array option that forbids indels between sites and held sites with single acceptable states as invariable for all methods. All specifications are available in supplementary figure S1, Supplementary Material online. We simulated 100 data sets for each method. Supplementary figure S3, Supplementary Material online, gives the input files used for the simulations.

Indel Statistics

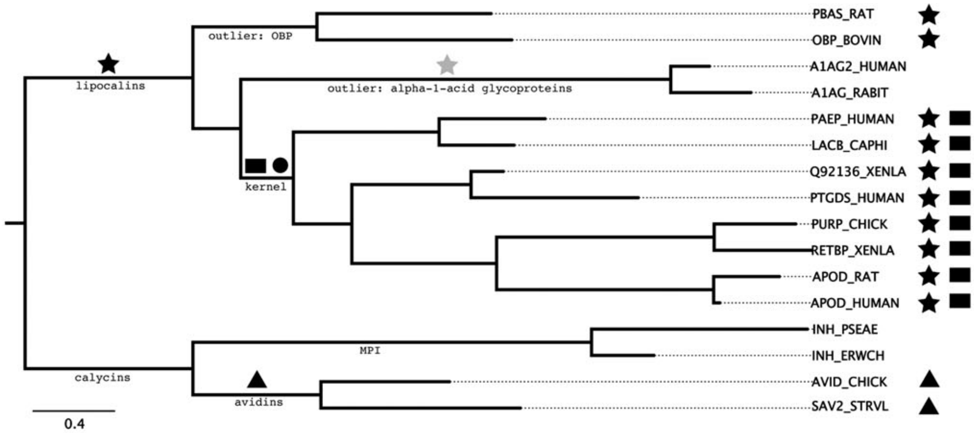
To test the minimum and maximum subsequence constraints (template) in iSGv2.0, we flagged insertions and deletions in ROSE and iSGv1.0 that broke minimum and maximum subsequence constraints. In order to do this, the template constraints needed to be introduced to each method. It was possible for iSGv1.0 by simulating within the iSGv2.0 template framework. However, we were unable to incorporate ROSE in the iSGv2.0 framework. For ROSE, to detect template-breaking indels, we inspected the true multiple alignment including ancestral sequences. From this alignment, we traversed all root-to-tip paths, examining the regions corresponding to the templated subsequences. When we found an unacceptable number of residues in a region, we counted one template-breaking indel. If the descendant sequences had a region shorter or longer than the corresponding template, we counted another template-breaking indel only if the indel pattern (gap columns) was different from the ancestral sequence.

Results and Discussion

Test 1: Insertions Alone or Deletions Alone

Our first test was to run insertion-only and deletion-only simulations. Indel lengths were fixed with 1, 2, 4, and 8 residues or bases. We measured the performance of each method by 1) the length of the true multiple alignment for insertions, where the number of sites inserted is equal to the alignment length minus 1,000, and 2) the number of characters remaining in the output sequence for deletions.

Figure 5 and supplementary figure S3, Supplementary Material online, show the test results. The number of internal nodes in the guide tree had an adverse effect on the performance of SIMPROT, iSGv1.0, ROSE, and MySSP (only in the case of insertions). As a side effect of their continuous modeling of indels, overestimation of deletions (fig. 5*A* and *B*) and underestimation of insertions (fig. 5*C* and *D*) are clearly shown with fewer numbers of internal nodes. These methods calculate the expected number of indel events without adjusting the sequence length when an event



★ *SCR1, Prosite PS00213:*
[DENG]-{A}-[DENQGSTARK]-x(0,2)-[DENQARK]-[LIVFY]-{CP}-G-{C}-W-[FYWLRH]-{D}-[LIVMTA]

■ *SCR2:*
[ILV]-[AILPV]-[ADEHK]-T-[DN]-Y-[DEK]-[NQEKST]-[FY]-[AILV]-[ILMFV]-[AQILMFV]-[CHLFY]

● *SCR3:*
[CILYV]-[LFY]-[AGSV]-R-[NDEST]-[QLKP]-[NDQEKT]-[GLPV]-[RNDLPS]-[ANDEGPS]-[DEPS]-[AEILTV]-[ILKMV]-[DQET]-[REK]-[ILF]

▲ *Avidin, Prosite PS00577:*
[DENY]-x(2)-[KRI]-[STA]-x(2)-V-G-x-[DN]-x-[FW]-T-[KR]

ROSE

```
                                                                     ★
OBP_BOVIN  -dgcnlldlcn--------------r-l--lsq------r------i---------dgvviseGFATQSLAGDWFGTgkgsa-svef-v-qplv--rpkdcffvafgegceqk-t-f-evqlpgthk-k----eg-rcnddagpgi
PBAS_RAT   -vfcklrslpk--------------y-g--l-k------aepfaggl---------esysttrGFEAEDLAGIWTGLashvp-nlnv-ftealg--r-svclpqespetcntv-a-f-geklagkvk-k----sn-nhqdktgtgv
A1AG_RABI  -rpevekk-kl--------------s-v--l-wcqg-aaa------iwalkc---dplvisrasaaasllagkwyglvrhdk-khlc-kgeh-g--r-dkcfyklsad----g-r-a-qngllsqav-g----ac-qcygklhddg
A1AG2_HUM  -lpqvdkk-kg--------------v-v--l-wcqgvlsa------vwamtc---dplvtetgsaavrllagkwyglarhdk-kflc-kgeh-g--rykkpfytlsaq----g-r-t-qksllshgm-g----rn-ecydkvddsg
LACB_CAPH  hstqsfqale---------------f-t--v-w------n------yygvgl--syaieeaesGLDRSHLPGLWLSQgahet-cgnm-fttt-t--sekdcffgmaag----v-g----vpllgsvv-e----va-tctsilyapr
PAEP_HUMA  hpgcifqalkp--------------l-t--v-w------h------afacel--sl---evitGLDTPHLPGLWLGLhadei-cgnm-yesn-a--tvkgcfynmaad----v-a-q-aiplksnlv-s----gg-tcqpilgspr
PTGDS_HUM  hg-----glir----------------------------l------kidgnc--arhtkvieeGLDVVHFSGKWYGFgaheh-qanfpllyc-katkknlclfglygm----l-v-f-kmnfirsql-s----dr-pcqeavgphk
Q92136_XE  hg-----glvr----------------------------l------midvnc--siegvkdidGLEVSHFPGQWYGFgshqs-radlpllvt-aakkhdmcffslfgk----l-a-f-qvkllgndl-d----cq-scqntvgtvk
APOD_HUMA  t------tler--------------i-vnfv-w------l------kahl-c--tgqagndgaSLIIDYAMGPWYGNgeyac-gsllekgpt-t--aatlcfavlngv----n-e-f-hirvvamhs-l----lt-gsgecvgaal
APOD_RAT   t------emkr--------------i-vdft-w------l------kahl-y--tgkagddgaSLIINYAMGPWYGAgeyac-gc-lekgpt-t--tctlcfsvlnkq----n-e-lttirvialqa-l----lt-gageclgpal
RETBP_XEN  le-----ilgk----------------q--v-w------c------agsa-g--ehlptsshaGLRGSYLKGKWYAKgknhc-pvkvtepae-q--arglcfyslkak----vfi-f-glqvtsstn-efgrgqt-lcpncespst
PURP_CHIC  lg-----ilgr--------------f-q--a-w------s------igsk-e-qengpgtsqqGPRVKYLKGKWYEKgqdtc-hvsvpspk--q--trglcfiklkaa----l-a-f-glkltgpss-gfgrgtk-hcpgcgskss
SAV2_STRV  -ttc-k-lllg----ayvnidraqdv-l--m-g------i------ytrsqh--g-------gesgeklvngkwmeliegpy-qqri-ivat-p--aveaqfniirlg----k-g-v-ytdiavtrggt----me-hsaagkgapg
AVID_CHIC  -sdc-e-ltlp----qfl--qsppds-l--l-g------v------ncyshr--d-------gegnaeflkgrwfkevsppa-hgrc-ivas-c--avdysffiisas----n-g-g-yaqiavergas----dn-mckqgrggph
INH_ERWCH  -lvf-g-mevgrchg----------l-l--l-e------v------hcnhlaa-sanvqa-dqglegdtvdgrwyrerkpfsasvel-gkks-c--nydeclfslppi----s-gag-yihlknkhnti----isltcghnkvagd
INH_PSEAE  -eif-n-kdfgkcqe----------ltl--l-n------s------rkshvs--ssngac-hksvtsetvdgdwlsdkepfaaycel-gkqe-c--acddclfylspf----g-a-v-yielkvqtdlt----gplrcthgrvpge

                                                ■                    ▲           ●
OBP_BOVIN  lg-k-----pengkpnq-v---g--yqnaffdllthynaffdvysiekeynhkqhdvkvgtsmlartvelsdaplgsmtrkkqnvvdpq----ypkdqy-slar-----e-vik
PBAS_RAT   id-a-----rpagtkge-d---s--lksdffmlytqydsylalyrcqatcvhrnceiavgartltrgpeqqqgslakflrnlqcmlgep----klyeklakisk-v---v-tin
A1AG_RABI  le-p-------lhlndtkf---g--klrdpdnkytdyskfflfkwcsgkefesrpqltvglqgscrvaqmhqgrlgrfvrltkppyltk----ahsllk-cag--------nad
A1AG2_HUM  ik-d-------lnladtkv---g--elwqpnnkytdyskfflfkgcpgryfekrsdltvglagscrvaqlqqghlgrfvkllevpygfr----thsllk-cas--------dvn
LACB_CAPH  vr-e-----ktvelgvkyg---i----phdyMLYTHYDSYYLVHtcrfkyntaavaitvgshKLARMSSITEGDVATFvkllvlklas-----niqvnd-natk-d---a-dls
PAEP_HUMA  mm-g-----klvklaveyg---g----nksyDLYTHYHSYYLVHacrfqydtnavsitvgssSLARLPARSDAQVGKFvaflilkltt-----kmlgfd-gtsk-q---t-nls
PTGDS_HUM  yl-r----lvvtqltvhrp---s--pdlgkyALLTHYEMYYYTYscravyqlveaciqvgkaALARVGGPTAGVLGLFgqlelerla---------gge-hplp-nkg-i-eas
Q92136_XE  ya-k---glkggelsvrnp---s--pnqgiyELFTHYNKYYYPYhckakyeinktaiqvgtsKLARIDGVSEGQLGLFgqlligrls---------iag-ppvd-s---p-dts
APOD_HUMA  lr-n-----keapfnigyd---p--gggsnyKVPTHYNQIYYPIeckavfmmnvaapvvgasALTRVEADLHGQTAGTtrilidkalga----dlapar-gark-p---g-tpe
APOD_RAT   lr-n-----kaepfnigfk---p--ggssdyKFPTDYNEIIYPVecdvvftsnvaapavgvsALVRVEANLHGQTAGTarilmdktaac----dqaavl-gakk-p---g-tpe
RETBP_XEN  ve-r-----tpeeanvrte---e--qqkadpKRPTHYKVTYSPYtcsvdfpdarvgnvvgpnEVSRRSYIDEGKLDQYtsapeastnaq----whwrpt-gaggpf---i-rle
PURP_CHIC  vf-a-----cpaqlhvrhn---q--eqkktpKRSTGYRVIYAPFtcavefakvriqavvqasEVDRNAYIADGKLGQYgsaleavttgd----thslpv-gsns-a---i-rle
SAV2_STRV  ka-a-yf--epvafaahfd-hgt--stepiwmtktlydsyyfvyach-k-gHQSCVKMVGEKTLNRgtglagstlnkfehklrel-vva----ssrtgk-pqea-k--tp-elt
AVID_CHIC  ktav-yl--hpilieltys-hgslvisgavwmmqtdydsytflyhcs-e-hNHHCSDIVGDKVLSRgsgknsetiqkfivgakel-ieg-npvpliepk-htqp-r--kp-aar
INH_ERWCH  qk-tn-k--avqlvyddqvpvtl--pmsvpwsdktayekhaymysck-a-chpncaqavggivltrspeilgenvakfeqsrtalkgdd----phlerr----n-g--apdyvh
INH_PSEAE  st-v--k--pidlasdeei-fql--slergwayattyekrpyiyace-v-chiscvdlvgdtkltreqqffgeviakfpgpassls-eav---kgvkkk-lhik-d--pssaae
```

iSGv2.0

```
                                                                   ★
INH_PSEAE  c---kdwt-------t---ll-glfc-gaep----vgeiwspsk---ssncgns--------hkv-idlasnwfappge-ngvvi--qsavemithtalafqa--vnhetyn--tlipttf-vake-gelvga-ssfepapni--
INH_ERWCH  p---rkwvq------t---ll-glfc-gatp----pgaawaphr---kpdngld--------mklhlemsgnwfgpmts-qssge--ppgievgslfpviyra--ikqevag--v-lpdsl-iakd-gelvil-hsypassds--
AVID_CHIC  v---ssanl------iltqll-glfc-gark-------acrt-pppwtt-sapv--------qrldw-tqenwynsfev-dpead---gdeqttv-rtlffev--lneedld--l-dptns-hanspttna---iss-lnlgs--
SAV2_STRV  m--------------vltlll-glfc-gakk-------agsq-t---ap-tvpdrgsggfgthhidw-vrtkrasafpa-ngdke---vdeqatk-rpmlfek--msdqdyg--k-ippse-yens-rggkfvntlg-verie--
PURP_CHIC  i---------hqfmylqrgllgglfc-gaqg-------lkgakk---vp-nNQN----------PSR-ILGLWHGMdav--vtgr---e------ry-palvv--akhlfevrtt-tsvpe-ssgr-he--imn--e-gkqrr--
RETBP_XEN  i------lglhelmflergllwglfc-gaqg-------v------------NQD-----------NR-ILGLWRGMsav-vdtse---e------rq-galel--akehrnivns-vsvte-gtgq-rd--ims--g-akqkr--
APOD_RAT   l------iasgqelvgffqll-glfc-gaae-------atva--------eNLD----------NAR-VLGVWHAVayl-qrlae---e------lr-vmfqi--dapcyag--i-mlfgt-esge-dh--iqr--g-gqprk--
APOD_HUMA  i------stlsqllvavfqll-glfc-gatk-------atvtap---lp-eNLD----------RTR-VFGVWHLIeyi-qrlge---e------lr-viski--dapsyag--v-mlsgl-etgr-en--iqr--g-tqhkk--
Q92136_XE  s------vr------vlhtll-glfc-gaec-------l------------NDD----------LAR-IAGNWYYMgkv-pagak---l------tg-avltq--sdeqycs--a-glage-gteq-dt--saqekg-iktkrhg
PTGDS_HUM  s------fs------vnfnll-glfc-ganc-------m------------NDD----------LAR-VGGKWYCMhli-ppgtr---v------vs-eilln--sdelyca--e-tlave-htmh-dg--napekg-stmtr--
PAEP_HUMA  fvtktilaw------ii--ll-glfc-gady-------nlsllp---ii-sDQQ----------SDK-LEGNWYWAlfm-spega---rskvarhlk-ashdt-fevdelck--l-llfgs-htna-gt--layqhi-tskar--
LACB_CAPH  f---lilan------laflll-glfc-gadc-------tlslpk---it-iDEQ----------SGK-VVGNWHWVahm-saagk---wsraachmr-admdi-feadllct--l-plfgd-havp-nd--mafqad-tgqlh--
A1AG2_HUM  l---ictiw------rtqgll-glfc-gaea--lk---tsavda---it-qgle----------liq-irgewhrmsqn-a-ivaea-fqmpelk-k-vlakl--pnstnci--l-flfas-kgrd-ga--tksaig-krrak--
A1AG_RABI  l---icaiw------ktagll-glfc-gaea-------psavec---ip-ngle----------liq-vkgewhrlsqn-a-ilpde-lqmpelk-k-fltkl--psdtkcn--l-cleag-kspd-ga--sklait-grrlk--
PBAS_RAT   s---tcmiw------lvlmll-glfclgaaa-------likvpi---vt-dNPA----------LPE-LVGLWHMLamasvrdgi---lptleasmn-lvleaq-apqlvvd--t-dlagpkkgem-ws--gpekng-lnkkr--
OBP_BOVIN  a---lcivw------villll-glfc-ga-cry-----vcdmpk---tn-eNSQ----------VRQ-LSGLWHKMiftha-ttp---pv---shmk-feltl--gpeddnp--m-eyfgl-pseq-sf--gpklng-grdrn--

                                            ■                       ▲           ●
INH_PSEAE  aiee-ekskgpgh-smssyqqaptwvtemihfr---lklfkiik-v-ekilekh-klldlsapnygaditlsdfglhslcvnkl--nsg--------------sn-----asagq----ivk--s------a-cwgl--gtivv
INH_ERWCH  lier-ecptppgy-haid-qkapswiteivsgd---lqlyivmr-i-krvqkrr-keldleymnsgpaltlsgftdhvrcaakl--fns--------------rn-----ltegq----tin--s------s-cwgl--q----
AVID_CHIC  s--k-eyqtssat-aada-kksaawlhledvtd---ygf-ryiqnv-gqYMIFA-EENGVMLFTRPpalgtgglsnldksqdglt-mae--------------kq---------n----vqedpq------y-cpvi--d----
SAV2_STRV  p--r-dystgpah-nirc-mtpmswlmradvhl---ysl-ryvnpv-grYLAFT-DENGVIGITRCspygcgglealdkaprglw-ldd-----htdetfgttve-----g---n----dqddtq------f-crml-------
PURP_CHIC  s--i-gfdalted-dr---ppsqfwVISTDYKQ---FSS-LYwleira-sgeks-ysedepLLGENPNLPEWAVDERmntfkdkl-hht----n----------fn-----s---q----htenen------a-e--e--i----
RETBP_XEN  a--v---qaqtae-dg---vkkqfwIISTDYKQ---FPP-FHwcal-a-tsekd-hyeddpVLGEWPDLLKWKVDKTlnaykgki-fkk----a----------ls-----l---p----ktdtsn------p-g--e--n----
APOD_RAT   s--lgnfrksmpa-ngar-pnrqnwVILTNYEE---YLF-FFwpad-d-tpvlv-hgatgkIYGEEQEVLKHAMTKLgkffkdrm-rdagirst---------gt-----l---n----rtqedq-------------k----
APOD_HUMA  i--lgnfrkalpa-ngar-pnrenwVILTNYEE---YLS-FFwpad-d-ppvli-tgatgeVLGEELDVLKHAMTKLlkffkdrl-rda----t----------se-----l---d----rtdedk-------------t----
Q92136_XE  ---i-atkssqqn-nrse-c-kkywILETNYDH---YCV-MYlaks-r-yamkd-egarepVFSEKPEPLQRHVDKGpvalnlkg-a------g----------lt-----m---g----etkeer------q-f--s--i----
PTGDS_HUM  ---m-aikvseht-ssrn-cakkcwILETNYEV---YCV-MY-aep-r-matge-ngtgepIFSEEPEPPHYHVDKGlcsvnlrg-a------g----------at-----v---g-------enc------r-n--n--v----
PAEP_HUMA  t--n-psrvsrsnldrke-dtkkwwLIKTNYEQ---FVV-FYlnsa-k-dsnga-cgygkeLLAEKPNGLYLQLDKStntykge-pdda----i----------li-----i---r----rsdgdggmkgknhqt--l--l----
LACB_CAPH  t--h-pmltphtklnkqe-dtrkwwILATDYED---FLL-FYitrn-k-ypnsa-sgrgkaLLGEEPNTRHLGLEKVlgtikgegpdda----g----------ltctslwv---l----kvdgeg------yqt--qgdq----
A1AG2_HUM  k--l-nnssigkd-rt-k-sggstdivvnkity---ekl-yfvcrd-r-ccpvqtnsgpdimvtdgrdvlqkaaqfvlvqwglvs-ak---------------------------ttmsvvreqag------n-i--l--f----
A1AG_RABI  n--i-ndsiigkd-rkag-sgnsaevvlnklas---dql-yyvnrd-r-cepap-gelldiklkehrdvl-aqnqfamitwgrvp-tk---------------------------tm--vvreqar------n-l--v--i----
PBAS_RAT   l--l-lldtetic-dqce-kvgqcwindtnwnepegygp-charrt-k-chkgk-mecgeiklcykldnflaqledmlkgyqqgsi-le---------------------------p----dtpr------------l--s----
OBP_BOVIN  m--v-pettstae-qkld-gkykglmldiffhe---ce--fykrel-k-ehlgr-rdggvylcmkgpdvvavdlgdmikgvqvrm-ln---------------------------g----aitk------------m--a----
```

occurs. DAWG, iSGv2.0, and EvolveAGene3 show no or very slight effects in indel numbers. For DAWG and iSGv2.0, this is because sequence lengths are adjusted dynamically along the branch. The unaffected results by EvolveAGene3 are likely due to the fact that this method treats branch lengths as the number of mutation-event tests that occur along a branch. For our purpose, we set the branch length to 8,000 mutation-event tests (1,000-character sequence with each site undergoing eight substitutions). EvolveAGene3 also forbids overlapping insertions and deletions, effectively reducing the deletion rates with larger deletions. This effect can be seen in supplementary figure S3 (I), Supplementary Material online, where "more" characters are left after simulations with larger deletion sizes. Because of the constant insertion and deletion probabilities set in EvolveAGene3 and our removal of codon constraints in EvolveAGene3, the lengths of the alignments are often shorter than other simulation methods.

These results show that iSGv2.0 and DAWG performed appropriately, producing consistent results regardless of the number of internal nodes. EvolveAGene3 also behaved appropriately according to its own indel model. iSGv1.0, ROSE, SIMPROT, and MySSP are all affected by the number of internal nodes, producing artificially high or low rates for deletions or insertions, respectively.

Test 2: Including Both Insertions and Deletions with Various P_{ins} and P_{del}

To further examine the effect of indel models, we simulated both insertions and deletions using the Zipfian distribution (Chang and Benner 2004) with five methods: DAWG, ROSE, SIMPROT, iSGv1.0, and iSGv2.0. We simulated three scenarios: 1) $P_{\mathrm{ins}} = 0.01$ and $P_{\mathrm{del}} = 0.03$, 2) $P_{\mathrm{ins}} = 0.02$ and $P_{\mathrm{del}} = 0.02$, and 3) $P_{\mathrm{ins}} = 0.03$ and $P_{\mathrm{del}} = 0.01$, where P_{ins} and P_{del} are the number of insertions and deletions per substitution, respectively. We chose to use the Zipfian distribution because it is an empirically determined length distribution for insertion and deletion events. For MySSP, which implements a length distribution that is normally distributed based on the mean indel length given by the user, we used the expected indel length of 2.082, which is based on the Zipfian distribution with a maximum indel size of 10. EvolveAGene3 was excluded from this test because changing P_{ins} and P_{del} fundamentally alters the indel creation method in EvolveAGene3. In this test, an event counter reporting the numbers of insertions and deletions that occurred during the simulation was added to each method. Because we were unable to obtain the source code for MySSP, we calculated the number of events as follows: Each gap in the root sequence in the true multiple alignment is the effect of an insertion in the descendant sequences, and likewise, each gap in the tip sequence is the result of a deletion in the ancestral sequence. To obtain the number of insertion and deletion events, we tallied the total number of gaps in the root and tip sequences, respectively, and divided that number by the mean indel size. We measured the quality of indel simulation by comparing the numbers of insertions and deletions generated. We calculated the coefficient of variation (σ^2/μ), which is dimensionless and makes results from different simulation methods comparable. If $\sigma^2/\mu \approx 0$, it means that the simulation method behaved similarly between the guide trees (no effect of different number of nodes). A larger σ^2/μ suggests that the simulation method performed differently between the guide trees with different number of nodes.

Figure 6 and table 2 show the results of this test. As expected, the number of insertions and deletions generated is affected by the insertion and deletion probabilities. For iSGv1.0 and ROSE, when $P_{\mathrm{ins}} \neq P_{\mathrm{del}}$, the number of internal nodes affects both the numbers of insertions and deletions. SIMPROT, as a result of their multiple-hit correction feature, shows much more drastic effects with the internal-node numbers than iSGv1.0 and ROSE when the insertion rate is larger than the deletion rate ($P_{\mathrm{ins}} = 0.03$ and $P_{\mathrm{del}} = 0.01$). Such effects are not shown when the deletion rate is larger ($P_{\mathrm{ins}} = 0.01$ and $P_{\mathrm{del}} = 0.03$). MySSP shows increasing numbers of indels for each test, with the most drastic change occurring when $P_{\mathrm{ins}} = 0.03$ and $P_{\mathrm{del}} = 0.01$. This behavior can be better understood using the results of Test 1, where in the case of insertions, the sequence length grows as more internal nodes are added, but the sequence length is stable for deletions under the same conditions.

Table 2 summarizes the degree of variation (σ^2/μ) in the numbers of insertions and deletions among Test 2 experiments. For iSGv1.0 and ROSE, we note that the variation in the number of insertion and deletion events is higher trending toward the dominant event-probability as shown in figure 6, whereas MySSP shows high variability in all tests, although it is most pronounced when the relative insertion rate is high. SIMPROT also shows high variation when $P_{\mathrm{ins}} = 0.03$ and $P_{\mathrm{del}} = 0.01$ or $P_{\mathrm{ins}} = P_{\mathrm{del}}$, although it is comparable with iSGv2.0 and DAWG when $P_{\mathrm{ins}} = 0.01$ and $P_{\mathrm{del}} = 0.03$. When $P_{\mathrm{ins}} = P_{\mathrm{del}}$, the indel models of iSGv1.0 and ROSE appear to be affected very little by internal-node numbers. Note that when $P_{\mathrm{ins}} = P_{\mathrm{del}}$, iSGv2.0 and DAWG show slightly larger σ^2/μ values than iSGv1.0 and ROSE. This is a consequence of the larger number of steps with indel events and sequence-length evaluations performed by these methods. Simulation as a random walk increases the variance in sequence length at each step. DAWG and iSGv2.0 take much larger numbers of steps (600 and 1,024, respectively)

←

FIG. 7.—Simulation of the calycin protein superfamily using ROSE and iSGv2.0. The phylogeny at the top is the guide tree used for the simulation. The signature motifs for each protein sequence (SCR1, SCR2, SCR3, and avidin) are listed by the UniProt protein IDs. Where the motifs are gained or lost are illustrated on the tree by black or gray symbols, respectively. The regular expression patterns defining these motifs are listed below the tree. The input specification data used for these simulations are given in supplementary figure S3, Supplementary Material online. In the output alignments, lowercase letters indicate nonmotif positions, whereas uppercase letters belong to motifs. Each motif is boxed in the alignment, and the identity of the motif is given by the corresponding symbols at the top of the alignment. See supplementary figure S5, Supplementary Material online for the results including iSGv1.0.

Table 3
Performance Comparison among iSGv1.0, ROSE, and iSGv2.0 for the Calycin Superfamily Simulation[a]

	iSGv1.0	ROSE	iSGv2.0	Input Alignment
Percent sequence identity	19.78	17.72	14.62	15.65
Percent motif positions conserved	61.82	61.88	80.98	—
Number of template violating indels[b]	13.18	19.91 [d]	0	—
Number of rejected indels[c]	NA	NA	0.38	—

[a] Statistics for iSGv1.0, ROSE, and iSGv2.0 are averages from 100 simulations.

[b] The number of indels that produced subsequences that were larger or smaller than the maximum or minimum values given by the template specified for iSGv2.0.

[c] A rejected indel occurs when a scan of the sequence returns no positions in which an indel can be placed because of subsequence size constraints imposed by the template. NA: not available.

[d] Approximate values inferred from the sequences as given in the true multiple alignment including internal nodes in the guide tree.

compared with at most eight steps taken by all other simulators. We confirmed this result by varying the number of steps taken by iSGv2 (data not shown).

Example Application for a Protein Superfamily Simulation

Figure 7 shows examples of "true alignment" output obtained from ROSE and iSGv2.0 (see supplementary fig. S5, Supplementary Material online, for the output including iSGv1.0). As shown in table 3, iSGv2.0 correctly conserved 80.98% of the sequence positions, whereas iSGv1.0 and ROSE, both of which cannot conserve sets of characters, correctly modeled only 61.82% and 61.88% of the positions, respectively. Of the different motifs, iSGv2.0 perfectly conserved all sites of SCR1 (see fig. 7). This is because this motif was present in the root alignment and conserved from the beginning of the simulation run along the lipocalin lineage. For other motifs, all substitutions were accepted until the motif became effective later in the tree, after which only substitutions conforming to the position-specific constraints were accepted.

We observed multiple side effects due to the restrictions imposed by the quaternary invariable and $I + \gamma$ arrays of iSGv1.0 and ROSE, respectively. As seen in figure 7 and supplementary figure S5, Supplementary Material online, conserved motifs in the multiple alignment appeared as "islands" where indels were absent. Additionally, invariable sites such as the GXW region of SCR1 were conserved for the entire column of the alignment, despite the fact that it should only be conserved among kernel lipocalins and the outlier lipocalin family of odorant-binding proteins. iSGv1.0 also simulated fewer indels, which is a side effect of the number of alignment positions that contain motif positions, reducing the number of accepting positions for indels. It appears that ROSE uses the absolute number of residues in the sequence to calculate the overall probability of an indel for a branch, regardless of the number of non-accepting sites for indels. Differences in the indel placements between iSGv1.0 and ROSE versus iSGv2.0 are also evident. As shown in figure 7, iSGv2.0 has a much higher number of indels along the N- and C-terminal regions of the alignment. This is because these regions had only weak constraints on their sizes: The N-terminal was constrained to 10–43 residues and the C-terminal 10 to 30 residues. During the simulation process, iSGv2.0 determined the size of the indel, and based on both template and motif constraints searched the sequences to find regions that could accept the indel. Most of the larger indels tended to fall in the least constrained regions. Because neither iSGv1.0 nor ROSE has such constraint capabilities, indels were placed wherever they were not forbidden by the quaternary invariable and $I + \gamma$ arrays. Furthermore, the superfamily fold could not be modeled by either iSGv1.0 or ROSE. They placed an average of 13.18 and 19.91 template-breaking indels, respectively (table 3). iSGv2.0 upheld the template restrictions. On average, 0.35 indels per simulation run were rejected using iSGv2.0 because there were limited acceptable positions for indels due to template constraints.

The input multiple alignment (supplementary fig. S1, Supplementary Material online) had an average pairwise sequence identity of 15.65% (table 3). The 20–35% range of sequence identity or lower is the so-called "twilight zone" of sequence identity (Rost 1999), which is often seen among proteins belonging to highly divergent families. iSGv1.0, ROSE, and iSGv2.0 simulated data sets in this range, with the average values over 100 runs of 19.78%, 17.72%, and 14.62%, respectively. The difference in sequence identities between iSGv1.0 and ROSE versus iSGv2.0 is explained by the global conservation of invariable sites by iSGv1.0 and ROSE, even for sequences without the lineage-specific motifs. iSGv1.0 and ROSE both showed a lower "percent motif positions conserved" (table 3) indicating that some positions were conserved by them even if they did not conform to the residue constraints for different protein families.

Conclusion

Good sequence evolution simulation requires not only realistic event simulation through substitution, insertion, and deletion but also needs realistic constraint enforcement and heterogeneous evolution among between subsequences and among subtrees. In this study, we showed that although many of current simulation methods introduce insertion and deletion events, only iSGv2.0 and DAWG have robust models. We introduced a formal model of functional constraints on substitution and indel events. We improved the modeling of sequence evolution by fixing indel evolution, incorporating novel functional constraints for motif conservation and subsequence-length preservation, and improved heterogeneous sequence evolution and lineage-specific evolution. iSGv2.0 also added modeling of coding and noncoding DNA evolutions.

We showed that the majority of indel-simulating programs incorporate indel models that do not account for sequence-length variations during the branch evolution. They introduce bias into the results of the sequence simulation, although such biases are not evident when insertion and deletion frequencies are equal. We also showed that adding subsequence-length constraints and motif constraints allows iSGv2.0 to correctly model superfamily evolution in the twilight zone of sequence similarity.

Supplementary Material

Supplementary figures S1, S2, S3, S4, and S5 are available at *Molecular Biology and Evolution* online (http://www.mbe.oxfordjournals.org/).

Acknowledgments

We would like to thank Dr Elisabeth Tillier (University of Toronto, Ontario Cancer Institute) for incorporating the functionalities necessary to perform our tests into SIMPROT and also for her comments. We are also grateful to Dr Steve Kachman (University of Nebraska) for his input. We thank the anonymous reviewers for their productive comments.

This work was supported in part by National Science Foundation AToL grant 0732863 to E.N.M., the Department of Education grant P200A040150 to the Department of Computer Science and Engineering at University of Nebraska—Lincoln (UNL), and by an Undergraduate Creative Activities and Research Experiences (UCARE) grant to K.A. from UNL.

Literature Cited

Attwood TK, Beck ME, Bleasby AJ, Parry-Smith DJ. 1994. PRINTS—a database of protein motif fingerprints. Nucleic Acids Res. 22:3590–3596.

Bradley RK, Holmes IH. 2007. Transducers: an emerging probabilistic framework for modeling indels on trees. Bioinformatics. 23:3258–3262.

Cartwright RA. 2005. DNA assembly with gaps (Dawg): simulating sequence evolution. Bioinformatics. 21:iii31–iii38.

Chang MSS, Benner SA. 2004. Empirical analysis of protein insertions and deletions determining parameters for the correct placement of gaps in protein sequence alignments. J Mol Biol. 341:617–631.

Chivers PT, Laboissiere MC, Raines RT. 1996. The CXXC motif: imperatives for the formation of native disulfide bonds in the cell. EMBO J. 15:2659–2667.

Dayhoff MO, Schwartz RM, Orcutt BC. 1978. A model for evolutionary change in proteins. In Atlas of protein sequence and structure. Washington (DC): National Biochemical Research Foundation. Vol. 5, p. 345–352.

Edgar RC. 2004. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinformatics. 5:113.

Felsenstein J. 2008. PHYLIP (Phylogeny Inference Package) Version 3.68. Distributed by the author. Department of Genetics, University of Washington, Seattle (WA).

Flower DR, North ACT, Sansom CE. 2000. The lipocalin protein family: structural and sequence overview. BBA. 1482:9–24.

Hall BG. 2008. Simulating DNA coding sequence evolution with EvolveAGene 3. Mol Biol Evol. 25:688–695.

Hasegawa M, Kishino H, Yano T. 1985. Dating the human–ape splitting by a molecular clock of mitochondrial DNA. J Mol Evol. 22:672–677.

Henikoff S, Henikoff JG. 1992. Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci. 15:10915–10919.

Jones DT, Taylor WR, Thornton JM. 1992. The rapid generation of mutation data matrices from protein sequences. Bioinformatics. 8:275–282.

Lassmann T, Sonnhammer E. 2002. Quality assessment of multiple alignment programs. FEBS Lett. 529:126–130.

Lo Conte L, Ailey B, Hubbard TJP, Brenner SE, Murzin AG, Chothia C. 2000. SCOP: a structural classification of proteins database. Nucleic Acids Res. 28:257–259.

Notredame C. 2007. Recent evolutions of multiple sequence alignment algorithms. PLoS Comput Biol. 3:1405–1408.

Pang A, Smith AD, Nuin PAS, Tillier ERM. 2006. SIMPROT: using an empirically determined indel distribution in simulations of protein evolution. BMC Bioinformatics. 6:236.

Pei J, Kim B-H, Grishin NV. 2008. PROMALS3D: a tool for multiple sequence and structure alignment. Nucleic Acids Res. 36:2295–2300.

Qian B, Goldstein RA. 2001. Distribution of indel lengths. Proteins. 45:102–104.

Raghava GPS, Searle SMJ, Audley PC, Barber JD, Barton GJ. 2003. OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy. BMC Bioinformatics. 4:47.

Rambaut A, Grassly NC. 1997. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Bioinformatics. 13:235–238.

Rosenberg MS. 2005. MySSP: non-stationary evolutionary sequence simulation, including indels. Evol Bioinform Online. 1:81–83.

Rost B. 1999. Twilight zone of protein sequence alignments. Prot Eng. 12:85–94.

Sigrist CJA, Cerutti L, Hulo N, Gattiker A, Falquet L, Pagni M, Bairoch A, Bucher P. 2002. PROSITE: a documented database using patterns and profiles as motif descriptors. Brief Bioinformatics. 3:265–274.

Stoye J, Evers D, Meyer F. 1997. Generating benchmarks for multiple sequence alignments and phylogenetic reconstructions. Proc Int Conf Intel Syst Mol Biol. 5:303–306.

Stoye J, Evers D, Meyer F. 1998. ROSE: generating sequence families. Bioinformatics. 14:157–163.

Strope CL, Scott SD, Moriyama EN. 2007. indel-Seq-Gen: a new protein family Simulator incorporating domains, motifs, and indels. Mol Biol Evol. 24:640–649.

Subramanian AR, Weyer-Menkhoff J, Kaufmann M, Morgenstern B. 2005. DIALIGN-T: an improved algorithm for segment-based multiple sequence alignment. BMC Bioinformatics. 6:66.

Thompson JD, Koehl P, Ripp R, Poch O. 2005. BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. Proteins. 61:127–136.

van Walle I, Lasters I, Wyns L. 2005. SABmark—a benchmark for sequence alignment that covers the entire known fold space. Bioinformatics. 21:1267–1268.

Varadarajan A, Bradley RK, Holmes IH. 2008. Tools for simulating evolution of aligned genomic regions with integrated parameter estimation. Genome Biol. 9:R147.

Yang Z. 1994. Estimating the pattern of nucleotide substitution. J Mol Evol. 39:105–111.

Sudhir Kumar, Associate Editor

Accepted July 29, 2009