REvolver: Modeling Sequence Evolution under Domain Constraints

Tina Koestler,*[,1,2,3] Arndt von Haeseler,[1,2,3] and Ingo Ebersberger*[,1,2,3]

[1]University of Vienna, Vienna, Austria
[2]Medical University of Vienna, Vienna, Austria
[3]Center for Integrative Bioinformatics Vienna (CIBIV), Max F. Perutz Laboratories, Vienna, Austria

***Corresponding author:** tina.koestler@univie.ac.at; ingo.ebersberger@univie.ac.at.

Associate editor: Sudhir Kumar

Abstract

Simulating the change of protein sequences over time in a biologically realistic way is fundamental for a broad range of studies with a focus on evolution. It is, thus, problematic that typically simulators evolve individual sites of a sequence identically and independently. More realistic simulations are possible; however, they are often prohibited by limited knowledge concerning site-specific evolutionary constraints or functional dependencies between amino acids. As a consequence, a protein's functional and structural characteristics are rapidly lost in the course of simulated evolution. Here, we present REvolver (www.cibiv.at/software/revolver), a program that simulates protein sequence alteration such that evolutionarily stable sequence characteristics, like functional domains, are maintained. For this purpose, REvolver recruits profile hidden Markov models (pHMMs) for parameterizing site-specific models of sequence evolution in an automated fashion. pHMMs derived from alignments of homologous proteins or protein domains capture information regarding which sequence sites remained conserved over time and where in a sequence insertions or deletions are more likely to occur. Thus, they describe constraints on the evolutionary process acting on these sequences. To demonstrate the performance of REvolver as well as its applicability in large-scale simulation studies, we evolved the entire human proteome up to 1.5 expected substitutions per site. Simultaneously, we analyzed the preservation of Pfam and SMART domains in the simulated sequences over time. REvolver preserved 92% of the Pfam domains originally present in the human sequences. This value drops to 15% when traditional models of amino acid sequence evolution are used. Thus, REvolver represents a significant advance toward a realistic simulation of protein sequence evolution on a proteome-wide scale. Further, REvolver facilitates the simulation of a protein family with a user-defined domain architecture at the root.

Key words: insertion, protein domain, protein family, sequence simulation, site-specific models, pHMM.

Introduction

Molecular sequences change over time and their rate and pattern of sequence change are influenced by a variety of different parameters, such as mutation rate or functional and structural constraints. Simulating the evolution of biological sequences is therefore a trade-off between simplifying assumptions to reduce complexity of the problem and biological reality. Several programs exist to simulate the evolution of proteins along a phylogenetic tree (Rambaut and Grassly 1997; Stoye et al. 1998; Fletcher and Yang 2009; Strope, Abel, et al. 2009). All either start with a user-provided sequence or generate a random sequence at the root. Seq-Gen (Rambaut and Grassly 1997) simulates the evolution of the root sequence only by substitutions and does not consider insertions and deletions. ROSE (Stoye et al. 1998) was the first program to close this gap by also modeling the insertion and deletion process. By default, both programs assume that sites evolve independently and identically. Although this is a fair assumption for sequences not assuming any structure or exerting any function, it is an obvious oversimplification when it comes to the simulation of sequence change in functional sequences, such as genes or gene products. As a result, relevant sites that remain unchanged over considerable evolutionary distances in real sequences may be altered by a simulator after only a few simulation steps. To cope with this problem, both programs consider substitution rate heterogeneity by randomly assigning rate scaling factors to individual sequence positions. Although this is valid for random root sequences, it is not when the evolution of real protein sequences should be simulated. In such cases, it cannot be avoided that a functionally relevant site, which is unlikely to change over time, is assigned a high substitution rate by chance. Even more problematic is the modeling of insertion and deletion events (indels) that are typically placed randomly in a sequence by the simulator. Moreover, indel lengths are often drawn from a single distribution. However, a biologically meaningful simulation requires that the placement of indels be guided by information, about where in a sequence insertions or deletions can be tolerated and where they are likely to interfere with the protein's function. Since the spacing between interacting amino acids in the native structure of a protein is important, the length distribution of indels may also vary between individual positions of a sequence (see Laity et al. 2001). INDELible (Fletcher and Yang 2009), SIMPROT

(Pang et al. 2005), and indel-Seq-Gen (iSG; Strope, Abel, et al. 2009) represent major steps toward more realistic simulation. These programs facilitate the manual assignment of different evolutionary parameters to specific segments of the sequence. This enables explicit differentiation between evolutionary constraints acting for example on functional protein domains and those acting on intervening linker regions. Despite this progress, two major limitations remain. First, it is not feasible to use these programs in large-scale studies where the evolution of hundreds or thousands of protein sequences is simulated, as there is no automatized procedure to extract meaningful constraints. Second, there is no standard operating procedure for inferring evolutionary constraints. This opens the door for ad hoc decisions that may later be hard to justify or reproduce. Considering sequence structure is an obvious solution for the second problem. The emergence of fast algorithms capable of evaluating the effects of mutations on the structure of the protein facilitated the development of programs integrating structural consequences of individual mutations into the simulation (e.g., Parisi and Echave 2001; Rastogi et al. 2006; Grahnen, Kubelka, et al. 2011; Grahnen, Nandakumar, et al. 2011; Lakner et al. 2011). Unfortunately, for the vast majority of sequences, the relevant information for deriving evolutionary constraints, that is, the structure is not available. Moreover, predicting the exact effect of individual mutations on structure and function of a protein and extrapolating this to the evolutionary behavior of individual sites of a protein is still hard. This limits a wide use of structure-informed constraints in simulated sequence evolution.

Here, we suggest a pragmatic approach to achieve a biologically meaningful simulation of sequence evolution. Homologous sequences have been evolving independently since they last shared a common ancestor. The comparison of such sequences reveals sites that remain entirely conserved over time, sites displaying only a subset of the amino acid alphabet, and sites that appear to be free to change. Moreover, it indicates the preferred positions of insertions and deletions as well as their respective length distributions. This pattern of sequence conservation and alteration represents the footprint of a constrained evolutionary process acting on these sequences. In principle, databases such as Pfam (Finn, Mistry, et al. 2010) or SMART (Letunic et al. 2009) provide exactly this information. They have been specialized in the collection and alignment of homologous protein sequences or protein domains and describe the characteristics of the resulting alignments by a profile hidden Markov model (pHMM; fig. 1). In these models, site-specific emission probability vectors reflect the frequencies of the 20 amino acids at the corresponding positions in real instances of the modeled domain. Similarly, the models provide site-specific insertion and deletion probabilities. Unfortunately, it is not straightforward to exploit this information for the simulation of sequence evolution. Traditionally, pHMMs are defined as generative models that produce instances of a domain or protein family rather than modeling its change. Consequently, time is not considered in the pHMM formalism. Our new simulator, REvolver, solves this problem by implementing the following key features:

- Emission probabilities of the pHMM are used as site-specific amino acid equilibrium frequencies in the substitution model.
- Insertions and deletions are placed preferentially at positions where they have been already observed in real instances.
- REvolver corrects for the formation of artificially large insertions due to repeated nested insertions.
- Evolution acts on the amino acid sequence AND on the relationship between the amino acids sequence and the constraints. Hence, the information about site-specific evolutionary constraints is maintained throughout the simulation.
- A mechanism counterbalancing the erosion of characteristic sites prevents a simulated sequence from losing its identity as a domain instance.

The Simulator

In the following sections, we describe the general procedure to simulate the evolution of a domain along a phylogenetic tree. The ancestral evolutionary instance consists of the amino acid sequence together with its state path through a pHMM. A typical pHMM is depicted in figure 1. It consists of match states (M), insertion states (I), deletion states (D), and a *Begin* and an *End* state. States are connected via transitions, where each transition has its individual transition probability (P). Match states and insertion states emit amino acids according to an emission probability vector $E = (e_1, \ldots, e_{20})$ for the 20 amino acids. A random path through a pHMM starts at the *Begin* state, passes through match, insertion and deletion states, and terminates at the *End* state. By that, an instance of the modeled domain/protein is generated. The resulting state path represents the relationship between the constraints and the specific amino acid positions.

Starting at a node in a phylogeny, the parent instance evolves along a branch leading to a child instance. Mutations result in changes in the amino acid sequence but can also alter the state path. Thus, the state path must evolve with the sequence. The procedure for the simulated evolution on one branch is repeated for each branch in the tree, resulting in protein sequences on the leaf nodes sharing a common ancestry together with their state paths. Next, we explain the realization of the individual mutations (substitutions, insertions, and deletions) with and without domain constraints. Note that in the context of this manuscript, we partition a protein sequence into domains and linker regions. We refer to a domain as a segment of a protein that is modeled by a pHMM and refer to the remainder of the protein as linker sequences. In our simulations, domain regions evolve under constraints inferred from the pHMM, whereas linker regions evolve free of constraints. If a protein contains

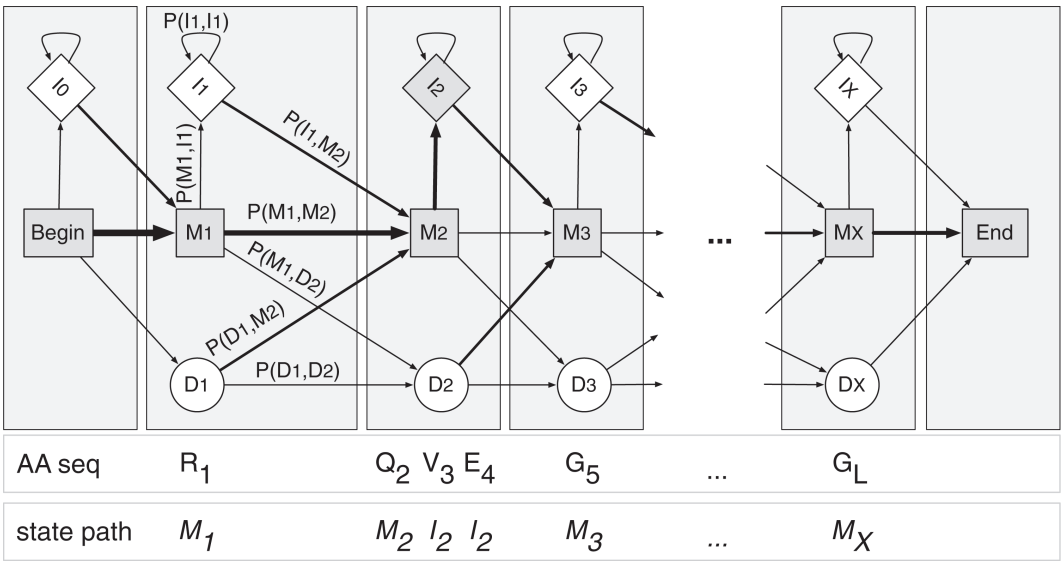

FIG. 1. Structure of a pHMM: The pHMM comprises match states (M_x), insertion states (I_x), deletion states (D_x), a *Begin* state, and an *End* state. The index x ranges from 0 to X, where X is the length of the pHMM. Since states (M_x, I_x, D_x) of the same position x (except for position 0 and X) have together 7 transitions to states $x+1$, the model is called Plan7 (http://hmmer.janelia.org/). Arrows indicate transitions between individual states, where the line weight is proportional to the transition probability $P(\text{State}_x, \text{State}_y)$. The amino acid sequence is indexed from 1 to the sequence length L. The respective states in the corresponding state path are shaded in gray in the pHMM. The sequence RQVEG...G are amino acids emitted from match (RQGG) and insertion (VE) states.

more than one segment, we perform the simulation on each segment separately. REvolver is based on Plan7 pHMMs produced by the program hmmscan from the HMMER3 software package (http://hmmer.janelia.org/; cf. fig. 1).

Simulation Procedure

To simulate substitutions, insertions, and deletions, we apply the Gillespie algorithm (Gillespie, 1977) as outlined in Algorithm 1. Substitutions are described by a continuous-time Markov chain that is characterized by a matrix Q of instantaneous rates q_{ij}, where q_{ij} is the product of the relative rate of substitution ρ_{ij} from amino acid i to amino acid j, and the amino acid frequency π_j. Currently, 14 standard amino acid substitution models are implemented in REvolver (table 1). In addition, REvolver can use any user-defined substitution model composed of the relative rate matrix $R = \{\rho_{ij}\}$ and the equilibrium frequencies π_j. The substitution rate for any amino acid i is given by $q_i = \sum_{j \neq i} q_{ij}$. Finally, the total substitution rate $\Lambda_S = \sum_{l=1}^{L} q_{i_l}$, where L is the sequence length and i_l the amino acid at position l of the sequence. In addition to substitutions, we simulate insertions and deletions. The rates for insertions λ_I and deletions λ_D are independent from each other. Since a sequence of length L has L possible positions to start a deletion, the total deletion rate is $\Lambda_D = L\lambda_D$. Insertions can occur before the first amino acid and after every amino acid. Consequently, the total insertion rate is $\Lambda_I = (L+1)\lambda_I$. The insertion position at the very beginning of a sequence is considered to be an immortal link (Thorne et al. 1991). Thus, an insertion can occur even when all amino acids were deleted in a previous step. Note that $\Lambda_I = (L_1+1)\lambda_I$ applies only for the first segment. The total insertion rate for the remaining segments is $L_n\lambda_I$, where L_n is the length of the nth segment, $n > 1$. Eventually, we set the total event rate $\Lambda = \Lambda_S + \Lambda_I + \Lambda_D$.

Table 1. Standard Protein Evolution Models Implemented into REvolver.

Substitution Model	Reference
JTT	Jones et al. (1992)
JTT_dcmut	Kosiol and Goldman (2005)
Dayhoff	Dayhoff et al. (1978)
Dayhoff_dcmut	Kosiol and Goldman (2005)
WAG	Whelan and Goldman (2001)
mtMAM	Yang et al. (1998)
mtART	Abascal et al. (2007)
mtREV	Adachi and Hasegawa (1996)
cpREV	Adachi et al. (2000)
Vt	Müller and Vingron (2000)
Blosum62	Henikoff and Henikoff (1992)
LG	Le and Gascuel (2008)
HIVb	Nickle et al. (2007)
HIVw	Nickle et al. (2007)

To simulate the evolutionary process along a branch of a tree (cf. Algorithm 1), we divide the branch into a number of time steps that are exponentially distributed. To this end, we draw a "waiting" time t_w from an exponential distribution with mean $1/\Lambda$ during which exactly one mutation occurs (von Haeseler and Schöniger 1998). t_{rem} is the remaining time, initialized with the branch length t. If t_w is smaller than or equal to t_{rem}, a mutation occurs. We next choose according to Λ_I, Λ_D, and Λ_S whether an insertion, deletion, or a substitution should occur. The sequence and the state are then changed, and we update Λ as follows: If the event was an insertion or deletion, we adjust the sequence length L by

adding or subtracting the length of the insertion or deletion, respectively, and recalculate Λ_I and Λ_D accordingly. An important property of REvolver is that once a sequence has been inserted, it undergoes the same evolutionary process as the root sequence, that is, it can be substituted, deleted, and the insertion can be extended. If a substitution occurred in which amino acid j replaced amino acid i, we exchange q_i by q_j to update Λ_S. Finally, we set $t_{\text{rem}} = t_{\text{rem}} - t_w$, draw a new t_w from the exponential distribution with the updated parameter Λ, and repeat until $t_w > t_{\text{rem}}$.

This general procedure is used for all sequences. The specific details of simulating unconstrained and constrained sequences are described in the next sections.

Algorithm 1 Outline of the simulation procedure

$\Lambda \leftarrow \Lambda_S + \Lambda_I + \Lambda_D$
$t_{\text{rem}} = t$
$t_w \sim \text{Exp}(\Lambda)$
while $t_w \leqslant t_{\text{rem}}$ **do**
  randomVariable $\sim$ Uniform()
  if randomVariable $\leqslant \Lambda_I/\Lambda$ **then**
   doInsertion()
else if randomVariable $\leqslant (\Lambda_I + \Lambda_D)/\Lambda$ **then**
   doDeletion()
  else
   doSubstitution()
  end if
  Λ = updateEventRate()
  $t_{\text{rem}} \leftarrow t_{\text{rem}} - t_w$
  $t_w \sim \text{Exp}(\Lambda)$
end while

Evolutionary Events for Unconstrained Segments (Linker)

In the following, we describe the simulation of substitutions, insertions, and deletions for unconstrained segments, where the evolutionary instance is simply the amino acid sequence.

Substitutions

REvolver simulates the substitution process in unconstrained segments based on a substitution model Q plus a parameter r that encodes variation in rate across sites (RAS). The substitution rate at a given site l is, thus, calculated as $q_i r_l$, where r_l is a rate scaling factor and i is the current amino acid at site l. We provide three types of RAS models, where r_l is always independently and identically distributed among sites: the scaling factor is (i) the same at all sites (default), (ii) drawn from a continuous gamma distribution, and (iii) drawn from a discrete gamma distribution. Both gamma distributions have a mean of 1 and shape parameter α. In the case of rate heterogeneity ((ii) and (iii)), rate scaling factors are assigned to each position l in the root sequence. Child nodes inherit the scaling factors from their parent node. Newly inserted positions receive a scaling factor from this gamma distribution. The sequence site l where the substitution occurs is chosen proportional to its substitution rate $q_i r_l$. The probability that amino acid i is substituted with amino acid j is proportional to q_{ij}/q_i for $i \neq j$ (Karlin and Taylor 1975).

Insertions and Deletions

Insertion and deletion positions are distributed uniformly along the unconstrained segments. To determine the length of an individual insertion or deletion, we draw a value from a probability distribution. Currently, we have implemented the geometric distribution and the Zipfian distribution (Benner et al. 1993; Chang and Benner 2004). The parameters for the distributions are user-defined. Once position and length of an insertion are determined, we sample the amino acids from the equilibrium frequency of the selected substitution model Q.

Evolutionary Events for Constrained Segments (Domains)

Next, we describe the simulation of substitutions, insertions, and deletions for a constrained segment. The evolutionary instance is now the amino acid sequence together with its state path through the pHMM (cf. fig. 1).

Substitutions

For each site l, the emission probabilities of the associated pHMM state are taken as the stationary amino acid frequencies of the user-selected substitution model Q. Thus, each site l in the domain gets assigned its own model Q_l. The substitution rate q_{i_l} at site l is therefore $\sum_{j \neq i} \rho_{ij} e_{j_{M_x}}$ for a match state or $\sum_{j \neq i} \rho_{ij} e_{j_{I_x}}$ for an insertion state, where $e_{j_{M_x}}$ and $e_{j_{I_x}}$ are the state-specific emission probabilities for amino acid j of the pHMM. The sequence site l where the substitution occurs is chosen proportional to the substitution rate q_{i_l}. The probability that amino acid i is substituted with amino acid j is proportional to q_{ij}/q_i for $i \neq j$ (Karlin and Taylor 1975).

Insertions

The probability of placing an insertion after position l in the amino acid sequence is $P(M_x, I_x)$ if l is associated with M_x, or $P(I_x, I_x)$ otherwise. The probability of placing an insertion before the first amino acid is $P(Begin, I_0)$. We apply a geometric distribution with parameter $1 - P(I_x, I_x)$ to determine the insertion length n. Simply adding the insertion to the sequence, however, poses one problem. Subsequent nested insertion events would allow insertions to grow to total lengths that do not adhere to the model. To counterbalance this effect, we have implemented the following procedure. If there are already k insertion states I_x in the state path, we only add the number of insertion states required to achieve length n rather than adding all n insert states. Thus, at one insertion event, only $n - k$ amino acids are inserted. Finally, we sample the amino acids proportional to the emission probabilities of I_x and insert them to the right of any amino acids that are already associated with state I_x.

Deletions

The site l where a deletion occurs is either associated with state M_x or state I_x. In the case of M_x, we enter D_x from the

respective previous state $x-1$ to realize the deletion. Recall, that D_x can be reached either via the transition $M_{x-1} \rightarrow D_x$ or via the transition $D_{x-1} \rightarrow D_x$. The deletion probability is, therefore, either $P(M_{x-1}, D_x)$ or $P(D_{x-1}, D_x)$. We replace M_x with D_x in the state path and remove the corresponding amino acid l from the sequence. Next, we determine the length of the deletion. Note that the pHMM does not provide an explicit deletion length distribution. Instead, it gives two choices to leave D_x: either we can move to M_{x+1} and terminate the deletion or we can move to D_{x+1} and extend the deletion. Thus, amino acids get deleted one by one, where in each step, we have the choice to terminate the deletion. If D_{x+1} is already present in the state path, we move to the last deletion state in a row D_{x+z}, where z is the number of successive deletion states and consider $P(D_{x+z}, D_{x+z+1})$ for a deletion extension. Alternatively, if the amino acid l marked for deletion is associated with I_x, we proceed as follows: Transitions from I states to D states are not allowed in Plan7 pHMMs. Therefore, we first assign the same deletion probability to each I state, namely, the mean deletion probability of all match states. Then we choose the deletion length either from a geometric or a Zipfian distribution with the same parameters as for unconstrained sequence parts. Note that the deletion length is limited by the number of consecutive I states. Finally, we remove the I states from the state path and the corresponding amino acids from the sequence. This, in principle, completes the simulation schema. However, we take into account one more detail.

Resurrection of M States

Deletions remove amino acids that are associated with I or M states. Insertions, on the other hand, only create I states. Consequently, on the long run, the state path gets depleted of M states until only I states remain. To compensate for this erosion, we allow the insertion of amino acids that are associated with lost M states. More formally, if I_x emits an amino acid and I_x is followed by D_{x+1}, we facilitate the resurrection of M_{x+1}. Thus, the new amino acid emitted by the I_x state can be assigned to the M_{x+1} state.

Let us illustrate this by the example in figure 2: Suppose the amino acid sequence is associated with a state path as follows:

sequence :	A	G	K				A
state path :	M_2	I_2	I_2	D_3	D_4	D_5	M_6

Furthermore, suppose that an insertion length of 5 was drawn from the geometric distribution with parameter $1 - P(I_2, I_2)$ to extend I_2. Since I_2 already appears two times in the state path, we extend this insertion by additional three amino acids. We emit amino acids (CQL) proportional to the emission probabilities in vector E_{I_2}, and insert them stepwise, starting with the C, after amino acid K (cf. fig. 2). The deletion states D_3, D_4, and D_5 follow directly after I_2, and thus the C is now given the chance to resurrect one of the corresponding match states: M_3, M_4, or M_5. We first choose the candidate for resurrection with probabilities proportional to the match state emission probability for C. Assume M_4 was selected, then we decide whether or not M_4 will be resurrected. The emission probability for C at M_4 is 0.8. Consequently, C will be associated with M_4 with probability 0.8 and with probability 0.2, it will stay with I_2. We then continue with the next amino acid in the insertion string, Q. Since we associated C with M_4, Q can either be associated with I_4 or M_5. In our example, we selected I_4. Finally, we insert L. With probability 0.7 (e_L at M_5), we associate L with M_5. The resulting sequence with the associated state path after the insertion is then:

sequence :	A	G	K		C	Q	L	A
state path :	M_2	I_2	I_2	D_3	M_4	I_4	M_5	M_6

and M_4 and M_5 are the newly populated match states.

Additional Features

Input

REvolver takes a user-defined phylogenetic tree in Newick format and a root sequence as input. If the root sequence is also user-specified, a protein sequence together with its protein domain annotation via hmmscan (http://hmmer.janelia.org/) is required. If the same amino acid in a protein is assigned to more than one domain, REvolver considers only the domain with the smallest e-value. Alternatively, the root sequence can be randomly generated. In this case, the user defines a domain architecture, that is, a linear order of domains from the pHMM database (e.g., Pfam or SMART) together with the lengths of any linker regions. The root protein can consist of any combination of domains and linkers. REvolver extracts the corresponding pHMMs from the database and generates a random instance for each domain. For unconstrained segments, the sequence is sampled proportional to the equilibrium frequency of the substitution model Q. Then the root sequence evolves along the tree.

When REvolver is invoked without any input, REvolver guides the user interactively through the setting of all required parameters and input files suggesting reasonable default values. Upon execution, the program generates a configuration file encoding these input parameters in xml format, which can be re-used, for example, when integrating REvolver into an automated workflow.

Output

After the simulation, REvolver outputs a multiple alignment of the simulated leaf node sequences with the options to include the root sequence or inner node sequences. Simulated sequences can be annotated with models from a pHMM database, for example Pfam or SMART, automatically. Moreover, we provide the option to present the domain architectures of the sequences visually.

Lineage-Specific Evolution

REvolver allows the specification of the substitution model and the insertion and deletion parameters individually for each branch in the tree. The model and the insertion and deletion rates will then apply to all domains and linkers.

Running Time

The simulation of evolution of constrained segments is obviously computationally more expensive than of

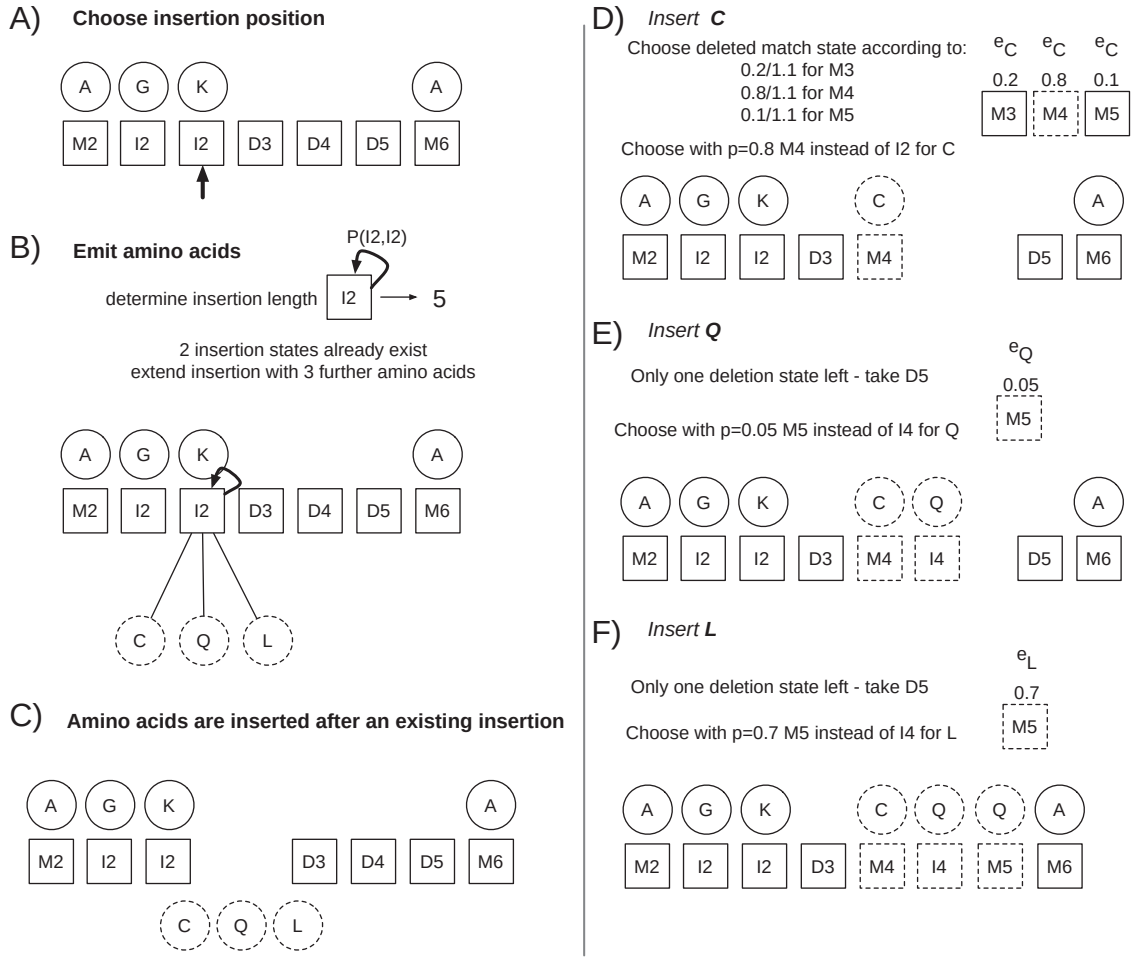

FIG. 2. A generic insertion scenario: circles represent the amino acid sequence, the corresponding state path is shown as squares. Dashed circles and dashed squares represent newly inserted amino acids and the corresponding states, respectively. The insertion position is chosen at amino acid K with the corresponding state I_2 (*A*). The geometric distribution with the transition probability $1 - P(I_2, I_2)$ as parameter determines the length of the insertion (*B*). Amino acids are randomly emitted according to their emission probabilities E_{I_2} at state I_2. The stepwise insertion of amino acids considers the emission probabilities e_i for individual amino acids i at deleted match states (M_3, M_4, and M_5) (*C*–*F*).

unconstrained segments. Nevertheless, REvolver runs in a reasonable time. For example, the simulated evolution of a root protein of 500 amino acids with two domains along a tree with 30 leaf nodes (total tree length: 24.17 expected substitutions per site) with equal insertion and deletion rates of 0.01 took 4.9 s (user + sys time) on an Intel quad core i5 PC (3.30 GHz). The simulation with the same setup, but without domain constraints ran for 1.2 s.

Availability

REvolver, the manual, and example files are available for download at www.cibiv.at/software/revolver. The source code is available upon request. The software is written in java, and thus runs on any platform where java 6 is installed. REvolver requires the HMMER3 software package, which is freely available at http://hmmer.janelia.org. pHMM databases for the REvolver simulations (e.g., Pfam or SMART) have to be downloaded from the appropriate sources. Alternatively, custom pHMM collections may be used.

Verification of the Implementation

REvolver is the first simulator of protein sequence evolution that uses pHMMs for automatically customizing evolutionary models. In the following, we evaluated the effect of pHMM informed constraints on simulated sequence change.

Simulation of Substitutions

The equilibrium frequencies of REvolver's site-specific substitution models are derived from the emission probabilities of the corresponding states in the pHMM. Consequently, if related sequences evolve long enough and are then aligned, the amino acid frequencies at individual positions should again reflect the emission probabilities of the corresponding states in the original pHMM. To demonstrate this property, we used the Pfam domain A1_Propeptide whose pHMM was trained on a gap free seed alignment of 85 sequences. We evolved a single domain instance along a star tree with 85 branches to obtain a corresponding simulated seed alignment. Every sequence position on each branch

was substituted on average 30 times. From the simulated sequences, we then constructed a pHMM and computed a similarity score to the original A1_Propeptide pHMM with hhalign (Söding 2005). The similarity score between the original pHMM and the pHMM inferred from the simulated data is 75.13, only slightly smaller than the score that is obtained when the original pHMM is compared to itself (80.83). In contrast, when we repeated the simulation, this time without domain constraints, the similarity score between the original pHMM and the pHMM based on the simulated sequences dropped to only 0.22. This demonstrates that REvolver's domain constraint maintains site-specific compositional properties of protein sequences.

Simulation of Insertions

The placement of insertions within a domain, as well as their individual length distributions, are guided by the transition probabilities in the domain pHMM. Insertions are placed preferentially at such positions where the probability for reaching an insert state is high. In the same way, the transition probability to leave the insert state is used as parameter for the insertion length distribution. To verify the implementation of this procedure, we tracked insertions in the simulated evolution of the ABC transporter domain (ABC_tran; PF00005). In 10,000 simulations, we each started with an instance of the ABC_tran domain at the root that consisted of only match states ($M_1 M_2 M_3, \ldots, M_{118}$). This sequence was then evolved under the WAG substitution model (Whelan and Goldman 2001) up to 0.5 expected substitutions per site with $\Lambda_I = \Lambda_D = 0.1$. Then we tracked the positions of insertions as well as their respective lengths on the state path level and compared the results with the expected positions and lengths given the pHMM (fig. 3*a*). We observed insertion hot spots in our simulations at match states 21, 22, 36, 47, 64, 80, and 84 (fig. 3*b*). The same match states are flagged as the most prominent insertion positions in the pHMM logo (cf. fig. 3*a*). However, note that insertions in our simulation were not restricted to these positions. They also occurred after other match states but with considerably lower frequency. Similar to the position of the insertions, their respective length distributions also meet the expectations given the pHMM. We observed the longest insertions (mean length of 31.78 aa) at M_{64} (fig. 3*c*). Similarly, insertions at M_{52} and M_{84} tend to be longer than those at other states. In summary, our results indicate that REvolver models insertions in such a way that both their distribution along the sequence and their lengths agree with what is seen in real sequences.

Benchmarking and Example Applications

Comparing REvolver to Other Simulation Programs

For the benchmarking of REvolver, we utilized the framework introduced by Strope, Scott, et al. (2007), which is based on the simulated evolution of G protein-coupled receptors (GPCR). The GPCR superfamily includes a vertebrate olfactory receptor protein family, characterized by, on average, 7 transmembrane (tm) regions, and an extracellular N-terminus. Strope, Scott, et al. (2007) collected 29 olfactory receptors, constructed an alignment, and inferred a maximum parsimony (MP) tree. The consensus sequence of the 29 proteins was then evolved on this MP tree. For the simulations, Strope and colleagues manually defined a variety of individual parameter settings, including the assignment of site-specific rates, invariant sites, individual rates and length distributions for insertions and deletions, and tree scaling factors for different protein segments. With these optimized settings, they compared iSG (Strope, Scott, et al. 2007), ROSE (Stoye et al. 1998), Seq-Gen (Rambaut and Grassly 1997), and SIMPROT (Pang et al. 2005) with respect to the following properties of the simulated sequences: (i) the preservation of transmembrane regions, (ii) the preservation of Pfam domains, and (iii) the maintenance of a significant sequence similarity to the GPCR superfamily.

We simulated the evolution of GPCRs with REvolver adhering as closely as possible to the procedure described by Strope, Scott, et al. (2007). To this end, we took the published MP tree topology and the alignment and estimated the number of substitutions on each branch with PAUP* (Wilgenbusch and Swofford 2003). The number of substitutions per site was obtained by dividing the number of substitutions per branch inferred from the MP tree by the alignment length. We constructed a consensus sequence from the alignment of 29 olfactory receptors with iSG, annotated this sequence with Pfam, and performed 1000 independent REvolver simulations starting from this sequence. The simulations were performed using the JTT substitution model (Jones et al. 1992). The insertion and deletion rates of 0.018 were chosen as the mean of 15 different insertion and deletion rates that were assigned to individual segments of the root protein in the analysis by Strope, Scott, et al. (2007). For the benchmark test, we analyzed the simulated sequences by (i) counting the number of transmembrane regions with a transmembrane prediction program (hmmtop v2.1; Tusnády and Simon 2001), (ii) determining the presence of Pfam domains (Finn, Mistry, et al. 2010) with hmmscan (http://hmmer.janelia.org/), and (iii) assessing their similarity to GPCRs as represented in Uniprot (The UniProt Consortium 2010) with BlastP (Altschul et al. 1990).

Table 3 displays the results of the benchmark test. (i) REvolver preserves on average 6.89 transmembrane regions, which is close to 7, the expected number for GPCRs. The mean observed number of transmembrane regions for sequences simulated with the other simulators are as follows: ROSE: 5.94, SIMPROT: 0.20, Seq-Gen: 6.84, and iSG: 7.03. Considering the standard deviations for the individual experiments, the differences between Seq-Gen, iSG, and REvolver are negligible. (ii) The average bit score between REvolver simulated sequences and the 7tm_1 Pfam domain (PF00001) is 102.8 (cf. table 3). In contrast, sequences simulated with the other programs achieve a mean bit score of no more than −5.1 (iSG). (iii) In the third part of the analysis we show that REvolver simulated sequences have a higher sequence similarity to members of the GPCR protein family than to any other protein in the Uniprot database. For each simulated sequence, the top 250 BlastP hits were only

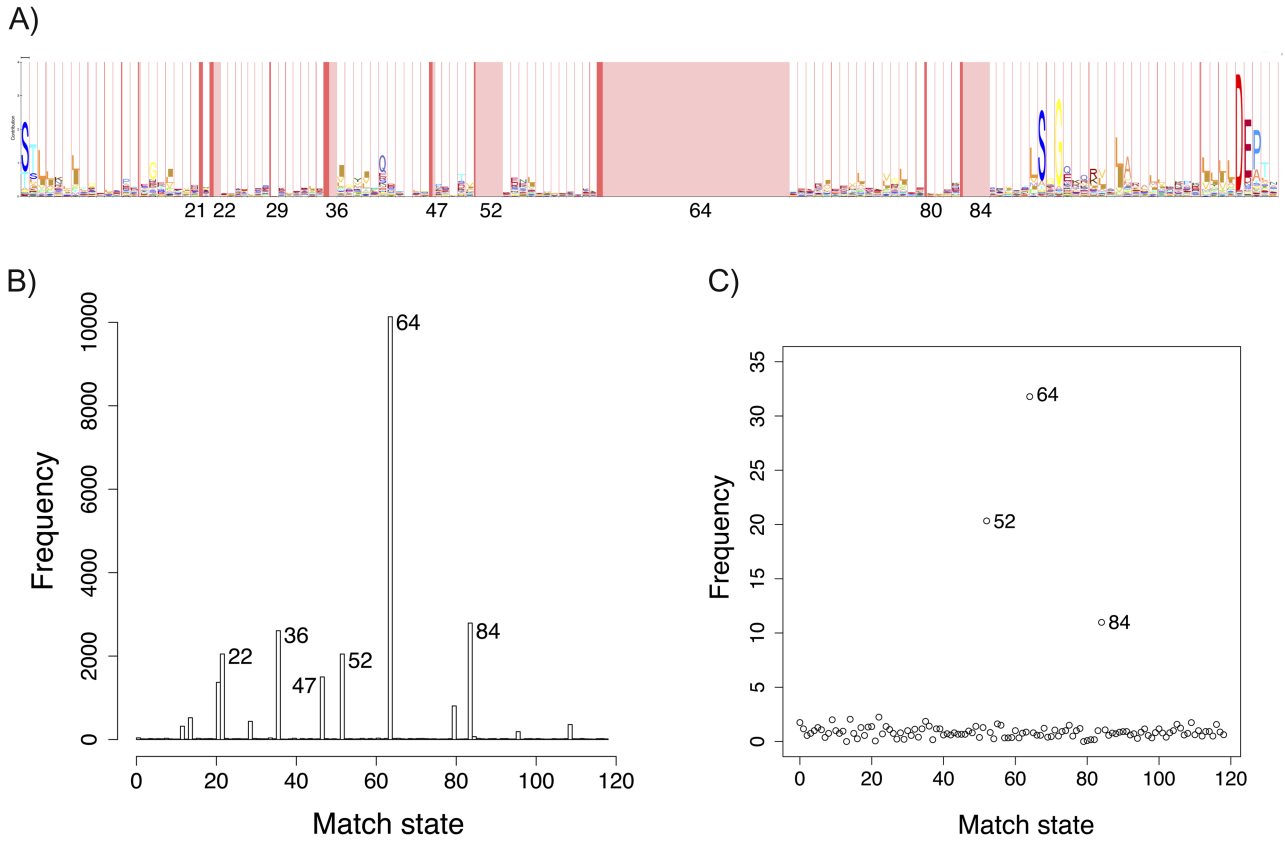

FIG. 3. Positions and lengths of insertions in the ABC_tran domain. (*A*) The pHMM logo (Schuster-Bockler et al. 2004) of the ABC_tran domain (http://pfam.sanger.ac.uk/family/PF00005) summarizes for each pHMM position information about emission probabilities, transition probability to enter an insertion state, and the probability to stay in an insertion state. The relative height of an amino acid at a certain match state reflects its emission probability. The thickness of dark pink bars represent how likely an insertion occurs at a given position, whereas the thickness of light pink bars represent the expected length of an insertion. (*B*) The histogram shows how often a pHMM position was chosen for an insertion event in 10,000 REvolver simulations starting from an ABC_tran root sequence. (*C*) The plot displays for each of the 118 positions of the ABC_tran pHMM the mean insertion length in the 10,000 simulations.

comprised of GPCRs. The mean bit scores lie in the range between those of ROSE and iSG (table 3).

In summary, REvolver performs comparable or even outperforms existing protein simulators in the maintenance of functional characteristics in the chosen benchmark data set. The major improvement however is that the parameterization to achieve this performance was done automatically and did not require any manual interaction. Thus, REvolver is able to deal with large scale data as demonstrated next.

Proteome-Wide Evaluation of Domain Content Preservation

We simulated the evolution of human proteins on a proteome-wide scale. For this purpose, we annotated 21,971 human proteins (Ensembl 51) with Pfam (Finn, Mistry, et al. 2010) and with SMART (Letunic et al. 2009) using hmmscan with default settings (http://hmmer.janelia.org/). This procedure identified 45,738 Pfam and 32,289 SMART domains. Then we took each human protein as root sequence, simulated its evolution over different evolutionary times T (scaled in expected substitutions per site) and annotated the resulting sequences again with hmmscan. Finally, we compared the domain content for each simulated sequence with that of the root sequence. We considered a domain to be preserved if it was present both in the root sequence and in the respective simulated sequence. The fractions of preserved domains for T, ranging from 0.1 to 1.5, are shown in figure 4*A* (Pfam) and *B* (SMART). The parameter settings for the individual rounds of simulations are summarized in table 2. In the first round, we set the insertion and deletion rates to 0 ($\lambda_I = \lambda_D = 0$). When simulating in the traditional way, that is, without domain constraints, only 15% of the Pfam and 9% of the SMART domains were preserved at $T = 1.5$. This figure changes substantially when we impose domain constraints. In this case, more than 90% of the Pfam and SMART domains were detected in the simulated sequences at $T = 1.5$. Subsequently, we assessed the effect of insertions and deletions. For the evolution without domain constraints, the percentages of retained domains decreased rapidly with increasing evolutionary time. At $T = 1.5$ only 1%/2% (Pfam/SMART) of the original domains were maintained with insertion and deletion rates of 0.05 and only 0.5%/1% with insertion and deletion rates of 0.1.

Here, the effect of domain constraints on the preservation of domains over time was even more pronounced. At $T = 1.5$ still 79%/74% (insertion and deletion rates

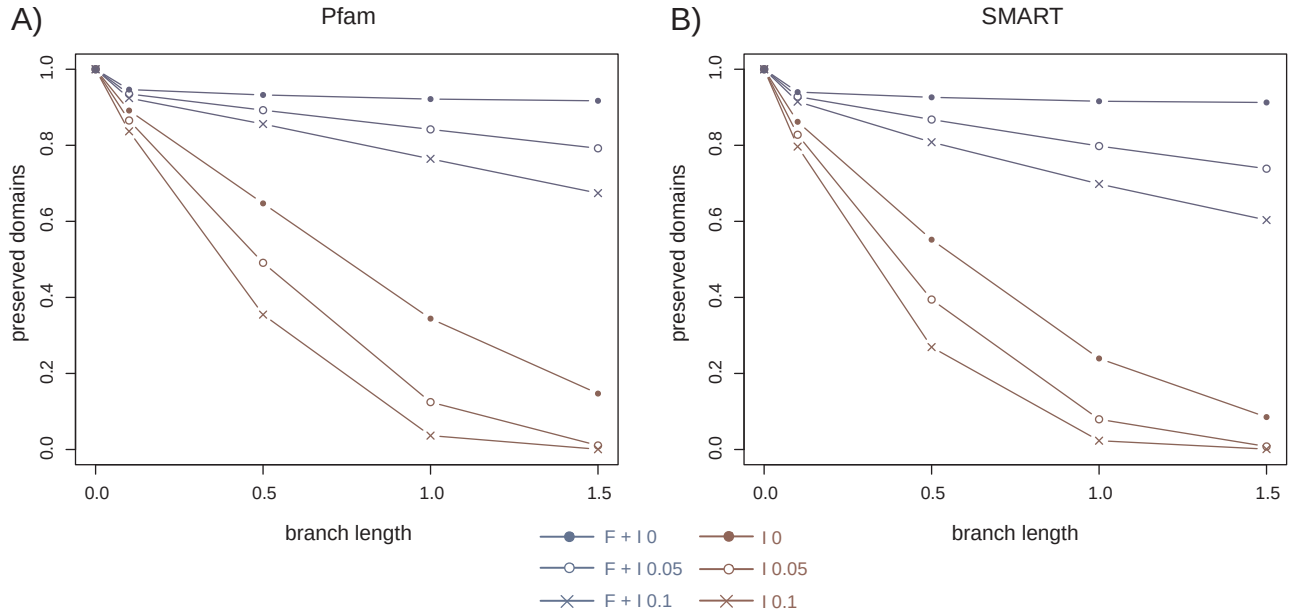

FIG. 4. Fraction of preserved Pfam (*A*) and SMART (*B*) domains. All human proteins were taken as root sequences and evolved with 0.1, 0.5, 1.0 and 1.5 expected substitutions per site. *F* denotes simulations with domain constraints (blue lines). Simulations without domain constraints are colored in red. *I* 0 stands for simulations without indels, *I* 0.05 for insertion and deletion rates of 0.05, and *I* 0.1 for insertion and deletion rates of 0.1 (cf. table 2).

of 0.05) and 67%/60% (insertion and deletion rates of 0.1) of the domains were preserved. Simulations under a RAS model are often used to account for sites under different evolutionary constraints in a protein. We therefore repeated our simulation for the unconstrained case using two different values for the shape parameter of the gamma distribution ($\alpha = 1$ and $\alpha = 0.5$). Despite the case of the RAS model, the increase in the number of retained domains was only marginal when insertions and deletions were included in the model (supplementary tables S1 and S2, Supplementary Material online). Without indels, simulations under domain constraints still outperformed the RAS model by a factor of 2–3.

Preservation of Structure

So far, we have shown that REvolver substantially increases the evolutionary stability of protein domains in the course of simulated sequence change. Although structural constraints are not explicitly captured in pHMMs (but see Eddy 1998), we next assessed whether sequences simulated under domain constraints are also structurewise more similar to the native protein than sequences simulated without constraints. For our analysis, we used the human SAP SH2 protein (Poy et al. 1999) and evolved it with and without domain constraints ($\Lambda_D = \Lambda_I = 0$). Then we assessed the rooted mean square distance (RMSD) between the structure of the native protein (1d4tA; Velankar et al. 2011) and the inferred structure of the simulated sequences. SARA (Grahnen, Kubelka, et al. 2011) was used for analyzing the RMSD between corresponding side chains in the two structures. Next, we used MODELLER (Eswar et al. 2006) to analyze the RMSD between the peptide backbones of two structures. This analysis was performed with three different insertion and deletion rates ($\Lambda_D = \Lambda_I = 0/0.05/0.1$). In all comparisons, the RMSD between the native structure and the inferred structure of the simulated sequence was smaller for the constrained simulation than for the unconstrained simulation (supplementary figs. S1 and S2, Supplementary Material online). A one-sided *t*-test ($\alpha = 0.05$) revealed that, except for a single case, the differences are significant.

Table 2. Parameter Settings for the Simulations of Human Protein Evolution.

	Insertion and Deletion Rates	Abbreviation
Unconstrained	**0**	***I* 0**
	0.05	***I* 0.05**
	0.1	***I* 0.1**
Constrained	**0**	***F* + *I* 0**
	0.05	***F* + *I* 0.05**
	0.1	***F* + *I* 0.1**

All simulations were performed for 0.1, 0.5, 1.0, and 1.5 expected substitutions per site under the WAG substitution model (Whelan and Goldman 2001). The geometric distribution ($p = 0.25$) was used to model indel lengths. The last column shows the abbreviations for the parameter setting used in fig. 4, where *F* labels simulations under domain constraints and *I* denotes the parameter for the insertion and deletion rates. The analysis was performed once using the Pfam database and once using the SMART database for protein domain annotation.

Simulation of Proteins with User-Defined Domain Architectures

REvolver is the first program that offers the possibility to simulate protein evolution with user-defined domain architectures. To exemplify this feature, we used REvolver to generate a random root sequence consisting of instances of an RLI domain (Possible metal-binding domain in RNase L inhibitor; PF04068), a Fer4 domain (4Fe–4S binding domain;

Table 3. Comparison of REvolver to Other Simulators.

	REvolver	iSG	ROSE	SIMPROT	Seq-Gen
tm regions	6.89±0.60	7.03±0.30	5.94±1.25	0.20±0.37	6.84±0.91
Pfam bit score	102.75	−5.09	−31.47	—	−7.18
Top *n* BlastP hits					
25	152.0	174.0	141.1	—	196.7
100	143.6	164.7	132.7	—	183.3
250	135.5	155.9	124.4	—	177.8

Results for the analysis of GPCR proteins. Values for iSG, ROSE, SIMPROT, and Seq-Gen were taken from Strope, Scott, et al. (2007). "tm regions" denotes the number of transmembrane regions. "Pfam bit score" shows the mean bit scores between simulated sequences and the Pfam domain 7tm_1. No score is given for SIMPROT because of missing 7tm_1 hits. The mean bit scores of the first 25, 100, and 250 BlastP hits are shown under "Top *n* BlastP hits." Values for SIMPROT are missing because the top scoring BlastP hits did contain non-GPCR proteins

PF00037), and two ABC_tran domains (ABC transporter; PF00005). The domains are separated by unconstrained segments of different lengths. The root sequence was then evolved under the WAG model along an arbitrary phylogeny displayed in figure 5, using insertion and deletion rates of 0.1. Figure 5 shows the input tree together with the resulting domain architectures of the simulated sequences at the leaves. Not all domains are preserved in all the simulated sequences. For example, at leaves G, F, and E, the Fer4 domain was lost. Domains also diverged in length due to insertions and deletions. Hence, REvolver produces sequences of similar, but not identical, domain architectures. The resulting pattern of presence and absence of protein domains resembles what can be observed in real protein families.

Discussion

In recent years, a number of approaches were developed to simulate evolutionary protein sequence change (e.g., Rambaut and Grassly 1997; Stoye et al. 1998; Pang et al. 2005; Rastogi et al. 2006; Fletcher and Yang 2009; Strope, Abel, et al. 2009; Grahnen, Nandakumar, et al. 2011; Lakner et al. 2011). With REvolver, we present a new versatile simulator that stands out from existing programs in

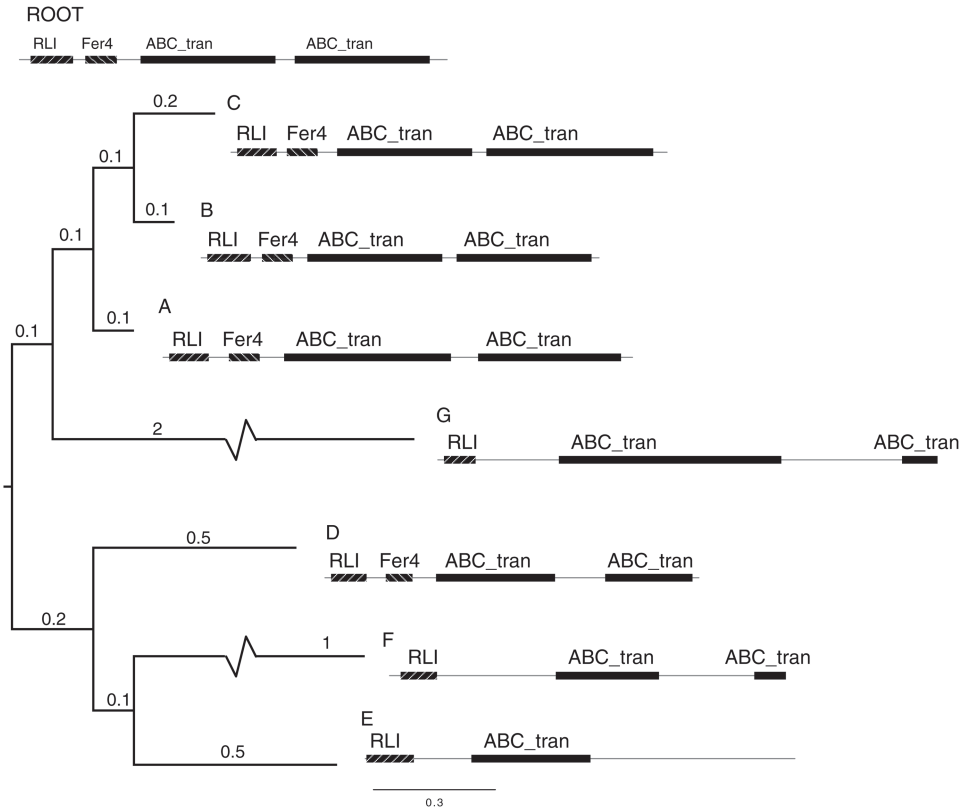

Fig. 5. Domain architectures of sequences evolved with REvolver. A root sequence with the specified domain architecture was evolved on the shown tree. The root sequence consists of one RLI (PF04068), one Fer4 (PF00037), and two ABC_tran (PF00005) domains (Finn, Mistry, et al. 2010) separated by linker regions. Domains were evolved under domain constraints, linker regions were evolved without domain constraints. Branch lengths are given in expected substitutions per site but are not drawn to scale for lengths $\geqslant 1$.

two relevant aspects: The maintenance of protein domains in the course of evolution, and the large-scale applicability due to the automatic inference of sequence-specific evolutionary constraints. We have shown that the pattern of sequence differences between homologous sequences, as captured in pHMMs, can be used to describe adequately the constrained evolutionary process to which a protein domain is subjected. REvolver is the first tool that integrates this information about protein sequence evolution in an automated fashion. To facilitate the use of pHMMs in sequence evolution simulations, we implemented several essential features. The first aspect is concerned with the modeling of insertions. We have derived the parameter for the geometric distribution used to model insertion lengths from the transition probability $P(I_x, I_x)$ of an insertion state. This transition probability was trained on an alignment of contemporary sequences. Consequently, sampling from the resulting geometric distribution results in insertions lengths that are observed in extant sequences. However, they do not necessarily represent the lengths of individual insertion events. Multiple nested insertions in the simulation would therefore result in much longer insertions than they were observed in the sequences used to train the model. To prevent the formation of such unrealistically long insertions, REvolver only extends insertions to the actually drawn random variable from the geometric distribution. Thus, the total length of an insertion in the sequence is always a value from the geometric distribution. The second aspect is concerned with the gradual erosion of *M* states due to the deletion process. We counterbalance this effect by facilitating the resurrection of *M* states via the insertion process. This is important to maintain the identity of the domains; otherwise, it would just be a matter of time until all match states have been lost and amino acids are all associated with insertion states. From the biological point of view, our procedure is also reasonable: Suppose, for example, that at one point during evolution, a functional site is deleted. This deletion may not abolish the functionality of the protein or domain but modify it. If at some point later in time, an amino acid is inserted at the previously deleted position that, by chance, has similar or the same properties as the original amino acid, the protein's function would be fully restored. In the current version of REvolver, we assess the probability that an inserted amino acid revives a previously lost *M* state using the probability that this *M* state emits exactly this amino acid. We can think of alternative ways of realizing the resurrection. One possibility would be to consider the inserted segment as a single entity rather than individual amino acids. The goal would then be to find the state path that most likely emitted that amino acid segment (Viterbi 1967). Insertion states and deleted match states would be valid states for the path, deletion states would be forbidden. However, for now, we decided to implement the stepwise insertion procedure since it is simpler and computationally less expensive.

Our comparison of REvolver to other simulators of protein sequence evolution has shown that REvolver solves two tasks in the benchmarking optimally, that is, the maintenance of 7 tm domains and maintaining a significant similarity of the simulated sequences to the GPCR protein family. However, in contrast to the other programs, for which 7 tm regions were explicitly defined and parameters had to be tweaked manually to obtain optimal performance, REvolver performed the parameterization automatically. The difference between the compared simulators becomes even more obvious in the third task, namely the maintenance of the similarity of the simulated sequences to the 7tm_1 pHMM (PF00001). This pHMM models a 7 tm receptor domain, which is the characteristic for the GPCR protein family (Palczewski et al. 2000). Although the similarity between the sequences generated with the existing simulators and the 7tm_1 pHMM is poor, sequences simulated with our program achieve average bit scores (102.8) that are only slightly lower than what is achieved on average when comparing real GPCRs to the pHMM (124.4). Thus, REvolver not only preserves the correct number of tm domains but also the intervening regions required for placing them in a functional context of a 7 tm receptor. This result suggests that REvolver may also conserve structural properties of protein domains, although they are not embedded in the pHMMs (but see Eddy 1998). To follow this issue up further, we simulated the evolution of the human SAP SH2 protein both with and without domain constraints and determined the RMSD between the structure of the native protein and the inferred structure of the simulated sequences. The results confirmed that, indeed, the simulation of sequence evolution under domain constraints not only maintains domain sequences but also has a positive influence on the preservation of their structure.

So far, we have demonstrated the use of REvolver only in the combination with pHMMs derived from public databases. However, REvolver simulations under domain constraints are applicable to all proteins even if they show no significant sequence similarity to any of the domains for which public pHMMs are available. Alternatively, it may be desired to use pHMMs more specific than those available in the public databases, for example, when a particular protein subfamily is analyzed. In such instances, the protocol is straightforward: For any given root sequence, homologous sequences can first be identified, for example, via a Blast search. The root together with a set of homologous sequences can then be aligned and used to construct and train a pHMM. REvolver then uses this custom pHMM to infer the evolutionary constraints for the root sequence. We have exemplified this procedure with the GPCR data set. To this end, we constructed a pHMM from the alignment of the 29 GPCRs. Next, we simulated the evolution of the GPCR protein family using this custom pHMM. The simulated sequences still retain most of the transmembrane regions, show a significant sequence similarity to the 7tm_1 domains and find only other GPCRs among the top BlastP hits (supplementary table S3, Supplementary Material online). This shows that even in the case of missing explicit information about protein specific features, REvolver still preserves most of them.

In summary, REvolver is a versatile tool for simulating evolutionary sequence change and improves in many aspects over existing simulators. Although not limited to it, one obvious application of REvolver is the generation of benchmark data sets for programs designed to trace and interpret the evolutionary signal in molecular sequences, for example, programs for sequence alignment, orthology prediction, or tree reconstruction (e.g., Felsenstein 2004; Notredame 2007; Remm et al. 2001). Testing the accuracy of these tools with real data is obviously problematic since the evolutionary history is frequently not known (cf. Chen et al. 2007). Benchmarking on sequences that have been evolved *in silico*, in principle, overcomes this problem. Still, the results are of little relevance if the scheme used for simulating sequence evolution is unrealistic (Kim and Sinha 2010). From this perspective, we expect that REvolver is a significant contribution to this field. We envision an even stronger impact when it comes to the benchmarking of programs that search for proteins with similar feature architecture (Koestler, von Haeseler and Ebersberger 2010) or that infer the function of a protein based on its domain content (Forslund and Sonnhammer 2008). The simulated evolution of a domain architecture along a tree is still in its infancies, as REvolver does not consider evolutionary events like domain shuffling and domain stealing. However, an integration of such mutation events will be a logical extension to REvolver's simulation scheme.

Supplementary Material

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

Acknowledgments

We would like to thank three anonymous reviewers for helpful comments. We kindly thank Dannie Durand for discussions and helpful comments on the manuscript, and Sebastian Schuster for critically reading the manuscript. This work was supported by the Wiener Wissenschafts-, Forschungs- und Technologie Fonds (WWTF). T.K. is supported by the DFG priority program SPP 1174 Deep Metazoan Phylogeny (HA 1628/9 to A.v.H.). A.v.H. appreciates the support from the Genome Research in Austria project Bioinformatics Integration Network III.

References

Abascal F, Posada D, Zardoya R. 2007. MtArt: a new model of amino acid replacement for arthropoda. *Mol Biol Evol.* 24:1–5.

Adachi J, Hasegawa M. 1996. Model of amino acid substitution in proteins encoded by mitochondrial DNA. *J Mol Evol.* 42:459–468.

Adachi J, Waddell PJ, Martin W, Hasegawa M. 2000. Plastid genome phylogeny and a model of amino acid substitution for proteins encoded by chloroplast DNA. *J Mol Evol.* 50:348–358.

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. 1990. Basic local alignment search tool. *J Mol Biol.* 215:403–410.

Benner SA, Cohen MA, Gonnet GH. 1993. Empirical and structural models for insertions and deletions in the divergent evolution of proteins. *J Mol Biol.* 229:1065–1082.

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.

Chen F, Mackey AJ, Vermunt JK, Roos DS. 2007. Assessing performance of orthology detection strategies applied to eukaryotic genomes. *PloS One* 2:e383.

Dayhoff M, Schwartz R, Orcutt B. 1978. A model of evolutionary change in proteins. *Atlas Protein Sequence Struct.* 5:345–352.

Eddy SR. 1998. Profile hidden Markov models. *Bioinformatics* 14:755–763.

Eswar N, Webb B, Marti-Renom MA, Madhusudhan MS, Eramian D, Shen M, Pieper U,Sali A. 2006. Comparative protein structure modeling using modeller. *Curr Protoc Bioinformatics* Chapter 5:Unit 5.6.

Felsenstein J. 2004. Inferring phylogenies. Sunderland (MA): Sinauer Associates.

Finn RD, Mistry J, Tate J, et al. (14 co-authors). 2010. The Pfam protein families database. *Nucleic Acids Res.* 38:D211–D222.

Fletcher W, Yang Z. 2009. INDELible: a flexible simulator of biological sequence evolution. *Mol Biol Evol.* 26:1879–1888.

Forslund K, Sonnhammer ELL. 2008. Predicting protein function from domain content. *Bioinformatics* 24:1681–1687.

Gillespie DT. 1977. Exact stochastic simulation of coupled chemical reactions. *J Phys Chem.* 81:2340–2361.

Grahnen JA, Kubelka J, Liberles DA. 2011. Fast side chain replacement in proteins using a coarse-grained approach for evaluating the effects of mutation during evolution. *J Mol Evol.* 73:23–33.

Grahnen JA, Nandakumar P, Kubelka J, Liberles DA. 2011. Biophysical and structural considerations for protein sequence evolution. *BMC Evol Biol.* 11:361.

Henikoff S, Henikoff JG. 1992. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A.* 89:10915–10919.

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

Karlin S, Taylor H. 1975. A first course in stochastic processes. 2nd ed. San Diego (CA): Academic Press.

Kim J, Sinha S. 2010. Towards realistic benchmarks for multiple alignments of non-coding sequences. *BMC Bioinformatics.* 11:54.

Koestler T, von Haeseler A, Ebersberger I. 2010. FACT: functional annotation transfer between proteins with similar feature architectures. *BMC Bioinformatics.* 11:417.

Kosiol C, Goldman N. 2005. Different versions of the dayhoff rate matrix. *Mol Biol Evol.* 22:193–199.

Laity JH, Lee BM, Wright PE. 2001. Zinc finger proteins: new insights into structural and functional diversity. *Curr Opin Struct Biol.* 11:39–46.

Lakner C, Holder MT, Goldman N, Naylor GJP. 2011. What's in a likelihood? Simple models of protein evolution and the contribution of structurally viable reconstructions to the likelihood. *Syst Biol.* 60:161–174.

Le SQ, Gascuel O. 2008. An improved general amino acid replacement matrix. *Mol Biol Evol.* 25:1307–1320.

Letunic I, Doerks T, Bork P. 2009. SMART 6: recent updates and new developments. *Nucleic Acids Res.* 37:D229–D232.

Müller T, Vingron M. 2000. Modeling amino acid replacement. *J Comput Biol A J Comput Mol Cell Biol.* 7:761–776.

Nickle DC, Heath L, Jensen MA, Gilbert PB, Mullins JI, Pond SLK. 2007. HIV-specific probabilistic models of protein evolution. *PloS One* 2:e503.

Notredame C. 2007. Recent evolutions of multiple sequence alignment algorithms. *PLoS Comput Biol.* 3:e123.

Palczewski K, Kumasaka T, Hori T, et al. (12 co-authors). 2000. Crystal structure of rhodopsin: A g Protein-Coupled receptor. *Science* 289:739 –745.

Pang A, Smith AD, Nuin PAS, Tillier ERM. 2005. SIMPROT: using an empirically determined indel distribution in simulations of protein evolution. *BMC Bioinformatics.* 6:236.

Parisi G, Echave J. 2001. Structural constraints and emergence of sequence patterns in protein evolution. *Mol Biol Evol.* 18:750–756.

Poy F, Yaffe MB, Sayos J, Saxena K, Morra M, Sumegi J, Cantley LC, Terhorst C, Eck MJ. 1999. Crystal structures of the XLP protein SAP reveal a class of SH2 domains with extended, phosphotyrosine-independent sequence recognition. *Mol Cell.* 4:555–561.

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

Rastogi S, Reuter N, Liberles DA. 2006. Evaluation of models for the evolution of protein sequences and functions under structural constraint. *Biophys Chem.* 124:134–144.

Remm M, Storm CE, Sonnhammer EL. 2001. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J Mol Biol.* 314:1041–1052.

Schuster-Bockler B, Schultz J, Rahmann S. 2004. HMM logos for visualization of protein families. *BMC Bioinformatics* 5:7.

Söding J. 2005. Protein homology detection by HMM–HMM comparison. *Bioinformatics* 21:951–960.

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

Strope CL, Abel K, Scott SD, Moriyama EN. 2009. Biological sequence simulation for testing complex evolutionary hypotheses: indel-Seq-Gen version 2.0. *Mol Biol Evol.* 26:2581–2593.

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.

The UniProt Consortium. 2010. Ongoing and future developments at the universal protein resource. *Nucleic Acids Res.* 39: D214–D219.

Thorne JL, Kishino H, Felsenstein J. 1991. An evolutionary model for maximum likelihood alignment of DNA sequences. *J Mol Evol.* 33:114–124.

Tusnády GE, Simon I. 2001. The HMMTOP transmembrane topology prediction server. *Bioinformatics* 17:849–850.

Velankar S, Alhroub Y, Alili A, et al. (33 co-authors). 2011. PDBe: protein data bank in europe. *Nucleic Acids Res.* 39:D402–D410.

Viterbi A. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans Inf Theory.* 13:260–269.

von Haeseler A, Schöniger M. 1998. Evolution of DNA or amino acid sequences with dependent sites. *J Comput Biol A J Comput Mol Cell Biol.* 5:149–163.

Whelan S, Goldman N. 2001. A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Mol Biol Evol.* 18:691–699.

Wilgenbusch JC, Swofford D. 2003. Inferring evolutionary trees with PAUP*. *Curr Protoc Bioinformatics* Chapter 6:Unit 6.4.

Yang Z, Nielsen R, Hasegawa M. 1998. Models of amino acid substitution and applications to mitochondrial protein evolution. *Mol Biol Evol.* 15:1600–1611.