Accepted for publication by Molecular Biology and Evolution

RESEARCH ARTICLE

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

Cory L. Strope[1],

Stephen D. Scott[1], and Etsuko N. Moriyama[2*],

[1]Department of Computer Science and Engineering, University of Nebraska-Lincoln

[2]School of Biological Sciences and Plant Science Initiative, University of Nebraska-Lincoln

*Corresponding author address:
N107 Beadle Center for Genetics Research, University of Nebraska – Lincoln,
Lincoln, NE 68588-0660
Phone: (402) 472-4979, Fax: (402) 472-3139, Email: emoriyama2@unl.edu

keywords: Protein superfamily, sequence simulation, domains, motifs, indels

running head: indel-Seq-Gen Protein Family Simulator

Abstract

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

Introduction

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

Rigorous evaluation of multiple alignment and phylogenetic methods, especially for highly diverged sequences with large numbers of amino acid insertions or deletions (indels), is possible if we have a tool that can simulate realistic sequence evolution incorporating indel events based on a valid biological model derived from empirical data. However, current simulation methods are mostly focused on reconstructing *substitution* events (for both nucleotides and amino acids) over entire protein sequences: e.g., `evolver` (Yang 1997) and the simulator in Molphy (Adachi 1996). They cannot accommodate the numerous evolutionary forces that act upon functional domains found in families of proteins. These domains frequently, but not always, coincide with structural domains, and are connected together by amino acid sequences that have no rigid structure (e.g., random coils). Conservation of these domain regions often imposes different evolutionary constraints, including different evolutionary models (e.g., amino acid frequencies, substitution rates) and indel parameterizations (e.g., maximum indel size, length distribution, and probability of opening an indel). Smaller functional units, or motifs, embedded within domain regions may require other

specific models, such as invariability of amino acid positions and prevention of indel events within the motif. Seq-Gen (Rambaut and Grassly 1997) introduced substructures that are allowed to have different lengths and evolve along different evolutionary trees. However, one cannot change evolutionary pressures acting upon the substructures, such as differing substitution models and motif conservation. SISSI (Gesell and von Haeseler 2006) accounts site-specific interactions in its nucleotide sequence simulation, enabling a process such as RNA evolution with secondary-structure constraints. To apply SISSI for protein sequence evolution, however, we need information on residue interactions in greater detail than we currently have.

While indels are a necessary component for portraying sequence evolution, their addition in sequence simulation poses a number of problems. One such set of problems relates to the indel creation itself, i.e., where to and where not to place indels, the distribution of indel lengths, and how to generate inserted sequences. The pioneering application for the addition of indel events during simulation is ROSE (Stoye, Evers, and Meyer 1998). ROSE implements indel events using a simple, strict model: Indels occur linearly with respect to evolutionary distance and with a user-defined length distribution. The sequence inserted is a random sequence based on the given amino-acid frequencies. Empirical indel models whose indel probabilities are non-linear with respect to evolutionary distance, such as Benner, Cohen, and Gonnet (1993) and Chang and Benner (2004), cannot be utilized in ROSE. User-input parameters are also global across the entire sequence, not allowing for domain-specific interactions.

Recently, several other sequence simulators including indels have been developed: SIMPROT (Pang et al. 2005), MySSP (Rosenberg 2005), EvolveAGene (Hall 2004), and DAWG (Cartwright 2005). SIMPROT can generate protein sequences, and like Seq-Gen, allows the user to create subsequences evolving using different parameters, in the end

concatenating these subsequences into a single sequence. However, SIMPROT does not include the options for the input of a root sequence, the preservation of sequence motifs, or changing amino acid frequencies between subsequences.

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

Methods

Protein Sequence Evolution Engine without Indels

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

Protein Family Creation

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

Motif conservation. The conservation of essential motifs in sequence generation can be accomplished by disallowing substitution events within the motifs. Invariable sites are introduced for this purpose. However, when indels are incorporated, the conservation of length-sensitive motifs becomes a problem. For example, ROSE allows for the preservation of motifs by joining the gamma-distribution rates and invariable sites into one mutation probability vector, which we call the I+γ array. Each site in the I+γ array specifies the substitution rate for the corresponding amino acid site. If the rate is 0.0, then the site is invariable. If the rate is less than 1.0, the site is not allowed to have any indel. Tying indels to the relative substitution frequencies is a drawback in this representation, since it cannot represent low-frequency indels in regions that have high substitution rates,

such as transmembrane regions. Neither can it represent highly variable sites in length-based motifs that depend on the distance between particular amino acids, such as the 'CXXC' motif. When shorter or longer, this motif loses its propensity for creating the disulfide bridges that are essential for Thioredoxin-fold proteins (Chivers, Laboissière, and Raines 1996).

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

Heterogeneous evolution among domains. Seq-Gen (Rambaut and Grassly 1997) introduced the idea of *partitions* to allow variations in evolutionary rates and patterns among domains. Each partition represents a subsequence of a protein, and each partition evolves independently as specified by the evolutionary tree. However, more variations in evolutionary patterns need to be incorporated to represent, for example, different indel rates in secondary-structure regions compared to non-structured coil regions (Thorne 2000). In iSG, the percentage of invariable sites, branch lengths (representing substitution rates), amino acid frequencies, substitution models, and indel rates can all be varied

between partitions. Such flexible options allow us to generate realistically complex protein families.

Indel Event Handling

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

Probability of indels. In iSG, the following two indel-models are used:

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

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

$$P(\text{indel}) = 0.224 - 0.0219e^{-0.01168d}. \quad (1)$$

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

Length distribution of indels. The distribution of gap lengths in protein sequences has been studied through the use of pairwise alignments (Chang and Benner 2004) as well as

by examining structural databases (Qian and Goldstein 2001; Goonesekere and Lee 2004). By default, iSG will create the normalized Zipfian distribution for indel lengths (Chang and Benner 2004):

$$N = 2628(X^{-1.821}), \quad (2)$$

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

Insertion sequence options. iSG incorporates two methods of amino acid sequence generation. The random model randomly chooses amino acids based on the given frequencies. The second method is unique because it chooses amino acids in the insertion sequence by incorporating the "neighbor preference" (Xia and Xie 2002) along with the amino acid frequencies. Neighbor preference was derived by studying neighboring amino acid positions in functional proteins in order to empirically determine the effect of the side chains in the acceptance of functionally beneficial amino acid changes. We use their 20X20 matrix to generate insertion sequences in a Bayesian fashion:

$$P(j \mid i) = \frac{P(i \mid j)P(j)}{P(i)} \quad (3)$$

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

Originally, neighbor preferences were built on the protein sequences derived from the

Escherichia coli K-12 genome. We also provide neighbor preferences calculated over all of the protein sequences contained in the Swiss-Prot database (Bairoch et al. 2005). The neighbor preferences are global over the entire simulation run of a family of sequences.

Root Sequence Options

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

- The range of columns in the multiple alignment to use.

- The number of sequences used. All (the default) or a given number of sequences can be selected from the multiple alignment using the bootstrap-sampling method (selecting randomly with replacement). The bootstrap-sampling option is useful, for example, when

using highly divergent sequences with a large number of gap sites. The simple consensus from such divergent sequences may include mainly gaps and very few amino acids. On the other hand, when there are many equally probable root sequences in S, the user may opt to randomly select one sequence as the root for each simulation run.

• The root sequence generation method. The default consensus method uses the majority-rule to choose the amino acid for the column, using a coin toss to break ties. The random method randomly chooses the amino acid representing the column based on the position-specific amino acid frequencies, with the exception of invariable columns. For invariable columns, the representative amino acid is chosen using the consensus method.

Implementation

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

Comparison of Simulation Methods

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

Beta-strand prediction. For the lipocalin family simulation, we used PSIPRED (Jones 1999) to predict the secondary structures. Realistically simulated sequences should have

eight beta-strands that correspond to the beta-barrel structure found in the lipocalin family proteins.

BLAST similarity search. In order to see how the simulated sequences preserved the similarities against the template proteins, we performed BLASTP (Altschul et al. 1997) protein similarity searches against the protein database UniProt (Bairoch et al. 2005), using the simulated sequences as the queries. We did not use the option to filter out the low-complexity sequences. The default E-value threshold (10) was used to cut off the search results.

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

Parametric bootstrap and phylogenetic analysis. Following the phylogenetic tree reconstructed from the template alignment (see Results section), 1000 simulated datasets were generated, each including the same number of protein sequences as in the template set. Phylogenetic trees were reconstructed from these datasets. The maximum parsimony and neighbor-joining (Saitou and Nei 1987) with JTT distance (Jones, Taylor and Thornton 1992) estimation methods, consensus trees, and bootstrap values were calculated using Phylip programs (version 3.65; Felsenstein 2005).

Results and Discussion

Simulation Setup

Template multiple alignments. We chose to model protein sequences based on two protein family datasets: (1) the vertebrate olfactory receptor family and (2) the lipocalin protein family.

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

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

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

Lipocalins: Lipocalin proteins are a family of small, globular proteins belonging to the calycin superfamily. They are often implicated in allergic reactions. Members of the lipocalin family are highly diverged, but all share a conserved beta-barrel conformation consisting of eight beta-strands.

The template multiple alignment and phylogeny for the lipocalin family including 23 sequences were obtained from an evolutionary study done by Sánchez et al. (2003). The sequences were split into 12 regions including five beta-strand regions as shown in Fig. 3c. Note that the region including four short beta-strand regions (B_{5678} in Fig. 3c) is treated as a single evolution unit. Branch lengths were estimated from each segment as described before. Four coil regions (indicated by 'C' in Fig. 3c) were very short (each with 4-6 amino acids in length). Therefore, their branch lengths were obtained based on their concatenated sequences. The amino acid frequencies specific to the beta-strand, coil,

or alpha-helix regions were calculated using those obtained from the template multiple alignment (75% weighting) combined with pseudocounts (25% weighting). The pseudocounts were obtained either from the conformational parameters by Chou and Fasman (1974) for beta-strands and alpha-helices, or from Jones, Taylor, and Thornton (1992) for the N-terminus, C-terminus, and coil regions.

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

1. TM1, 2, 4 and 6 contain more gaps than TM3, 5 and 7, and

2. The loop regions between TM4 and TM7 (EC4, CY3, and EC5) are more diverged than other regions.

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

Comparison of Simulated Sequences between iSG and Other Methods

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

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

One thousand datasets were simulated using each method. The true multiple alignments showing the actual indel positions were collected. We compared the true alignments

generated by iSG and ROSE against the profile HMM built from the template GPCR alignment (Supplementary Fig. 3). No significant differences were shown between the normal score distributions obtained from iSG (mean = 75.03, variance = 17.20) and ROSE (mean = 72.30, variance = 16.72), indicating that the sequences generated by the two methods using the set of the parameters we chose were approximately equivalent with respect to the template multiple alignment.

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

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

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

Conservation of transmembrane regions. To check if the TM regions were retained in the reconstructed datasets, we predicted TM regions from simulated GPCR sequences. The numbers of predicted TM regions were 7.03±0.30 by iSG, 5.94±1.25 by ROSE, 0.20±0.37 by SIMPROT, and 6.84±0.91 by Seq-Gen. Clearly iSG has the greatest accuracy in reproducing the seven TM regions. The N-terminus (EC1) was correctly predicted to be extracellular 94% of the time in simulated datasets of iSG. Conversely, with ROSE and Seq-Gen, only 44% and 55% of simulated sequences, respectively, were predicted to have the extracellular N-terminus correctly. These results clearly show that iSG's ability to allow different amino acid frequencies between domains is effective in preserving important features of transmembrane proteins more accurately than other methods. This is important since results from Panchenko and Madej (2005) suggested

patterns found in non-domain regions, such as loop or N/C-terminal regions, were more important in recognizing superfamily members than considering only the conserved core regions.

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

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

BLAST and PFAM search results. In order to further examine how simulated sequences preserved functionally important sequence properties, we used the simulated sequences as queries for BLASTP protein similarity search. Table 1 summarizes the search results. The top 250 hits obtained by all methods except SIMPROT were correctly from various members of vertebrate olfactory receptors. Seq-Gen performed the best of

the three methods indicated by the highest average scores. The absence of indels in Seq-Gen generated sequences must have caused a low percentage of gaps in the alignments. iSG performed nearly as well as Seq-Gen, and outperformed ROSE with higher scores and lower percentages of gaps. The lower percentage of gaps found in BLAST alignments with iSG sequences than those with ROSE sequences can be explained by the use of amino acid frequencies specific to subsequences (e.g., TM, extracellular, and cytosolic regions) in iSG. It must have produced amino acids that are most likely to appear in each region of GPCRs. As a result, fewer gaps were required in the iSG alignment making better hit scores than in ROSE. Contrary to iSG, ROSE, and Seq-Gen, simulated sequences by SIMPROT did not find any vertebrate olfactory receptor protein under the E-value threshold used with the BLASTP search. Within the threshold, SIMPROT sequences averaged between 3-5 unrelated hits, with the average highest score = 30 (E-value = 1.5).

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

For simulated lipocalin proteins, BLASTP and PFAM search did not produce as many significant hits as we observed with GPCR sequences, which implies the difficulty in simulating beta-strand proteins. As shown in Table 2, the BLAST results were comparable among the simulation methods except for SIMPROT. Only about two

lipocalins were found per simulated sequence query. Interestingly, slightly more different lipocalin sequences were found among simulation sets produced by iSG than ROSE and Seq-Gen. This is probably a result of the multiple alignment root sequence option in iSG changing the root sequences between datasets. Against the PFAM Lipocalin profile HMM entry, iSG had the lowest scores. However, iSG produced the highest number of sequences (33.62%) that showed a hit to the Lipocalin profile, compared to ROSE and Seq-Gen (25.68% and 22.41%, respectively). Again, simulated sequences by SIMPROT did not find any lipocalin profile HMM.

Note that, as described earlier, the root sequences used for ROSE and Seq-Gen were constructed by the "consensus method" we incorporated in iSG. In this method, a consensus sequence was constructed based on the reference multiple alignment and this sequence was used as the root sequence for ROSE and Seq-Gen. This may explain the similar performances observed by BLASTP and PFAM among iSG, ROSE, and Seq-Gen. Neither ROSE nor Seq-Gen in their original methods included this capability.

Phylogenenetic analysis with parametric bootstrap. Neighbor-joining phylogenies were reconstructed based on the T-Coffee as well as the "true" multiple alignments of simulated sequences. For both GPCR and lipocalin simulations using the parameters described earlier, the topologies of the consensus trees obtained from all four methods were identical to the original phylogenies reconstructed from the template multiple alignments, and all nodes were supported with 99% or better bootstrap values. To explore the effect of indels on phylogenetic reconstruction, we simulated another 100 datasets of GPCRs and lipocalins using iSG, with twice higher substitution rates and with and without indels as shown in Table 3. When substitution rates were not changed ("Substitution rate X 1") indels did not significantly affect the phylogenetic reconstructions. However, when the substitution rates were doubled ("Substitution rate X

2"), for both GPCR and lipocalin simulations, the branch supports became lower when indels were incorporated in the sequence evolution. For example, the number of internal branches supported by 90% or higher was 9 for the lipocalin phylogeny based on T-Coffee alignments when indel was not incorporated. However, the same number decreased to only 1 when indels were incorporated. Similar results were obtained for GPCR simulations although the effect was less drastic. The average branch support value (80.2%) for simulated lipocalin phylogenies using T-Coffee multiple alignments with indels was 10 percent lower than when no indel was incorporated. Note also that when indels were incorporated, phylogenies based on T-Coffee alignments were less supported than those based on the true multiple alignments. Phylogenies based on simulated GPCR sequences were consistently more supported than those based on simulated lipocalin sequences, and incorporating indels showed only small decreases in branch supports. This is because the lipocalin template sequences are more diverged (the average pairwise sequence identity = 13.7%) than GPCR template sequences (the average pairwise sequence identity = 35.4%). Thus, our simulations generated more diverged lipocalin-like sequences than GPCR-like sequences.

Conclusion

indel-Seq-Gen, or iSG, is a new method for simulating protein sequence evolution and generating realistic protein families. It has a unique ability to simulate heterogeneous evolution of multi-domain protein families in addition to the incorporation of indel evolution. iSG is highly flexible and various sequence features can be easily included in the simulation. For example, iSG allows highly variable regions in the sequence while still maintaining low indel rates. This is useful in simulating motifs such as 'CXXC' in Thioredoxin-fold proteins or zinc-fingers. This is not possible in other simulation methods. The use of a template multiple alignment is also a unique strength of iSG. With

these new functions, iSG can be used effectively for testing the performance of various molecular evolution and bioinformatics methods when they are applied to extremely divergent heterogeneous protein analysis. Since each subsequence can be simulated with a different evolutionary tree, lateral gene transfer and recombination detection methodologies can also be examined. For protein families with well-known features, iSG can be used to generate synthetic model sequences. Such model sequences can be used to search their candidate members from databases.

Supplementary Material

Supplementary Figures 1 - 4 and Table 1 are available at Molecular Biology and Evolution online.

Acknowledgments

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

Literature Cited

Adachi J, Hasegawa M. 1996. MOLPHY version 2.3: programs for molecular phylogenetics based on maximum likelihood. Computer Science Monographs 28:1–150.

Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucl. Acids Res. 25:3389-3402.

Bairoch A, Apweiler R, Wu CH, et al. (15 co-authors). 2005. The universal protein resource (UniProt). Nucl. Acids Res. 33:D154–D159.

Bateman A, Coin L, Durbin R, et al. (13 co-authors). 2004. The Pfam protein familiesdatabase. Nucl. Acids Res. 32:D138–D141.

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

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

Chou PY, Fasman GD. 1974. Prediction of protein conformation. Biochemistry 13:222-245.

Eddy, SR. (1998). Profile hidden Markov models. Bioinformatics 14:755-763.

Edgar RC, Sjolander K. 2004. COACH: profile-profile alignment of protein families

using hidden Markov models. Bioinformatics 20:1309–1318.

Felsenstein J. 2005. PHYLIP (PHYLogeny Inference Package) Version 3.65. Distributed by the author. Department of Genome Sciences, University of Washington, Seattle.

Flower DR, North ACT, Attwood TK. 1993. Structure and sequence relationships in the lipocalins and related proteins. Prot. Sci. 2:753-761.

Fuchs T, Glusman G, Horn-Saban S, Lancet D, Pilpel Y. 2001. The human olfactory subgenome: from sequence to structure and evolution. Hum. Genet. 108:1–13.

Gesell T, Haeseler A. 2006. In silico sequence evolution with site-specific interactions along phylogenetic trees. Bioinformatics 22:716-722.

Gilad Y, Man O, Glusman G. 2005. A comparison of human and chimpanzee olfactory receptor gene repertoires. Genome Research 15:224–230.

Gooneskere NCW, Lee B. 2004. Frequency of gaps observed in a structurally aligned protein pair database suggests a simple gap penalty function. Nucl. Acids Res. 32:2838–2843.

Grassly NC, Adachi J, Rambaut A. 1997. PSeq-Gen: an application for the Monte Carlo simulation of protein sequence evolution along phylogenetic trees. CABIOS 13:559-560.

Hall BG. 2004. Comparison of the accuracies of several phylogenetic methods using protein and DNA sequences. Mol. Biol. Evol. 22:792–802.

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

Jones, DT. 1999. Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. 292:195-202.

Notredame C, Higgins DG, Heringa J. 2000. T-Coffee: A novel method for fast and accurate multiple sequence alignment. J. Mol. Biol. 302:205–217.

Panchenko AR, Madej T. 2000. Structural similarity of loops in protein families: toward the understanding of protein evolution. BMC Evolutionary Biology 5.

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.

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

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

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

Saitou N, Nei M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. 4:406–425.

Sánchez D, Ganfornina MD, Gutiérrez G, Marín A. 2003. Exon-intron structure and evolution of the lipocalin gene family. Mol. Biol. Evol. 20:775-783.

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

Swofford D. 2003. PAUP*: Phylogenetic analysis using parsimony (*and other methods). Version 4. Sunderland, MA: Sinauer Associates.

Thorne J. 2000. Models of protein sequence evolution and their applications. Curr. Opin. Genet. Dev. 10:602-605.

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

Xia X, Xie Z. 2002. Protein structure, neighbor effect, and a new index of amino acid dissimilarities. Mol. Biol. Evol. 19:58–67.

Yang Z. 1993. Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. Mol. Biol. Evol. 10:1396–1401.

Yang Z. 1997. PAML: a program package for phylogenetic analysis by maximum likelihood. Comput. Appl. Biosci. 13:555-556.

Table 1

The results of BLASTP and PFAM search when using simulated GPCR sequences as the query

Method	Number of top hits	avg[a]	min[a]	max[a]	%ID[b]	%SIM[b]	%GAP[b]
BLASTP							
iSG	25	174.0	115.0	303.2	34.3	55.9	3.2
	100	164.7	106.0	303.2	33.1	54.6	3.3
	250	155.9	95.8	303.2	32.1	53.5	3.4
ROSE	25	141.1	93.6	258.8	35.3	53.1	7.5
	100	132.7	85.4	258.8	34.1	52.1	7.6
	250	124.4	74.4	258.8	33.0	51.0	7.6
Seq-Gen	25	196.7	132.2	324.8	34.5	55.9	0.0
	100	187.3	122.2	324.8	33.5	54.8	0.4
	250	177.8	110.2	324.8	32.5	53.7	0.5
SIMPROT[c]	-	-	-	-	-	-	-
PFAM[d]							
iSG	7tm_1	-5.09	-99.5	158.8			
ROSE	7tm_1	-31.47	-114.9	97.7			
Seq-Gen	7tm_1	-7.18	-129.2	160.1			
SIMPROT[c]	7tm_1	-	-	-			

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

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

[c]SIMPROT queries found only 3-5 unrelated hits by BLASTP. PFAM profile HMM search did not find 7tm_1.

[d]Only scores for the most commonly found PFAM model, 7tm_1, are shown.

Table 2

The results of BLASTP and PFAM search when using simulated lipocalin sequences as the query

Method	Number of BLASTP hits[a]	Number of different lipocalin hits[b]	Average PFAM score[c]	%hits[d]
iSG	2.22 (43.36)	60	-30.12	33.62
ROSE	2.25 (42.16)	56	-18.34	25.86
Seq-Gen	2.31 (45.87)	56	-14.14	22.41
SIMPROT	0 (-)	0	-	0

[a]The average number of BLASTP hits to lipocalins for each simulated sequence. The average bit score for these lipocalin hits is shown in parentheses.

[b]The total number of different lipocalin sequences found by BLASTP using all simulated sequence queries.

[c]The average score against the 'Lipocalin' profile HMM among queries. SIMPROT queries did not obtain 'Lipocalin' hit.

[d]The proportion of simulated sequences that are matches to the PFAM Lipocalin model.

Table 3

Phylogenetic analysis using simulated GPCR and lipocalin sequences with various substitution and indel rates

% support[b]	Substitution rate X 1[a]				Substitution rate X 2[a]			
	No indel		With indel[a]		No indel		With indel[a]	
	Lipocalin[c]	GPCR[c]	Lipocalin[c]	GPCR[c]	Lipocalin[c]	GPCR[c]	Lipocalin[c]	GPCR[c]
$x = 100$	14 (14)	25 (25)	11 (12)	25 (26)	0 (3)	22 (21)	0 (0)	15 (18)
$90 \le x < 100$	6 (6)	1 (1)	9 (8)	1 (0)	9 (12)	4 (5)	1 (15)	11 (8)
$80 \le x < 90$	0 (0)	0 (0)	0 (0)	0 (0)	5 (1)	0 (0)	7 (1)	0 (0)
$70 \le x < 80$	0 (0)	0 (0)	0 (0)	0 (0)	1 (0)	0 (0)	5 (2)	0 (0)
$x < 70$	0 (0)	0 (0)	0 (0)	0 (0)	5 (4)	0 (0)	7 (2)	0 (0)
Average % support[d]	98.9 (99.0)	100 (100)	98.6 (98.8)	100 (100)	80.2 (90.4)	99.5 (99.5)	70.9 (86.4)	98.9 (99.7)

[a]Substitution and indel rates used in iSG are the same as those used in Supplementary Fig. 1 or twice higher.

[b]Branch supporting values from the parametric bootstrap analysis with 100 simulations.

[c]The number of internal branches with a given range of bootstrap supporting values. T-Coffee was used to generate multiple alignments from simulated sequences. The results based on the true multiple alignments were given in parentheses.

[d]The average supporting values (%) obtained from all internal branches.

Figure 1: A flow chart showing the process of iSG protein sequence simulation. In this example, parameterization of evolutionary information is done based on the vertebrate olfactory receptor family. Steps 1–5 illustrate the process of obtaining the template multiple alignment, phylogeny, and various parameters. Step 6 shows the sample input for iSG with each line showing parameters used for a different subsequence (domain). See Supplementary Fig. 2 for the example input file. Two evaluation methods, phylogenetic analysis and alignment comparison using profile HMMs, are shown in dashed boxes.

Figure 2: Multiple alignments of the template and simulated GPCR sequences. Each pixel represents one position in the multiple alignment. The color of the pixel represents: a gap (white), an amino acid from a transmembrane region (gray), and an amino acid not from a transmembrane region (black). The template alignment (a) includes 29 GPCR sequences, and the 15 subsequence regions are indicated above. For five simulated datasets produced by iSG (b) and ROSE (c), the "true" multiple alignments and multiple alignments reconstructed by T-Coffee are shown.

Figure 3: Distribution of predicted beta-strands along the lipocalin sequences. (a) Proportions of predicted beta-strands are plotted for each amino acid position of the alignments. For the reference alignment, the proportions are based on 23 lipocalin sequences. For simulated sequences, average proportions were plotted based on 5 simulated datasets using their true alignments obtained from each simulation method. Seven regions are mainly gaps with amino acids inserted into a few sequences (less than

half the sequences have an amino acid). The proportions of predicted beta-strands in these seven regions are represented by a single dot. These regions correspond to those in which indels are allowed (3, 7, 9, B and F as well as the start and end regions in the reference alignment, see the region labels in figure b). For each method, the average numbers of positions predicted as beta-strands are calculated from subsequence regions simulated as beta-strands (regions 2, 4, 6, 8, A, C, E, G in figure b) and coils (regions 1, 3, 5, 7, 9, B, D, F, H in figure b). These values are shown as 'Mean-beta' and 'Mean-coil', respectively, in each plot. (b) The reference lipocalin alignment obtained from Sánchez et al. (2003) is schematically shown with a single pixel representing one amino acid. The grey pixels represent beta-strands. The alignment is 213 positions long and consists of 23 lipocalins. Seventeen regions are labeled 1 through H. (c) Twelve subsequence regions used for simulation. The region B_{5678} concatenated short consecutive regions from 9 to G.

Figure 1

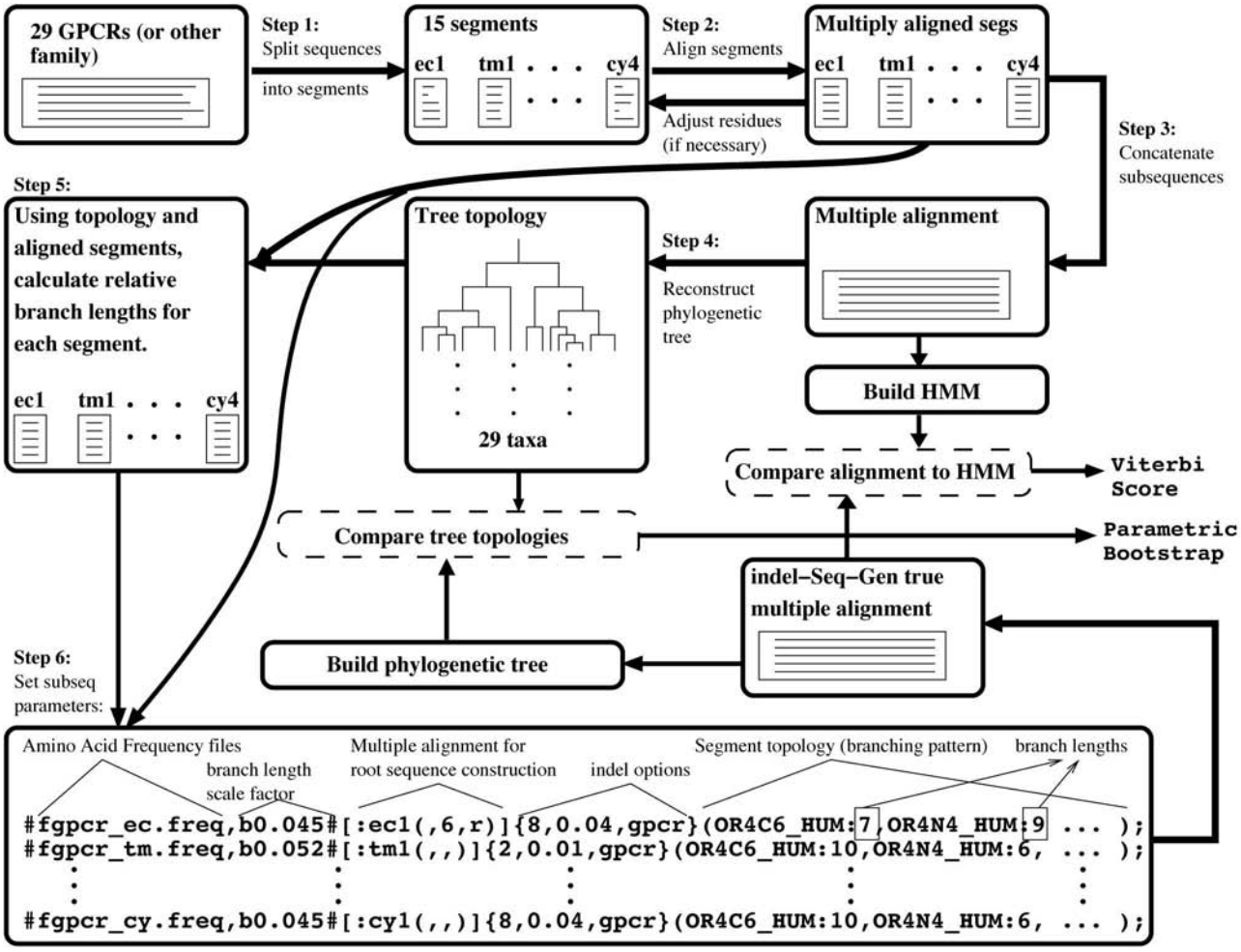

29 GPCRs (or other family)
Step 1:
Split sequences into segments
15 segments
ec1 tm1 . . . cy4
Step 2:
Align segments
Adjust residues (if necessary)
Multiply aligned segs
ec1 tm1 . . . cy4
Step 3:
Concatenate subsequences
Step 5:
Using topology and aligned segments, calculate relative branch lengths for each segment.
ec1 tm1 . . . cy4
Tree topology
29 taxa
Step 4:
Reconstruct phylogenetic tree
Multiple alignment
Build HMM
Compare alignment to HMM
Viterbi Score
Compare tree topologies
Parametric Bootstrap
indel−Seq−Gen true multiple alignment
Build phylogenetic tree
Step 6:
Set subseq parameters:
Amino Acid Frequency files
branch length scale factor
Multiple alignment for root sequence construction
indel options
Segment topology (branching pattern)
branch lengths
#fgpcr_ec.freq,b0.045#[:ec1(,6,r)]{8,0.04,gpcr}(OR4C6_HUM:7,OR4N4_HUM:9 ...);
#fgpcr_tm.freq,b0.052#[:tm1(,,)]{2,0.01,gpcr}(OR4C6_HUM:10,OR4N4_HUM:6, ...);
#fgpcr_cy.freq,b0.045#[:cy1(,,)]{8,0.04,gpcr}(OR4C6_HUM:10,OR4N4_HUM:6, ...);

Figure 2

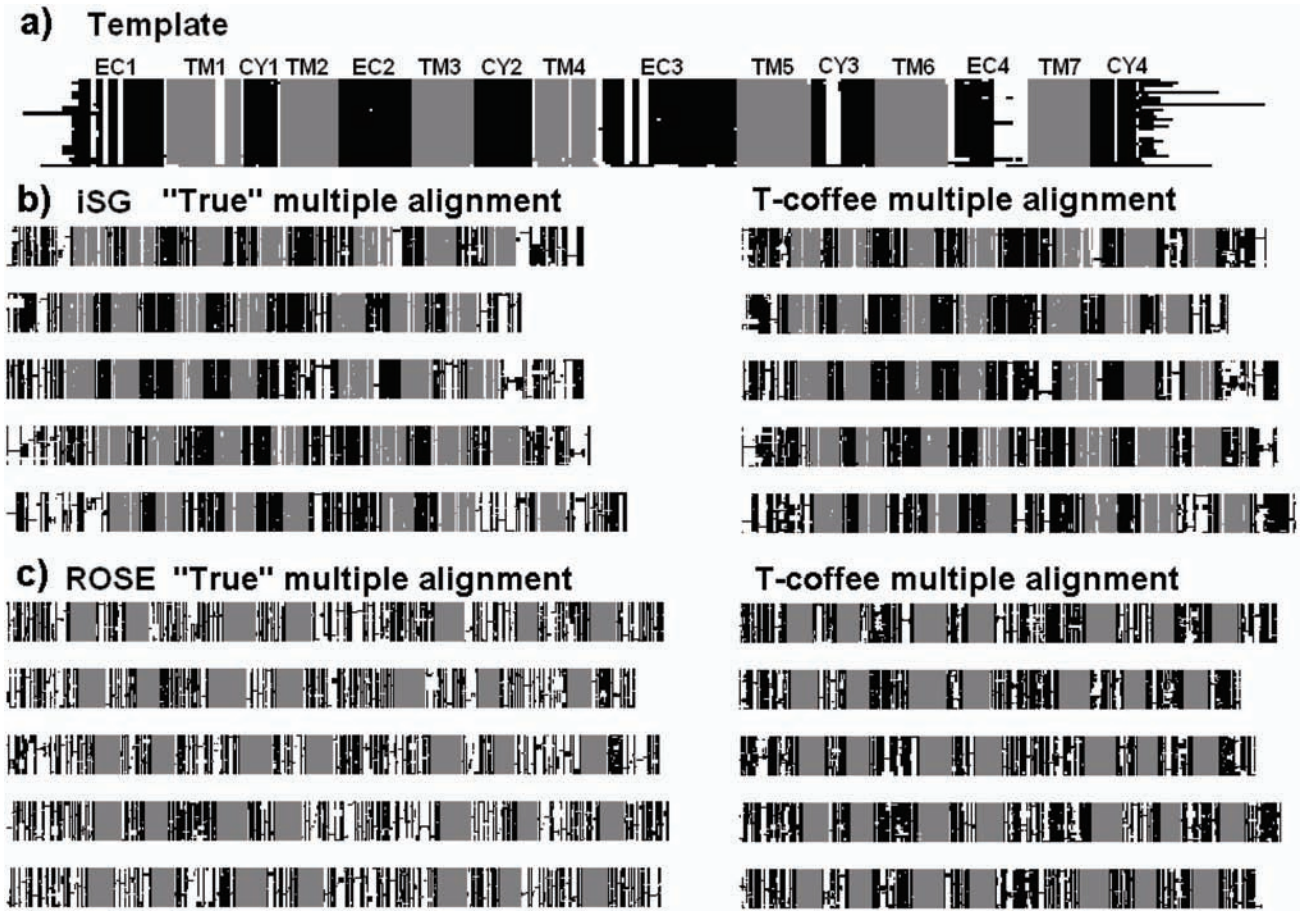
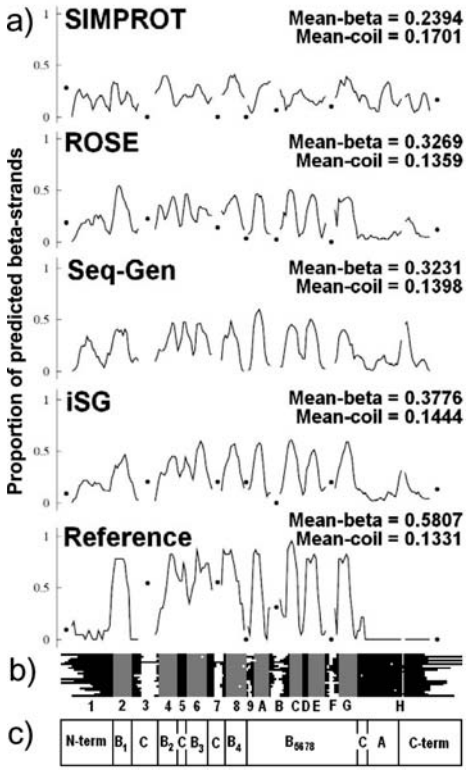

a) Template
EC1
TM1
CY1
TM2
EC2
TM3
CY2
TM4
EC3
TM5
CY3
TM6
EC4
TM7
CY4
b) iSG "True" multiple alignment
T-coffee multiple alignment
c) ROSE "True" multiple alignment
T-coffee multiple alignment

Figure 3


a)
SIMPROT
Mean-beta = 0.2394
Mean-coil = 0.1701
ROSE
Mean-beta = 0.3269
Mean-coil = 0.1359
Seq-Gen
Mean-beta = 0.3231
Mean-coil = 0.1398
iSG
Mean-beta = 0.3776
Mean-coil = 0.1444
Reference
Mean-beta = 0.5807
Mean-coil = 0.1331
Proportion of predicted beta-strands
b)
1 2 3 4 5 6 7 8 9A B CDE F G H
c)
N-term
B1
C
B2
C
B3
C
B4
B5678
C
A
C-term