BIOINFORMATICS

Vol. 14 no. 2 1998
Pages 157–163

Rose: generating sequence families

Jens Stoye[1,3], Dirk Evers[2] and Folker Meyer[2]

[1]Research Center for Interdisciplinary Studies on Structure Formation (FSPM) and [2]Technische Fakultät, University of Bielefeld, Postfach 100 131, 33501 Bielefeld, Germany

Received on August 27, 1997; revised on October 7, 1997; accepted on October 14, 1997

Abstract

Motivation: *We present a new probabilistic model of the evolution of RNA-, DNA-, or protein-like sequences and a software tool, Rose, that implements this model. Guided by an evolutionary tree, a family of related sequences is created from a common ancestor sequence by insertion, deletion and substitution of characters. During this artificial evolutionary process, the 'true' history is logged and the 'correct' multiple sequence alignment is created simultaneously. The model also allows for varying rates of mutation within the sequences, making it possible to establish so-called sequence motifs.*

Results: *The data created by Rose are suitable for the evaluation of methods in multiple sequence alignment computation and the prediction of phylogenetic relationships. It can also be useful when teaching courses in or developing models of sequence evolution and in the study of evolutionary processes.*

Availability: *Rose is available on the Bielefeld Bioinformatics WebServer under the following URL: http://bibiserv.TechFak.Uni-Bielefeld.DE/rose/ The source code is available upon request.*

Contact: *folker@TechFak.Uni-Bielefeld.DE*

Introduction

It is useful, for many reasons, to have a family of sequences with well-known evolutionary history. This kind of data is used in the study of evolutionary processes, in the evaluation of multiple sequence alignment methods, and in the reconstruction of phylogenetic trees. Other applications in computational molecular biology may also benefit from its availability. Unfortunately, nature does not provide 'benchmark' problems well suited for all these applications since there is no way to learn the exact phylogeny of the sequences involved. Therefore, it is common practice to create sequence data artificially, trying to be as close to the real world as possible.

The simulation of evolutionary processes at the molecular sequence level has a long tradition. Starting with the model of Jukes and Cantor (1969), several generalizations and alterations have been presented (e.g. Kimura, 1980; Felsenstein, 1981; Hasegawa *et al.*, 1985; Schöniger and von Haeseler, 1995). These models were designed for the study of molecular evolution at the sequence level, focusing on a well-founded statistical basis rather than on producing sequence families most similar to those usually considered in molecular biology. The early models even ignored the well-known fact of insertions and deletions (indels) during evolution. Some models (Thorne *et al.*, 1991, 1992) consider indels, but still have some other restrictions.

To create most realistic sequence families, we have added indels and 'sequence motifs' [patterns in a family of related sequences (Wu and Brutlag, 1995)] to the so-called HKY model (Hasegawa *et al.*, 1985) which only allows the description of arbitrary rate substitutions in DNA sequences. We also extended the underlying alphabet to cover amino acid sequences. An evolutionary process is simulated by iterated mutation of a 'common ancestor sequence' following the edges of a given 'mutation guide tree'. This way, the topology of the tree induces the relationship of the sequences. The mutations are performed by insertion, deletion and substitution of single characters or whole subsequences. Figure 1 sketches the creation process of a family of four sequences. In addition to knowing the exact evolutionary distance of the sequences, our approach provides us with their whole evolutionary history and the true alignment. Therefore, in contrast to biological applications, it is easily possible to verify predictions about alignments and phylogenetic relationships drawn from the sequences simply by comparing the predicted phylogeny to the tree that was used in the creation process.

In fact, we can go one step further and evaluate the adequacy of mathematical models such as maximum parsimony or sum-of-pairs multiple alignment. Given a program that calculates the best solution according to the model on a data set generated by Rose, we may contrast these results to the 'true' phylogeny or alignment.

The data created by our tool Rose (random model of sequence evolution) have been extensively tested with the Divide-and-Conquer Alignment (Stoye, 1997; Stoye *et al.*,

[3]*Present address: Department of Computer Science, University of California, Davis, CA 95616-8562, USA*

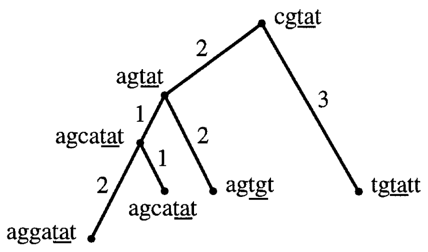


Fig. 1. Example of a creation process of four sequences from a common ancestor 'cgtat'. The underlined part denotes a sequence motif with smaller substitution probability.

1997) and GeneFisher (Giegerich *et al.*, 1996; Meyer and Schleiermacher, 1996) software packages.

Systems and methods

For reasons of speed, efficiency and portability, Rose was developed under UNIX using the ANSI C programming language. The software has been tested on various UNIX platforms e.g. DEC, HP, LINUX-PC, SGI, Sun. The actual program development was carried out on a Sun Sparcstation using gcc and Sun SPro C compilers, as well as bison and flex to build the input parser. The publicly available version runs on a Sun Enterprise 3000 server.

Algorithm

The model

Our procedure requires the following input:

(i) an alphabet A
of size l, e.g. the DNA alphabet {A, C, G, T} or the 20 character amino acid alphabet;

(ii) a root sequence s or an average sequence length n
(if no root sequence is specified, a random sequence of length $n \geq 1$ is generated);

(iii) character frequencies $f = (f_1, \ldots, f_l)$
satisfying $\sum_{i=1}^{1} f_i = 1$ used for insertions and the creation of the root sequence (if not specified);

(iv) a mutation guide tree T or a sequence distance d_{av}
the tree may be supplied with edge lengths (otherwise all edges are assumed to have uniform length 1), if no tree is entered, a binary mutation guide tree of user-defined average pairwise sequence distance d_{av} (see the subsection on adjusting the edge lengths) is created;

(v) a mutation matrix M
of size $l \times l$ representing pairwise mutation frequencies used for substitutions;

(vi) insertion and deletion probability functions
representing the probability of an indel event p_{ins} or p_{del}, combined with indel length functions l_{ins} and l_{del}, respectively; and

(vii) a mutation probability vector v
of length n allowing one to specify regions of different mutation rate, e.g. to specify sequence motifs.

Given these parameters, Rose generates
(i) a family of sequences $s_1, \ldots, s_m$
containing sequences with average length n and average pairwise evolutionary distance d_{av};

(ii) a multiple sequence alignment A
of the sequences $s_1, \ldots, s_m$ that is correct with respect to the creation process, i.e. it reflects the 'true' evolutionary history of $s_1, \ldots, s_m$; and finally

(iii) a relatedness tree T'
showing the phylogenetic relationship of the created sequences. T' is the smallest subtree of T which contains all the nodes corresponding to the generated sequences (and possibly some additional inner nodes which can be seen as extinct ancestors).

An outline of the algorithm is given here:

```
Rose(A, s, n, f, T, $d_{av}$)
begin
  if undefined(s)
        s := create_root_sequence(A, n, f);
  fi
  if undefined(T)
        T := create_guide_tree($d_{av}$) ;
  fi
  T.seq := s ; //copy root sequence to root of tree
  traverse(T); //recursively mutate sequences along tree
  print_sequences(T); //generate output
  print_alignment(T);
  print_tree(T);
end
```

where sub-function *traverse* is implemented as follows:

```
traverse(T)
foreach subtree T′ of T do
  T′.seq := evolve(T.seq)
  traverse(T′)
od
```

In the following subsections, we take a closer look at the different steps of Rose.

The root sequence

The implementation of function *create_root_sequence* is straightforward: if no pre-given root sequence is specified, each of the n positions in the root sequence is independently filled by a random process that returns letter A_i $(1 \leq i \leq l)$ with probability f_i.

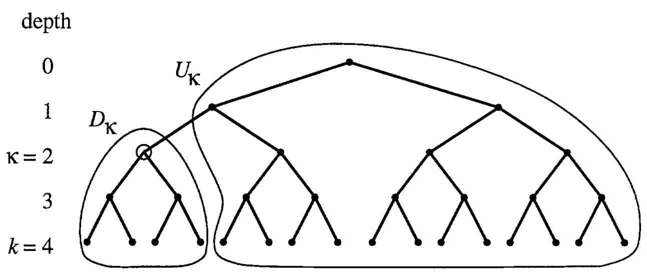


Fig. 2. Uniform binary tree of depth $k = 4$ with $K = 2^5 - 1 = 31$ nodes. For a node in depth $\kappa = 2$ (marked by the circle), those nodes contributing to D_κ and U_κ, respectively, are shown.

Rose works with arbitrary alphabets and any matching list of frequencies. For amino acid sequences, we implemented as default values the normalized frequencies of the amino acids given in Dayhoff *et al.* (1979), and for nucleotides we use the frequencies given in Agarwal and States (1996).

The mutation guide tree

The general behavior of *create_guide_tree* is similar to that of *create_root_sequence*: if no tree T is specified, Rose computes a uniform binary tree with $k = 1023$ nodes whose edge labels are adjusted such that the average sequence distance (i.e. the expected length of a shortest path between two randomly chosen nodes in the tree) meets the user-defined value d_{av} (see below). After the tree is created, either only from the leaves or from the leaves and inner nodes (chosen by the user), the required number of sequences is selected uniformly. So, in the latter case it can happen that, at the same time, an inner node sequence and a sequence from the corresponding subtree is chosen.

Obviously, it is possible to save space and computation time by pruning the unnecessary edges in the tree before performing the evolutionary process if not all of the sequences are contained in the final sequence family.

Adjusting the edge lengths. Assume a binary uniform tree of depth k with $K = 2^{k+1} - 1$ nodes and constant length b of every edge (see Figure 2). For the moment, let $b = 1$. Then, the average sequence distance d_{av} is the sum of all pairwise distances in the tree divided by $K(K-1)$, the number of pairs of distinct nodes. Consider, therefore, a node in level κ of the uniform binary tree, $0 \le \kappa \le k$. In the example of Figure 2, we have chosen $\kappa = 2$. The corresponding node is indicated by a circle.

In each level i, $0 \le i \le k - \kappa$ of the subtree 'below' the observed node, there are 2^i nodes with distance i. The sum of distances to all these nodes is:

$$\begin{aligned} D_\kappa &:= \sum_{i=0}^{k-\kappa} (2^i \cdot i) \\ &= 2^{k-\kappa+1}(k-\kappa-1) + 2 \end{aligned}$$

Additionally, there are κ nodes 'above' the observed node, each being the starting point of a subtree. Summing the distances to all these nodes gives:

$$\begin{aligned} U_\kappa &:= \sum_{i=1}^{\kappa}\left(i + \sum_{j=1}^{k-(\kappa-i)} 2^{j-1} \cdot (i + j)\right) \\ &= 2^{k-\kappa+1}(\kappa - k + 3) \\ &\quad + 2^{k+1}(\kappa + k - 3) + \kappa \end{aligned}$$

Thus, the total sum of distances from a node in level κ to all other nodes is:

$$\begin{aligned} N_\kappa &:= D_\kappa + U_\kappa \\ &= 2^{k-\kappa+2} + 2^{k+1}(\kappa + k - 3) + \kappa + 2 \end{aligned}$$

Averaging this value over all pairs of distinct nodes, we obtain:

$$\begin{aligned} d_{av} &:= \frac{\sum_{\kappa=0}^{k} (2^\kappa \cdot N_\kappa)}{K(K-1)} \\ &= 2 \cdot 2^{k+1} \frac{4 + k(1 + 2^{k+1}) - 2^{k+2}}{(2^{k+1}-1)(2^{k+1}-2)} \end{aligned}$$

which approximates

$$2\,\frac{k \cdot 2^{k+1} - 2^{k+2}}{2^{k+1}} = 2(k-2)$$

for sufficiently large k.

Similarly, if all edges have length b, we get:

$$d_{av} \approx 2b(k-2)$$

Hence, to obtain sequences of a pre-given relatedness, we simply have to alter the edge length b:

$$b \approx \frac{d_{av}}{2(k-2)}$$

For example, to obtain sequences of an average distance $d_{\mathrm{av}} \approx 250$ PAM, the edge length of our default tree with 1023 $= 2^{9+1} - 1$ nodes has to be set to $b \approx \frac{250}{2(9-2)} \approx 18$.

Note that in the above calculation we assumed that the sequences are selected from both the internal nodes and the leaves of the mutation guide tree. In case sequences are selected only from the leaves, a similar calculation leads to the formula:

$$b \approx \frac{d_{av}}{2(k-1)}$$

Creation of child sequences

We now take a closer look at the implementation of function *evolve*, the core of Rose. The following steps are used to create a new 'descendant' sequence s_{new} from a given ancestor sequence s_{old}:
evolve(s_{old})

1. The mutation function *mutate* for the given alphabet is applied to every position i in s_{old}:

$$s_{\text{new}}[i] = mutate(s_{\text{old}}[i], b)$$

where b is the length of the branch leading to the new node. The mutation matrix is selected with respect to b as described below.

2. One or more subsequences are deleted from s_{new}, taking into account the deletion probability p_{del} and the deletion length function l_{del}:

$$perform\_deletions(p_{\text{del}}, l_{\text{del}})$$

3. One or more sequences are inserted at arbitrary positions in s_{new}:

$$perform\_insertions(p_{\text{ins}}, l_{\text{ins}})$$

Function *mutate* makes use of the mutation probability matrix M. An entry $M[i,j]$ is interpreted as the probability for the jth letter of the alphabet A being substituted by the ith letter. Hence, the sum of each column of M should be $\sum_{i=1}^{l} M[i,j] = 1$ for all $j = 1, \ldots, l$. The diagonal values $M[i,i]$ determine the degree of stability: for example, a value of $M[i,i] = 0.99$ for all $i = 1, \ldots, l$ will result in an average mutability of 1% accepted mutations per unit of branch length.

In case the mutation matrix M is the probability matrix of one accepted amino acid substitution per hundred sites (1 PAM) given in Dayhoff *et al.* (1979), which is our default for proteins, we denote this new unit of measure for the distance of a child sequence from its ancestor including insertions and deletions by 1 PAM* where the parameters for insertions and deletions have to be specified additionally.

Evolutionary rates of more than 1 PAM* are obtained by applying the creation procedure repeatedly. As Schöniger and von Haeseler (1995) have shown, the use of a custom matrix (such as PAM 10) helps to save time when the number of substitutions exceeds an upper bound. At each step along an edge of the guide tree, depending on the mutation rate, the decision is made either to use pre-computed PAM* matrices repeatedly or to compute a new custom matrix.

Sequence motifs

Up to this point, we have assumed a constant rate of mutation over the whole length of the sequences. This is not very realistic: the mutation rate of genomic sequences found in nature is not constant for all positions in the genome. Mutations in regions with strong functional and/or structural importance are less often observed than elsewhere.

Therefore, we have generalized the function *evolve*: we allow the use of different rates of mutation for different regions of the sequence by a vector v of length n with values $v_i \geq 0$ which linearly increase/decrease the degree of variability at position i of the root sequence. A value $v_i = 1$ yields exactly the variability given by the edge length. Values $v_i < 1$ suppress mutations ($v_i = 0$: no mutation) and higher values $v_i > 1$ allow the specification of regions of particular high mutation rate, e.g. so-called hot spots. The vector v is inherited by child sequences. Indels are forbidden in regions with $v_i < 1$, thus establishing conserved sequence motifs. Inserted regions have a variability of 1.

Creation of indels

It is obvious that the exact mechanism of insertion and deletion is crucial for the simulation of evolution. Unfortunately, there is neither a well-established model (like HKY for nucleotide substitution) nor consensus as to the number of indels that corresponds to a certain evolutionary distance. We therefore chose to accommodate a wide range of possibilities with a function that we call inverted gap function. The following pseudocode shows the selection and creation of insertions; deletions are handled analogously (except for an additional test if the mutation probability of all deleted characters is ≥ 1).

```
perform_insertions($p_{\text{ins}}$,$l_{\text{ins}}$)
  begin
    do T.dist times
      if random_number_between_zero_and_one() < $p_{ins}$
        pos := choose_random_position(length($s_{old}$));
        len := compute_insertion_length($l_{ins}$);
        if $v_i$ >= 1
          do_insertion(pos, len);
        fi
      fi
    od
  end
```

The starting position for the insertion in *choose_random_position* is selected uniformly among the positions 1, …, *length*(s_{old}). To allow a high degree of variability, Rose accepts any quantized length function $l_{\text{ins}} = l_{ins}^{(1)}, \ldots, l_{ins}^{(q_{ins})}$ with $\sum_{i=1}^{q_{ins}} l_{ins}^{(i)} = 1$. Then, length len in $(1, \ldots, q_{ins})$ is selected with probability $l_{ins}^{(len)}$.

Function *do_insertion* finally is similar to the creation of the root sequence; the characters inserted maintain the initial character distribution.

Implementation

Input/output formats

The user input is via an HTML forms interface, the user can also choose to feed a file with all the input information into Rose using a simple tag value format. The format and the parameters are further described in our online manual http://bibiserv.TechFak.Uni-Bielefeld.DE/rose/manual.html.

Resource requirements

On a Sun Ultra I 167 MHz CPU, Rose used the following resources:

Table 1.

Protein			DNA		
# seqs	s	Mbyte	# seqs	s	Mbyte
10	1.8	1.1	10	3.3	1.4
100	9.8	1.9	100	18.6	3.8
500	24.9	4.2	500	49.5	9.6

Here, the created protein sequences have an average length of 250 letters and an average relatedness of 250 PAM*; the DNA sequences have an average length of 1000 letters and an average relatedness of 50.

Examples

The following examples show some of the features and demonstrate the versatility of Rose.

A protein sequence family. In Figure 3a, a sample family with $m = 4$ sequences of average length $n = 50$ is shown. This family is created with the default settings of Rose: a uniform binary mutation guide tree of depth $k = 9$ and uniform edge length $b = 18$ PAM*. The probability for insertions and deletions is set to $p_{\text{ins}} = p_{\text{del}} = 0.3\%$, and the insertion and deletion length functions are exponentially decreasing with a maximal length value of 10.

The alignment given in Figure 3b is the 'true' alignment corresponding to the creation process of the sequences. Figure 3c shows an optimal alignment according to the PAM 250 substitution matrix (Dayhoff *et al.*, 1979) (in distance form with values between 0 and 24) and gap function $g(l) = 8 + 12l$ computed with the program MSA (Lipman *et al.*, 1989; Gupta *et al.*, 1995). While the overall optimal alignment is correct, the exact location of the gaps does not coincide in all cases. This suboptimality of true alignments regarding the standard alignment score functions is also shown

```
(a)  FSAEAALVSPGKGDDEQVPNKDKCVYHGHKDGKRMNVKTPPTGPLVVGVHQ
     YEGANEVGATCEESSYCYVKEQAIQVKESQECTDFARHEVKSFRGVPGKLTEVIPVPL
     YGAAHPVGDPIKLGSLFLNHYESKGHTAAMCLLGMKTELIEPIEVQA
     SGVTEPVPNPVPATGIKLDKYTREENCLGMCLMGMGPPMVTIGEVGI

(b)  FSAEAALVSP--------GKGDDEQVPNKDKCVYHGHKDGKRMNVKTPPTGPLVVGVHQ
     YEGANEVGATCEESSYCYVKEQAIQVKESQECTDFARHEVKSFRGVPGKLTEV-IPVPL
     YGAAHPVGDP--------IKLGSLFLNH---YESKGHTAAMCLLGMKTELIEP-IEVQA
     SGVTEPVPNP--------VPATGIKLDK---YTREENCLGMCLMGMGPPMVTI-GEVGI

(c)  FSAEAALVSP--------GKGDDEQVPNKDKCVYHGHKDGKRMNVKTPPTGPLVVGVHQ
     YEGANEVGATCEESSYCYVKEQAIQVKESQECTDFARHEVKSFRGVPGKLTE-VIPVPL
     YGAAHPVGDP--------IKLGSLFL---NHYESKGHTAAMCLLGMKTELIE-PIEVQA
     SGVTEPVPNP--------VPATGIKL---DKYTREENCLGMCLMGMGPPMVT-IGEVGI
```

Fig. 3. (**a**) Sample family of random sequences obtained with Rose for $n = 50$ and $m = 4$. (**b**) 'True' alignment of these sequences. (**c**) A score-optimal alignment according to PAM 250 substitution matrix and gap function $g(l) = 8 + 12l$ computed with the program MSA. While the overall optimal alignment is correct, the exact location of the gaps does not coincide in all cases.

```
(a)  AGTG------ACTATAAT---CG---GAGGACAG--
     ATTCTGT---CCTATAAT---CG---GAGAAAAGCC
     AGTCTGT---ACTATAATGTTGG---GAGGAAAAGC
     AGTCCGTTGC--TATAAT---GG---GAGGAAAACC
     AATCTGT---AGTATAAT---GGTGTGAGGAAAGCC

(b)  AGT----GACTATAAT---CGGAGGACAG--
     ATTCTG-TCCTATAAT---CGGAGAAAAGCC
     AGTCTG-TACTATAATGTTGGGAGGAAAAGC
     AGTCCGTTGCTATAAT---GGGAGGAAAACC
     AATCTG-TAGTATAATGGTGTGAGGAAAGCC
```

Fig. 4. DNA example with TATAAT motif: (**a**) the 'true' and (**b**) an optimal alignment.

by the (distance) scores for both alignments: the 'true' alignment has an alignment score of 5184, while the optimal alignment has a 'better' score of 5166.

A simple DNA sequence family with motif. The use of motifs in sequence families created by Rose is demonstrated in Figure 4. The upper part shows the 'true' alignment of a family of five DNA sequences which contains a conserved TATAAT motif obtained with Rose using a mutation vector disallowing mutations within the motif, while outside the mutability remains normal. Figure 4b shows a score-optimal alignment of these sequences computed with MSA [unit substitution cost with gap function $g(l) = 2 + l$]. It is considerably shorter than the 'true' alignment. The parsimony objective underlying the sum-of-pairs scoring of MSA fails here.

A protein sequence family with varying mutation rate. Finally, we present a protein example where we fixed the root sequence and the mutation guide tree. We also varied the mutability along the sequence.

As root sequence, we took the human hemoglobin alpha sequence. The mutation guide tree is shown in Figure 5. The true alignment of our 'artificial globins' is shown in Figure 6. The histogram above the alignment shows the mutation probability along the sequence allowing a higher mutation rate between the α helices than within.

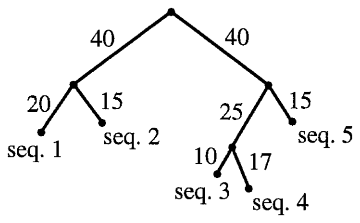
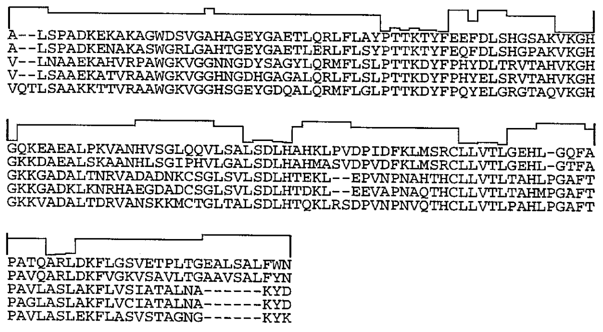


Fig. 5. Relatedness tree for the sequences shown in Figure 6.



```
A--LSPADKEKAKAGWDSVGAHAGEYGAETLQRLFLAYPTTKTYFEEFDLSHGSAKVKGH
A--LSPADKENAKASWGRLGAHTGEYGAETLERLFLSYPTTKTYFEQFDLSHGPAKVKGH
V--LNAAEKAHVRPAWGKVGGNNGDYSAGYLQRMFLSLPTTKDYFPHYDLTRVTAHVKGH
V--LSAAEKATVRAAWGKVGGHNGDHGAGALQRLFLSLPTTKDYFPHYELSRVTAHVKGH
VQTLSAAKKTTVRAAWGKVGGHSGEYGDQALQRMFLGLPTTKDYFPQYELGRGTAQVKGH

GQKEAEALPKVANHVSGLQQVLSALSDLHAHKLPVDPIDFKLMSRCLLVTLGEHL-GQFA
GKKDAEALSKAANHLSGIPHVLGALSDLHAHMASVDPVDFKLMSRCLLVTLGEHL-GTFA
GKKGADALTNRVADADNKCSGLSVLSDLHTEKL--EPVNPNAHTHCLLVTLTAHLPGAFT
GKKGADKLKNRHAEGDADCSGLSVLSDLHTDKL--EEVAPNAQTHCLLVTLTAHMPGAFT
GKKVADALTDRVANSKKMCTGLTALSDLHTQKLRSDPVNPNVQTHCLLVTLPAHLPGAFT

PATQARLDKFLGSVETPLTGEALSALFWN
PAVQARLDKFVGKVSAVLTGAAVSALFYN
PAVLASLAKFLVSIATALNA------KYD
PAGLASLAKFLVCIATALNA------KYD
PAVLASLEKFLASVSTAGNG------KYK
```

Fig. 6. Family of 'globins' created by using human hemoglobin alpha as the root sequence. The mutability vector is shown in the form of a histogram above the alignment.

Discussion and conclusion

The data sets created by Rose are artificial sequence families that contain both indels and motifs. The evaluation of multiple sequence alignment tools and phylogenetic reconstruction tools is possible with these benchmarks.

Previous models were mainly designed to understand evolutionary processes better rather than create nature-like sequence families. While such studies need a rigorous probabilistic foundation, they are quite far from a realistic simulation of the biological truth. Even the most sophisticated model (Thorne *et al.*, 1992) including indels of complete blocks cannot describe overlapping insertions and deletions as two evolutionary events since fragments cannot vary over time. Our 'fragments' (i.e. inserted or deleted regions) can vary over time and hence overlap. Our model is based on empirically verified parameters. It is not *a priori* clear by which parameters the most natural results can be obtained and there does not seem to exist a single set of evolutionary parameters describing the whole variety one finds in nature. Therefore, with Rose, the user is free to set whatever parameters seem reasonable for the actual purpose.

While we have removed a number of limits that existed so far, there are still some limitations: while we do not assume that the characters of the sequences evolve independently and with the same rate in the whole family, we have not yet included a feature that simulates different rates of evolutionary pressure in different branches of the tree, enabling different lineages to evolve independently within our tree. This has been observed by a number of biologists (Greer, 1981, 1990; Schulz *et al.*, 1986; Benner *et al.*, 1994). While we are planning to include this feature in a future release of Rose and extend the scope of our model even further, it is important to note that all results have to regard the adequacy of the chosen evolutionary parameters, and that simulations can only aid the evaluation of algorithms. What matters in the end is the success on real biological sequences.

Acknowledgements

The authors wish to thank Robert Giegerich for encouraging us to work on this approach and Marcie McClure for making a good point. We would also like to thank the referees for valuable hints. The work was partially supported by the German Ministry for Education and Sciences (BMBF), the Ministry of Science of North Rhine Westfalia (MWF-NRW), the German Research Council graduate program (DFG-GK) Strukturbildungsprozesse, and the German Academic Exchange Service (DAAD).

References

Agarwal,P. and States,D.J. (1996) A Bayesian evolutionary distance for parametrically aligned sequences. *J. Comp. Biol.*, **3**, 1–17.

Benner,S.A., Cohen,M.A. and Gonnet,G.H. (1994) Amino acid substitution during functionally constrained divergent evolution of protein sequences. *Protein Eng.*, **7**, 1323–1332.

Dayhoff,M.O., Schwartz,R.M. and Orcutt,B.C. (1979) A model of evolutionary change in proteins. In Dayhoff,M.O. (ed.), *Atlas of Protein Sequence and Structure.* National Biomedical Research Foundation, Washington, DC, Vol. 5, Suppl. 3, pp. 345–352.

Felsenstein,J. (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.*, **17**, 368–376.

Giegerich,R., Meyer,F. and Schleiermacher,C. (1996) GeneFisher-Software support for the detection of postulated genes. In *Proceedings of the Fourth Conference on Intelligent Systems for Molecular Biology, ISMB 96.* AAAI Press, Menlo Park, CA, pp. 68–78.

Greer,J. (1981) Comparative model-building of the mammalian serine proteases. *J. Mol. Biol.*, **153**, 1027–1042.

Greer,J. (1990) Comparative modeling methods: applications to the family of the mammalian serine proteases. *Proteins: Struct. Funct. Genet.*, **7**, 317–334.

Gupta,S.K., Kececioglu,J.D. and Schäffer,A.A. (1995) Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment. *J. Comp. Biol.*, **2**, 459–472.

Hasegawa,M., Kishino,H. and Yano,T. (1985) Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.*, **22**, 160–174.

Jukes,T.H. and Cantor,C.R. (1969) Evolution of protein molecules. In Munro,H.N. (ed.), *Mammalian Protein Metabolism.* Academic Press, New York, Vol. 3, pp. 21–132.

Kimura,M. (1980) A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*, **16**, 111–120.

Lipman,D.J., Altschul,S.F. and Kececioglu,J.D. (1989) A tool for multiple sequence alignment. *Proc. Natl Acad. Sci. USA*, **86**, 4412–4415.

Meyer,F. and Schleiermacher,C. (1996) GeneFisher. http://bibiserv.TechFak.Uni-Bielefeld.DE/genefisher/

Schöniger,M. and von Haeseler,A. (1995) Simulating efficiently the evolution of DNA sequences. *Comput. Applic. Biosci.*, **11**, 111–115.

Schulz,G.E., Schiltz,E., Tomasselli,A.G., Frank,R., Brune,M., Wittinghofer,A. and Schirmer,R.H. (1986) Structural relationships in the adenylate kinase family. *Eur. J. Biochem.*, **161**, 127–132.

Stoye,J. (1997) DCA: divide and conquer multiple sequence alignment, Version 1.0. http://bibiserv.TechFak.Uni-Bielefeld.DE/dca/

Stoye,J., Moulton,V. and Dress,A.W.M. (1997) DCA: an efficient implementation of the divide-and-conquer approach to simultaneous multiple sequence alignment. *Comput. Applic. Biosci.*, in press.

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

Thorne,J.L., Kishino,H. and Felsenstein,J. (1992) Inching toward reality: an improved likelihood model of sequence evolution. *J. Mol. Evol.*, **34**, 3–16.

Wu,T.D. and Brutlag,D.L. (1995) Identification of protein motifs using conserved amino acid properties and partitioning techniques. In *Proceedings of the Third Conference on Intelligent Systems for Molecular Biology, ISMB 95.* AAAI Press, Menlo Park, CA, pp. 402–410.