SISSI: Simulating Sequence Evolution with Site-Specific Interactions

SISSI Manual

Tanja Gesell
Center for Integrative Bioinformatics Vienna,
Max F. Perutz Laboratories,
Dr. Bohr-Gasse 9/6, A-1030 Vienna, Austria.
email: `tanja.gesell (at) mfpl.ac.at`

Arndt von Haeseler
Center for Integrative Bioinformatics Vienna,
Max F. Perutz Laboratories,
Dr. Bohr-Gasse 9/6, A-1030 Vienna, Austria.
email: `arndt.von.haeseler (at) mfpl.ac.at`

Contents

1 Introduction

The program SISSI (Simulating Site-Specific Interactions along Phylogentic Trees) is a software tool which generates any number of data sets of related sequences with site-specific interactions for a given phylogeny, user defined sytems of neighbourhoods and instantaneous rate matrices. Different programs have been designed to simulate nucleotide sequences along a tree (Schöniger and von Haeseler, 1995; Rambaut and Grassly, 1997; Grassly *et al.*, 1997; Yang, 1997; Stoye *et al.*, 1998; Nicholas *et al.*, 2000; Tufféry, 2002). SISSI incorporating site-specific interactions complements the available simulation approach. The basic idea is to use separate models for each site defined by the interactions with other sites in the sequence. It follows that the rates of substitution may be not only variable in time (rate heterogeneity), but also correlated. With different null models, specific annotations and adequate parameters for the neighbourhoods, there are a lot of models imaginable for our simulation procedure, which will by and by implemented in SISSI.

SISSI is a command-line program which allows users to specify the parameter values.

SISSI is available free of charge from

`http://www.cibiv.at/software/sissi/`

The methods are described in detail in the following papers:

- Gesell and von Haeseler (2006) In silico sequence evolution with site-specific interactions along phylogenetic trees. *Bioinformatics.* **22(6)**:716-722
  `http://bioinformatics.oxfordjournals.org/cgi/content/abstract/22/6/716`

Important Notice:

Some programs to generate some special neigbourhood systems as input are available. Furthermore, on request special versions of SISSI, which can be faster (e.g. for overlapping dependcies of dinucleotides) or if necessary are easy to implement (e.g. codons or motifs).

For the purpose of this manual, we assume that the user is familar with the theory and practice of molecular evolution as well as the use of their computer system.

2 Requirements

SISSI runs on UNIX/Linux, Windows and Mac OS systems, including Mac OS X. It should be easily compiled. SISSI is a command–line–controlled program written in ANSI C. The switches and parameters that control the program are supplied on the command line.

3 Running SISSI

sissi [parameters] < [trees] > [sequencies]

-[parameters] are the parameters for the program.
-[trees] is the tree file
-[sequences] is the name of the file that will contain the simulated sequences. The sequences produced by SISSI are written to the standard output and can thus redirected to the output file using > [filename]. Other information and results are written to standard error and thus will appear on the screen.

4 Command-line options

Users can specify parameters through a set of command-line options, which are extremely useful to start a batch job. Run '`sissi -h`' to print out a short description of available options:

```
sissi  [OPTIONS][-nFORMAT NEIGHBOURHOODSYSTEMFILE][-fSTATE FREQUENCIES][treefile]

POSSIBLE OPTIONS:
 Miscellaneous options:
      -l #   = Sequence length              [default = 100].
      -a #   = Simulated datasets per tree  [default = 1].
      -d     = Generate DNA sequences       [default = RNA].
      -k     = Use an ancestor-sequence-file[default = random].
      -h     = Print this help message
      -z #   = Seed for random number generator [default = system generated].
      -y     = SPRNG random generator [default = dkiss random generator].
      -i     = Optimise running time for q(x) (see info at the end)
      -t #   = Transition-Transversion, TS/TV equal 0.5 is JC/Fel [default=0.5]
      -q     = Quiet

 Neighbourhood System:
  nFORMAT neighbourhoodsystem file [default = no correlations]:
      -nn  = sissi0.1 format (see example bsubtilis401.nei)
      -nr  = ctFile as neighbourhoodsystem (see example B.subtilis.ct2)
      -nc  = sissi0.2 format (see example codon.cnei)
        Note:  You must generate a neighbourhoodsystem for your sepecial case.

 Nucleotid Model:
   (F81 model -f option
```

```
  JC 69: Note, neighbourhood system has no effect [default]
  K80 -t option
  HKY use -t option and -f option
  Tamura use -rr option and -f option
  GTR use -rr and -f option
  Complex combinations for higher dimensions.)
fSTATE frequencies [default = all equal]:
    -fs #A #C #G #U  =  4 'single' nucleotide frequencies  (nk=0)
    -fd #AA ... #UU     = 16 'doublet' nucleotide frequencies (nk=1)
    -ft file: (#AAA...#UUU) = 64 'triplet' frequencies) (nk=2)
rGTR [default = all equal]:
    -rr # 5  = GTR abcdef: Not possible with t option!
    -rd # 24 = 24 rates extended GTR nk=1
    -rt # 95 = 95 rates extended GTR nk=2
    -ro        =  ovelapping sites
    -rc # 2    = CpG: #1 CpG->TpG; #2 CpG -> CpA
    -ra        = Take mononucleotide frequencies for nk=1
    -rb        = Take mononucleotide frequencies for nk=2
Rate Heterogeneity:
 eSTATE Rateheterogeneity [default = all equal]:
   -ee         = file site-specific scaling factor (ssf) (#1...#l = seqlen)
   -en         = file normalise ssf (#1...#l = seqlen
   -ed         = file ssf in the format (#pos #ssf
   -em         = file normalise ssf (#pos #ssf)

 wOPTION Write additional information [default=none]:
   -wa = Write ancestral sequences for each node
   -wpfilename = Print in FILES
   -wh = Write Hamming distance after each node
   -wo = Print note for simulating branch length with no subst.
   -wt = Print special treefiles with additional information
                <intern.htree>          with internal nodes
               <internlength.htree>    with internal nodes and branch length
               <hammingdistance.htree> with internal nodes and Hamming distance
   -wb = Binary Alphabet in FILES (with wp option):
              0: Purine     R : A || G
              1: Pyrimidine Y : C || U || T
 sOPTION scaling branch length or tree [default=none]:
   -sb = Branch length scaling factor [default = 1.0].
   -st = Total tree scale [default = use branch lengths].
         !don't use it with zero branch length in the tree
oOPTION Output File format [default=Phylip]:
   -of = Fasta output
   -or = relaxedPhylip output
   -oc = Clustal output
```

```
 vOPTION Verbose [default=none]:
    -vf = Print frequencies
    -vq = Print rate matrices
    -vn = Print neighborhood in sissi0.1 format
    -vt = Print number of substitutions and Hamming distance
    -va = Print note for no substitutions
    -vr = Print expected number of substitutions per site
    -vz = Print random generator seed
TREEFILE: Name of the treefile [default=generates no sequences!]

Sequence length (-l) must agree with the length of the neighbourhood system!
Some faster versions are available on request for sepecial Ns.
```

5 Input Neighbourhood System File Format

For the representation of site-specific interactions we are using a neighbourhood system. Neighbourhood systems allow for an universal description of arbitrarily complex dependencies among sites. In the paper we give some illustrative simple examples how to apply neighbourhood systems to define various structural elements. To include the neighbourhood, the input to SISSI is a text file containing the neighbourhood system for each site.

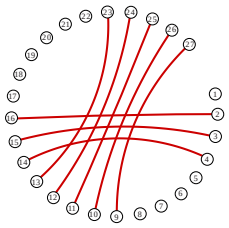
5.1 -nn sissi0.1 format

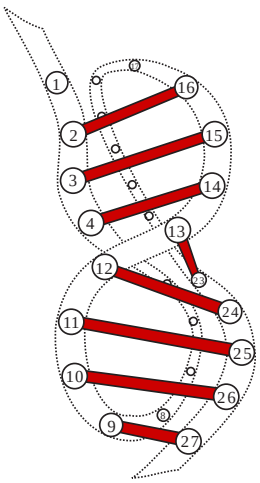
5.1.1 Example: RNA Pseudoknot

Input neighbourhood file - pseudo.nei:

$\mathcal{N}$(Pseudoknot):

$N_1 = \{\}$ $N_2 = \{16\}$ $N_3 = \{15\}$ $N_4 = \{14\}$ $N_5 = \{\}$ $N_6 = \{\}$ $N_7 = \{\}$ $N_8 = \{\}$ $N_9 = \{27\}$ $N_{10} = \{26\}$ $N_{11} = \{25\}$ $N_{12} = \{24\}$ $N_{13} = \{23\}$ $N_{14} = \{4\}$ $N_{15} = \{3\}$ $N_{16} = \{2\}$ $N_{17} = \{\}$ $N_{18} = \{\}$ $N_{19} = \{\}$ $N_{20} = \{\}$ $N_{21} = \{\}$ $N_{22} = \{\}$ $N_{23} = \{13\}$ $N_{24} = \{12\}$ $N_{25} = \{11\}$ $N_{26} = \{10\}$ $N_{27} = \{9\}$






```
Pos    0|
Pos    1| 15:(1.000000)
Pos    2| 14:(1.000000)
Pos    3| 13:(1.000000)
Pos    4|
Pos    5|
Pos    6|
Pos    7|
Pos    8| 26:(1.000000)
Pos    9| 25:(1.000000)
Pos   10| 24:(1.000000)
Pos   11| 23:(1.000000)
Pos   12| 22:(1.000000)
Pos   13|  3:(1.000000)
Pos   14|  2:(1.000000)
Pos   15|  1:(1.000000)
Pos   16|
Pos   17|
Pos   18|
Pos   19|
Pos   20|
Pos   20|
Pos   22| 12:(1.000000)
Pos   23| 11:(1.000000)
Pos   24| 10:(1.000000)
Pos   25|  9:(1.000000)
Pos   26|
```

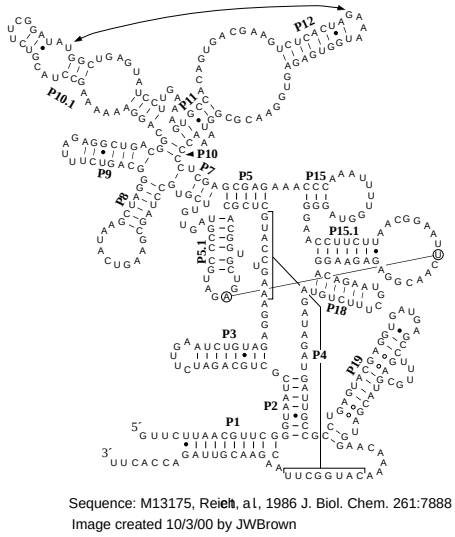
where the first number in a line gives the current site and after the pipe are written the neighbours for this site. In bracket are the correlationsvalue, which will don't use in this version of SISSI and must set to one.

5.2 -nr Output of RNA structure prediction programs as a input

For RNA structure SISSI cann read ct files:
The files containing data about the base pairs in an RNA secondary structure (see `http://monod.nyu.edu/rna/analysis/program_description.html`). Furthermore, for example in the Vienna RNA package `http://www.tbi.univie.ac.at/~ivo/RNA/` (Hofacker *et al.*,

1994) two utilities convert back and forth between secondary structure in bracket format (as used in the Vienna package) and ct files (as used by mfold (Zuker, 2000)) . Thus, SISSI can read directly the output of existing structure predicting programs or structural databases, files with and without pseudoknots of the RNAse P Database (Brown, 1999). For Examples compare the files bsub.nei, bsub.ct and bsub.2ct in the directory ExampleNeighbourhoods.

5.2.1 Example: ct file of the RNAse P Database



Input neighbourhood file:

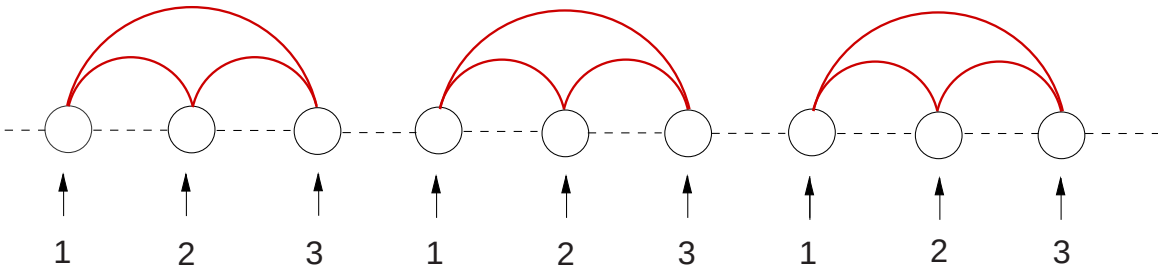
```
401 ENERGY = 0 B.subtilis RNase P RNA
   1 G        0     2     0     1
   2 U        1     3     0     2
   3 U        2     4     0     3
   4 C        3     5   395     4
   5 U        4     6   394     5
   6 U        5     7   393     6
   7 A        6     8   392     7
   8 A        7     9   391     8
   9 C        8    10   390     9
  10 G        9    11   389    10
  11 U       10    12   388    11
  12 U       11    13   387    12
  13 C       12    14   386    13
  14 G       13    15   384    14
   ....
```

5.3 -nc sissi0.2 format

For more complex neighbourhood systems we use categories for sites, where the category of the site follows the position number. Here are some examples:

5.3.1 Example: Codons

Codons have categories of the codon positions 1,2 or 3 with three different instantaneous rate matrices of the same dimension (64x64).

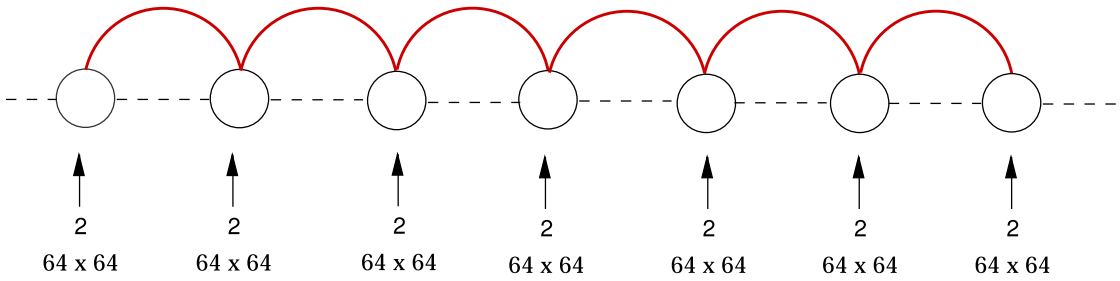


Input neighbourhood file:

```
Pos    0-1|    1:(1.000000)    2:(1.000000)
Pos    1-2|    0:(1.000000)    2:(1.000000)
Pos    2-3|    0:(1.000000)    1:(1.000000)
Pos    3-1|    4:(1.000000)    5:(1.000000)
Pos    4-2|    3:(1.000000)    5:(1.000000)
Pos    5-3|    3:(1.000000)    4:(1.000000)
Pos    6-1|    7:(1.000000)    8:(1.000000)
Pos    7-2|    6:(1.000000)    8:(1.000000)
Pos    8-3|    6:(1.000000)    7:(1.000000)
Pos    9-1|   10:(1.000000)   11:(1.000000)
Pos   10-2|    9:(1.000000)   11:(1.000000)
Pos   11-3|    9:(1.000000)   10:(1.000000)
Pos   12-1|   13:(1.000000)   14:(1.000000)
Pos   13-2|   12:(1.000000)   14:(1.000000)
.............
.............
```

5.3.2 Example: Overlapping Dependencies

One of the most famous examples for this neighbourhood system are CpGs: The process can be defined through a corresponding model and the following neighborhood system, calling direct site-neighbors:



Input neighbourhood file:

```
Pos    0-2|  999:(1.000000)    1:(1.000000)
Pos    1-2|    0:(1.000000)    2:(1.000000)
Pos    2-2|    1:(1.000000)    3:(1.000000)
Pos    3-2|    2:(1.000000)    4:(1.000000)
Pos    4-2|    3:(1.000000)    5:(1.000000)
Pos    5-2|    4:(1.000000)    6:(1.000000)
Pos    6-2|    5:(1.000000)    7:(1.000000)
Pos    7-2|    6:(1.000000)    8:(1.000000)
Pos    8-2|    7:(1.000000)    9:(1.000000)
Pos    9-2|    8:(1.000000)   10:(1.000000)
.....
.....
```

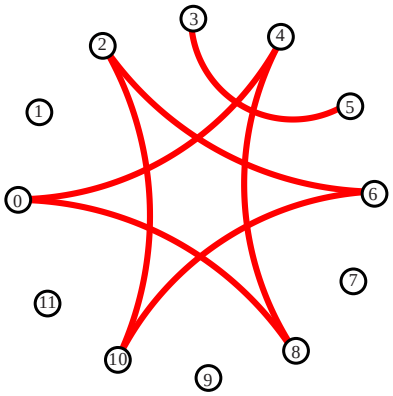
5.3.3 Example: Complex and Artificial

with triplets and categories for sites.

Input neighbourhood file:

```
Pos  0 – 1|  4:(1.000000)  8:(1.000000)
Pos  1 – 0|
Pos  2 – 1|  6:(1.000000) 10:(1.000000)
Pos  3 – 0|  5:(1.000000)
Pos  4 – 2|  0:(1.000000) 8:(1.000000)
Pos  5 – 0|  3:(1.000000)
Pos  6 – 2|  2:(1.000000) 10:(1.000000)
Pos  7 – 0|
Pos  8 – 3|  0:(1.000000)  4:(1.000000)
Pos  9 – 0|
Pos 10 – 3|  2:(1.000000)  6:(1.000000)
Pos 11 – 0|
```

Circle plot $\mathcal{N}$(artificial example):



Important Notice:

Some programs to generate some special neigbourhood systems as input are available. In addition on request special versions of SISSI, which can be faster based on limited to special neighbourhood system (e.g. for overlapping dependcies of dinucleotides) are available or if necessary are easy to implemented (e.g. codons or motifs)l.
The rate matrices for combination of sites $n_k = 0, n_k = 1$, and $n_k = 2$ are implemented. Further cardinalities are easy to implemented or on request (tanja@cs.uni-duesseldorf.de).

6 Substitution process

6.1 Model options

With different null models, specific annotations and adequate parameters for the neighbourhoods, there are a lot of models imaginable for our simulation procedure, which will by and by implemented in SISSI. The following models are implemented (fStates and rRates), e.g.:

- JC69 (Jukes and Cantor, 1969).
- K2P (Kimura, 1980).

- F81 (Felsenstein, 1981).
- HKY85 (Hasegawa *et al.*, 1985).
- TN93 (Tamura and Nei, 1993).
- GTR - General Time Reversible (e.g., Tavaré, 1986).

Irreversible models are not fully implemented yet.*

Dependency models are imaginable through possible combinations, e.g

- SH94 (Schöniger and von Haeseler, 1994).
- MU95 (Muse, 1995).
- RH95 (Rzhetsky and Nei, 1995).

Codonmodels are also available through possible comibinations.
CpGs as a special case hav its own option: **-rc # 2 CpG− >TpG; CpG− >CpA**

6.2 Relative State Frequencies

This options specifiy the equilibrium distributions of subsequences, which are relevant for the constraints of the neighbourhood system.
These frequencies need not sum up to one, as the program will do the normalization. Therefore, however, the vector has to have an entry for every nucleotide. The default (when no frequencies are specified) will be that all frequencies are equal.
In addition, the stationary frequencies over the appropriate state space for the respectiv model Q_k can be counted using these special file types, based on a given structural alignment or input by the user.

-fs $< statefrequencies >$: Single Frequencies $(n_k = 0)$ in order of:
a c g u
a c g t

-fd $< statefrequencies >$: Doublet Frequencies $(n_k = 1)$ in order of:
aa ac ag au ca cc cg cu ga gc gg gu ua uc ug uu
aa ac ag at ca cc cg ct ga gc gg gt ta tc tg tt

-ft $< statefrequencies >$: Triplet Frequencies File $(n_k = 2)$ in order of:
for subsequences with $n_k > 1$

6.3 Site-Specific Scaling Factor

SISSI can simulate with rate heterogeneity (rate homogeneity, default) with an vector, including a rate for each site:
The input file include the site-specific rates in one line corresponding to the sequence length, or the positions with the rate in two columns, like in the output of IQPNNI.
Furthermore, with the site-specific factor invariable sites are possible.

7 Input Tree File Format

The tree format is the same as used by PHYLIP (also called the "Newick" format). This is a nested set of bifurcations defined by brackets and commas. Here are two examples:

(((Taxon1:0.2, Taxon2:0.2):0.1, Taxon3:0.3):0.1, Taxon4:0.4);
((Taxon1:0.1, Taxon2:0.2):0.05, Taxon3:0.3, Taxon4:0.4);

The first is a rooted tree because it has a bifurcation at the highest level. The next tree is unrooted - it has a trifurcation at the top level. Each tree should finish with a semicolon. Any numbers of trees may be in the input file separated by a semi-colon and a new-line. Whilst PHYLIP only allows taxon names of up to 10 characters. SISSI can read trees with taxon names of up to 256 characters. Unless the -o option is set (see below), the output file will conform to the PHYLIP format and the names will be truncated to 10 characters. Note that this could cause some taxon names to be identical and this can cause problems in some phylogenetic packages.

8 Ancestor sequence file

This option allows the user to use a supplied sequence as the ancestral sequence at the root (otherwise a random sequence is used). $< ancestorsequencefile >$ is a file with the nucleotides of the ancestor. The ancestor sequence must have the same length like the neighbourhood system and option -l.

8.1 Additional Information

This option allows the user to obtain additional information for each of the internal nodes in the tree.

-wa Write ancestral sequences for each node. The sequences are written out along with the sequences for the tips of the tree in relaxed PHYLIP format.

-wh Write hammingdistance between the sequences for each node

-wo Give a note for simulating branch length with hammingdistance=0

-wt Write trees with intern nodes in < *intern.htree* >

For further parameter options see 4.

9 An example of performing simulations using SISSI:

More Examples are in the directory ExampleFiles:

sissi -fs 0.422360 0.105590 0.236025 0.236025 -fd 0.000423 0.004228 0.012685 0.169133 0.004228 0.000423 0.262156 0.000423 0.012685 0.262156 0.000423 0.042283 0.169133 0.000423 0.042283 0.016915 -nn ../Neighbourhoods/bsubtilis401.nei -l401 ../Trees/example.tree

Further Examples are given in the directory ExamplFiles.

10 Credits

Some parts of the code were taken from Seq-Gen. Rambaut, A. and Grassly, N. C. (1997) Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Comput. Appl. Biosci. 13: 235-238

The source code for a random generator are taken from SPRNG (Scalable Pseudo Random Number Generator) library package Mascagni, Michael and Srinivasan, Ashok (2000) SPRNG: A Scalable Library for Pseudorandom Number Generation. ACM Trans. Math. Software, 26:436-461, DOI: 10.1145/358407.358427

Bugs found by: - Bernhard Misof, Zoologisches Forschungsinstitut Bonn, Germany (-d Option)

Acknowledgement

We would like to thank Andrew Rambaut for allowing us to use some code from Seq-Gen Thanks to Heiko Schmidt for various help.
Financial support from the Wiener Wissenschafts-, Forschungs- and Technologiefonds (WWTF) is greatly appreciated

References

Brown, J. W. (1999) The ribonuclease P database. *Nucleic Acids Research*, **27**, 314.

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

Gesell, T. and von Haeseler, A. (2006) In silico sequence evolution with site-specific interactions along phylogenetic trees. *Bioinformatics*, **22**, 716–722.

Grassly, N., Adachi, J. and Rambaut, A. (1997) PSeq-Gen: An application for the Monte Carlo simulation of protein sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, **13**, 559–560.

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

Hofacker, I., Fontana, W., Stadler, P., Bonhoeffer, S., Tacker, M. and Schuster, P. (1994) Fast folding and comparison of RNA secondary structure. *Monatshefte f. Chemie*, **125**, 167–188.

Jukes, T. H. and Cantor, C. R. (1969) Evolution of protein molecules. In Munro, H. N. (ed.), *Mammalian Protein Metabolism*, volume 3, pp. 21–123, Academic Press, New York.

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.

Meyer, S. and von Haeseler, A. (2003) Identifying site-specific substitution rates. *Mol. Biol. Evol.*, **20**, 182–189.

Muse, S. V. (1995) Evolutionary analyses of DNA sequences subject to constraints on secondary structure. *Genetics*, **139**, 1429–1439.

Nicholas, J. S., Hoyle, D. C. and Higgs, P. G. (2000) RNA sequence evolution with secondary structure constraints: Comparison of substitution rate models using maximum-likelihood methods. *Genetics*, **157**, 399–411.

Rambaut, A. and Grassly, N. C. (1997) Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, **13**, 235–238.

Rzhetsky, A. and Nei, M. (1995) Tests of applicability of several substitution models for DNA sequence data. *Mol. Biol. Evol.*, **12**, 131–151.

Schöniger, M. and von Haeseler, A. (1994) A stochastic model for the evolution of autocorrelated DNA sequences. *Mol. Phylogenet. Evol.*, **3**, 240–247.

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

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

Tamura, K. and Nei, M. (1993) Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Mol. Biol. Evol.*, **10**, 512–526.

Tavaré, S. (1986) Some probabilistic and statistical problems on the analysis of DNA sequences. *Lec. Math. Life Sci.*, **17**, 57–86.

Tufféry, P. (2002) CS-PSeq-Gen: Simulating the evolution of protein sequence under constraints. *Bioinformatics*, **18**, 1015–1016.

Yang, Z. (1994) Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximative methods. *J. Mol. Evol.*, **39**, 306–314.

Yang, Z. (1997) PAML: a program package for phylogenetic analysis by maximum likelihood. *Computer Applications in BioSciences*, **13**, 555–556.

Zuker, M. (2000) Calculating nucleic acid secondary structure. *Curr. Opin. Struct. Biol.*, **10**, 303–310.