Seq-Gen

Sequence-Generator: An application for the Monte Carlo simulation of molecular sequence evolution along phylogenetic trees.

Version 1.3.2

Andrew Rambaut and Nick C. Grassly

Supported by The Royal Society

Department of Zoology,
University of Oxford,
South Parks Road,
Oxford OX1 3PS, U.K.

Bug fixed in version 1.3.2 - 7 Jan 2005

- The PAM (Dayhoff) and Blosum matrices were interchanged so specifying one would result in the other being used instead.

Bug fixed in version 1.3.1 - 4 Nov 2004

- Specified nucleotide frequencies were being ignored and equal frequencies being used instead.

New Features in version 1.3 - 30 Aug 2004

- Added amino acid simulation to Seq-Gen. This replaces PSeq-Gen which was not being updated but also adds a number of other amino acid models.
- Removed the limit on tree size. The only limit now is the available memory.

- Updated to the latest version of the MT19937 random number generator.

Citation

If you use this program in a publication please cite the following reference:

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.

Introduction

Seq-Gen is a program that will simulate the evolution of nucleotide or amino acid sequences along a phylogeny, using common models of the substitution process. A range of models of molecular evolution are implemented including the general reversible model. Nucleotide/Amino acid frequencies and other parameters of the model may be given and site-specific rate heterogeneity may also be incorporated in a number of ways. Any number of trees may be read in and the program will produce any number of data sets for each tree. Thus large sets of replicate simulations can be easily created. It has been designed to be a general purpose simulator that incorporates most of the commonly used (and computationally tractable) models of molecular sequence evolution. The paper cited above contains details of the algorithm and a short discussion about the uses of Seq-Gen.

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

Requirements

Seq-Gen is a command-line controlled program written in ANSI C. It should be easily compiled and run on any UNIX system or workstation (which includes Mac OS X). This paper describes the use of Seq-Gen on a UNIX machine. The application requires an amount of memory proportional to the size of each simulated sequence data set.

The Mac OS X version of Seq-Gen now functions in the same way as the UNIX version using the 'Terminal' application. There is however a new graphical user-interface that can be used to run Seq-Gen on Mac OS X (and

hopefully soon, Windows and Linux) written by Thomas Wilcox. This is available in the Mac OS X package for Seq-Gen:

http://evolve.zoo.ox.ac.uk/software/Seq-Gen/

Acknowledgements

A.R is supported by a Royal Society University Research Fellowship and previously was supported by grant 50275 from The Wellcome Trust. N.C.G. is also supported by a Royal Society University Research Fellowship. We would like to thank Ziheng Yang for allowing us to use some invaluable code from PAML.

The models of substitution

All the models of molecular substitution implemented in Seq-Gen are time-reversible Markov models, and assume evolution is independent and indentical at each site and along each lineage. Almost all models used in the maximum likelihood reconstruction of phylogenies using nucleotide sequences are processes of this type (but see Yang, 1994). Selecting either a nucleotide or amino acid model of substitution will determine which type of data is produced.

Nucleotide models of substitution

The Hasegawa, Kishino and Yano (HKY) model (Hasegawa et al., 1985) allows for a different rate of transitions and transversions as well as unequal frequencies of the four nucleotides (base frequencies). The parameters requires by this model are transition to transversion ratio (TS/TV) and the base frequencies. There are a number of simpler models that are specific cases of the HKY model. If the base frequencies are set equal (by not specifying base frequencies) then the model becomes equivalent to the Kimura 2-parameter (K2P) model (Kimura, 1980). If the TS TV rates are set to be equal (by not specifying a TS/TV ratio) as well, then it becomes equivalent to the Jukes-Cantor (JC69) model (Jukes and Cantor, 1969).

The F84 model (Felsenstein and Churchill, 1996), as implemented in DNAML in the PHYLIP package (Felsenstein, 1993), is very similar to HKY but differs slightly in how it treats transitions and transversions. This model requires the same parameters as HKY.

Finally, the general reversible process (GTR) model (e.g. Yang, 1994) allows 6 different rate parameters and is the most general model that is still

consistent with the requirement of being reversible. The 6 parameters are the relative rates for each type of substitution (i.e. A to C, A to G, A to T, C to G, C to T and G to T). As this is a time-reversible process, the rate parameter of one type of substitution (e.g., A to T) is assumed to be the same as the reverse (e.g., T to A).

Amino acid models of substitution

A number of empirical models of amino acid substitution are included with Seq-Gen. These include JTT (Jones et al, 1992), WAG (Whelan & Goldman, 2001), PAM (Dayhoff et al, 1978), Blosum62 (Henikoff & Henikoff, 1992), mtREV (Adachi & Hasegawa, 1996) and cpREV (Adachi & Hasegawa, 2000). These models specify empirical relative rates of substitution and equilibrium amino acid frequencies. Alternatively, the frequencies can be specified or set to be equal. The GENERAL model allows the user to specify the relative rates of substitution and amino acid frequencies.

Site-specific rate heterogeneity

Site-specific rate heterogeneity allows different sites to evolve at different rates. Two models of rate heterogeneity are implemented. The first is a codon-based model in which the user may specify a different rate for each codon position. This can be used simulate the protein-coding sequences for which the third codon position evolves faster than the first and second because a substitution at this position is considerably less likely to cause an amino-acid substitution. Likewise, the first codon position is expected to evolve slightly faster than the second. Obviously this can only be used with nucleotide models of substitution.

The second model of rate heterogeneity assigns different rates to different sites according to a gamma distribution (Yang, 1993). The distribution is scaled such that the mean rate for all the sites is 1 but the user must supply a parameter which describes its shape. A low value for this parameter (<1.0) simulates a large degree of site-specific rate heterogeneity and as this value increases the simulated data becomes more rate-homogeneous. This can be performed as a continuous model, i.e. every site has a different rate sampled from the gamma distribution of the given shape, or as a discrete model, i.e. each site falls into one of N rate categories approximating the gamma distribution. For a review of site-specific rate heterogeneity and its implications for phylogenetic analyses, see Yang (1996).

Seq-Gen also implements the invariable sites model. With this model, a specified proportion of sites are expected to be invariable across the whole

tree. The expected number of substitutions then fall on the remaining variable sites.

The final way of introducing site-specific rate heterogeneity is to specify a number of partitions and give these partitions relative rates. See section 'Input File Format', below, for details about how to do this.

Compilation and Execution

Seq-Gen is written in ANSI C and should compile on most UNIX systems and workstations. In this manual I will describe the process of installation and compilation on a UNIX system. This applies to Mac OS X if run under the Terminal. Alternatively a Mac OS X package is available that contains a User-Interface program to run Seq-Gen.

Compilation on UNIX

A simple Makefile is included in the package. You should edit this and change the name of the compiler (by default this is **cc**) and add any flags for optimisation on your system (an example is given for SUN compilers). Once this is done, type:

```
make
```

The program will then compile and you will have an executable program: **seq-gen**.

Running Seq-Gen

To run Seq-Gen you type:

```
seq-gen [parameters] < [trees] > [sequences]
```

where [parameters] are the parameters for the program (described below), [trees] is the tree file and [sequences] is the name of the file that will contain the simulated sequences. The tree file must contain one or more trees in PHYLIP format (see below).

The sequences produced by Seq-Gen are written to the standard output (and can thus redirected to the output file using > [filename]). Other information and results are written to the standard error and thus will appear on the screen.

Seeding the random number generator

Seq-Gen uses a pseudo-random number generator called MT19937 devised by Makoto Matsumoto (see the website >http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html< for more details). Like all pseudo-random number generators, this produces a sequence of random numbers for a given 'seed' number. This is why it is pseudo-random: for a particular seed, the algorithm will always give the same sequence of numbers (and thus the same simulated sequences out of Seq-Gen).

The -z option can be used to specify a seed (a non-zero, positive integer). When Seq-Gen is called using the default seed, the actual seed used is printed to the screen. If this number is noted down and then specified using the -z option (and exactly the same simulation settings) then Seq-Gen will generate exactly the same simulated data. Thus by recording the options and seed, it is possible to regenerate a simulated data set. This is useful if the simulated datasets are extremely large - they can be deleted and then reconstructed if required. If you do this, keep a copy of the exact version of Seq-Gen you used because this technique may not work when the seed came from a different version of Seq-Gen.

By default, Seq-Gen uses as its seed the time taken from the system clock (it actually combines the number of seconds that have passed since 1970 with a millisecond timer). This means that if sequential runs of Seq-Gen are done very quickly (e.g., short runs using a script) the default seed could be very similar or even identical. This has serious consequences for the independence of the simulated data. Basically, it is inadvisable to call Seq-Gen many times using a script, running one simulation per call. If the -n option is used to produce multiple simulated data sets from a single call, this will insure adequate independence. If it is necessary to script short runs of Seq-Gen we have two suggestions to avoid the above problem. Firstly, the -z option could be used to call Seq-Gen multiple times in a sequence by deriving a seed for each call using another random number generator. Secondly, adequate time could be allowed to lapse between calls to Seq-Gen (either by using simulating many data sets and then just using the first or by calling a time wasting function in your scripting language). We can't predict whether either of these solutions will work properly so please use caution.

Input 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 number 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, Seq-Gen 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.

Optionally, the user can supply a sequence alignment as input, as well as the trees. This should be in relaxed PHYLIP format. The trees can then be placed in this file at the end, after a line stating how many trees there are. The file may look like this:

```
4 50
Taxon1 ATCTTTGTAGTCATCGCCGTATTAGCATTCTTAGATCTAA
Taxon2 ATCCTAGTAGTCGCTTGCGCACTAGCCTTCCGAAATCTAG
Taxon3 ACTTCTGTGTTTACTGAGCTACTAGCTTCCCTAAATCTAG
Taxon4 ATTCCTATATTCGCTAATTTCTTAGCTTTCCTGAATCTGG
1
(((Taxon1:0.2,Taxon2:0.2):0.1,Taxon3:0.3):0.1,Taxon4:0.4);
```

Note that the labels in the alignment do not have to match those in the tree (the ones in the tree will be used for output) - there doesn't even have to be the same number of taxa in the alignment as in the trees. The sequence length supplied by the alignment will be used to obtain the simulated sequence length (unless the -l option is set). The -k option also refers to one of the sequences to specify the ancestral sequence.

Data partitions with different trees

The user can input different trees for different partitions of the dataset. A partition is a set of contiguous sites that has evolved under a single tree. Using multiple partitions with different trees, a recombinant history for the sequences can be simulated. Assuming a 1000 bp sequence length and 2 partitions consisting of 400bp and 600bp respectively, the following input treefile could be used:

```
[400](((Taxon1:0.2,Taxon2:0.2):0.1,Taxon3:0.3):0.1,Taxon4:0.4);
[600]((Taxon1:0.1,Taxon3:0.2):0.05,Taxon2:0.3,Taxon4:0.4);
```

Note the partition lengths in square brackets before each tree. These must sum to the specified total sequence length (given by the -l option). Multiple sets of partition trees may be input with different trees, numbers of partitions and partition lengths. Seq-Gen will work out the number of partitions for each replicate by the partition lengths (the maximum number of partitions must be given by the -p option.

For example:

```
[400](((Taxon1:0.2,Taxon2:0.2):0.1,Taxon3:0.3):0.1,Taxon4:0.4);
[600]((Taxon1:0.1,Taxon3:0.2):0.05,Taxon2:0.3,Taxon4:0.4);
[300](((Taxon1:0.2,Taxon2:0.2):0.1,Taxon3:0.3):0.1,Taxon4:0.4);
[400]((Taxon1:0.1,Taxon3:0.2):0.05,Taxon2:0.3,Taxon4:0.4);
[300]((Taxon1:0.1,Taxon2:0.2):0.05,Taxon3:0.3,Taxon4:0.4);
```

will generate 2 datasets, the first consisting of 2 partitions (400bp and 600bp) and the second consisting of 3 partitions (300bp, 400bp and 300bp).

Data partitions with different rates

The user can also input the same tree for all partitions of the dataset and then specify a relative rate of evolution for each. This allows partition rate heterogeneity. The relative rates should have a mean of 1.0 (although, if they don't the program will scale them so that they do).

For example:

```
[300, 0.5](((Taxon1:0.2,Taxon2:0.2):0.1,Taxon3:0.3):0.1,Taxon4:0.4);
[400, 1.75](((Taxon1:0.2,Taxon3:0.2):0.1,Taxon2:0.3):0.1,Taxon4:0.4);
[300, 0.75](((Taxon1:0.2,Taxon2:0.2):0.1,Taxon3:0.3):0.1,Taxon4:0.4);
```

will generate 3 partitions (300bp, 400bp and 300bp) with relative rates of 0.5, 1.75 and 0.75 along the same tree.

Output File Format

The default format for the output files was chosen for its simplicity and for the wide range of programs that use it. All of the programs in the PHYLIP package that accept molecular sequences can analyse multiple data sets in the format produced by Seq-Gen. Seq-Gen can now also generate NEXUS format output (see the -o option, below) for use with the PAUP program (Swofford, 1993). A PAUP command block (or any other text) can be inserted between each simulated dataset to automate the analysis process (see the -x option, below).

Parameters to control Seq-Gen

Options and parameters to Seq-Gen are supplied on the command line. The general format is a minus sign followed by a code letter. If required, the values of any parameters come after the code, separated from both code and each other with either a comma or a space. Some options act like switches and require no parameters. The case of the options is ignored.

Model

This option sets the model of nucleotide or amino acid substitution. For nucleotides there is a choice of either *F84*, *HKY* (also known as *HKY85*) or *GTR* (markov general reversable model - also known as *REV*). The first two models are quite similar but not identical. They both require a transition transversion ratio and relative base frequencies as parameters. Other models such as *K2P*, *F81* and *JC69* are special cases of *HKY* and can be obtained by setting the nucleotide frequencies equal (for *K2P*) or the transition transversion ratio to 0.5 (for *F81*) or both (for *JC69*). A transition transversion ratio of 0.5 is the equivalent to equal rates of tansitions and transversions because there are twice as many possible transversions.

For amino acid models there is the choice of *JTT*, *WAG*, *PAM*, *BLOSUM*, *MTREV* or *CPREV*. By default these models specify both the relative rates of substitution and amino acid frequencies. The *GENERAL* model is the amino acid equivalent of the general time reversible and a relative rate matrix and amino acid frequences can be specified using the -r and -f options, respectively.

The usage is:

-m <MODEL>

Where <MODEL> is the model name: HKY, F84, GTR, JTT, WAG, PAM, BLOSUM, MTREV, CPREV or GENERAL. For compatibility with older versions REV is treated as synonymous with GTR.

Length of Sequences

This option allows the user to set the length in nucleotides or amino acids that each simulated sequence should be.

-l <SEQUENCE_LENGTH>

Where <SEQUENCE_LENGTH> is an integer number greater than zero. If an alignment is supplied as input and this option is not set, then Seq-Gen will use the length of the sequences in the alignment.

Number of Replicate Datasets

This option specifies how many separate datasets should be simulated for each tree in the tree file.

-n <NUMBER_OF_DATASETS>

Where <NUMBER_OF_DATASETS> is an integer number that corresponds to the number of datasets to be simulated.

Number of Data Partitions

This option specifies how many partitions of each data set should be simulated. Each partition must have its own tree and a number specifying how many sites are in the partition. See 'Input file format', above, for details. Multiple sets of trees are being inputted with varying numbers of partitions, then this should specify the maximum number of partitions that will be required.

-p <NUMBER_OF_PARTITIONS>

Where <NUMBER_OF_PARTITIONS> is an integer number that corresponds to the number of partitions for each dataset.

Scale branch lengths

This option allows the user to set a value with which to scale the branch lengths in order to make them equal the expected number of substitutions per site for each branch. Basically Seq-Gen multiplies each branch length by this value.

-s <SCALE>

Where <SCALE> is a decimal number greater than zero. For example if you give an value of 0.5 then each branch length would be halved before using it to simulate the sequences.

Scale tree length

This option allows the user to set a value which is the desired length of each tree in units of subsitutions per site. The term *tree length* here is the distance from the root to any one of the tips in units of mean number of subsitutions

per site. This option can only be used when the input trees are rooted and ultrametric (no difference in rate amongst the lineages). This has the effect of making all the trees in the input file of the same length before simulating data.

-d <SCALE>

Where <SCALE> is a decimal number greater than zero. The option multiplies each branch length by a value equal to SCALE divided by the actual length of the tree.

Codon-Specific Rate Heterogeneity

Using this option the user may specify the relative rates for each codon position. This allows codon-specific rate heterogeneity to be simulated. The default is no site-specific rate heterogeneity. This option can only be used with nucleotide substitution models.

-c <CODON_POSITION_RATES>

Where <CODON_POSITION_RATES> is three decimal numbers that specify the relative rates of substitution at each codon position, separated by commas or spaces.

Gamma Rate Heterogeneity

Using this option the user may specify a shape for the gamma rate heterogeneity called alpha. The default is no site-specific rate heterogeneity.

-a <ALPHA>

Where <ALPHA> is a real number >0 that specifies the shape of the gamma distribution to use with gamma rate heterogeneity. If this is used with the -g option, below, then a discrete model is used, otherwise a continuous one.

Discrete Gamma Rate Heterogeneity

Using this option the user may specify the number of categories for the discrete gamma rate heterogeneity model. The default is no site-specific rate heterogeneity (or the continuous model if only the -a option is specified).

-g <NUM_CATEGORIES>

Where <NUM_CATEGORIES> is an integer number between 2 and 32 that specifies the number of categories to use with the discrete gamma rate heterogeneity model.

Proportion of Invariable Sites

Using this option the user may specify the proportion of sites that should be invariable. These sites will be chosen randomly with this expected frequency. The default is no invariable sites. Invariable sites are sites that cannot change as opposed to sites which don't exhibit any changes due to chance (and perhaps a low rate).

-i <PROPORTION_INVARIABLE>

Where <PROPORTION_INVARIABLE> is an real number >= 0.0 and <1.0 that specifies the proportion of invariable sites.

Relative State Frequencies

This option is used to specify the equilibrium frequencies of the four nucleotides or twenty amino acids. If simulating nucleotides, the default (when no frequencies are specified) will be that all frequencies are equal. When simulating amino acids the default frequencies will be set to the empirical values associated with the specified substitution model (with the exception of the GENERAL model which has a default of equal frequencies).

-f <STATE_FREQUENCIES>

Where <STATE_FREQUENCIES> are either four decimal numbers for the frequencies of the nucleotides A, C, G and T or 20 numbers for the amino acid frequencies (in the order ARNDCQEGHILKMFPSTWYV) separated by spaces or commas.

-fe

This results in all the frequencies being set equal. This can be used to force equal frequencies for empirical amino acid models.

Transition Transversion Ratio

This option allows the user to set a value for the transition transversion ratio (TS/TV). This is only valid when either the HKY or F84 model has been selected. The default is to have a TS/TV of 0.5 which gives equal instantaneous rates of transitions and transversions. Thus omitting the -t option with the -mHKY option results in the F81 model (or the JC69 if the -f option is also omitted).

-t <TRANSITION_TRANSVERSION_RATIO>

Where <TRANSITION_TRANSVERSION_RATIO> is a decimal number greater than zero.

General Reversable Rate Matrix

This option allows the user to set values for the relative rate of substitutions between nucleotide or amino acid states. This is only valid when either the (nucleotides) or (amino acids) model has been selected.

-r <RATE_MATRIX_VALUES>

Where <RATE_MATRIX_VALUES> are decimal numbers for the relative rates of substitutions from (for nucleotides) A to C, A to G, A to T, C to G, C to T and G to T respectively, separated by spaces or commas. For amino acids there are 190 rate required representing the upper (off-diagonal) triangle of a 20x20 matrix with amino acids in the order ARNDCQEGHILKMFPSTWYV. The matrix is symmetrical so the reverse transitions equal the ones set (e.g. C to A equals A to C) and therefore only six values need be set.

Ancestral Sequence

This option allows the user to use a supplied sequence as the ancestral sequence at the root (otherwise a random sequence is used).

-k <ANCESTRAL_SEQUENCE_NUMBER>

Where <ANCESTRAL_SEQUENCE_NUMBER> is an integer number greater than zero which refers to one of the sequences supplied as input (see 'Input File Format', above).

Random Number Seed

This option allows the user to specify a seed for the random number generator. Using the same seed (with the same input) will result in identical simulated datasets. This is useful because you can then delete the (often large) simulated sequence files to save disk space. To recreate a set of simulations, you must use exactly the same model options. The default is to obtain a seed from the system clock which will be displayed on the screen allowing it noted down.

-z <RANDOM_NUMBER_SEED>

Where <RANDOM_NUMBER_SEED> is an integer number.

Output file format

This option selects the format of the output file. The default is PHYLIP format.

-op

PHYLIP format.

-or

Relaxed PHYLIP format: PHYLIP format expects exactly 10 characters for the name (padded with spaces if the name is actually less than 10). With this option the output file can have up to 256 characters in the name, followed by a single space before the sequence. The longer taxon names are read from the tree. Some programs can read this and it keeps long taxon names.

-on

NEXUS format: This creates a NEXUS file which will load into PAUP. It generates one DATA block per dataset. It also includes the simulation settings as comments which will be ignored by PAUP.

Insert Text File into Output

This option allows the user to specify text file which will be inserted into the output file after every dataset. This allows the user to insert a PAUP command block or a tree (or anything else) into the file to automate the analysis.

-x <TEXT_FILE_NAME>

Where <TEXT_FILE_NAME> is the name of a file. For Macintosh users this file must be in the same folder as the Seq-Gen program (I find it convenient to copy the Seq-Gen program and move it into the folder in which my datafile are and then delete it afterwards). For UNIX users, this can be specified with a path or should be in the current directory (the one you were in when you run Seq-Gen). This option plus NEXUS format (-on option) replaces the previously included separate program, Phy2Nex.

Write Ancestral Sequences

This option allows the user to obtain the sequences for each of the internal nodes in the tree. The sequences are written out along with the sequences for the tips of the tree in relaxed PHYLIP format (see above).

-wa

Write Site Rates

This option allows the user to obtain the relative rate of substitution for each site as used in each simulation. This will go to stderr (or the screen) and will be produced for each replicate simulation.

-wr

Minimum Information

This option prevents any output except the final trees and any error messages.

-q

Help

This option prints a help message describing the options and then quits.

-h

An example of performing simulations using Seq-Gen

An example phylogeny is included with this package (called 'example.tree'). This is an unrooted tree in PHYLIP format (see 'Input file format', above). To use this tree to simulate 3 sets of sequences 50 nucleotide long using the HKY model, a transition-transversion ratio of 3.0 and unequal base frequencies, type the following:

```
seq-gen -mHKY -t3.0 -f0.3,0.2,0.2,0.3 -l40 -n3 < example.tree > example.dat
```

This produces a PHYLIP format data file called 'example.dat'; which looks something like this:

```
4 50
Taxon1 ATCTTTGTAGTCATCGCCGTATTAGCATTCTTAGATCTAA
Taxon2 ATCCTAGTAGTCGCTTGCGCACTAGCCTTCCGAAATCTAG
Taxon3 ACTTCTGTGTTTACTGAGCTACTAGCTTCCCTAAATCTAG
Taxon4 ATTCCTATATTCGCTAATTTCTTAGCTTTCCTGAATCTGG
4 50
Taxon1 AGAACACAAGTCCCAAATAACCGAACTGGAGCGGGCAGTT
Taxon2 AAGACACAGGCCTAAACTGAGCGTACTGGAACGAGTCGTT
Taxon3 AAGACACAGGTCTCACTTGACTGCGTTGAAACGGTCCGTT
Taxon4 AAGACCCAGACTGTAACTTGTCGGACTGGAATGGACCGTT
4 50
```

Taxon1 CAGCTGAGGCATTACGAAGCGCCCGGCCGGCCGGACGATT
Taxon2 TAACTGAGACAGTACGAAACGCGCAATGGGCCCCAAAACC
Taxon3 CGGTTAGGACATGACGAATCGCCCAGCGGGCCTCCGGACC
Taxon4 CAACTGGAATGTTACCAGCTGCCTAGGGTGCTCCGAGGAC

References

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

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

Dayhoff, M.O., Schwartz, R.M., Orcutt, B.C. (1978) A model of evolutionary change in proteins. In Dayhoff, M.O. (ed.) Atlas of Protein Sequence Structur., Vol5, Suppl. 3, National Biomedical Research Foundation, Washington DC, pp. 345-352.

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

Felsenstein, J. (1993) Phylogeny Inference Package (PHYLIP), Version 3.5. Department of Genetics, University of Washington, Seattle.

Felsenstein, J. and Churchill, G. (1996) A hidden markov model approach to variation among sites in rate of evolution. Molecular Biology and Evolution, 13, 93-104.

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

Henikoff, S., and Henikoff, J. G. (1992) ???. PNAS USA 89:10915-10919.

Jones, D.T., Taylor, W.R. and Thornton, J.M. (1992) The rapid generation of mutation data matrices from protein sequences. CABIOS. 8: 275-282

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

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.

Swofford, D. L. (1993) Phylogenetic analysis using parsimony (PAUP), Version 3.1.1. Illinois Natural History Survey, Champaign.

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.

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

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. (1994) Estimating the pattern of nucleotide substitution. J. Mol. Evol., 39, 105-111.

Yang, Z. (1996) Among-site rate variation and its impact on phylogenetic analyses. Tr. Ecol. Evol., 11, 367-372.

Version History.

Bug fixed in version 1.3.2 - 7 Jan 2005

- The PAM (Dayhoff) and Blosum matrices were interchanged so specifying one would result in the other being used instead.

Bug fixed in version 1.3.1 - 4 Nov 2004

- Specified nucleotide frequencies were being ignored and equal frequencies being used instead.

New Features in version 1.3 - 30 Aug 2004

- Added amino acid simulation to Seq-Gen. This replaces PSeq-Gen which was not being updated but also adds a number of other amino acid models.
- Removed the limit on tree size. The only limit now is the available memory.

- Updated to the latest version of the MT19937 random number generator.

New Features in version 1.2.7 - 19 Nov 2003

- Replaced the random number generator with the high quality Mersenne Twister. When Seq-Gen was originally written, the computation of random numbers was a significant burden but with the speed of current computers this is no longer an issue.

New Features in version 1.2.6 - 4 Dec 2002

- Improved resolution of the automatic seeding of the random number generator by adding some milliseconds to it. Thus runs of Seq-Gen that are less than a second apart will have different seeds. This probably only matters on UNIX machines using scripts to do multiple (short) runs.

New Features in version 1.2.5 - 25 Sep 2001

- New option to write the relative rate used for each for each site

New Features in version 1.2.4 - 6 Jul 2001

- Can now specify a relative rate for each partition. The partitions are specified in the tree files but all the partitions can be given the same tree but different rates.

Bug fixed which resulted in missing 'Begin Data' in NEXUS files when creating a single set of sequences.

New Features in version 1.2.3 - 6 Apr 2001

- Added feature write ancestral sequences (-w option).
- Improved the interface of the Macintosh version. Can now drag a tree onto the application - this tree will then be selected as input in the opening dialog box. This box has been made wider to allow longer command-lines.
- New Macintosh Carbon version that will run on MacOS X and MacOS 9.0 this can be found in the Macintosh package along with a version that will run on pre-MacOS 9 computers.

Bug fixed in Macintosh which would result in some of the end of a long command line being ignored.

Bug fixed in Version 1.22 - 4 Feb 2001

Fixed a bug which prevented unrooted trees from loading (complained about polytomies in the tree).

Bug fixed in version 1.21

Fixed a bug which prevented single partitions being simulated (i.e. most people's simulations). Updated make file in UNIX version.

New Features in version 1.2

- Invariable sites model. You can now specify a proportion of invariable sites using the **-i** option, outlined below.
- The default model is now HKY (instead of F84). I think it is now more widely used and the computational difference between them has become small.
- If you don't specify a TS/TV under either the HKY or F84 models, then the TS/TV is chosen to make the instantaneous rate of TSs and TVs equal. This has the effect of collapsing both models to K2P or F81 (depending on whether the base frequencies are equal or note). For the F84 this TS/TV will be 0.5 but for HKY this will be dependent on the base frequencies (for equal base frequencies this will be 0.5). TS/TV used to always default to an arbitrary value of 2.0.
- Output in NEXUS format. You can choose the format of the output using the &#o option followed by a code specifying PHYLIP, relaxed PHYLIP or NEXUS. When creating NEXUS format files, the name of a file containing a PAUP command block (or any other text) can be specified and this will be inserted into the output after every simulated dataset. This is done with the **-x** command.
- Simulate different partitions of the data under different trees. This allows the simulation of a recombinant history. The trees for each partition are given with the length of the partition in square brackets before it. The **-p** option specifies the number of partitions (the **-p** option used to specify the output format).
- Will now detect and disallow trees containing polytomies. Polytomies can be simulated by Seq-Gen if they are resolved arbitrarily with zero internal branches. This can be done automatically by TreeEdit, available at http://evolve.zoo.ox.ac.uk/software/TreeEdit/.
- Version 1.2 no longer has a Mac 68K executable. It should still compile for these machines but I don't have the time to support it.
- The default seed to the random number generator now has more resolution. Previously Seq-Gen used the system time in seconds. This means that if two short runs of Seq-Gen we executed in less than a second, the same random number seed could be used resulting in two identical datasets. It now adds a higher resolution clock to the seed. It

will also print out the seed used so that you can check for this problem. You can also specify your own seed using the **-z** option. If you write down the seed outputted, you can use this to recreate an exact set of simulations. This is useful because you can then delete the (often large) simulated sequence files to save disk space. To recreate a set of simulations, you must use exactly the same model options.

Bug fixed in version 1.1

WARNING: a very important bug has been fixed in this release. Many apologies to anyone to whom this is relevent. All versions of Seq-Gen prior to this have not reassigned the gamma rate categories for each site between replicate simulations. This means that the same site will have the same rate (in both the discrete and continuous model) between replicates. This will reduce the amount of variability in a set of simulations.