User's guide for CodonRates 1.0

Tae-Kun Seo, Ph.D.
Professional Programme for
Agricultural Bioinformatics,
University of Tokyo,
1-1-1 Yayoi Bunkyo-Ku Tokyo
113-8657 Japan
TEL) +81-3-5841-1139
FAX) +81-3-5841-5068
E-mail) seo@iu.a.u-tokyo.ac.jp

Jul 16, 2005

INTRODUCTION

What is CodonRates ?

CodonRates is a program to estimate divergence times and 'absolute rates' of synonymous and nonsynonymous substitution with protein-coding genes. The conventional programs, such as PAML (Yang 1997), HYPHY (Kosakovsky Pond and Muse 2000), are available for the estimation of the 'amounts' of nonsynonymous and synonymous substitution and their ratio. The evolutionary amount is confounded with time and rate, and *CodonRates* is designed to separate the evolutionary amount into time and synonymous and nonsynonymous rates under Bayesian framework without relying on molecular clock. Other information than sequence data, such as fossil constraints and prior distribution for the root time, are required. For the application of *CodonRates*, see Seo et al. (2004). For the more general things about Bayesian estimation of divergence times, see Thorne et al. (1998), Kishino et al. (2001) and Thorne and Kishino (2002).

How to get and how to compile CodonRates

You can get *CodonRates* from Tae-Kun Seo upon email request (seo@iu.a.u-tokyo.ac.jp). The web page for this program is being constructed. For Windows users, *CodonRates* software package includes an executable file (codonrates.exe), source files and a project file for Microsoft's Visual Studio 6.0. For unix users, a makefile is provided. At unix prompt, you could compile the source files by typing ‘make’ if your unix system has C++ compiler.

FORMATS AND OPTIONS

I will explain how to use *CodonRates* via an example. The distributed package of *CodonRates 1.0* includes the files for this example analysis. Make sure that *testseq1.txt*, *testseq2.txt*, *testtree.txt*, *codonrates_option.txt* and executable file *codonrates* (*codonrates.exe* in Windows system) are all in the same directory in which you perform the analysis. Because *CodonRates* does not consider the uncertainty of tree topology, the tree you assume should be saved in *testtree.txt*.

Suppose that you assume the tree shown in Figure 1. You have six ingroup sequences (TaxonA - F) and one outgroup sequence (TaxonG). You have the prior information for the ingroup root time. The mean and standard deviation of this prior are 80.0 and 5.0 million years ago (MYA) respectively. This prior information might have come from fossil information or previous studies. In addition to the root time information, you have the fossil information such that the most recent common ancestor (MRCA) of TaxonD, E and F is between 10 and 20 MYA and that the MRCA of TaxonC, D, E, and F is older than 30 MYA. In the following, I will explain how to set up these fossil information and the prior distribution of root time to estimate divergence times and evolutionary rates.

In the analysis using *CodonRates*, the current time – in which all sequences are sampled – is set to be 0 MYA and time increases to the past. The current version (1.0) of *CodonRates* can deal with only contemporaneous samples.

Sequence and tree files

Figure 2 shows the format of sequence file. Sequence file should begin with the number of sequences and the length in nucleotides (i.e. three times the number of codons) in the first line. A taxon name should begin after ‘>’ character in the new line, and sequence data should follow sequentially in the next line. Any spaces, tabs and new lines that are inserted within sequence data is ignored. The current version (1.0) of *CodonRates* accepts only one of A,C,T,G or - as a sequence character. ‘-’ represents gap or missing character. In this manual, I will use “...” when I want to omit lines or paragraphs within files.

As shown in Figure 3, the tree file – *testtree.txt*, in this example – should begin with the number of taxa in the first line. The topology should come in the next line. The outgroup should be located in the

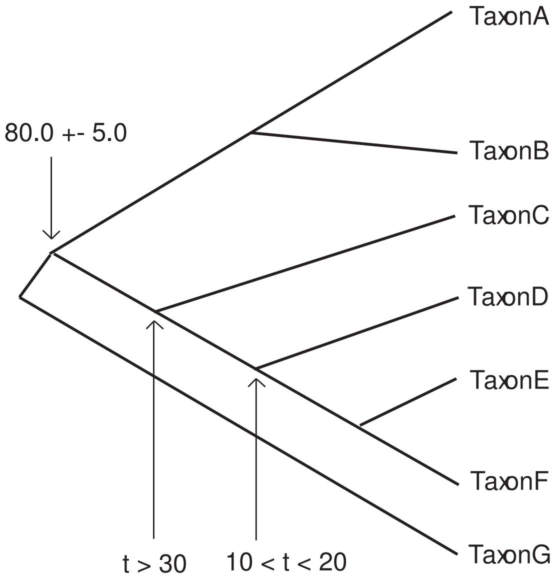

Figure 1: testtree.txt

```
 7 1500
>TaxonA
CGCCTGGGCACTCGCTTACCGGTTCGG...
....
>TaxonB
CGTCTGTGCACCCGCCTACCCGTGCGT...
....
....
>TaxonG
CGTCTAAGCACCCGCTTACCTGTCCGT...
```

Figure 2: testseq1.txt

```
 7
((TaxonA,TaxonB), (TaxonC,(TaxonD,(TaxonE,TaxonF))), TaxonG);
```

Figure 3: testtree.txt

rightmost place within the outmost parenthesis as follows.

(Ingroup1,Ingroup2,OutGroup);

In the above, ingroup root is trifurcating to Ingroup1, Ingroup2, and OutGroup. Ingroup1, 2 and OutGroup can be bifurcating or multifurcating subtrees or a single taxon. If ingroup root is more than trifurcating, then the tree should be as follows.

(Ingroup1,Ingroup2,$\cdots$,IngroupN,OutGroup);

Option file

Options are stored in 'codonrates_option.txt' as shown in Figure 4. The message after "//" are ignored by *CodonRates*. Do not remove or add '<' and '>' characters in the option file, because they are used to denote the starting and ending of each option item. Also, do not make any change between '<' and '>' characters. Except <mcmc option>and <minor option>, all option should be placed in the same line after '>' character.

(1) <number of genes>

In this option, you set the number of 'genes'. You specify here whether your analysis is single-gene or multi-gene analysis. When you set the number larger than 1, the multiple genes are assumed to have common divergence times but have separate sets of synonymous and nonsynonymous rates over time. The number of entries in the following options should be same as the number you set here: <gene name>, <sequence file>, <hess file>, <estimation output file>, <mcmc output file>, <codon table>, <codon frequency>, and 3rd-6th lines (prior median and std of Root SynRate and Root NonsynRate) of <mcmc option>. The order of entries of these options should be consistent. Depending on your job (see <Job >option below), some options may have nothing to do with your current job. Even in this case, you should fully specify the entries of these options.

In our example of analysis, we are using two genes which are saved in *testseq1.txt* and *testseq2.txt*. Number 2 is set here.

```
<number of genes>  2
<gene name>  testseq1 testseq2
<sequence file>  testseq1.txt testseq2.txt
<tree file>  testtree.txt
<hess file>  testseq1brhess.txt testseq2brhess.txt
<estimation output file>  testseq1est.txt testseq2est.txt
<mcmc output file>  testseq1mcmc.txt testseq2mcmc.txt
<omega option>  0 // 0:  separate omega, 1:  global omega
<codon table>  2 2 // see below
<codon frequency>  3 3 // see below
<Job>  2 // see below
<mcmc option>
   -1 10000 1000 1000 // burn in, # of interval, # of sample
   80 5.0 // (1)mean (2)std dev of Gamma prior root time
   0.01 0.01 // prior median of Root SynRate
   0.01 0.01 // prior std of Root SynRate
   0.01 0.01 // prior median of Root NonsynRate
   0.01 0.01 // prior std of Root NonsynRate
   0.01 0.01 //(1)prior NuMean(syn), (1)prior NuMean(nonsyn)
   100.0 0.0001 0.0001 // initial (1)root time, (2)nu_s, (3)nu_n
   1 // 0:  no constraint, 1:  constraint
<minor option>
   0 // ShowDetailOptimization, 0:  minimum to screen, 1:  more to screen

###### (option info) ######
<Job>  0:  Br only, 1:Hess Only, 2:  Br&Hess, 3:  prior, 4:  posterior
<codon table>  1:Universal codon 2:vertebrate MT 3:yeast 4:mold
5:invertebrate MT
<codon frequency>  0:1/61, 1:F1x4, 2:F3x4 3:empirical
```

Figure 4: codonrates_option.txt

(2) <gene name>

The analysis using *CodonRates* is separated into two steps: (1) Getting maximum likelihood estimates (MLE's) of parameters and calculating first and second derivative of MLE's for the approximation of likelihood function, and (2) Getting approximation of posterior distribution using Markov chain Monte Carlo (MCMC) procedure. The gene names you specify here should be same in both steps of the analysis. The gene names are stored in the result files from the first step of the analysis. The second step of the analysis reads in the result files from the first step and check if the stored gene names are same as the entries you specify here. This prevents users from using inappropriate result files from the first step in the analysis of the second step.

(3) <sequence file>

Sequence file names are specified here.

(4) <tree file>

Specify tree file here. Tree file should be common to all genes in the multi-gene analysis.

(5) <hess file>

To save the computational time, *CodonRates* employs the approximation of likelihood function in the MCMC step (see the Appendix of Seo et al. 2004). The information for the approximation is stored in the files you specified here. In the last part of these files, you are supposed to set the time constraints which are used in MCMC step. For the example of setting time constraints, see the 'EXAMPLE OF ANALYSIS' section.

(6) <estimation output file>

Maximum likelihood estimates (MLE's) of synonymous and nonsynonymous branch lengths and their sums, transition/transversion ratio (κ), ω's, are summarized in these files. Note that *CodonRates* measures synonymous(nonsynonymous) branch length in terms of synonymous (nonsynonymous) substitution per codon, not per synonymous (nonsynonymous) site. So, the ω is not equal to the simple ratio of nonsynonymous branch length over the synonymous branch length. If we pay attention to the fact that 1st and 2nd sites are nonsynonymous and 3rd sites are mostly synonymous, the number of nonsynonymous sites is roughtly two times the number of synonymous sites. This fact should be considered if you want to transform the simple ratio of nonsynonymous branch length over synonymous branch lengths to the ω (For detail, see Seo et al. 2004).

(7) <mcmc output file>

The summaries of prior (case of 3 in <Job>option) or posterior distributions (case of 4 in <Job>option) of times and rates are saved in the files you specify here. The samples from posterior distributions are saved in the files named "file names you specify here" + "stored.txt". For example, you have two entries *testseq1mcmc.txt* and *testseq2mcmc.txt* in this option. Then, the summaries of posterior distribution will be saved in these files, and the posterior samples will be saved in '*testseq1mcmc.txt***stored.txt**' and '*testseq2mcmc.txt***stored.txt**' respectively.

(8) <omega option>

'separate omega' means that different lineages have different ω values (dN/dS ratios) which represent selective pressure in codon model (Goldman and Yang 1994). With 'global omega', all lineages have common ω values. In the current version (1.0) of *CodonRates*, you can estimate divergence times only with 'separate omega' option. But, if you are not interested in time/rate estimates and if you want to obtain just MLE's under the codon model in which ω is common over all lineages, you can use 'global omega' option.

(9) <codon table>and <codon frequency >

The number of entries in these options should be same as the number you specify in <number of gene>. Codon frequencies in the codon model (Yang 1997;Yang 1998; see also Seo et al. 2004) are assumed to be one of equal (1/61 for universal codon or 1/60 for vertebrate mitochondrial protein-coding genes), proportional to the multiplication of average nucleotide frequencies (F1x4: considering common frequencies of A,C,T,G for three codon sites), proportional to the multiplication of nucleotide frequencies at each codon site (F3x4: considering different frequencies of A,C,T,G in three different codon sites), or same as empirical codon frequencies (empirical).

(10) <Job>

By setting 0 ('Br Only'), you can get MLE's of parameters, such as branch lengths, transition/transversion ratio (κ), ω's, and etc. The summaries of the results are saved in the files that are specified in <estimation output file>. The information for MLE's are also saved in the files that are specified in <hess file>, but the first and second derivatives (referred to as Hessian matrix in this manual) which are required for the approximation of the likelihood function (see the Appendix of Seo et al. 2004) are not saved in <hess file>with 'Br Only' option. Next, by setting 1 ('Hess Only'), keeping other options unchanged, and running program, you can calculate the Hessian matrices and save them in the files that are specified in <hess file>. By setting 2 ('Br&Hess'), you can do two jobs specified with 0 and 1 consecutively. If your data set is extremely huge or the memory of your computer is extremely small, separate running is recommended.

By setting 3 ('prior'), you can get the approximation of prior distributions of times and rates. In this prior approximation, the information from sequence data is not considered. Prior distributions of rates and times, prior distributions of rate change parameters ν_n and ν_s, and the time constraints given by fossil information are considered in the prior approximation. It is recommended to examine the prior distributions before obtaining posterior distributions to check if prior distributions you assume are reasonable.

By setting 4 ('posterior'), you can combine the information from sequence data with prior distributions and get the approximation of posterior distributions.

(11) <mcmc option>

This is the option for the approximation of prior (3 of <Job>) or posterior (4 of <Job>) distributions. If your <Job> is one of 0, 1 and 2, you can fill the entries of this option with arbitrary numbers except for the seed number. Seed number should be negative integer and it is used to determine random starting point for MLE's optimization (0 or 2 of <Job>) or for MCMC procedure (3 or 4 of <Job>). Note that this option should be properly filled no matter what your <Job> is.

In the first line of this option, you should specify the seed of random number, the 'number of burn-in', the 'number of interval' and the 'number of samples'. The 'number of burn-in' is the number of steps you discard in the beginning of MCMC procedure so that your result is not affected by the starting point. The 'number of samples' means the number of samples you want to choose from prior (3 of <Job>) or posterior (4 of <Job>) distributions. The 'number of interval' is the number of steps in MCMC between each sample so that the consecutive samples are not correlated each other.

In the second line, you are supposed to set the mean and standard deviation (std) of gamma distribution of prior root time. You can arbitrarily set these values if your <Job> is one of 0, 1 and 2 because these entries have nothing to do with obtaining MLE's nor calculating 1st and 2nd derivatives. However, I recommend you to set the means and std that will be used in the approximation of prior (3 in <Job>) or posterior (4 in <Job>) distributions even when your <Job>is one of 0, 1 and 2. Based on the mean and std you specify here, the proper recommendations for the prior distributions of root rates are generated and saved in the files specified in <estimation output file> when <Job>is one of 0, 1 and 2 (see Figure

9). If you arbitrarily fill these entries when your <Job>is one of 0, 1 and 2, you should manually calculate the prior medians of root rates as suggested in Figure 9.

In 3rd-6th line, prior medians and standard deviations of synonymous and nonsynonymous root rates should be specified for MCMC running. These entries have nothing to do with obtaining MLE's (0 or 2 in <Job>) nor calculating 1st and 2nd derivatives (1 or 2 in <Job>). But, even in these cases, you should feel these entries and the number of entries should be consistent with the number specified in <number of gene>. You can specify arbitrary numbers here (0.01 in our example) when your <Job> is one of 0,1, and 2. When your <Job> is 3 or 4, you can find the suggestions for these values in the last part of the files specified in <estimation output file> (see Figure 9).

In the 7th line, you specify the means of the prior distributions of ν_s and ν_n. These quantities represent how much synonymous and nonsynonymous rates change per unit time. *CodonRates* assumes bivariate log-normal distribution as a model of rate change. That is, for a branch with time duration t beginning in node i and ending in node j, the logarithms of the rates at the node j follow a bivariate normal distribution of the form:

$$\begin{pmatrix} \log r_n^j \\ \log r_s^j \end{pmatrix} \sim N\left(\begin{pmatrix} \log r_n^i \\ \log r_s^i \end{pmatrix}, \begin{pmatrix} \nu_n t & 0 \\ 0 & \nu_s t \end{pmatrix} \right), \tag{1}$$

where r_n^i and r_s^i are nonsynonymous and synonymous rates at a node i respectively. When ν_s and ν_n are both equal to zero, this model is reduce to molecular clock because it does not allow rate variation over time. *CodonRates* assumes that the distributions of ν_s and ν_n are exponential, in which standard deviations are same as the means. I recommend to use the same values for these two entries, because it enables you to compare the posterior distributions of ν_s and ν_n. With the same priors, if the means of posteriors are significantly different, you could say that one rate varies more over time than the other rate. For the means of prior distributions of ν_s and ν_n, it seems to be good to use the values of moderate range. Because ν_s (ν_n) represents the variance of the synonymous (nonsynonymous) rate change per unit time in log unit (Eq. 1), a small mean allows the small amount of rate variation over time *a priori*. This makes the analysis similar to the conventional molecular clock approach. A large mean allows the large amount of rate variation over time *a priori* and it may cause difficulties in obtaining posterior distribution of ν parameters when there are small number of taxa. For the means of the prior distributions of ν_s and ν_n, it seems to be good to use the values whose multiplication with the prior mean of root time is between 0.5

and 2.5 (Thorne, personal communication). In this example, you set the prior mean of ν_s (ν_n) to be 0.01. The multiplication of 80 (the mean of root prior specified in the second line of <mcmc option>) and 0.01 is 0.8 and 0.8 is within the range of (0.5, 2.5).

In the 8th line, you specify starting points of root time and ν_s and ν_n in the MCMC step. It is recommended to run MCMC multiple times with different starting points and different random seed number. You should get similar results with the multiple MCMC runs. If your multiple MCMC runs produce different results, you need to increase the 'number of burn-in', the 'number of interval' and the 'number of samples' in the first line of <mcmc option>.

In the 9th line, you specify whether you have time constraints provided with fossil information. Note that the gamma distribution of root time (the 2nd line of <mcmc option>) is not regarded as time constraint. If <Job> is one of 0, 1, and 2, this entry is meaningless (but you must have an arbitrary entry here). If <Job> is 3, or 4, and if you set 1 in this option, you should specify time constraints in the end of the files specified in <hess file>. For the example of setting time constraints, see Figure 10.

(12) <minor option>

Currently, only *ShowDetailOptimization* option is implemented in <minor option>. If you set 0, you can see the minimum display to the screen.

EXAMPLE OF ANALYSIS

In this section, I will explain how to estimate divergence times and rates with *CodonRates* via an example. Make sure that you have *testseq1.txt*, *testseq2.txt*, *testtree.txt*, *codonrates_option.txt* and executable file *codonrates.exe* in the directory (C:\example) where you want to perform the analysis. I show the example of the analysis at Windows(or DOS) system.

Step 1 : Estimation of branch lengths and Hessian matrices

After setting *codonrates_option.txt* as Figure 4, type *codonrates* and hit enter to get the MLE's and Hessian matrix.

```
C:\example>codonrates  +  Enter
```

Results of Step 1

MLE's of branch lengths and ω's are saved in *testseq1est.txt* and *testseq2est.txt* that are specified in <esti-

```
Version:  CodonRates1.0
testseq1 :  gene name
Model:  codon model(Vertebrate Mt, EmpiricalCodon)
LogLike = -5967.21343293
Ts/Tv = 5.51634
Estimated Tree(total branch):
((TaxonA:0.598737,TaxonB:0.440491):0.409126,(TaxonC:0.654248,(TaxonD:...
Estimated Tree(nonsyn branch):
((TaxonA:0.051010,TaxonB:0.105283):0.066670,(TaxonC:0.068498,(TaxonD:...
Estimated Tree(syn branch):
((TaxonA:0.547727,TaxonB:0.335208):0.342456,(TaxonC:0.585750,(TaxonD:...
......
```

Figure 5: testseq1est.txt (1)

mation output file>. The information for Hessian matrices are saved in *testseq1hess.txt* and *testseq2hess.txt* that are specified in <hess file>.

Figures 5 - 9 show the results of testseq1 which are saved in *testseq1est.txt*. As shown in Figure 5, the version of *CodonRates*, gene name, codon table and type of codon frequencies are shown in the beginning of the file. Next, the maximum log-likelihood value and MLE of transition/transversion ratio (κ) are shown. After these values, three trees are generated: the tree of total branch lengths, the tree of nonsynonymous branch lengths, and the tree of synonymous branch lengths. Note that synonymous (nonsynonymous) branch lengths are measured in unit of synonymous (nonsynonymous) substitution per ‘codon’, not per ‘synonymous (nonsynonymous) site’. Thus, the total branch length is simply the sum of synonymous and nonsynonymous branch lengths, but the ω is not simply the ratio of nonsynonymous branch length over the synonymous branch length. For detail about this issue, see Seo et al. (2004).

Next, you see the information for the numbering of nodes as in Figure 6. “Node W: X->, <-Y, <-Z” means that “Node X is the ancestor of node W, and nodes Y and Z are the descendants of node W”. For example, node 10 is the ancestor of node 6. Node 0 and 1 are two descendants of node 6.

Next, as in Figure 7, you see the detail information for MLE’s of branch lengths, ω’s, synonymous and nonsynonymous branch lengths. The first column denotes the indices of branches. The numbers of nodes which connect the branch are shown in the second column. For example, two ending nodes of branch 0 is node 6 and node 0. After Figure 7, you can see the information for the frequencies of A,C,G,and T (Figure 8).

```
......
Node 0:  TaxonA
Node 1:  TaxonB
Node 2:  TaxonC
Node 3:  TaxonD
Node 4:  TaxonE
Node 5:  TaxonF
Node 6:  10->, <-0, <-1
Node 7:  8->, <-4, <-5
Node 8:  9->, <-3, <-7
Node 9:  10->, <-2, <-8
Node 10:  11->, <-6, <-9
Node 11:  TaxonG
......
```

Figure 6: testseq1est.txt (2)

```
......
 Br#  Node--Node     TotalBr       Omega       SynBr    NonsynBr
   0     6--0     0.59873696  0.04728670  0.54772659  0.05101037
   1     6--1     0.44049105  0.15947329  0.33520810  0.10528296
   2     9--2     0.65424809  0.05937556  0.58575040  0.06849769
   3     8--3     0.59038654  0.02368092  0.56407819  0.02630835
   4     7--4     0.29501405  0.02918493  0.27897846  0.01603559
   5     7--5     0.45375484  0.00684958  0.44771505  0.00603979
   6    10--6     0.40912582  0.09884828  0.34245598  0.06666985
   7     8--7     0.46819915  0.00890587  0.46012846  0.00807069
   8     9--8     0.39116902  0.00477970  0.38752105  0.00364797
   9    10--9     0.32007606  0.02734894  0.30371675  0.01635931
  10    10--11    0.23089796  0.03967679  0.21416257  0.01673539
......
```

Figure 7: testseq1est.txt (3)

```
......
Site# & sitewise base frequencies(T,C,A,G)
1 0.19914286 0.28171429 0.26114286 0.25800000
2 0.27085714 0.24885714 0.23571429 0.24457143
3 0.27142857 0.25971429 0.24571429 0.22314286
Total base frequencies of T, C, A, and G
0.24714286 0.26342857 0.24752381 0.24190476
......
```

Figure 8: testseq1est.txt (4)

```
......
Sum of lengths from root to tip (Syn, Nonsyn)
TaxonA: 0.89018257 0.11768021
TaxonB: 0.67766407 0.17195281
TaxonC: 0.88946715 0.08485700
TaxonD: 1.25531599 0.04631563
TaxonE: 1.43034472 0.04411357
TaxonF: 1.59908131 0.03411776
Median:  1.07274928 0.06558631
Standard dev of Syn,Nonsyn br lengths from root to tip
0.35935626 0.05360216

### Suggestions for the prior distributions of root rates
### When mean of prior root time is 80,
 - suggested exponential distributions of root prior rates..
     Median of SynRate = 0.0134094
     Std of SynRate = 0.0193456 <= Median of SynRate / Log(2.0)
     Median of NonsynRate = 0.000819829
     Std of NonsynRate = 0.00118276 <= Median of NonsynRate / Log(2.0)
  - suggested gamma distributions of root prior rates..
     Median of SynRate = 0.0134094
     Std of SynRate = 0.00449195
     Median of NonsynRate = 0.000819829
     Std of NonsynRate = 0.000670027
When mean of prior root time is NOT 80,
  - suggested exponential distributions of root prior rates..
     Median of SynRate = 1.07275/(time you want to use)
     Std of SynRate = Median of SynRate / Log(2.0)
     Median of NonsynRate = 0.0655863/(time you want to use)
     Std of NonsynRate = Median of NonsynRate / Log(2.0)
  - suggested gamma distributions of root prior rates..
     Median of SynRate = 1.07275/(time you want to use)
     Std of SynRate = 0.359356/(time you want to use)
     Median of NonsynRate = 0.0655863/(time you want to use)
     Std of NonsynRate = 0.0536022/(time you want to use)
......
```

Figure 9: testseq1est.txt (5)

In the last part of *testseq1est.txt*, the sums of synonymous and nonsynonymous branch lengths from ingroup root to each tip are listed as in Figure 9. This information is provided so that the users can determine the prior distributions of root rates which are specified in the 3rd - 6th lines of <mcmc option>. For example, the sum of synonymous (nonsynonymous) branch lengths from ingroup root to TaxonA is 0.89018257 (0.11768021), and the sum of synonymous (nonsynonymous) branch lengths from ingroup root to TaxonB is 0.67766407 (0.17195281). The medians of synonymous and nonsynonymous sums are 1.07274928 and 0.06558631. The standard deviation of these sums are 0.35935626 and 0.05360216. These medians and standard deviations are used to determine the prior distributions of root rates. If your prior mean of root time is 80 MYA which is specified in the second line of <mcmc option>, the medians of synonymous and nonsynonymous prior root rate can be obtained by the medians of synonymous (1.07274928) and nonsynonymous (0.006558631) sum of branch lengths divided by the mean of root time (80 MYA). That is, 0.0134094 (=1.07274928/80) and 0.000819829 (=0.006558631/80) are suggested for the medians of synonymous and nonsynonymous root rates. If you want to assume the exponential distributions for root rates, the standard deviation (std) is directly calculated with median (std = median/log(2.0)). If you want to assume the gamma distributions for root rates, the std can be obtained in a similar way to obtaining medians. In this example, we use the exponential distributions for root rates. The suggestion shown in Figure 9 is used in the 3rd-6th line of <mcmc option>in the 'step 2' of *CodonRates* analysis.

Analysis Step 2 : Markov chain Monte Carlo

(1) Setting the time constraints

The information for the 1st and 2nd derivatives is stored in *testseq1brhess.txt* and *testseq2brhess.txt* which are specified in <hess file> option. When you have time constraints, you should specify time constraints in these files. The part that you should change is the last part starting with "#_of_Constraint" (Figure 10). The last part of Figure 10 shows the example of the format of time constraints which is automatically generated by *CodonRates*. If you set 0 for "#_of_Constraint", all lines below it are ignored. When you set time constraints, you should specify node numbers and their lower and upper bounds. If the upper bound is not available, specify default upper bound (10000000000, or 1.0e10. Note that there are ten zeros.). If you set different value other than 10000000000 by mistake, then the node will be regarded as constrained by two sides. With Figures 1 and 6, we see that node 8 is between 10 and 20 MYA and node 9 is older

```
testseq1 :  gene name, version CodonRates1.0
Br0(Eta)= 0.95484837313374993
Br1(Eta)= 0.86246057220545413
Br2(Eta)= 0.94395230083081283
Br3(Eta)= 0.97686688897671348
Br4(Eta)= 0.97164267318545217
...
...
9th____syn_1stD= 0.0039339265928787937
9th_nonsyn_1stD= 0.010632184623383441

#_of_Constraint 0 //0:  no constraint
 Node#  Lower_Bound  Upper_Bound  (default Upper_Bound:  10000000000 )
   0       0.0          2.0
   1       0.0      10000000000
```

Figure 10: testseq1brhess.txt

than 30 MYA. The last part of all files specified in <hess file>(*testseq1brhess.txt* and *testseq2brhess.txt*) should be changed as follows.

```
...
#_of_Constraint 2 //0:  no constraint
 Node#  Lower_Bound  Upper_Bound  (default Upper_Bound:  10000000000 )
   8       10.0         20.0
   9       30.0     10000000000
```

Because *CodonRates* uses 1.0e10 as an upper bound of time, it is important that the actual node times should be far less than this upper bound. This means that time unit must be carefully selected. In our example, the unit of time is million years. If you use the unit of day instead of million years, 80 million years corresponds to ≈2.92e10 days. This is larger than the default upper bound and will cause problems in running 3 or 4 in <Job > option. I recommend the user to use the moderate unit of time to avoid this problem.

(2) Setting the prior root time and rates.

Following the suggestion shown in Figure 9 (see the explanation for Figure 9), you can specify the prior median and standard deviation of synonymous and nonsynonymous rates at root node in 3rd - 6th lines of <mcmc option>. If you want to use the exponential distributions for root rates, you should change <mcmc option> of *codonrates_option.txt* which is shown in Figure 4 as follows.

```
......
<mcmc option>
   -1 10000 1000 1000 // burn in, # of interval, # of sample
   80 5.0 // (1)mean (2)std dev of Gamma prior root time
   0.0134094 0.0107195 // prior median of Root SynRate
   0.0193456 0.0154649 // prior std of Root SynRate
   0.000819829 0.0016494 // prior median of Root NonsynRate
   0.00118276 0.00237958 // prior std of Root NonsynRate
   0.01 0.01 //(1)prior NuMean(syn), (1)prior NuMean(nonsyn)
   10.0 0.0001 0.0001 //(1)initial root, (2)initial nu_s, (3) initial nu_n
   1 // 0:  no constraint, 1:  constraint
......
```

After you do (1) and (2) above, change 2 to 4 in <Job> option of Figure 4, and run the program as follows.

C:\example>codonrates   +   Enter

Results of Step 2

The summaries of posterior distributions of times and rates are saved in the *testseq1mcmc.txt* and *testseq2mcmc.txt* which you specified in <mcmc output file>option. Samples from posterior distributions are stored in the files named *testseq1mcmc.txt***stored.txt** and *testseq2mcmc.txt***stored.txt** (see the explanation for <mcmc output file>). In the following, I will explain the summaries of posterior distributions stored in *testseq1mcmc.txt*.

(1) Posterior distributions of ν parameters (Figure 11)

ν_n and ν_s are quantifying measure for nonsynonymous and synonymous rate change per unit time (Seo et al. 2004). By investigating the posterior distribution of the logarithm of their ratio, we can see which rate has more variation over time. Figure 11 shows the posterior mean and standard deviation of ν_n, ν_s, $\log(\nu_n/\nu_s)$ and their 95% credibility intervals. When the prior distributions of ν_n and ν_s are same exponential distributions as in our example, it is possible to show analytically that $\log(\nu_n/\nu_s)$ is symmetrically distributed around 0 and its 95% confidence interval is (-3.664, 3.664) *a priori*. In our analysis, the posterior mean of log of ν_n/ν_s is 0.6667, which means that data show more variation in nonsynonymous rate change over time. But this pattern is not significant, because 95% credibility interval, (-0.7568, 2.098), still contains zero.

(2) Concordance statistic (Figure 12)

Concordance statistic (S) shows whether synonymous and nonsynonymous rate change over time in a

```
Version:  CodonRates1.0
gene name:  testseq1

Post mean Nu(Nonsyn) = 0.02680+-0.01379 (0.008029,0.06062)
Post mean Nu(Syn) = 0.01393+-0.007562 (0.004307,0.03377)
Post mean log(Nu_n/Nu_s)= 0.6667+-0.7371 (-0.7568,2.098)
     Probability of tail part which is less than zero 0.189
     When two priors are same, mean of log(Nu_n/Nu_s)= 0.0 +- 1.814,
(-3.664, 3.664)
......
```

Figure 11: testseq1mcmc.txt (1)

```
......
Null S   0.4998+-0.05754 (0.3847,0.6153)
Post prob.  of concordance(S) = 0.4703, Lower P-value = 0.30892, Upper
P-value = 0.68596
......
```

Figure 12: testseq1mcmc.txt (2)

correlated way or uncorrelated way (see the Eq.11 of Seo et al. 2004). Under the null hypothesis in which two rate changes are uncorrelated, the distribution of S is symmetrically distributed around 0.5 (see 'Null S' in Figure 12). In our analysis, the test statistic (S) is 0.4703 and is less than 0.5, which means a sign of discordance. But, the discordance is not significant because Lower P-value and Upper P-value are not small. Lower P-value and Upper P-value are Pr(Null_S <S) and Pr(Null_S >S), which can be used in testing significance of S statistic.

(3) Posterior distributions of times (Figure 13)

In the first column, you see the node indices. Posterior median and standard deviation, 95% lower credibility interval (LowerCI), 95% upper credibility interval (UpperCI) and posterior mean come after each node index. In our example, node 0 - node 5 are terminal nodes whose times are definitely known and it is inappropriate to consider the posterior distributions of times for these nodes. The entries for these nodes are filled arbitrarily with 0.0.

(4) Posterior distributions of rates (Figure 14)

After the summaries of divergence times, the informations for posterior distributions of synonymous rates, nonsynonymous rates and ω's are listed. In Figure 14, I show only the summaries of synonymous rates.

```
 ......
Posterior Distribution of Time
 Node#   PostMedian         Std      LowerCI      UpperCI     PostMean
     0   0.00000000  0.00000000   0.00000000   0.00000000   0.00000000
     1   0.00000000  0.00000000   0.00000000   0.00000000   0.00000000
     2   0.00000000  0.00000000   0.00000000   0.00000000   0.00000000
     3   0.00000000  0.00000000   0.00000000   0.00000000   0.00000000
     4   0.00000000  0.00000000   0.00000000   0.00000000   0.00000000
     5   0.00000000  0.00000000   0.00000000   0.00000000   0.00000000
     6  32.80146709  3.93296767  25.84578676  41.17976115  33.03257530
     7   8.40265521  1.04933583   6.55107905  10.75221856   8.46155385
     8  19.56812157  0.56458946  17.96422180  19.97631811  19.39794191
     9  38.09491346  3.02490672  32.90050645  44.75360551  38.27941899
    10  70.01737988  4.45501252  61.41336963  79.21430414  70.18967966
......
```

Figure 13: testseq1mcmc.txt (3)

The output format of nonsynonymous rates and ω's are same. The interpretation of each column is same as that of Figure 13.

(5) Posterior distributions of branch lengths (Figure 15)

With the posterior distributions of times and synonymous and nonsynonymous rates, it is possible to get the posterior distributions of synonymous and nonsynonymous branch lengths and their sums. In Figure 15, I show the summaries of posterior distributions of synonymous branch lengths. The format of nonsynonymous and total branch lengths are same. The interpretation is same as that of Figure 13. You can compare the posterior median or mean of branch lengths with the MLE's shown in Figure 7.

(6) Concordance probability in each branch (Figure 16)

Figure 16 shows the posterior probability of concordance in each branch. The overall posterior probability of concordance (S) over all lineages is shown in Figure 12. For more detail see pp. 1204-1205 of Seo et al. (2004).

(7) Phylogenies using posterior medians of branch lengths (Figure 17)

Using the medians of posterior distributions of synonymous and nonsynonymous branch lengths and their sums, you can reconstruct phylogenetic trees. You can compare these trees with the trees obtained by the maximum likelihood method (Figure 5).

ACCESSORY PROGRAMS

```
......
Posterior Distribution of SynRate
 Node#  PostMedian         Std     LowerCI     UpperCI    PostMean
     0  0.01939554  0.00605413  0.00888019  0.03262758  0.01968643
     1  0.00908357  0.00359794  0.00332419  0.01740333  0.00942820
     2  0.01378686  0.00540698  0.00539586  0.02630600  0.01436944
     3  0.02901143  0.00904943  0.01345732  0.04819077  0.02964508
     4  0.03796983  0.01091897  0.01908860  0.06083694  0.03863984
     5  0.04864552  0.01438592  0.02481506  0.08379906  0.04975854
     6  0.01117686  0.00232493  0.00710531  0.01592465  0.01127750
     7  0.04067557  0.00691798  0.02878923  0.05639852  0.04094444
     8  0.02815093  0.00548217  0.01966965  0.04121968  0.02881824
     9  0.01400474  0.00308646  0.00842458  0.02079673  0.01411743
    10  0.00859803  0.00266094  0.00400228  0.01435577  0.00878680
......
```

Figure 14: testseq1mcmc.txt (4)

```
......
Posterior Distribution of Synonymous Branch Length
 Br#  PostMedian         Std     LowerCI     UpperCI    PostMean
   0  0.50326498  0.07615143  0.34951384  0.65839522  0.50420112
   1  0.33669864  0.05465397  0.23968057  0.45116819  0.33874713
   2  0.53964027  0.09297034  0.37493839  0.73642544  0.54219961
   3  0.56551794  0.08711695  0.39176976  0.73981102  0.56659981
   4  0.33153555  0.04532937  0.24029228  0.42009133  0.33187331
   5  0.37582254  0.05603974  0.27688539  0.49460132  0.37829253
   6  0.37108250  0.06110246  0.26115142  0.49152872  0.36999259
   7  0.38185341  0.06821082  0.25751939  0.52909068  0.38287965
   8  0.39875539  0.06957091  0.27627631  0.55087020  0.40220785
   9  0.36037149  0.06639836  0.23460976  0.48920372  0.36250567
....
```

Figure 15: testseq1mcmc.txt (5)

```
......
Concordance Probabilities
 Br#       Prob.
   0  0.18100000
   1  0.38200000
   2  0.46900000
   3  0.50300000
   4  0.46100000
   5  0.46100000
   6  0.75000000
   7  0.72600000
   8  0.57000000
   9  0.20000000
....
```

Figure 16: testseq1mcmc.txt (6)

```
......
/// Tree of posterior syn brL (median)
((TaxonA:0.503265,TaxonB:0.336699):0.371082,(TaxonC:0.539640,(TaxonD:···

/// Tree of posterior nonsyn brL (median)
((TaxonA:0.054966,TaxonB:0.100141):0.055423,(TaxonC:0.055540,(TaxonD:···

/// Tree of posterior total brL (median)
((TaxonA:0.558797,TaxonB:0.436880):0.427047,(TaxonC:0.594838,(TaxonD:···
......
```

Figure 17: testseq1mcmc.txt (7)

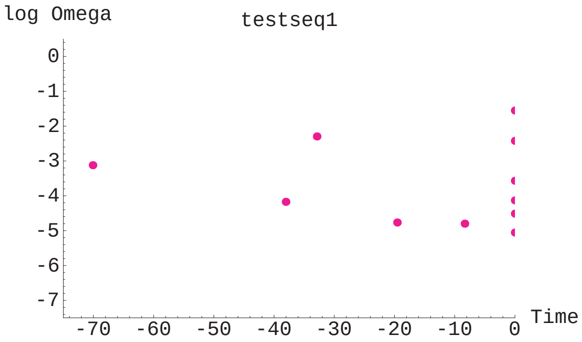

Figure 18: Omega plot generated by Mathematica4.0

There are two accessory programs, *GetRt1.0* and *CalCon1.0*. These two programs should be run in the same directory where you finish MCMC run (case of 3 or 4 in <Job> option). The former uses the files specified in <mcmc output file> (*testseq1mcmc.txt* and *testseq2mcmc.txt* in our analysis) and the latter uses the files specified in <mcmc output file> + "stored.txt" (*testseq1mcmc.txtstored.txt* and *testseq2mcmc.txtstored.txt* in our analysis). Do not make any change in these files after you finish MCMC run.

(1) GetRT1.0

This program generates graphic files of the posterior medians of times, ω's, synonymous and nonsynonymous rates . The graphic files are written in postscript format. If you know the postscript language, you can edit the files and change the colors, fonts, thickness of lines, position of legend, and etc. Redirect the message from the screen to the file as follows.

```
C:\example>getrt > ratentime.txt   +   Enter
```

If you know how to use Mathematica (Wolfram Research Inc.), you can use *ratentime.txt* file to generate more figures (e.g. Figure 18). Figure 19 and 20 show the files generated by *GetRT1.0*

(2) CalCon1.0

The concordance between synonymous and nonsynonymous rates within gene (Figure 12) can be extended to the concordance between synonymous or nonsynonymous rates or ω's of different genes. This program is used to calculate the concordance statistic (S) between the rates and ω's of different genes and p-values of the statistics. Redirect the message from the screen to the file as follows.

```
C:\example>calcon > concordance.txt   +   Enter
```

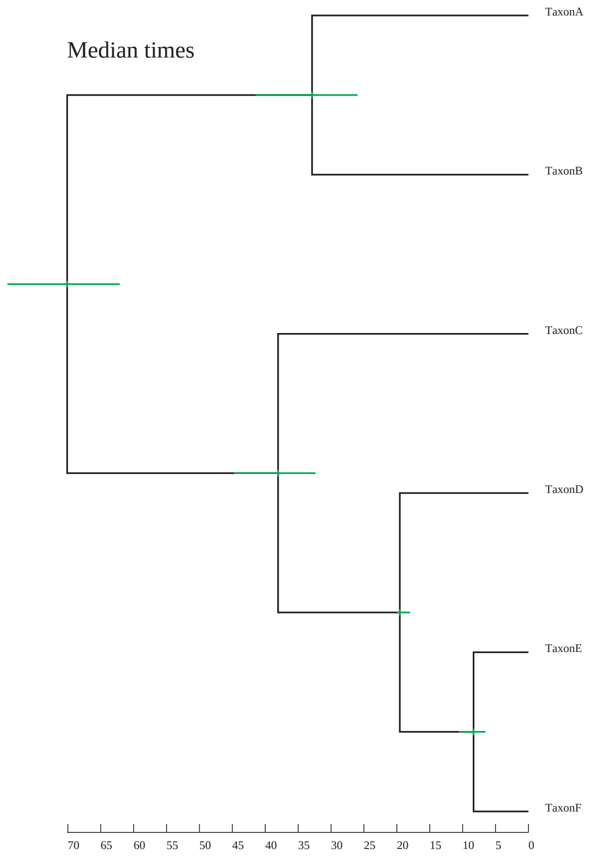

Figure 19: Posterior median of times and 95% C.I.

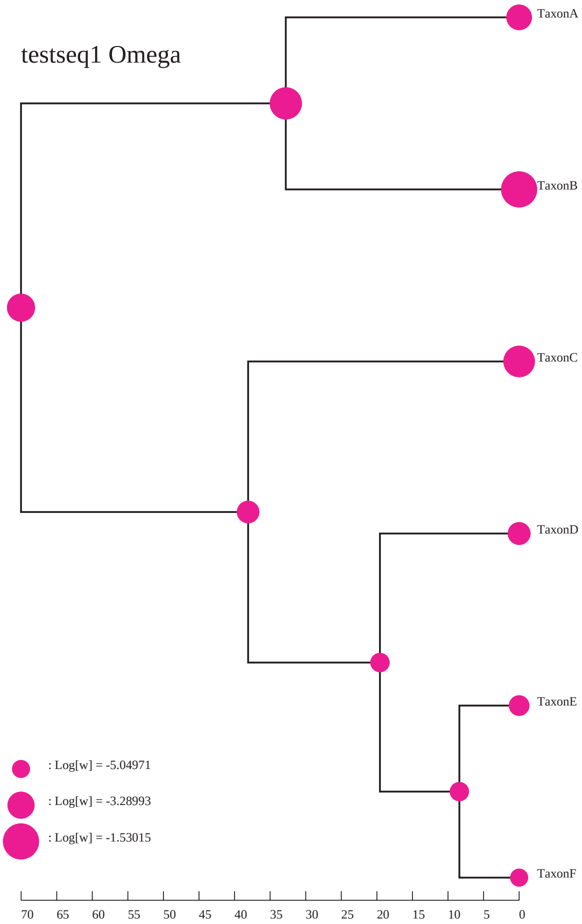

Figure 20: Posterior median of ω value

```
Version:  CalCon1.0
testseq1 -- testseq2
  - concordance of omega
     Null S = 0.499726 +- 0.0737806(0.359, 0.641)
     S = 0.4558, LowerPValue = 0.28714, UpperPValue = 0.7117
  - concordance of synrate
     Null S = 0.499926 +- 0.0526148(0.4067, 0.5933)
     S = 0.5897, LowerPValue = 0.96376, UpperPValue = 0.03538
  - concordance of nonsynrate
     Null S = 0.499801 +- 0.0662898(0.3746, 0.6232)
     S = 0.537, LowerPValue = 0.67344, UpperPValue = 0.32448
......
```

Figure 21: concordance.txt (1)

```
....
testseq1
  - concordance of syn and nonsyn
     Null S = 0.499684 +- 0.0573253(0.3847, 0.6115)
     S = 0.4703, LowerPValue = 0.30926, UpperPValue = 0.68438
testseq2
  - concordance of syn and nonsyn
     Null S = 0.500084 +- 0.0531438(0.405, 0.595)
     S = 0.508, LowerPValue = 0.54642, UpperPValue = 0.4515
......
```

Figure 22: concordance.txt (2)

In *concordance.txt*, you can find the S statistic and its significance level (Figure 21). In the beginning, you can see the concordance of ω, synonymous rate (synrate) and nonsynonymous rate (nonsynrate) of two genes (testseq1 and testseq2 in our example), and their lower and upper p-values. The interpretation is same as that of Figure 12.

Next, you see the concordance of synrate and nonsynrate within gene (Figure 22). This information is already shown in the files specified in <mcmc output file> (Figure 12). Because the values are obtained with simulation, you might see the slight difference between them.

Finally, you can see the tables of concordances and their upper and lower p-values for ω and synonymous and nonsynonymous rates (Figure 23). These tables are just duplication of the information shown in Figure 21. In Figure 23, I show the tables for ω as an example.

```
....
S_Omega
 **********    testseq1    testseq2
   testseq1  ----------  0.45580000
   testseq2  0.45580000  ----------
LowerPValue of S_Omega
 **********    testseq1    testseq2
   testseq1  ----------  0.28714000
   testseq2  0.28714000  ----------
UpperPValue of S_Omega
 **********    testseq1    testseq2
   testseq1  ----------  0.71170000
   testseq2  0.71170000  ----------
UpperPValue + LowerPValue
 **********    testseq1    testseq2
   testseq1  ----------  0.99884000
   testseq2  0.99884000  ----------
......
```

Figure 23: concordance.txt (3)

ACKNOWLEDGEMENT

I thank Jeffrey L. Thorne and Hirohisa Kishino for helpful discussion for the implementation of the program, Ziheng Yang for allowing me to use codes of PAML, and members in Thorne's group (Jiaye Yu, Stephane Aris-Brosou and Sang-Chul Choi) for giving me suggestion for compiling, debuging and testing the program.

LITERATURE CITED

Goldman, N., and Yang, Z. 1994. A codon-based model of nucleotide substitution for protein–coding DNA sequences. Mol. Biol. Evol. **17**:32–43.

Kishino, H., J.L. Thorne, and W.J. Bruno 2001. Performance of a divergence time estimation method under a probabilistic model of rate evolution. Mol. Biol. Evol. **18**:352–361.

Kosakovsky Pond SL, MuseSV(2000) HYPHY: hypothesis testing using phylogenies.Version 0.99b. http://www.hyphy.org

Seo T.-K., J.L. Thorne and H. Kishino 2004. Estimating absolute rates of synonymous and nonsynonymous nucleotide substitution in order to characterize natural selection and date species divergences. Mol. Biol. Evol. 21(7):1201–1213.

Thorne, J.L., H. Kishino, and I.S. Painter 1998. Estimating the rate of evolution of the rate of molecular evolution. Mol. Biol. Evol. **15**: 1647–1657

Thorne, J.L., and H. Kishino 2002. Divergence time and evolutionary rate estimation with multilocus data. Syst. Biol. **51**: 689–702.

Yang, Z. 1997. PAML: a program package for phylogenetic analysis by maximum likelihood CABIOS 13:555-556

Yang, Z. 1998. Likelihood ratio tests for detecting positive selection and application to primate lysozyme evolution. Mol. Biol. Evol. **15**:568-573.

—– temp

The function of this program is separated into two parts. First, the program obtains the maximum likelihood estimates (MLE's) of synonymous and nonsynonymous amounts of substitution and it approximates the uncertainty of these estimates. Second, the program infers divergence times and evolutionary rates by combining the information of maximum likelihood estimates with the information of time constraints such as fossil data. For more details, please see Seo et al.(2004).