PAML FAQ

Ziheng Yang

Last updated: 5 January 2005 (not all items are up to date)

Table of Contents

PAML FAQ 1
Table of Contents 1
Data Files 3
Why don't paml programs read my files correctly? 3
Does paml read phylip format? 3
Does paml read nexus (PAUP) format? 3
"This is a rooted tree, without clock. Check." What does the message mean? 4
How do I prepare an unrooted tree? 4
How do I analyze multiple data sets? 4
Should I remove alignment gaps and ambiguity characters in my analysis? 4
Windows, UNIX, and MAC OS X basics 4
Common mistakes and pitfalls 5
Alignment of protein-coding DNA sequences (for codeml) 5
Programs pausing under MS Windows 6
Window auto-close under MS Windows 6
MAC memory allocation problem 6
How can I put up with so many versions of PAML? 6
Windows Essentials 7
How to turn on file extensions? 7
How to use Task Manager? 7
BASEML 7
How do I interpret the parameters under the REV (GTR) and UNREST models? 7
How do I interpret the rates and branch lengths under models for multiple genes (Mgene)? 8
What does the mutation rate from the TipDate analysis mean? 8
How does baseml/codeml deal with ambiguity characters and alignment gaps? 8
CODEML 9
Some general notes about use of codon models to detect positive selection 9
How can I get codeml to accept more that one data set at a time? 10
What are the limits in terms of number of sequences in the alignment? 10
Are my sequences too divergent for a sensible ML analysis? 10
How do I test positive selection along specific lineages? 11

Can I conclude positive selection if I found significantly higher dN/dS ratios for my branches of interest than for other branches? 11
What is the effect of assuming the same selective pressure among lineages? 11
Can I obtain a list of probabilities for all sites and all categories under an NSsites model? 11
How can I run multiple NSsites models in one analysis? 11
What does the "mean w" in the rst file mean when I run an NSsites model? 11
The different NSsites models identified different sites. Which model should I believe? 12
How can I make NSsites models 7 and 8 converge? 12
Can I specify certain branches in a tree file to have independent ω parameters? 12
Can I specify certain sites in an alignment to have independent ω parameters? 12
Can I estimate dN and dS between all pairs of taxa in an alignment? 13
Why did I get duplicated ω estimates when I used many categories under M3? 13
Why do the inferred site classes not match the estimated proportions under the NSsites models? 13
My data set is too large. Can I still analyze them? 14
How do I generate approximate branch lengths for codon-based analysis? 14
How do I run the branch-site models (A & B in Yang&Nielsen 2002 MBE)? 14
I got strange estimates of ω (dN/dS) like –1, 89. What do they mean? 15
How can I estimate an amino acid substitution matrix from my own data, like mtmam.dat and wag.dat? 15
YN00 16
Why did I get this message: "1967 nucleotides, not a multiple of 3!?" 16
EVOLVER 16
Where is the control file for evolver? 16
Can I run evolver with command-line arguments? 16
Can I get the ancestral sequences generated during the simulation in evolver? 16
How do I simulate data under codon models of variable dN/dS (omega) ratios among sites as did Anisimova et al. (2001 MBE)? 17
How do I simulate data sets to be analyzed by paup? 17
What is the difference between evolver and seq-gen? 17

Data Files

Why don't paml programs read my files correctly?

Data files used by programs in paml are all plain text files. These include the control files (such as baseml.ctl), sequence data files, and tree files used by baseml, codeml, and yn00. Those programs typically read the control file first, the sequence data file, and then go through and evaluate the trees in the tree file. You can run the programs on a few example data sets included in the package and get familiar with the output. Then you should be able to tell whether each of these steps is correctly executed by watching the screen output. If there is a problem, you should at least be able to tell whether it occurred before or after the data files are correctly read.

You should open and check your control file, sequence file, and tree file and make sure they are plain text files. A common problem is caused by incorrectly inserted line breaks or missing line breaks. For example, the control file is processed on a line-by-line basis. Sometimes a line break is inserted in the middle of a long line by a text editor or email program, and then the file does not work anymore. Some text editors such as MS word might save a text file without proper line breaks, or more precisely, the file seems to have cartridge returns (CR) but no line feeds (LF). When you display the file contents on the screen using the command type or more, only the last line is shown. Such files are not readable by paml programs. The problem can also occur during file transfers between DOS/Windows and UNIX. You can use different text editors and try to save the file again to see whether it works. In MS Word, you can save the file (File-Save As) by choosing "Text Only with Line Breaks (*.txt)". Also the more command in my versions of Windows XP does not seem to work properly and often does not display the content of a proper text file properly.

See also the next two questions about phylip and paup formats.

Does paml read phylip format?

Yes, paml programs read both the sequential and interlaved formats used in phylip. However there are the following complications. First, paml requires that you have the option variable I on the first line if the interlaved format is used. Second, phylip assumes 10 characters in a sequence name, while paml assumes 30 characters (I think). My suggestion is that you make sure to separate the sequence name from the sequence by at least two spaces, in which case both paml and phylip programs can read the same data file with no problem. For example the following works in both paml and phylip.

```
4 20
sequence_1    TCATT CTATC TATCG TGATG
sequence_2    TCATT CTATC TATCG TGATG
sequence_3    TCATT CTATC TATCG TGATG
sequence_4    TCATT CTATC TATCG TGATG
```

The following works with phylip but not with paml since there is no space between the sequence name and the sequence.

```
4 20
sequence_1TCATTCTATCTATCGTGATG
sequence_2TCATTCTATCTATCGTGATG
sequence_3TCATTCTATCTATCGTGATG
sequence_4TCATTCTATCTATCGTGATG
```

Does paml read nexus (PAUP) format?

PAML reads nexus format if there are no comments in the sequences. PAML reads the sequence data only and ignores everything else.

"This is a rooted tree, without clock. Check." What does the message mean?

How do I prepare an unrooted tree?

In PAML, an unrooted tree should have at least a trifurcation at the root. So for example, ((1,2),3) is considered a rooted tree, while the unrooted version is (1,2,3). Similarly (((1,2),3),4) is a rooted tree since the root has two descendent nodes. The unrooted version is ((1,2),3,4), or (1,2,(3,4)), both of which are equivalent.

If the model you use in baseml or codeml does not distinguish between rooted trees, but you use a rooted tree, the programs will issue a warning message "This is a rooted tree. Please check!" in the output. For most models, the likelihood values are still correct even if you use a rooted tree, but the lengths of the two branches around the root are not stable, as only their sum is estimable. For other models, neither the likelihood nor the parameter estimates are correct. So really you should heed the message and use an unrooted tree in the analysis.

If you use PAUP to produce a tree, note that PAML ignores PAUP specifications like "[U]" and reads only the parenthesis notation. So you have to edit the paup tree to remove a pair of parenthesis so that the outmost pair of parentheses group together three clades (that is, there is a trifurcation at the root).

How do I analyze multiple data sets?

Programs baseml, codeml, and yn00 have an option variable (ndata) for analyzing multiple data sets in one go. You specify ndata = 100 if there are 100 data sets in the sequence file. You can generate multiple data sets in a file using the simulation program evolver.

Should I remove alignment gaps and ambiguity characters in my analysis?

Ambiguity characters (such as Y for pyrimidines T or C, ? for any nucleotide or amino acid) can be accommodated by the likelihood programs (baseml and codeml). The idea used is due to Joe Felsenstein and has been in use for some years. It was described in Yang (2000 J. Mol. Evol. 51: page 424 bottom). Using ambiguity characters will increase the computation (both memory and running time) compared with removing them.

Alignment gaps are more difficult. PAML does not have any methods of dealing with them properly. The two options are (1) to remove them, which you could do by manually removing them or by choosing cleandata = 1; and (2) to treat them as ambiguity (undetermined) nucleotides or amino acids. The latter is the default behavior if gaps are in the data. Neither is ideal. One obvious effect is that both strategies under-estimate sequence divergences, while the effects on other analyses might not be so clear. Personally I think sites at which most sequences have data except for one or two sequences should perhaps be kept while sites at which all sequences except one or two have alignment gaps had better be removed.

Windows, UNIX, and MAC OS X basics

Paml programs do not have a good interface, so knowing how to double click or drag a file icon is not enough. It would be much easier if you know some basic operations supplied by your operating system, being it Windows, UNIX, or OS X. You should know how to create a folder (directory), change to a different folder, copy files etc. Some basic commands for Windows and UNIX are listed below. It is probably worthwhile if you get a small UNIX basics book; it won't take you more than an hour or two to get familiar with the simple UNIX commands, and the benefit is that you will be able to use UNIX, MAC OS X, or Windows.

Suppose that the programs are extracted in a folder (directory) named paml. (It is normally named something like paml3.13, so you should replace the folder name paml here with whatever name you are using.) After you managed to compile the programs, the executables are in the folder paml/src. You can use ls (dir on Windows) to list the files. Since the control and data files

are in the folder paml and not in paml/src, I suggest that you move (mv on UNIX and move on Windows) all the executables one level up into the paml folder.

```
mv baseml basemlg codeml evolver ..

mv pamp chi2 ..
```

The ".." at the end of the command means one level up on the folder tree, which from the folder paml/src means paml/. Then you change directory to paml as well.

```
cd ..
```

Again note the "..". Now you should be in the paml folder. Use ls to list the files again. You can run a program by typing the program name.

```
codeml
```

If the current folder is not on your search path, you will have to use ./codeml to tell the OS that the command file is in the current folder.

In general, you can run the program from any folder by specifying the path of the program. For example, on UNIX (linux or MAC OS X), this might be one of the following ("~/" means the root folder of your account)

```
~/paml/codeml

~/paml3.13/src/codeml
```

../../paml3.13/codeml etc.

Note that programs like codeml require a control file and assumes it is in the same folder as the program.

Windows	UNIX	Function
cd, chdir	cd, chdir, pwd	Sets and displays current directory (folder)
copy	cp	Copies files
del	rm	Deletes files
dir	ls	Lists files
exit	exit	Exits from the command processor
find	fgrep	Searches for a string in files
help	man	Gets help
md	mkdir	Makes a new directory
more	more, less	Displays file contents by screenfuls
path	set PATH	Sets search path for commands
print	lpr	Prints files
rd, rmdir	rmdir, rm -r	Removes directories
ren	mv	Renames a file
time, date	date	Displays or sets time and date
type	cat	Displays the contents of a file
xcopy	cp	Copies files and subdirectories
ftp	ftp	Starts an ftp session
telnet	telnet	Starts a telnet session
	Ctrl-Z, followed by bg	Puts a foreground job into the background
	fg	Brings a job to the foreground
	nice, renice	Be nice to others by running your jobs at a lower priority

Common mistakes and pitfalls

Alignment of protein-coding DNA sequences (for codeml)

Some programs can be used to construct an alignment of protein-coding DNA sequences by using the alignment of the translated protein sequences. However, it has been noticed that some versions of some of those programs (for example BioEdit) have bugs. They corrupt the DNA sequences and use one single codon for each amino acid irrespective of the codon in the original DNA sequence during the "back translation". As a result, only 20 codons are used even in a large data set which has all the 61 sense codons present. If you use CodonFreq = 3, the observed codon frequencies are used to specify codon substitution rates, and with most of the codons missing, the Markov chain is not connected anymore. Thus codeml will have problems. The problem is easily spotted by looking at the codon usage tables in the codeml output. Advice: check your alignment carefully when you use those programs to automate the process.

Programs pausing under MS Windows

If you click on the Windows command prompt (the dos box) more than once when a program such as baseml or codeml is running, you might select some text in the window by accident. If that happens, the program will stop running until you hit the "Enter" key to copy your selection onto the clipboard. I have found this irritating but it is presumably a major feature of Microsoft Windows. The behavior differs depending on, for example, whether you have turned on "smart copy". When the program is not running, it won't be using any CPU time. This can watch from task bar (Click on the Start bar and choose Task Manager).

Window auto-close under MS Windows

Run the Windows version from a DOS/Windows command box by typing the program names such as `baseml`. Do not run the programs by double clicking on the file names from Windows 95/98/2000/NT Explorer. Otherwise, the window will close automatically when the programs finish or abort and you won't have the chance to see any error messages.

MAC memory allocation problem

When your data set is large, you may see a message like "oom ", which stands for "out of memory". If you think your data set should be manageable by the program/computer, you can change the memory that is allowed by the operating system for the program to use. If you select the file name and choose "File-Get information", you should see a pop up window. You can increase numbers in this window.

I understand that this is not a problem anymore under MAC OS X.

How can I put up with so many versions of PAML?

I have seen some people having many different versions of paml on their computer. Besides being embarrassing to me, they can also be confusing. Here is some advice, which may and may not be useful. You could create a directory called bin/ under your home directory. My home directory is D:\ and I am going to assume that below. So the full path name of the bin folder is D:\bin. In your case, it might be C:\Documents and Settings\GoodUser\bin. Rename the executables baseml.exe and codeml.exe to include the version numbers and then move them into the bin/ folder. For example, baseml3.13.exe is baseml.exe from paml v3.13. Add the bin folder D:\bin onto your search path for commands. One way of doing this is Start – Control Panel – System: Advanced - Environment Variables. Edit and add the full name of the bin folder to variable value for variable name path. Note that the different directory names are separated by ';'.

On UNIX/Linux/MAC OSX, you could similarly create a bin folder under your home folder. Some systems automatically do it for you. Again you can move the executables into that folder and make sure that the folder is on your search path. You can modify the initialization files to change the path. For example, if you use the c shell, you can modify .cshrc.

After this is done, you can use the command baseml3.13 from any directory to run the baseml program from paml v.3.13. I should point out that the versions may require slightly different file formats or older versions might not recognize new control variables.

Tip 1. You can move other command-line programs such as MrBayes (mb) into the bin folder as well, and then you can run it from any other folder by simply typing mb.

Tip 2. I do not like the long folder names given by Windows. After a fresh install of XP, "My Documents" is in "C:\Documents and Settings\Ziheng\". It is long to type and the space in the name can cause problems. The first thing I do is to right click on My Documents - Properties – Move Target to reassign a folder (a separate data drive D:\) as my default folder.

Windows Essentials

I use Windows XP as an example in the following. You should be able to find similar options in Windows 2000.

How to turn on file extensions?

The default windows installation hides file extensions for known file types in Windows Explorer. This is confusing, as being unable to see the full file name, you might be editing the wrong file. To show the file extensions in Windows Explorer, choose

Tools – Folder Options, click on View, and clear the box for "Hide extensions for known file types". Then OK.

How to use Task Manager?

MS Windows seems to be very slow and unstable when you run a computation-intensive job at the background. However, the system seems fine if you run such jobs at low priority. You change the job priority using Task Manager. To start Task Manager, right click on the task bar and choose Task Manger. Another way is to press the three keys simultaneously: Ctrl – Alt – Del. Here are a few things I do when I start Task Manager.

View – Update Speed – Low.

View – Select Columns: I add CPU Percentage and Base Priority.

Options – Hide when minimized.

I will then never close Task Manager but rather minimize it. There will then be an icon on the task bar with bright green indicating that the CPU is busy and dark green indicating no CPU usage.

To change the priority of a job, click the Processes tab. Right click on the job (process), choose Set Priority – Low. If you set the command prompt (cmd) to Low priority, then jobs started from that command prompt window will all have the Low priority.

BASEML

How do I interpret the parameters under the REV (GTR) and UNREST models?

The five parameters under REV are a, b, c, d, e, as defined in my 1994 (JME 39:105-111) paper. The 11 values under the UNREST model are the eleven substitution rates when you read the rate matrix (Q) row by row, disregarding the diagonals. The nucleotides are ordered T, C, A, and G. So the eleven values are q_{TC}, q_{TA}, q_{TG}, q_{CT}, q_{CA}, q_{CG}, q_{AT}, q_{AC}, q_{AG}, q_{GT}, q_{GC}, with q_{GA} = 1 fixed. Again it is the notation used in that paper. Note that PAUP uses a different ordering of the nucleotides, and the matrix output does not match the baseml output. Note also that the equilibrium base

frequencies under UNREST are calculated from the substitution rate matrix, and so for a fair comparison between REV and UNREST, you should use nhomo = 1 for REV (GTR), so that the base frequencies are estimated by maximum likelihood rather than by using the observed frequencies (nhomo = 0). Again see my JME paper for the argument.

How do I interpret the rates and branch lengths under models for multiple genes (Mgene)?

The detailed descriptions of the Mgene models are in Yang (1996 J. Mol. Evol. 42:587-596). The models assume that branch lengths are proportional among genes (or site partitions). So suppose the branch lengths in gene 1 are b1, b2, b3, ... Then the branch lengths for gene 2 are b1 * r2, b2 * r2, b3 * r2, ..., and the branch lengths for gene 3 are b1 * r3, b2 * r3, b3 * r3, and so on. The rate for the first gene r1 = 1 is fixed, so that r2, r3, etc. are relative rates. The output branch lengths in the programs are for the first gene, and so is the tree length (defined as the sum of the branch lengths along the tree, measured by the expected number of substitutions per site along the tree).

The model requires far fewer branch length parameters than a model that uses separate branch lengths for each gene (the Mgene = 1 option).

What does the mutation rate from the TipDate analysis mean?

The example files are named exampleTipDate.* in the folder examples/clock. The .rst file has results for 2 sets of TipDate runs (D and G). You can check Andrew Rambaut's Bioinformatics paper about his TipDate program, but I think the numbers at the end of the sequence names in the sequence data file are years, so the first sequence Brazi82 was determined in 1982.

baseml and codeml are the same in the analysis. The only difference is that in baseml the branch lengths are measured by the expected number of nucleotide substitutions per nucleotide site, which in codeml, they are defined as the expected number of nucleotide substitutions per codon. Since one codon approximately has 3 nucleotide sites, the branch lengths in codeml are about 3 times as large as those from baseml. The mutation rate is calculated by dividing the branch length (distance) by time, and so the same relationship holds. That is, in the exampleTipDate.rst file, the mutation rate from baseml is about 0.000774 nucleotide substitutions per site per year, and the mutation rate from codeml is about 0.00249 nucleotide substitutions per codon per year. So those two estimates are pretty much the same.

(1) baseml output:
Mutation rate = 7.74e-004 +- 1.00e-004
Node 1 Time 82.00 Node 18 Time 24.09 +- 6.66
Node 2 Time 83.00 Node 19 Time 31.65 +- 6.10

(2) codeml output
Mutation rate = 2.49e-003 +- 3.44e-004
Node 1 Time 82.00 Node 18 Time 22.88 +- 7.32
Node 2 Time 83.00 Node 19 Time 32.00 +- 6.39

How does baseml/codeml deal with ambiguity characters and alignment gaps?

baseml/codeml treats alignment gaps as ambiguity characters (undetermined nucleotides or amino acids) and not as insertion/deletion events, which is unsatisfactory. So the following is about ambiguity characters. Ambiguity characters are dealt with in the likelihood calculation according to an idea of J. Felsenstein's, mentioned in Yang (2000 JME 51: 423-432).

baseml and codeml for amino acids: If the model involves unequal nucleotide or amino acid frequencies, and those frequencies are calculated from the sequence data and not estimated by ML iteration, ambiguities are resolved through an iteration. So nucleotide Y in the data is counted as x% T and (100-x)% C, with x given by p_T/(p_T + p_C). Since the frequencies p_T, p_C, p_A, and p_G are unknown and being calculated, an iteration is used until the frequencies stabilize.

This of course does not apply to baseml (nhomo = 1), which estimates base frequencies by iteration.

codon models: For codon frequency modes F1x4, F3x4, and F61, the different codons have different frequencies, which are estimated empirically from the sequence data. Again codeml goes through an iteration to resolve ambiguities. So codon TTY is considered x% TTT and (100-x)% TTC, with x given by the relative frequencies of TTT and TTC. As codon frequencies are unknown, an iteration is used. Similarly, codon T-- (treated as TNN or T??) is resolved into 16 compatible codons, in proportion to their frequencies. Note that under F1x4 and F3x4, the codon frequencies are calculated using base frequencies, and so the iteration really updates the three base frequencies under F1x4 or 9 base frequencies for codon positions under F3x4.

If you choose verbose = 2 in codeml.ctl, the program will print out the codon frequencies in the format required by evolver (with frequencies for 64 codons, including 0's for stop codons), which you can inspect.

CODEML

Some general notes about use of codon models to detect positive selection

Many messages I have got ask general questions about the suitability of codon models for specific data sets. For example, are the codon models useful for my data, or I have run such and such analyses, but are the results reliable? So here are some general notes, mainly based on the following review papers and simulation studies:

Anisimova et al. 2001 The accuracy and power of likelihood ratio tests to detect positive selection at amino acid sites. Mol. Biol. Evol. **18:** 1585-1592.

Anisimova, M., J. P. Bielawski and Z. Yang, 2002 Accuracy and power of Bayes prediction of amino acid sites under positive selection. Mol. Biol. Evol. **19:** 950-958;

Yang, Z., 2001 Adaptive molecular evolution, pp. 327-350 in *Handbook of statistical genetics*, edited by D. Balding, M. Bishop and C. Cannings. Wiley, New York.

Yang, Z., 2002 Inference of selection from multiple species alignments. Curr. Opinion Genet. Devel. **12:** 688-694.

Yang, Z., and J. P. Bielawski, 2000 Statistical methods for detecting molecular adaptation. Trends Ecol. Evol. **15:** 496-503.

(a) Assessment of information content in your data. The analysis uses the number of silent and replacement changes to "decide" whether there is an excess of replacement changes relative to silent changes. So you need many changes to make an inference. For example, in the site-based analysis (the NSsites models), you need many sequences to accumulate changes at each site. With a typical human nuclear locus, even as many as 500 sequences may not contain enough variation for the codon based methods to have any power. Similarly, if you see a very big ω (dN/dS) for a branch (in a branch-based analysis using the variable model in codeml.ctl), but the branch length is very small, the estimate is not reliable. It is obvious that neither highly similar sequences nor very divergent sequences are informative. It is hard to specify exact values. You can look at the simulation papers to get a rough idea. Here are a few questions Chung-I Wu asked about applying codon models to identify amino acid sites under selection (the NSsites models). Treat the answers with suspicion.

1) How many species are needed?
   I suppose the absolute minimum is 4 or 5 if the sequence divergence is optimal. 10 would be good, while 20 would be much better. This will depend on how divergent the sequences are.

2) How far should the total distance among these species be? For example, dS should be > 0.5 in total?
   The optimum sequence divergence depends on the number of sequences, and a big tree

with many sequences can tolerate more changes. I think the method will be reasonable if dS summed over all branches on the tree is > 0.5.

3) How much selection can be detected given the parameters of 1) and 2)?
The method seems able to identify one or two sites under strong selection. When more sites are under selection but the pressure is weak, the LRT might say selected sites exist but the method tends to have trouble identifying them.

(b) Robustness analysis: One advice is that you vary the model somewhat to see whether your main results are sensitive to the detailed assumptions made in the model. For example, you can look at the effect of codon usage by changing CodonFreq. If you are running the NSsites models, you should try a few of them to see whether the results are consistent. Also look at the general consistency of results between different models. For example, the simplest model (called M0 in some papers, specified by model = 0 NSsites = 0) should be the first you should apply, and estimates of branch lengths, κ, and ω under the model should be consistent with estimates under more sophisticated models. I must say that too many times, the user of the program seemed to have no idea about very basic statistics of their data, such as the divergence levels, the base composition and codon usage bias, even though such information is always in the output.

(c) Computational issues: Some models are known to have convergence problems. For example, they may get got stuck at a corner of the parameter space, with some parameter estimates to be at 0, the lower bound set by the program. The problem can also be exacerbated by lack of information in the data, when the sequences are highly similar or highly divergent. You should run the program at least twice. Unfortunately the parameter-rich models tend to be more problematic computationally and they take a long time even if there is no difficulty in the iteration algorithm. Examples include the branch-based free-ratios model (model = 1 NSsites = 0), and some of the site-based models (model = 0, NSsites = 7 or 8). The branch-site models have difficulties as well (model = 2 NSsites = 2 or 3).

I typically run M0 (model = 0 NSsites = 0) first to get the branch lengths, and then copy the tree with branch lengths into the tree file and then use them as initial values when I run other difficult models.

Also creating a file of initial values (named in.codeml) might help.

How can I get codeml to accept more that one data set at a time?

You use the control variable ndata.

By default, the line is commented out with the asterisk * at the beginning of the line. Make sure you remove it.

What are the limits in terms of number of sequences in the alignment?

The simple answer is a few hundred sequences for codon-based analysis (codonml), one or two thousand sequences for nucleotide or amino acid based analysis (baseml and aaml), if you have a good PC or workstation (as of June 2001). The computation increases more quickly with the number of sequences than with the number of sites in the sequence. How large data sets the programs can handle will depend on many factors including the complexity of the model, how fast and how much memory you have on your machine. (Yang. 2000. J. Mol. Evol. 51:423-432)

Are my sequences too divergent for a sensible ML analysis?

In computer simulations, saturation of substitutions does not seem to be a big problem, as the performance of the models does not seem to deteriorate until the sequences become very divergent, say with 10 or 50 nucleotide substitutions per nucleotide site along the tree (Yang 1998. Syst. Biol. 47:125-133; Anisimova et al. 2001. Mol. Biol. Evol. in press). Also a large tree with many branches will be able to tolerate more changes than a small tree. Similarly maximum likelihood joint analysis of all sequences is more tolerant of multiple substitutions than pairwise distance methods. However, high sequence divergence is often associated with other problems,

such as difficulty in alignment, different codon usage biases or nucleotide compositions in different sequences.

How do I test positive selection along specific lineages?

Codonml (codeml with seqtype = 1) has a few models that assign and estimate different ω (= dN/dS) ratios for different branches in the phylogeny. The free-ratios model (model = 1) fits one ω ratio for each branch in the tree, and the two-ratio or three-ratios models (model = 2) let you decide how many ratios you want and which ratio each branch should have. The program will then estimate those ω ratios by ML. The reference is Yang (1998 Mol. Biol. Evol. 15:568-573). The example files are in the folder paml/examples/lysozyme/ in the package. Also look at the notes under "Branch labels" in the section "Tree file and representations of tree topology" in the documentation.

Can I conclude positive selection if I found significantly higher dN/dS ratios for my branches of interest than for other branches?

If your estimates of ω for the branches of interest are not > 1, you might not be able to conclude positive selection based on the likelihood ratio test alone. Relaxed selective constraint along the lineages of interest is an alternative compatible explanation. Furthermore, under Ohta's hypothesis of slightly deleterious mutations, purifying selection is more effective in large populations than in small populations, and so differences in population sizes along lineages provide another compatible hypothesis. If amino acid changes are slightly deleterious, we expect them to be removed from the population at a higher rate in a large population than in a small population. As a result, we expect to see a smaller dN/dS ratio in a large population than in a small one, even if there is no difference between the two lineages in selective pressure or gene function. For example, the dN/dS ratios in many nuclear genes are lower in rodents than in primates or artiodactyls (Ohta. 1995 J. Mol. Evol. 40:56-63; Bielawski et al. 2000 Genetics 156:1299-1308; Yang & Nielsen. 1998 J. Mol. Evol. 46:409-418).

What is the effect of assuming the same selective pressure among lineages?

The lineage-based analysis assumes the same ω (dN/dS) ratio among sites, and detects selection only if the ω averaged over all sites is greater than one. This is expected to make the test conservative and lack power.

Can I obtain a list of probabilities for all sites and all categories under an NSsites model?

Yes, the list is in the file rst. I used such output to make the plot published in Yang & Bielawski (2000 TREE 15: 496-503).

How can I run multiple NSsites models in one analysis?

You can run multiple NSsites models in one go by specifying several models on the NSsites line in codeml.ctl:

```
NSsites = 0 1 2 3 7 8
```

The above specification forces codeml to run M0, M1, M2, M3, M7, and M8 in one go. Note that when more than one NSsites model is specified in this way, the number of categories (`ncatG`) used will match those used in Yang *et al.* (2000), and the ncatG variable in the control file you specify will be ignored.

What does the "mean w" in the rst file mean when I run an NSsites model?

It is the approximate posterior mean of ω (omega). Suppose you use M3 (discrete) with 3 site classes and the estimated w's are w0 = 0.1, w1 = 1.7, and w2 = 3.5. Suppose for a particular

site, the posterior probabilities corresponding to the three site classes are 0.8, 0.15, and 0.05. Then "mean w" for the site is calculated as 0.8 * 0.1 + 0.15 * 1.7 + 0.05 * 3.5 = 0.51. Note that this value is different from any of the three estimated ω values in the M3 model. This is an intuitive approach and is based on the idea that ω is really a continuous number. This calculation ignores sampling errors in estimates of parameters (the frequencies and ω values for the three classes in the above example), and so should not taken with caution. The same warning applies to the posterior probabilities calculated by the program.

The different NSsites models identified different sites. Which model should I believe?

Different NSsites models such as M3 and M8 always produce different parameter estimates and possibly different lists of sites under positive selection. My experience with real data indicates that the models are almost always consistent about which sites might be under diversifying selection even if the lists are different. However, I have heard complaints about the fact that the lists are not exactly the same. There is no easy answer to this question. The list in the main output file is based on a cutoff of P = 50% and sites with P > 95% and P > 99% are marked with * and ** respectively. Typically if a site is in the list under one model, it will have a substantial probability under another model as well. The results might not be so different if you view them in this way. The problem of identifying sites is difficult and is prone to errors, partly because we are making so many inferences at the same time. The situation is similar to getting the top few students from a class. The more you include in your list, the poorer the quality.

How can I make NSsites models 7 and 8 converge?

Those models have remained computationally difficult to use. When the p and q parameters of the beta distribution are extreme (either very small, say < 0.05, or very large, say > 100), the beta distribution has extreme shapes, and it is difficult to discretize the distribution. It is not clear when I can get a stable algorithm working. I have come across two data sets, for which I cannot make the program converge. In both cases, parameters p and q for the beta distribution are extreme.

When the estimates of the two parameters are not so extreme, the program works most often fine. The most effective thing to do seems to try different starting values by using the file in.codeml, and see whether the iteration reaches the same place, as it should.

Can I specify certain branches in a tree file to have independent ω parameters?

Yes. See the question "How do I test positive selection along specific lineages?".

Can I specify certain sites in an alignment to have independent ω parameters?

Yes, you can use prior information such as the protein 3-D structure to partition sites (codons) in the gene into several different classes. You can then specify models that assign and estimate different ω (dN/dS) parameters for the different site partitions. The models are implemented in the program in the framework of analyzing multiple protein-coding genes for the same set of species, using the G option of the sequence data file. You then change the Mgene variable in the control file codeml.ctl. Look at "Option G" in the section "Sequence data format" in the documentation. The reference is Yang & Swanson (2002 Mol. Biol. Evol. 19:49-57).

Note that these models are different from the NSsites models, which do not require or make use of prior information to partition sites into classes. The NSsites models assume that different sites in the sequence have different ω ratios, but that we do not know which sites are conserved or which sites are evolving fast. The NSsites models add noise to the model by assuming a statistical distribution (such as a general discrete distribution, M3, or beta, M7) to account for the variability.

Can I estimate dN and dS between all pairs of taxa in an alignment?

Yes, you use runmode = -2. The ML iteration is still slow, and each pairwise comparison takes several seconds. Some people use yn00 for this purpose.

Why did I get duplicated ω estimates when I used many categories under M3?

If you use a large number (say, >5) for ncatG when you run the discrete model (NSsites = 3), you will find that some proportions become 0 or some ω ratios are identical. The model effectively collapse into one of fewer categories. The discrete-model is a typical "finite-mixture" model, and it is fairly well-known that you can't fit more than a few categories to real data sets. Even if you simulate long sequences with millions of codons and assume 20 distinct ω ratios (which you can easily do with evolver), you will find that you won't be able to fit more than say 5 or 6 categories when you analyze the data using codeml. If you use more categories, the model will just fall back to one of fewer categories. Three or four categories are more than enough for typical datasets. Often even the three-category model falls back to two categories, although you will rarely end up with one single category.

Why do the inferred site classes not match the estimated proportions under the NSsites models?

Thanks to Chris Woelk for asking the question originally. Here are more details about the question. The parameter estimates under M2 (NSsites = 2) suggest the following proportions and ω (dN/ds) values:

```
Frequencies for categories (K=3)
0.721330 0.000000 0.278670
dN/dS rate ratios for categories (K=3)
0.000000 1.000000 0.156520
```

Yet when I counted up the number of sites assigned to Class 3, there were 86 of them. 86/517 = 0.166. This is not equivalent to the frequency 0.278670 cited above.

Answer: Those proportions are not expected to match, and your case is not extreme. Identification of the site classes for each site is to predict the realized values of random variables. The statistical nature of the problem is the same as in the following simple case. Suppose I tell you that nine boys wear black hats and one boy wears a white hat. I then ask you to guess the colour of each of the ten boys' hats. Your best bet will be "black", for each and every boy. The colour for each boy is black with 90% probability and white with 10% probability. Because we ignore the small probability of 10% in each case, the combined result is an intrinsic "bias" in the exercise (I am not very sure whether bias is the right word).

There is nothing wrong with the exercise, as the probabilities are all correct (given that your estimates of parameters are reliable). And it is also the best we can do, if people want to know the most likely site class for each site (you can try to come out with a better solution to the hat colour problem I posed.) However, you should bear in mind that it is almost certain that the inference at one or more sites is wrong, and also that you should not use the inferred site classes to perform further statistical analysis (or otherwise you need to bear in mind the bias). For example if you use your "inferred" colours of hats to "estimate" the proportion of black hats, you get 10/10 rather than 9/10.

By the way, the calculation is done using the Bayes theorem, and the method is known as the (empirical) Bayes method. It is a proper and standard statistical method. Quite a few analyses in paml use it, and my comments above should apply to all those cases. Those include (1) the inference of the rate (or rate class) for sites under models of variable substitution rates among sites (like the gamma model), (2) inference of most likely ancestral nucleotides, amino acids, or codons (that is, ancestral reconstruction). Ancestral reconstruction has been so popular and many people are using reconstructed characters as if they were observed data, and so those methods are at least not proper.

Similarly the most likely rates for individual sites predicted by the gamma-rates models (Yang 1994) or the most likely ancestral character states predicted by the empirical Bayes method (Yang et al. 1995) all involve such systematic biases. Ideally they should not be used as data to construct statistical tests.

More comments due to a post at the PAML discussion group (#######Get the URL.). The question is whether the average posterior probabilities match the MLEs? In the simple hat-colour example, each boy has 90% and 10% probabilities for hat colours and the average over the boys match the observed proportions. The answer to this question is rather mystifying. If the branch lengths are estimated, there is a match, while if the branch lengths are fixed, there is no match. I do not understand why this is the case.

My data set is too large. Can I still analyze them?

How do I generate approximate branch lengths for codon-based analysis?

For large data sets, it might not be too bad to fix the branch lengths in the codon-based analysis as I did (Yang 2000). However, you should be aware that the definitions of branch lengths under nucleotide and codon models are different if you intend to use nucleotide-based analysis to get branch lengths for the codon models. In nucleotide based analysis (DNAML, fastDNAML, PAUP and baseml), the branch length is defined as the expected number of nucleotide substitutions per nucleotide site, while in codon based analysis, it is defined as the expected number of nucleotide substitutions per codon. Since a codon usually has 3 nucleotide sites, you really need to multiple the nucleotide based branch lengths by 3 to get the branch lengths for codon models. Of course, you could also look at how sensitive the codon based analysis is to the branch lengths you use. If you put the option variables GC on the first line of the sequence data file (see the documentation on sequence data format) and run baseml with the options

```
model = 4
Mgene = 4
```

the baseml program will fit an HKY model but accounts for differences among the three codon positions. In the output file, there will also be some output like the following:
"Tree with branch lengths for codon models:
(((rabbit:0.243994, rat:0.560065):0.069631, human:0.210058):0.086463, goat-cow:0.214479, marsupial:1.020316);"

The branch lengths here are the sums across the three codon positions. You can then use this tree and branch lengths in the codon-based analysis. That is what I did for Yang (2000. Maximum likelihood estimation on large phylogenies and analysis of adaptive evolution in human influenza virus A. J. Mol. Evol. 51:423-432).

Please note that branch lengths estimated from PAUP or DNAML in PHYLIP are not directly usable. Like those estimated by baseml, they are defined as the expected number of nucleotide substitutions per nucleotide site. You can multiple all of them by 3 before applying them. Perhaps I should add this as an option.

Another possibility is to use the codon model M0 to get branch lengths, and then run other NSsites models such as NSsites = 1, 2, 3, 7, 8 with those branch lengths fixed. The branch lengths estimated under M0 are probably better than those you can get from a nucletide-based analysis. This way, the M0-M3 comparison will be biased in favour of M0, but since M0 seems always rejected, it should not be a big concern. The computational need for M0 seems feasible with method = 1 when the data set has <2000 sequences.

How do I run the branch-site models (A & B in Yang&Nielsen 2002 MBE)?

The options are

Model A: (model = 2, NSsites = 2)
Model B: (model = 2, NSsites = 3)

They should be in codeml since version 3.12. I have not got time to document them properly, but the output should be self-explanatory.

I got strange estimates of ω (dN/dS) like –1, 89. What do they mean?

Normally -1 means error generated by dS = 0 (absence of synonymous changes), and 89 and 99 are preset upper limits, again generated by dS = 0, so all of those strange values mean infinity. You can report dN and dS if they are in the output.

The likelihood ratio test implemented in the program is fine whether or not the estimated dS = 0. It is quite possible to have dN/dS = infinity but the likelihood ratio test is not significant; that is, a dN/dS ratio estimated to be infinity may not be significantly different from 1. Note that 10 nonsynonymous changes and 0 synonymous changes are stronger evidence for a high dN/dS than 1 nonsynonymous change and 0 synonymous changes, even though both 10/0 and 1/0 are infinity. In a sense the LRT is able to distinguish between the two situations.

How can I estimate an amino acid substitution matrix from my own data, like mtmam.dat and wag.dat?

You choose the option variable

model = 9 :REVaa(nr=189)

You can also use fix_alpha = 0 to fit a REVaa+Gamma model.

If the data set is large, you might want to prepare an in.codeml file to start the iteration from a good starting point. Here is a way of preparing the file to run the REVaa+Gamma model. No clock (clock = 0) is assumed.

Step 1: Use model = 3 to generate good starting values for branch lengths and for alpha.

```
      model = 3
aaRatefile = mtmam.dat for mt proteins or wag.dat for nuclear proteins
 fix_alpha = 0
      alpha = 0.5
     method = 1
```

Step 2: Run REVaa to get good initial values for the substitution rate matrix.

Copy the tree with branch lengths obtained in step 1 into the tree file and choose fix_blength =1 to use them as initial values. Other options are

```
      model = 9
 fix_alpha = 1
      alpha = 1
     method = 0
```

The program will re-estimate the branch lengths as well as the substitution rate parameters in the rate matrix. All those estimates should then be copied into the in.codeml file, plus the alpha estimate from step 1. See notes in step 3.

Step 3: Run REVaa+Gamma. This model involves one more parameter, alpha, than REVaa without the gamma. Copy the parameter estimates under REVaa from step 2 into a file named in.codeml. Add another value for alpha at the end. You had better use the alpha estimate from step 1. Other options are

```
      model = 9
 fix_alpha = 0
```

```
 alpha = 0.5
method = 0
```

This sounds too complicated. The output from this run include a lower-diagonal matrix, which you can copy into a file suitably named, say, MyFavariateProteinMatrix.dat. Look at mtmam.dat or jones.dat etc. for the format. You can then use model = 2 or 3 for other analysis without re-estimating the matrix all over again. As a check, you should obtain identical (except for small rounding errors) branch lengths and log likelihood values when you use model = 2 or 3 but using the rate matrix you estimated under model = 9.

YN00

Why did I get this message: "1967 nucleotides, not a multiple of 3!?"

Both codeml (seqtype = 1) and yn00 require that the sequence length is a multiple of 3. This is done on purpose because I discovered that some people wanted to calculate dS and dN from intron or noncoding sequences. If your sequences are really coding sequences, you can edit the file either to delete one or two nucleotides or add one or two alignment gaps at the beginning or end of the sequence.

Neither codeml nor yn00 can be used for gene prediction.

EVOLVER

Where is the control file for evolver?

Evolver does not have a control file as the other programs do. You run the program and it will display a naive text menu. If you choose to simulate sequence data sets, the program will use data files MCbase.dat, MCcodon.dat, and MCaa.dat to simulate nucleotide, codon, and amino acid sequences, respectively. Print out a copy of the data file and change parameters as necessary.

Can I run evolver with command-line arguments?

Yes, only for simulating data sets. In this case evolver takes 2 arguments. The first has to be one of 5, 6, or 7 to simulate base, codon, or amino acid sequences, respectively. The second argument will be the data file to be used in the simulation. So here are all the four ways of running evolver:

```
evolver
evolver 5 MyMCbaseFile
evolver 6 MyMCcodonFile
evolver 7 MyMCaaFile
```

Can I get the ancestral sequences generated during the simulation in evolver?

Yes. If evolver does not print out a file named ancestral.seq, you need to change the source code evolver.c and recompile the program. Look for the following line in the routine Simulate():

```
int verbose=0;
```

and change it into

```
int verbose=1;
```

Then recompile. The ancestral sequences will be collected into a file named ancestral.seq. If you do not like this name, you can change it in the source code (next line below in the same routine) when you change verbose. Note that this change makes the program even more effective in filling your hard disk.

How do I simulate data under codon models of variable dN/dS (omega) ratios among sites as did Anisimova et al. (2001 MBE)?

(1) You need to define a variable and recompile the program evolver. Look at the beginning of the file evolver.c. There are the following lines.

```
/*
   #define CodonNSbranches 1
   #define CodonNSsites 1
*/
```

Change these lines into

```
#define CodonNSsites 1
```

Then type make or look at the paml/src/readme.txt for compiling instructions. I suggest that you rename the executable something like evolverNSsites so that it will not be confused with the default evolver in the distribution.

Alternative to changing the source code, you can use the following command to compile evolver:

```
cc -DCodonNSsites -o evolverNSsites -O2 evolver.c tools.c eigen.c -lm
```

You might need to chaneg cc into gcc, and you might need to remove the flag -lm.

(2) After the change, the program does not display the text menu anymore and will simulate data sets taking input from MCcodonNSsites.dat (not MCcodon.dat). That file is in the v3.1 distribution.

Again watch out for the screen output to make sure that the input data file is read correctly.

How do I simulate data sets to be analyzed by paup?

If you choose the PAUP* format, the program will look for files with the following names: paupstart (which the program copies to the start of the data file), paupblock (which the program copies to the end of each simulated data set), and paupend (which the program incorporates at the end of the file. This makes it possible to use PAUP* to analyze all data sets in one run.

What is the difference between evolver and seq-gen?

There is not much difference between evolver and seq-gen for simulating nucleotide sequences. Either one is fine. evolver can also simulate amino acid and codon sequences.