A worked example of estimating ω and testing for adaptive evolution in six parasite species

Supplementary material for:

A beginners guide to estimating the non-synonymous to synonymous rate ratio of all protein-coding genes in a genome.

Daniel C. Jeffares*, Bartłomiej Tomiczek, Victor Sojo and Mario dos Reis

In this tutorial we will estimate the non-synonymous to synonymous rate ratio ($\omega = d_N/d_S$) for 3,261 one-to-one orthologous gene alignments from six *Plasmodium* species (*P. berghei*: PB, *P. chabaudi*: PC, *P. yoelii*: PY, *P. falciparum*: PF, *P. vivax*: PV and *P. knowlesii*: PK), as well as performing some tests of adaptive evolution. The data are as described in ***(1).*** The files needed to run the tutorial are provided in the supplementary data file worked_example_files.zip. The sequence files are available at the PlasmoDB database (www.plasmodb.org), as described below. This tutorial assumes that you have some basic knowledge of working in a Unix system, and that you have PAML, Clustal Omega and PAL2NAL installed and working correctly. Please make sure you are working with the latest version of PAML (4.5 as of August 2012).

Downloading and formatting data (sequence files and ortholog assignments).

All genes and ortholog assignments were obtained from PlasmoDB (http://www.plasmodb.org), as follows:

1. Search using 'Identify Genes based on Orthology Phylogenetic Profile', using the select all option (all 6 species).
2. Add search step 'Transform by Orthology', selecting Syntenic Orthologs Only?: Yes.
3. Select 'Download 39586 Genes' and collect
   a. Protein sequences
   b. CDS sequences
   c. The ortholog list (containing the fields IDs, Species, Ortholog count, Paralog count, Ortholog Group).

We then need to collect the ortholog file into a comma-separated format for the ortholog_processor_aligner.pl script. We used a perl script to select and list *only* ortholog groups with one ortholog in each species and no paralogs, collect_ortho_groups_plasmodb.pl, which outputs a CSV file of 3286 ortholog groups. We then split the DNA and proteins sequences into one file per sequence, using a custom perl script, split_seqs.pl.

Alignment of orthologs

We then aligned all *protein* sequences using Clustal Omega, then convert to nucleotide sequences with PAL2NAL. This was achieved using the ortholog_processor_aligner.pl script:

```
./ortholog_processor_aligner.p l\
 -l 1to1_orthologs.txt2\
-i plasmoseqs/ -o clustal_alns -c -p
```

Estimating ω for all genes in a genome

First we will estimate average ω values (model M0) for each ortholog. File *all.paml* contains 3,261 blocks, each block corresponding to an alignment in PAML format for each ortholog from the six *Plasmodium* genomes. File *tree.txt* contains the species tree for the *Plasmodium* species. File *codeml-M0.ctl* is the control file needed by CODEML to run the analysis. Open your favourite text editor and look into *codeml-M0.ctl*, which is reproduced here:

```
      seqfile = all.paml      * sequence data filename
     treefile = tree.txt      * tree structure file name
      outfile = mlc           * main result file name

      runmode = 0  * 0: user tree;  1: semi-automatic;  2: automatic
                   * 3: StepwiseAddition; (4,5):PerturbationNNI; ...

      seqtype = 1  * 1:codons; 2:AAs; 3:codons-->AAs
    CodonFreq = 2  * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table

        ndata = 3261 * number of gene alignments to be analysed
        clock = 0  * 0:no clock, 1:clock; 2:local clock

        model = 0  * models for codons: 0:one, 1:b, 2:2 or more dN/dS
                   * ratios for branches

      NSsites = 0  * 0:one w;1:neutral;2:selection;
                   * 3:discrete;4:freqs;
                   * 5:gamma;6:2gamma;7:beta;8:beta&w;9:beta&gamma;

        icode = 0  * 0:universal code; 1:mammalian mt; 2-10:see below

    fix_omega = 0  * 1: omega or omega_1 fixed, 0: estimate
        omega = .4 * initial or fixed omega for codons

    cleandata = 0  * remove sites with ambiguity data (1:yes, 0:no)?
```

A detailed explanation of this file and all the options available is given in the PAML manual. The CodonFreq option is important, you should use options 1:F1x4 or 2:F3x4, which use nucleotide frequencies at the different codon positions to estimate overall codon frequencies. For long alignments of a few dozen species, option 3 is useful. Do not use it if any codon frequencies are equal to zero. Note the ndata option, this tells CODEML how many blocks should be analysed from the sequence file (*all.paml*). The two most important options are model and NSsites. The first option tells CODEML whether ω should be allowed to vary among branches in the tree, and the second option tells CODEML whether ω should be allowed to vary among sites. In both cases we have set these options to zero. This means that we will run the analysis with model M0 for all branches of the tree: a single ω for all sites and for all branches.

Go into the directory where the necessary files are stored (*all.paml*, *tree.txt*, *codeml-M0.ctl*), and in the command line type

```
codeml codeml-M0.ctl
```

This will start the analysis with CODEML. The analysis of all 3,769 genes took about 10h in a 1.5GHz 64-bit laptop. If your laptop or desktop computer has several cores (say 2 or 4), or if you have access to a high performance computer cluster, you can run the analyses in parallel. For example, if your computer has 4 cores, you can split the large *all.paml* file into four smaller files (say *all01.paml*, *all02.paml*, etc.) with around 800 alignments in each. Save each file in its own directory, together with copies of the control and tree files. You can then run CODEML four times in parallel and the analysis time would be brought down substantially (to about 2.5h in the 1.5GHz 4-core 64-bit laptop above).

The main results of the analysis are written to the *mlc* file (set with the `outfile` option in the control file). You can look into this file while the analysis is running, to see the results for the genes that have already been analysed. The output is divided into several blocks, each block corresponding to each ortholog. For the first block, look for a line looking like

```
lnL(ntime:  9  np: 11): -11050.757142      +0.000000
```

This is the log-likelihood of the gene alignment under site-model M0 with the same ω for all branches. The log-likelihood value will be very important in tests of adaptive evolution, as we will see below. CODEML estimates 11 parameters in this case: 9 branch lengths in the tree, the transition/transversion rate ratio κ, and ω. The likelihood value estimated in your computer may be slightly different for smallest decimal figures. Scroll down the file and look for a line looking like

```
(((PCHAS_010150: 0.094157, (PY02270: 0.057877, ...
```

This is the tree with branch lengths (in codon substitutions per site) for the first gene. If you scroll further down, you will see a few lines looking like

```
kappa (ts/tv) =  2.50350
omega (dN/dS) =  0.11532
```

These are the maximum likelihood estimates of the transition/transversion ratio κ (kappa), and ω (omega). You can see that the value of ω is much smaller than one. This is typical of most genes. As discussed in the introduction, most sites in most proteins are under strong purifying selection and usually average ω < 1.

Once the analysis has completed, make sure you are in the directory were the analysis run and in the command line type

```
grep omega mlc > all.omega.txt
```

This will collect all the ω values calculated for each one of the 3,769 gene alignments. You should be able to import this file into your favourite spreadsheet program for statistical analysis. Figure S1, shows a histogram of the ω values calculated in this step.

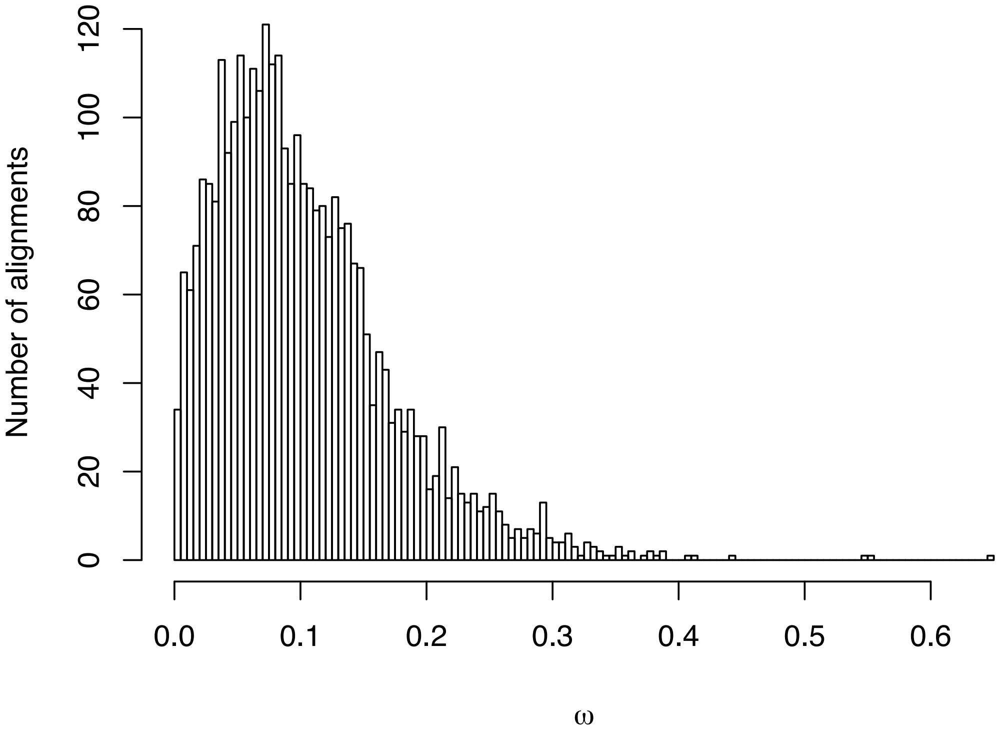

Figure S1. Histogram of ω values calculated for 3,261 orthologs in six *Plasmodium* species. The ω values range from 3.3×10^{-4} to 0.646, with a mean of 0.102. This shows the high degree of purifying selection normally acting on most codon sites on most proteins.

Statistical tests of adaptive evolution among codon sites (using 'site models')

File *PBANKA_030280.pal2nal.paml* is a 65-codon alignment for a fast evolving gene in the parasite genomes. We will test this gene for positive selection (site-model). File *codeml-M1278.ctl* is the control file needed for this analysis. Open it in your favourite text editor:

```
 seqfile = PBANKA_030280.pal2nal.paml * sequence data filename
treefile = tree.txt      * tree structure file name
 outfile = mlc-M1278           * main result file name

 runmode = 0  * 0: user tree;  1: semi-automatic;  2: automatic
              * 3: StepwiseAddition; (4,5):PerturbationNNI
```

```
      seqtype = 1  * 1:codons; 2:AAs; 3:codons-->AAs
    CodonFreq = 2  * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table

        clock = 0  * 0:no clock, 1:clock; 2:local clock;

        model = 0  * models for codons: 0:one, 1:b, 2:2 or more dN/dS
                   * ratios for branches

      NSsites = 0 1 2 7 8 * 0:one w;1:neutral;2:selection;
                   * 3:discrete;4:freqs;
                   * 5:gamma;6:2gamma;7:beta;8:beta&w;9:beta&gamma;

        icode = 0  * 0:universal code; 1:mammalian mt; 2-10:see below

    fix_omega = 0  * 1: omega or omega_1 fixed, 0: estimate
        omega = .4 * initial or fixed omega for codons

    cleandata = 0  * remove sites with ambiguity data (1:yes, 0:no)?
```

Note the changes we have made when compared to the previous control file (*codeml-M0.ctl*). In particular note the option NSsites = 0 1 2 7 8. This option tells CODEML to test 5 models of ω variation among sites. Model M0 is the standard one ω for all sites model. Models M1 and M7 have a fraction of sites with ω < 1 and a fraction with ω = 1. In models M2 and M8 account for positive selection and in both models an extra class of sites with ω > 1 is allowed. Models 1 and 2, and 7 and 8, form two pairs of models that can be used to test for positive selection using the likelihood ratio test (LRT). The two sets of models are complementary and differ simply in how sites with ω < 1 are treated. Make sure you are in the appropriate directory and in the command line type

```
codeml codeml-M1278.ctl
```

This will start the analysis, which should take about 10min to run. Once the analysis has finished, open the *mlc-M1278* file and look at the results. From the command line type

```
grep lnL mlc-M1278
```

This will generate the following output

```
lnL(ntime:  9  np: 11):  -1146.298093      +0.000000
lnL(ntime:  9  np: 12):  -1129.989551      +0.000000
lnL(ntime:  9  np: 14):  -1118.240939      +0.000000
lnL(ntime:  9  np: 12):  -1128.247979      +0.000000
lnL(ntime:  9  np: 14):  -1117.885273      +0.000000
```

These are the log-likelihoods for the 5 models. You can see that the worst likelihood is for model M0: -1146.3. Model M0 is nested within model M1, so we can use the LRT to compare these two models. The LRT statistic is $\chi^2 = 2\times(-1130 + 1146.3) = 16.3$. The degrees of freedom are $k = 12 - 11 = 1$. We can use the CHI2 program from the PAML package to assess whether the value χ^2=16.3 with k=1 is statistically significant. In the command line type

```
chi2
```

And you should see the following output:

```
Chi-square critical values

                         Significance level

 DF    0.9950   0.9750   0.9000   0.5000   0.1000   0.0500   0.0100   0.0010

  1    0.0000   0.0010   0.0158   0.4549   2.7055   3.8415   6.6349  10.8276
  2    0.0100   0.0506   0.2107   1.3863   4.6052   5.9915   9.2103  13.8155
```

You can see that the critical value for one-degree of freedom and significance level $\alpha=0.001$ is 10.83, our value is 16.3, so the inclusion of a class of neutral sites with $\omega = 1$ is statistically justified. We can now compare model M1 vs. M2 and model M7 vs. M8. For both comparisons the degrees of freedom are $k = 14 - 12 = 2$. The LRT statistics are $\chi^2 = 2\times(-1118.2 + 1130) = 11.8$ for M1 vs. M2; and $\chi^2 = 2\times(-1117.9 + 1128.2) = 10.3$. As you can see from the table above, both values are significant at the $\alpha=0.01$ level.

CODEML will perform Bayesian identification of codon sites (columns in the alignment) that are under positive selection. It uses two methods, the naïve empirical Bayes (NEB) and the Bayes empirical Bayes (BEB) methods. The BEB method is the newest and should be preferred to oldest, simplistic NEB. Scroll down the mlc-M1237 file until you find the BEB block for model M2 (there will be another block for model M8 further down the file). It should look like this:

```
Bayes Empirical Bayes (BEB) analysis (Yang, Wong &
Nielsen 2005. Mol. Biol. Evol. 22:1107-1118)
Positively selected sites (*: P>95%; **: P>99%)
(amino acids refer to 1st sequence: PBANKA_030280)

            Pr(w>1)     post mean +- SE for w

    10 L      0.929         8.361 +- 2.481
    19 Q      0.502         4.889 +- 4.022
    21 T      0.535         5.178 +- 4.057
    24 N      0.993**       8.868 +- 1.614
    25 V      0.864         7.829 +- 3.040
    28 D      0.999**       8.915 +- 1.497
    31 V      0.672         6.256 +- 3.878
    32 D      0.724         6.707 +- 3.742
    33 D      0.998**       8.912 +- 1.506
    36 V      0.754         6.948 +- 3.636
    38 T      0.918         8.257 +- 2.594
    40 S      0.967*        8.662 +- 2.026
    43 V      0.956*        8.577 +- 2.171
    47 S      0.515         4.973 +- 4.016
```

For model M2 and the BEB method, CODEML identified 14 sites under positive selection. The first column is the site number (column number) in the alignment. Then the amino acid in the first sequence is shown. Then the posterior probability Pr that ω

> 1 for the given codon site is shown (results are significant if Pr > 0.95). The last column is the posterior mean of ω with standard errors. For genes involved in an arms-race with the host immune system, codons with ω > 1 are usually evolving fast to change the surface shape of antigenic proteins, so as to avoid the antibodies produced by the host.

If you set seqfile = paml.all in the control file above, you could run all the positive selection site-models on all 3,261 alignments, but this is expected to be very slow. Our perl script ***codeml_site_models.pl*** can run the analysis for the five models for thousands of genes, and will perform the likelihood ratio test and give the significance level in each case.

Statistical tests of ω variation among lineages (using 'branch models')

A different test that can be performed is to check whether ω is different for different branches in the phylogeny. We will test whether (((2,(3,1)),6),5,4); is statistically different along the branch that links *Plasmodium falciparum* (the sixth species in the alignments) to the rest of the phylogeny. Open *tree.txt* with your favourite text editor, and add a #1 tag to species six:

```
(((2,(3,1)),6 #1),5,4);
```

Now open *codeml-M0.ctl*, and modify it so it looks like:

```
      seqfile = PBANKA_030280.pal2nal.paml * sequence data filename
     treefile = tree.txt      * tree structure file name
      outfile = mlc-M0-BS     * main result file name

      runmode = 0  * 0: user tree;  1: semi-automatic;  2: automatic
                   * 3: StepwiseAddition; (4,5):PerturbationNNI; ...

      seqtype = 1  * 1:codons; 2:AAs; 3:codons-->AAs
    CodonFreq = 2  * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table

        ndata = 1 * number of gene alignments to be analysed
        clock = 0  * 0:no clock, 1:clock; 2:local clock

        model = 2  * models for codons: 0:one, 1:b, 2:2 or more dN/dS
                   * ratios for branches

      NSsites = 0  * 0:one w;1:neutral;2:selection;
                   * 3:discrete;4:freqs;
                   * 5:gamma;6:2gamma;7:beta;8:beta&w;9:beta&gamma;

        icode = 0  * 0:universal code; 1:mammalian mt; 2-10:see below

    fix_omega = 0  * 1: omega or omega_1 fixed, 0: estimate
        omega = .4 * initial or fixed omega for codons

    cleandata = 0  * remove sites with ambiguity data (1:yes, 0:no)?
```

Note that now `model = 2`. This tells CODEML that we want to test variation in ω among lineages, but not among sites (`NSsites = 0`). In the command line type

```
codeml codeml-M0.ctl
```

Once the analysis finishes, look for the log-likelihood in the *mlc-M0-BS* file:

```
lnL(ntime:  9  np: 12):  -1145.987056      +0.000000
```

We know from our previous run that the log-likelihood for one ω for all sites and all branches is -1146.3. This is the null model of no ω variation among branches. The LRT statistic is χ^2= 2×(-1146.3 + 1146) = 0.6 with *k* = 12 – 11 = 1 degrees of freedom. This is not statistically significant according to the table provided by CHI2. So we concluded that for this gene, there is no evidence of variation in ω for the branch leading to *Plasmodium falciparum*. Our perl script ***codeml_branch_models.pl*** will calculate the branch test for thousands of genes.

Statistical tests of adaptive evolution among sites and lineages (branch-site models)

We will now show how to do a test of positive selection among sites and among lineages. Open *codeml-M0.ctl*, edit the file, and rename it to *codeml-M0-BS.ctl* so it looks like:

```
     seqfile = PBANKA_030280.pal2nal.paml * sequence data filename
    treefile = tree.txt      * tree structure file name
     outfile = mlc-M2-BS-H0  * main result file name

     runmode = 0  * 0: user tree;  1: semi-automatic;  2: automatic
                  * 3: StepwiseAddition; (4,5):PerturbationNNI; ...

     seqtype = 1  * 1:codons; 2:AAs; 3:codons-->AAs
   CodonFreq = 2  * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table

       ndata = 1 * number of gene alignments to be analysed
       clock = 0  * 0:no clock, 1:clock; 2:local clock

       model = 2  * models for codons: 0:one, 1:b, 2:2 or more dN/dS
                  * ratios for branches

     NSsites = 2  * 0:one w;1:neutral;2:selection;
                  * 3:discrete;4:freqs;
                  * 5:gamma;6:2gamma;7:beta;8:beta&w;9:beta&gamma;

       icode = 0  * 0:universal code; 1:mammalian mt; 2-10:see below

   fix_omega = 1  * 1: omega or omega_1 fixed, 0: estimate
       omega = 1 * initial or fixed omega for codons

   cleandata = 0  * remove sites with ambiguity data (1:yes, 0:no)?
```

In the command line type

```
codeml codeml-M2-BS.ctl
```

This will run the null model of the branch-site test of positive selection. We allow ω to vary among sites, but we have a fixed class of sites where ω = 1 for the background and foreground (the *P. falciparum*) branches. No sites with ω > 1 are allowed. Once the analysis has completed, look at the log-likelihood value in the *mlc-M2-BS-H0* file:

```
lnL(ntime:  9  np: 13):  -1129.989551      +0.000000
```

Now open codeml-M2-BS.ctl and edit the following options:

```
outfile = mlc-M2-BS-H1  * main result file name
fix_omega = 0  * 1: omega or omega_1 fixed, 0: estimate
```

Now run codeml again. We now allow ω > 1 for some sites in the foreground branch, but ω = 1 for the same sites in the background branch. The output will be written to the *mlc-MS-BS-H1 file*. Look for the log-likelihood value:

```
lnL(ntime:  9  np: 14):  -1129.989562      +0.000000
```

The likelihood ratio statistic is 2×(-1130 + 1130) = 0. There is no improvement in the likelihood when we allow $\omega > 1$ in the foreground branch. In the branch-site test of positive selection, the LRT test statistic does not follow a χ^2 distribution since some of the required regularity conditions are not met (Yang and dos Reis 2011). However, using a χ^2 with one degree of freedom makes the test conservative. Using the CHI2 table with can see that the test is not statistically significant in this case. Our perl script ***codeml_branch_site_models.pl*** will calculate the branch-site test for thousands of genes.

Summary

This tutorial gave a brief overview of using CODEML to estimate ω and to test for adaptive evolution. The PAML package provides many test examples, and the manual provides invaluable information. Please do consult this resources before embarking in your quest for positive selection.

References

1. Essien, K., Hannenhalli, S., and Stoeckert, C. J. (2008) Computational analysis of constraints on noncoding regions, coding regions and gene expression in relation to Plasmodium phenotypic diversity, *PLoS ONE 3*, e3122.