PAML (Phylogenetic Analysis by Maximum Likelihood)

A program package by Ziheng Yang

(Demonstration by Joseph Bielawski)

1. Three inference tasks

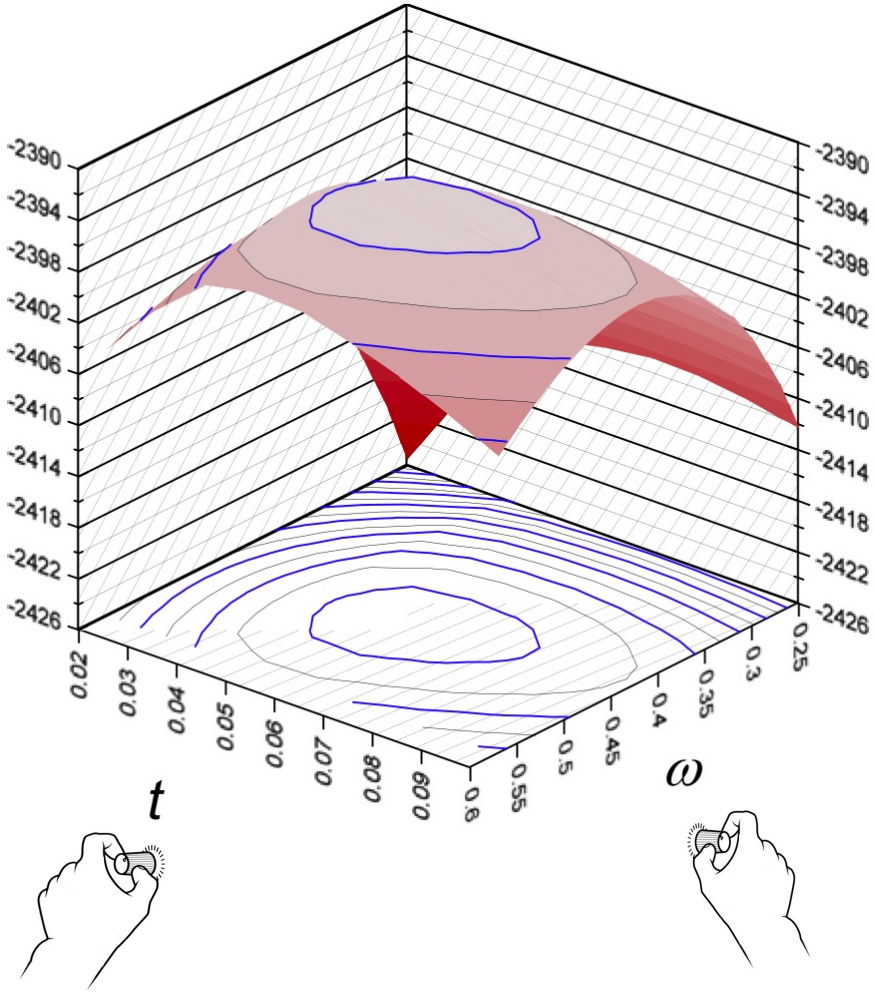
3 analytical tasks

task 1. parameter estimation (e.g., ω)

task 2. hypothesis testing

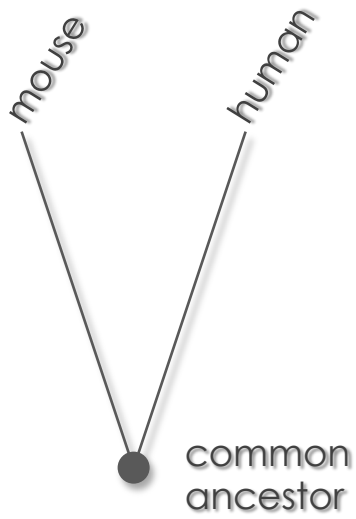
task 3. make predictions (e.g., sites having $\omega > 1$)

task 1: parameter estimation



Parameters: t and ω

Gene: acetylcholine α receptor



lnL = -2399

3 analytical tasks

task 1. parameter estimation (e.g., ω)

task 2. hypothesis testing

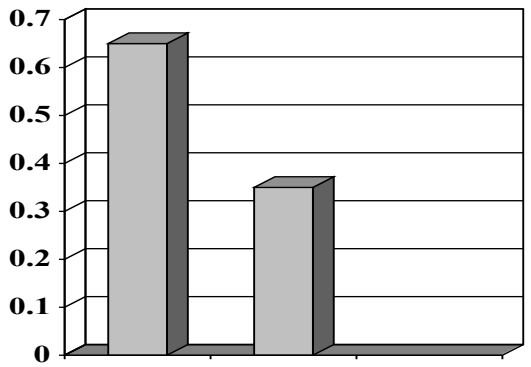
task 3. make predictions (e.g., sites having $\omega > 1$)

task 2: likelihood ratio test for positive selection

H_0: variable selective pressure but NO positive selection (M1a)

H_1: variable selective pressure with positive selection (M2a)

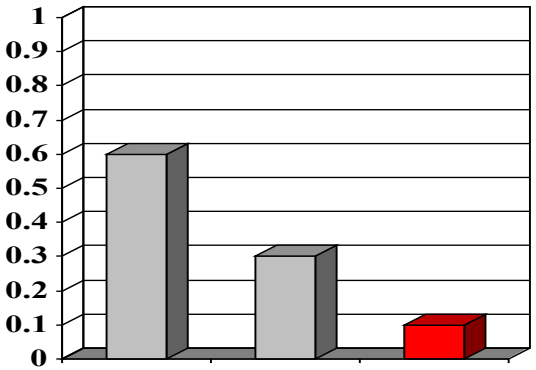
Model 1a (**M1a**)



$\hat{\omega}$ = 0.5 (ω = 1)

(NP = 2)

Model 2a (**M2a**)



$\hat{\omega}$ = 0.5 (ω = 1) $\hat{\omega}$ = 3.25

(NP = 4)

Likelihood ratio test:

test stat: $\mathbf{2\Delta l} = 2(l_{H1} - l_{H0})$

distribution: χ^2

d.f. 4 - 2 = 2

3 analytical tasks

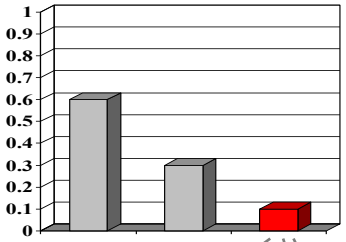
task 1. parameter estimation (e.g., ω)

task 2. hypothesis testing

task 3. make predictions (e.g., sites having $\omega > 1$)

task 3: which sites have *dN/dS* > 1

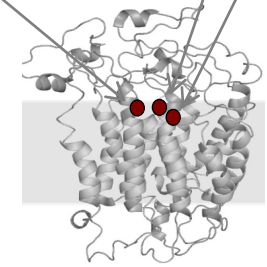
model:
5% have $\omega > 1$



Bayes' rule:
site 4, 12 & 13

```
GTG CTG TCT CCT GCC GAC AAG ACC AAC GTC AAG GCC GCC TGG GGC AAG GTT GGC GCG CAC
... ... ... G.C ... ... ... T.. ..T ... ... ... ... ... ... ... ... ... .GC A..
... ... ... ..C ..T ... ... ... ... A.. ... A.T ... ... .AA ... A.C ... AGC ...
... ..C ... G.A .AT ... ..A ... ... A.. ... AA. TG. ... ..G ... A.. ..T .GC ..T
... ..C ..G GA. ..T ... ... ..T C.. ..G ..A ... AT. ... ..T ... ..G ..A .GC ...
```

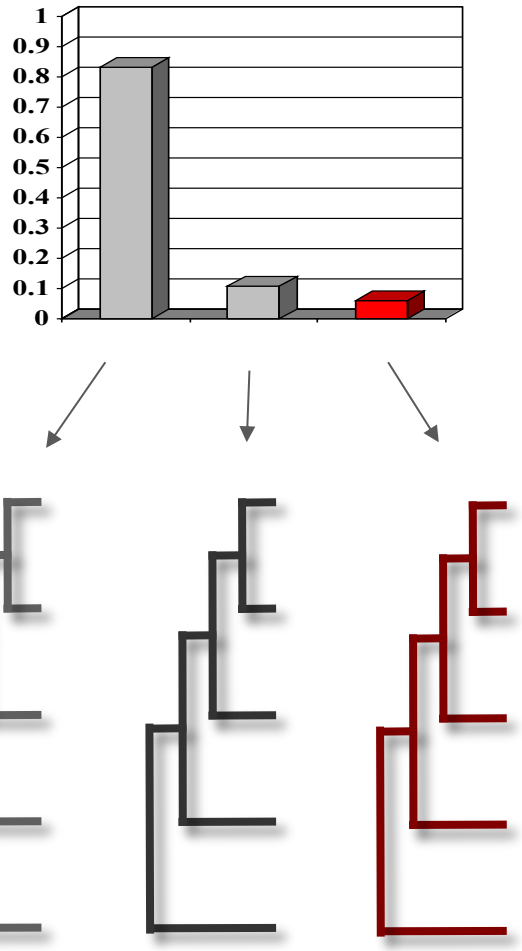
structure:
sites are in contact



review the mixture likelihood (model **M3**)

$$P(\mathbf{x}_h) = \sum_{i=0}^{K-1} p(\omega_i) P(\mathbf{x}_h \mid \omega_i)$$

Total probability **Prior** **Likelihood**



ω_0= 0.03 ω_1= 0.40 ω_2= 3.25

p_0= 0.85 p_1= 0.10 p_2= 0.05

Bayes' rule for identifying selected sites

- Site class 0: ω_0 = .03, 85% of codon sites
- Site class 1: ω_1 = .40, 10% of codon sites
- ? → **Site class 2: ω_2 = 3.2, 05% of codon sites**

Prior probability of hypothesis (ω_2)

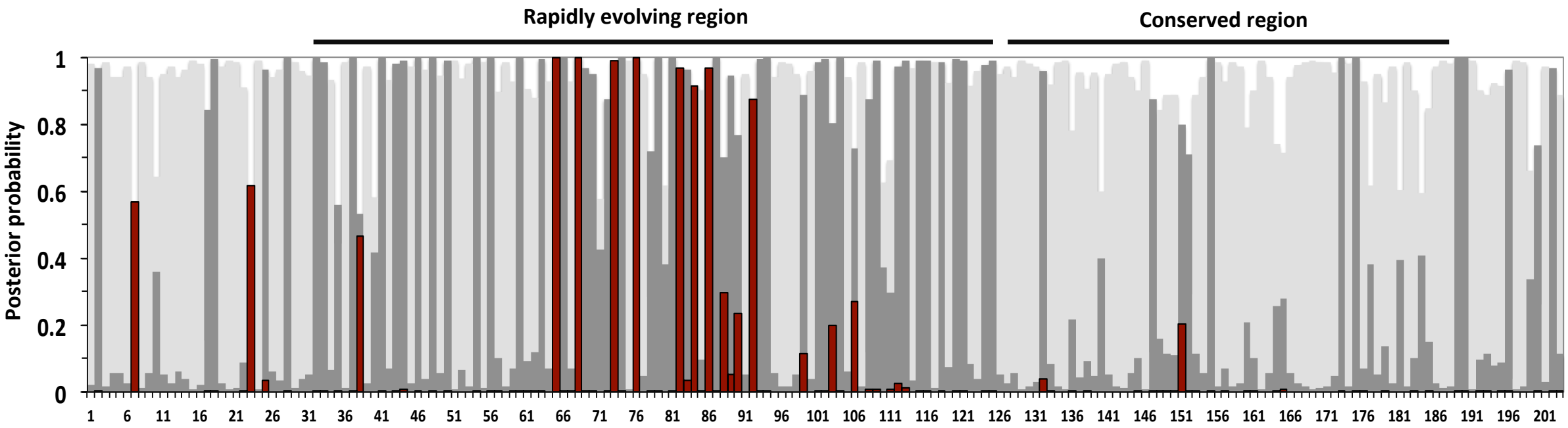
Likelihood of hypothesis (ω_2)

$$P(\omega_2 \mid x_h) = \frac{P(\omega_2)P(x_h \mid \omega_2)}{\sum_{i=0}^{K-1} P(\omega_i)P(x_h \mid \omega_i)}$$

Posterior probability of hypothesis (ω_2)

Marginal probability (Total probability) of the data

task 3: Bayes rule for which sites have dN/dS > 1



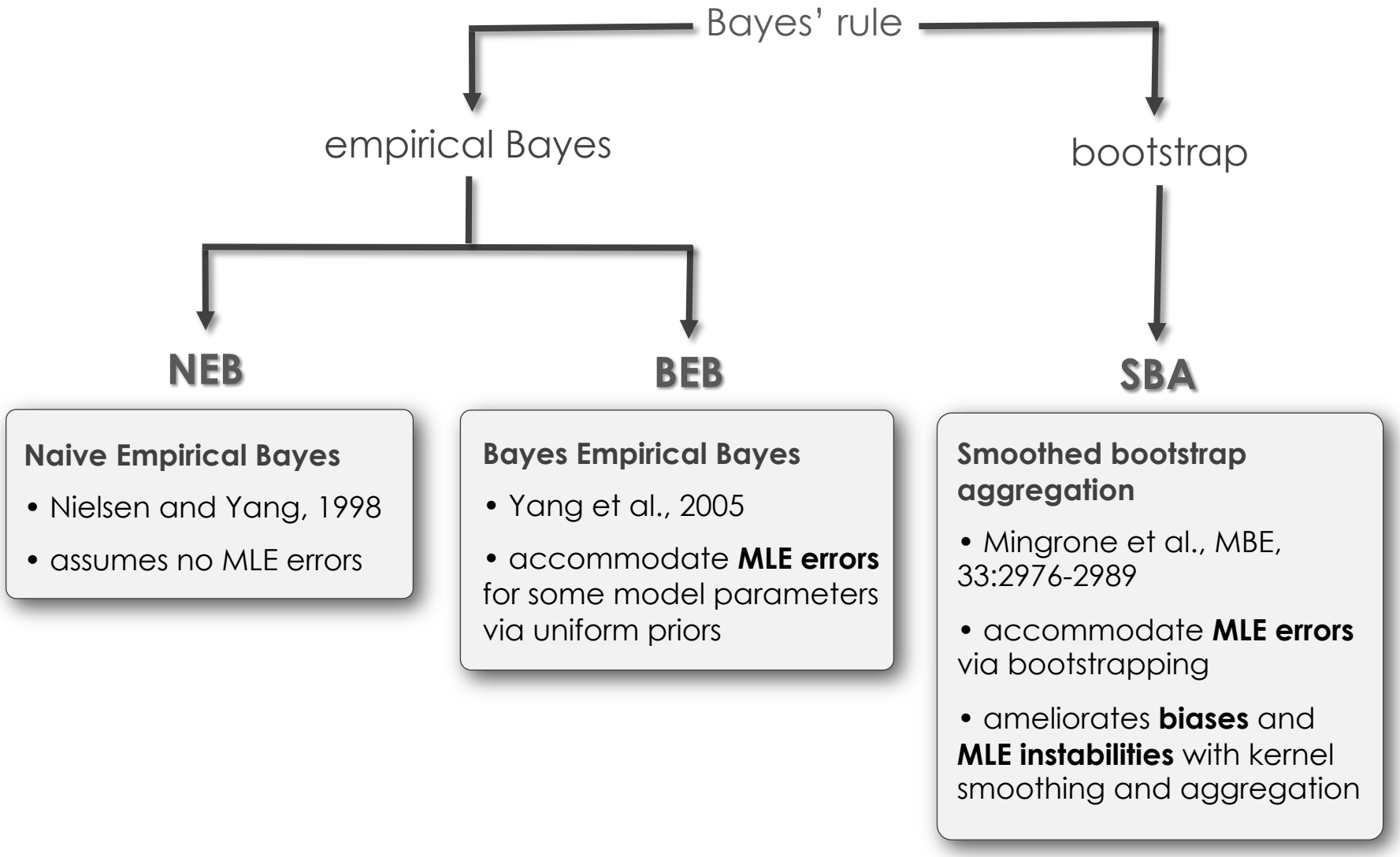
Site class 0: ω_0 = .03 (strong purifying selection)

Site class 1: ω_1 = .40 (weak purifying selection)

Site class 2: ω_2 = 3.25 (positive selection)

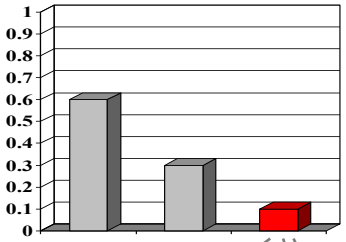
NOTE: The posterior probability should NOT be interpreted as a "*P*-value"; it can be interpreted as a measure of relative support, although there is rarely any attempt at "calibration".

task 3: Bayes rule for which sites have *dN/dS* > 1



tasks 1-3…

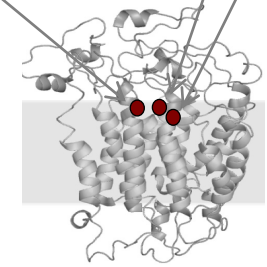
model:
5% have $\omega > 1$



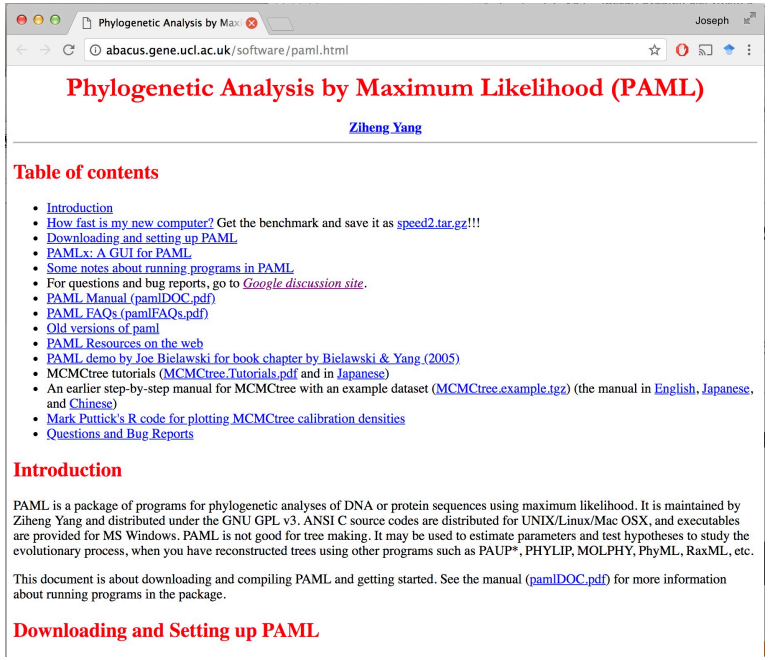
Bayes' rule:
site 4, 12 & 13

```
GTG CTG TCT CCT GCC GAC AAG ACC AAC GTC AAG GCC GCC TGG GGC AAG GTT GGC GCG CAC
... ... ... G.C ... ... ... T.. ..T ... ... ... ... ... ... ... ... ... .GC A..
... ... ... ..C ..T ... ... ... ... A.. ... A.T ... ... .AA ... A.C ... AGC ...
... ..C ... G.A .AT ... ..A ... ... A.. ... AA. TG. ... ..G ... A.. ..T .GC ..T
... ..C ..G GA. ..T ... ... ..T C.. ..G ..A ... AT. ... ..T ... ..G ..A .GC ...
```

structure:
sites are in contact



Software: both **PAML** and **HyPhy** are great choices for model-based inference!



Phylogenetic Analysis by Maximum Likelihood (PAML)

Ziheng Yang

Table of contents

- Introduction
- How fast is my new computer? Get the benchmark and save it as speed2.tar.gz!!!
- Downloading and setting up PAML
- PAMLx: A GUI for PAML
- Some notes about running programs in PAML
- For questions and bug reports, go to *Google discussion site*.
- PAML Manual (pamlDOC.pdf)
- PAML FAQs (pamlFAQs.pdf)
- Old versions of paml
- PAML Resources on the web
- PAML demo by Joe Bielawski for book chapter by Bielawski & Yang (2005)
- MCMCtree tutorials (MCMCtree.Tutorials.pdf and in Japanese)
- An earlier step-by-step manual for MCMCtree with an example dataset (MCMCtree.example.tgz) (the manual in English, Japanese, and Chinese)
- Mark Puttick's R code for plotting MCMCtree calibration densities
- Questions and Bug Reports

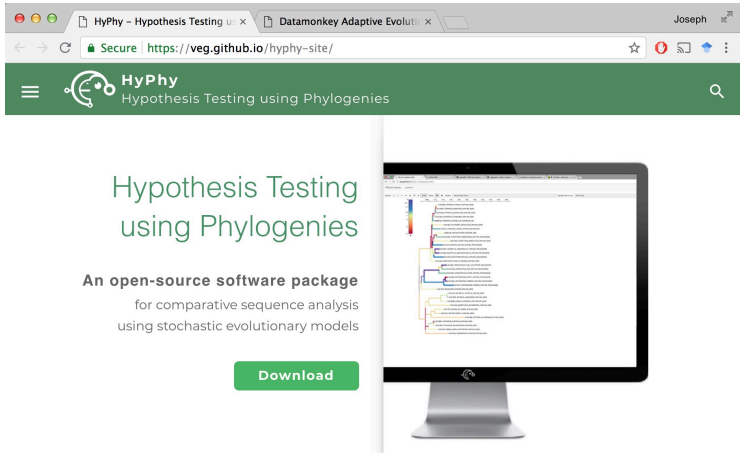
Introduction

PAML is a package of programs for phylogenetic analyses of DNA or protein sequences using maximum likelihood. It is maintained by Ziheng Yang and distributed under the GNU GPL v3. ANSI C source codes are distributed for UNIX/Linux/Mac OSX, and executables are provided for MS Windows. PAML is not good for tree making. It may be used to estimate parameters and test hypotheses to study the evolutionary process, when you have reconstructed trees using other programs such as PAUP*, PHYLIP, MOLPHY, PhyML, RaxML, etc.

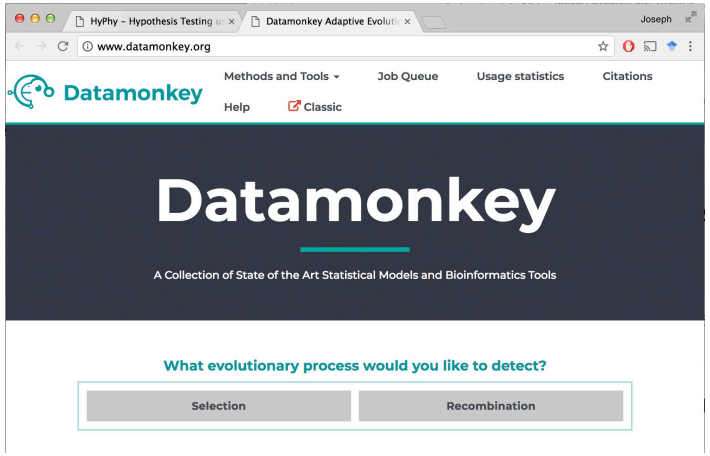
This document is about downloading and compiling PAML and getting started. See the manual (pamlDOC.pdf) for more information about running programs in the package.

Downloading and Setting up PAML

http://abacus.gene.ucl.ac.uk/software/paml.html



https://veg.github.io/hyphy-site/



http://www.datamonkey.org/

2. Brief introduction to PAML

programs in the package…

baseml	for nucleotide data (bases)
basemlg	continuous-gamma for nucleotides
codeml	**for amino acid & codons data**
evolver	simulation, tree distances
yn00	d_N and d_S by YN00
chi2	chi square table
pamp	parsimony (Yang and Kumar 1996)
mcmctree	Bayes MCMC tree (Yang & Rannala 1997). SLOW

Running PAML programs

1. Sequence data file
2. Tree file
3. Control file (*.ctl)

jpbielawski — -bash — 98×39

```
     0.083510     1.429014
     0.000010     0.400000
    50.000000   999.000000

Iterating by ming2
Initial: fx=   790.048189
x=  0.08351  1.42901

  1 h-m-p  0.0008 1.5892  53.4319 +CCYCYYCYY

a      0.002851     0.002852     0.002853     0.002852
f   786.714752   786.714671   786.714928   786.714815
        2.850987e-03    0.173056        1.552250        786.714752
        2.851077e-03    0.173059        1.552254        786.715025
        2.851167e-03    0.173062        1.552257        786.714972
        2.851257e-03    0.173064        1.552261        786.714775
        2.851347e-03    0.173067        1.552265        786.715034
        2.851437e-03    0.173070        1.552269        786.714792
        2.851527e-03    0.173073        1.552273        786.714784
        2.851617e-03    0.173076        1.552277        786.714819
        2.851707e-03    0.173079        1.552281        786.714959
        2.851797e-03    0.173081        1.552285        786.714638
        2.851887e-03    0.173084        1.552289        786.714695
        2.851977e-03    0.173087        1.552292        786.714803
        2.852067e-03    0.173090        1.552296        786.714769
        2.852157e-03    0.173093        1.552300        786.714804
        2.852247e-03    0.173095        1.552304        786.714764
        2.852337e-03    0.173098        1.552308        786.715002
        2.852427e-03    0.173101        1.552312        786.714815
        2.852517e-03    0.173104        1.552316        786.714900
        2.852607e-03    0.173107        1.552320        786.714754
        2.852697e-03    0.173110        1.552324        786.714922
Linesearch2 a4: multiple optima?
C   786.714671 10 0.0029    41 | 0/2
  2 h-m-p  0.0050 0.2387  30.7213 -------------..  | 0/2
  3 h-m-p  0.0000 0.0081 142.5083 -------- | 0/2
  4 h-m-p  0.0002 0.1084   2.2204 ++C     786.707806  0 0.0035    76 | 0/2
  5 h-m-p  0.0160 8.0000   1.9177 +CCYCY
```

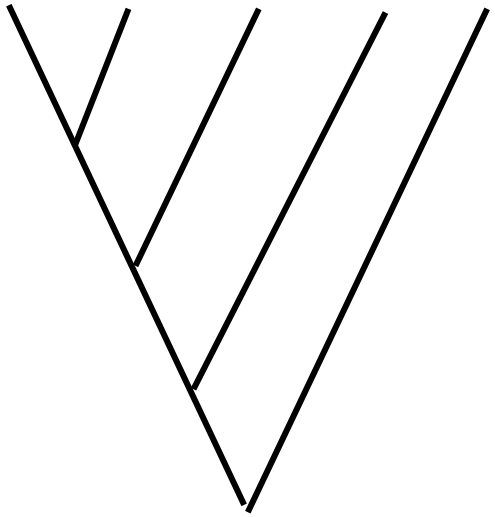
1. sequence file (modified “PHYLIP” format)

```
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
```

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

2. tree file (“Newick” format)

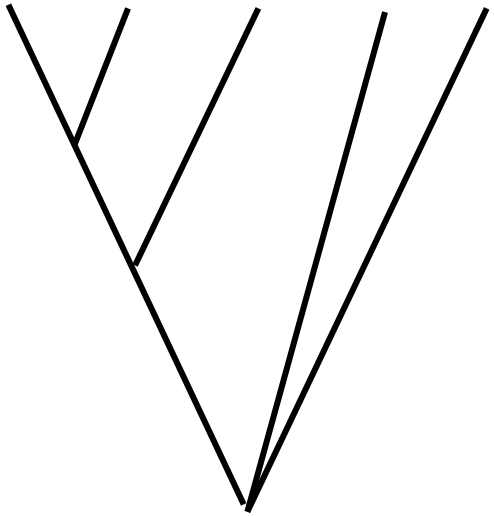

```
((((1 , 2), 3), 4), 5)
```



This is a rooted tree (root is degree = 2)

2. tree file (“Newick” format)


```
(((1 , 2), 3), 4, 5)
```



This is an **unrooted** tree (basal node is degree = 3)

Running PAML programs

1. Sequence data file
2. Tree file
3. **Control file (*.ctl)**

jpbielawski — -bash — 98×39

```
     0.083510     1.429014
     0.000010     0.400000
    50.000000   999.000000

Iterating by ming2
Initial: fx=   790.048189
x=  0.08351  1.42901

  1 h-m-p  0.0008 1.5892  53.4319 +CCYCYYCYY

a      0.002851     0.002852     0.002853     0.002852
f    786.714752   786.714671   786.714928   786.714815
        2.850987e-03    0.173056        1.552250        786.714752
        2.851077e-03    0.173059        1.552254        786.715025
        2.851167e-03    0.173062        1.552257        786.714972
        2.851257e-03    0.173064        1.552261        786.714775
        2.851347e-03    0.173067        1.552265        786.715034
        2.851437e-03    0.173070        1.552269        786.714792
        2.851527e-03    0.173073        1.552273        786.714784
        2.851617e-03    0.173076        1.552277        786.714819
        2.851707e-03    0.173079        1.552281        786.714959
        2.851797e-03    0.173081        1.552285        786.714638
        2.851887e-03    0.173084        1.552289        786.714695
        2.851977e-03    0.173087        1.552292        786.714803
        2.852067e-03    0.173090        1.552296        786.714769
        2.852157e-03    0.173093        1.552300        786.714804
        2.852247e-03    0.173095        1.552304        786.714764
        2.852337e-03    0.173098        1.552308        786.715002
        2.852427e-03    0.173101        1.552312        786.714815
        2.852517e-03    0.173104        1.552316        786.714900
        2.852607e-03    0.173107        1.552320        786.714754
        2.852697e-03    0.173110        1.552324        786.714922
Linesearch2 a4: multiple optima?
C   786.714671 10 0.0029    41 | 0/2
  2 h-m-p  0.0050 0.2387  30.7213 ------------..  | 0/2
  3 h-m-p  0.0000 0.0081 142.5083 -------- | 0/2
  4 h-m-p  0.0002 0.1084   2.2204 ++C     786.707806  0 0.0035    76 | 0/2
  5 h-m-p  0.0160 8.0000   1.9177 +CCYCY
```

3. codeml.ctl (the infamous "control file")

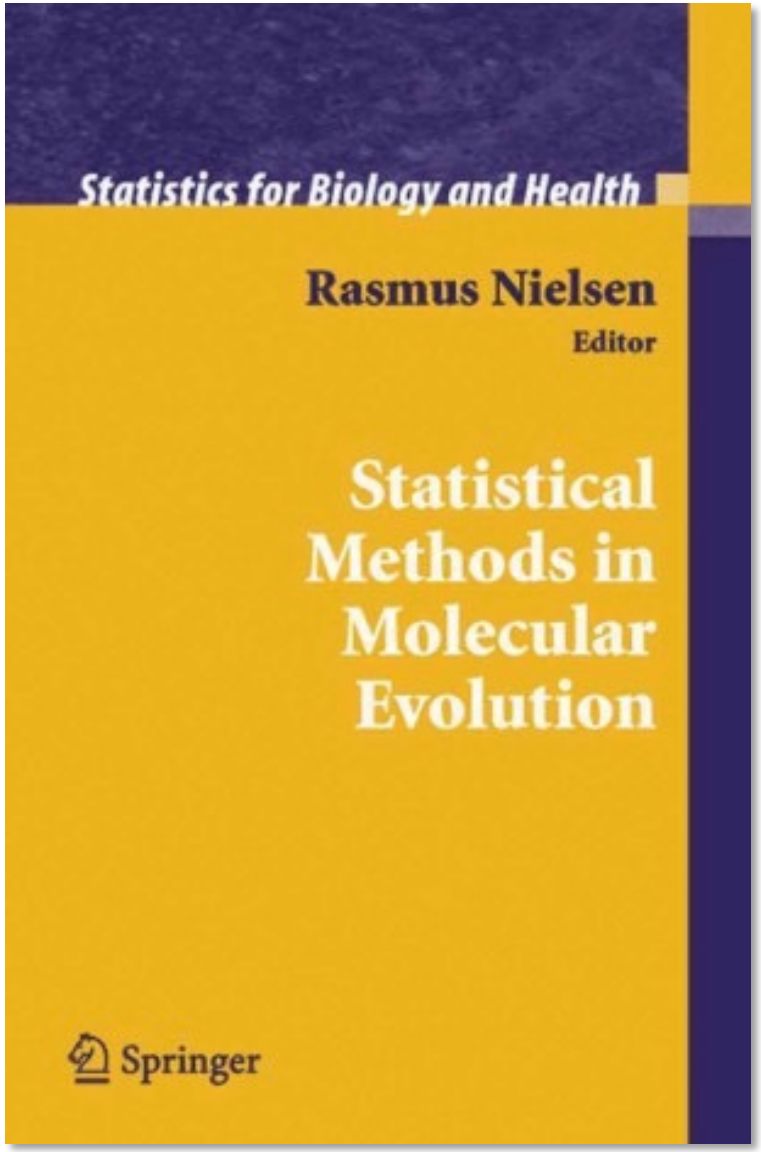
```
  seqfile = seqfile.txt   * sequence data filename
 treefile = tree.txt      * tree structure file name
  outfile = results.txt   * main result file name

    noisy = 9      * 0,1,2,3,9: how much rubbish on the screen
  verbose = 1      * 1:detailed output
  runmode = 0      * 0:user defined tree

  seqtype = 1      * 1:codons
CodonFreq = 2      * 0:equal, 1:F1X4, 2:F3X4, 3:F61

    model = 0      * 0:one omega ratio for all branches

  NSsites = 0      * 0:one omega ratio (M0 in Tables 2 and 4)
                   * 1:neutral (M1 in Tables 2 and 4)
                   * 2:selection (M2 in Tables 2 and 4)
                   * 3:discrete (M3 in Tables 2 and 4)
                   * 7:beta (M7 in Tables 2 and 4)
                   * 8:beta&w; (M8 in Tables 2 and 4)

    icode = 0      * 0:universal code

fix_kappa = 0      * 1:kappa fixed, 0:kappa to be estimated
    kappa = 2      * initial or fixed kappa

fix_omega = 0      * 1:omega fixed, 0:omega to be estimated
    omega = 5      * initial omega

                   *set ncatG for models M3, M7, and M8!!!
   *ncatG = 3      * # of site categories for M3 in Table 4
   *ncatG = 10     * # of site categories for M7 and M8 in Table 4
```

IMPORTANT NOTES:

1. Don't use exercise .ctl files for real data analysis (*they have been modified a little*).

2. Don't use your friends .ctl file for your analysis (*even if he claims it's set up correctly)*

3. The PAML lab



5

Maximum Likelihood Methods for Detecting Adaptive Protein Evolution

Joseph P. Bielawski[1] and Ziheng Yang[2]

[1] Department of Biology, Dalhousie University, Halifax, Nova Scotia B3H 4J1, Canada, j.bielawski@dal.ca
[2] Department of Biology, University College London, Gower Street, London WC1E 6BT, United Kingdom, z.yang@ucl.ac.uk

5.1 Introduction

Proteins evolve; the genes encoding them undergo mutation, and the evolutionary fate of the new mutation is determined by random genetic drift as well as purifying or positive (Darwinian) selection. The ability to analyze this process was realized in the late 1970s when techniques to measure genetic variation at the sequence level were developed. The arrival of molecular sequence data also intensified the debate concerning the relative importance of neutral drift and positive selection to the process of molecular evolution [17]. Ever since, there has been considerable interest in documenting cases of molecular adaptation. Despite a spectacular increase in the amount of available nucleotide sequence data since the 1970s, the number of such well-established cases is still relatively small [9, 38]. This is largely due to the difficulty in developing powerful statistical tests for adaptive molecular evolution. Although several powerful tests for nonneutral evolution have been developed [33], significant results under such tests do not necessarily indicate evolution by positive selection.

A powerful approach to detecting molecular evolution by positive selection derives from comparison of the relative rates of synonymous and nonsynonymous substitutions [22]. Synonymous mutations do not change the amino acid sequence; hence their substitution rate (d_S) is neutral with respect to selective pressure on the protein product of a gene. Nonsynonymous mutations do change the amino acid sequence, so their substitution rate (d_N) is a function of selective pressure on the protein. The ratio of these rates ($\omega = d_N/d_S$) is a measure of selective pressure. For example, if nonsynonymous mutations are deleterious, purifying selection will reduce their fixation rate and d_N/d_S will be less than 1, whereas if nonsynonymous mutations are advantageous, they will be fixed at a higher rate than synonymous mutations, and d_N/d_S will be greater than 1. A d_N/d_S ratio equal to one is consistent with neutral evolution.

Accessing the files

1. If you are doing the lab AT THE WORKSHOP: On the virtual machines we will be using in the 2022 workshop, there will be a symlink in your home directory named "moledata" that takes you to the course data files. There you will find directories for the various labs (e.g., MSAlab, revbayes, PamlLab, etc.).

To view the list of labs type:

```
ls ~/moledata
```

To view the contents of the Paml Lab type:

```
ls -l ~/moledata/PamlLab
```

This will reveal the directories for each excercise:

```
ex1
ex2
ex3
ex4
```

The files are already on the virtual machine you are using. *However, you will want to run each exercise in a separate directory that you will create*. So, create a new directory. The name of the new directory is up to you, but pick something informative (e.g., ~/PAML_ex1).

To *copy* the files required for exercise 1 just type:

```
cp ~/moledata/PamlLab/ex1/* ~/PAML_ex1
```

Now you are ready to do exercise 1 within `~/PAML_ex1`

2. If you are doing the lab INDEPNDENTLY of the workshop: You can do this lab by downloading all the necessary files from an archive here, or you can download the files individually for each exercise as you need them here.

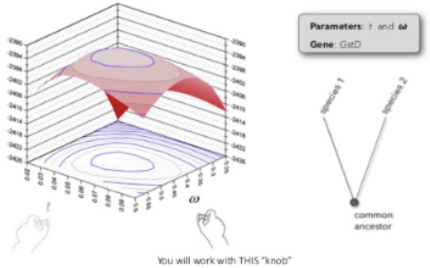
Either way, it is still *"best practice" to run each exercise in a separate working directory that you will create* (*e.g.*, `PAML_ex1`), *and work with copies of the required files within that directory*.

Step-by-step protocols

awarnach.mathstat.dal.ca

Exercise 1

The objective of this activity is to use CODEML to evaluate the likelihood of the *GstD1* sequences for a variety of ω values. Plot log-likelihood scores against the values of ω and determine the maximum likelihood estimate of ω. Check your finding by running CODEML's hill-climbing algorithm.



1. Find the input files for Exercise 1 (**ex1_codeml.ctl, seqfile.txt**) and familiarize yourself with them. Pay close attention to the contents of the modified control file called **ex1_codeml.ctl**.
2. Remember to create a directory where you want your results to go, and place all your files within it. Now open a terminal, move to the directory that contains your files. When you are ready to run CODEML, delete the **ex1_** prefix (the control file must be called **codeml.ctl**). Now you can run CODEML.
3. Familiarize yourself with the results (see annotations in ex1_HelpFile.pdf). If you have not edited the control file the results will be written to a file called **results.txt**. Identify the line within the results file that gives the likelihood score for the example dataset.
4. Now *change and save* the control file and re-run CODEML for a different fixed value of ω. The control file "quick guide" might be helpful here (quick guide). The objective is to compute the likelihood of the example dataset given a fixed value of ω. *Change the control file as follows*:
   - Change the name of your result file (via `outfile=` in the control file) or you will overwrite your previous results!

results "help-files"

Exercise 1 help file: This file contains an annotated portion of the results output by codeml for a maximum likelihood analysis of a pair of sequences. The box contains the portion of the results file that is most relevant to completing exercise 1. These lines of the output can be found at the end of the results file.

```
.
.
.

pairwise comparison, codon frequencies: Fcodon.


2 (Sim) ... 1 (Mel)
lnL = -786.354023
  0.17748  2.24589

t= 0.1775  S=    44.6  N=   555.4  dN/dS= 0.0010  dN= 0.0008  dS= 0.7866
```

This line indicates a pairwise comparison. "Sim" and "Mel" are the sequence labels provided in the sequence file. 1 and 2 indicate the order of these sequences in that file.

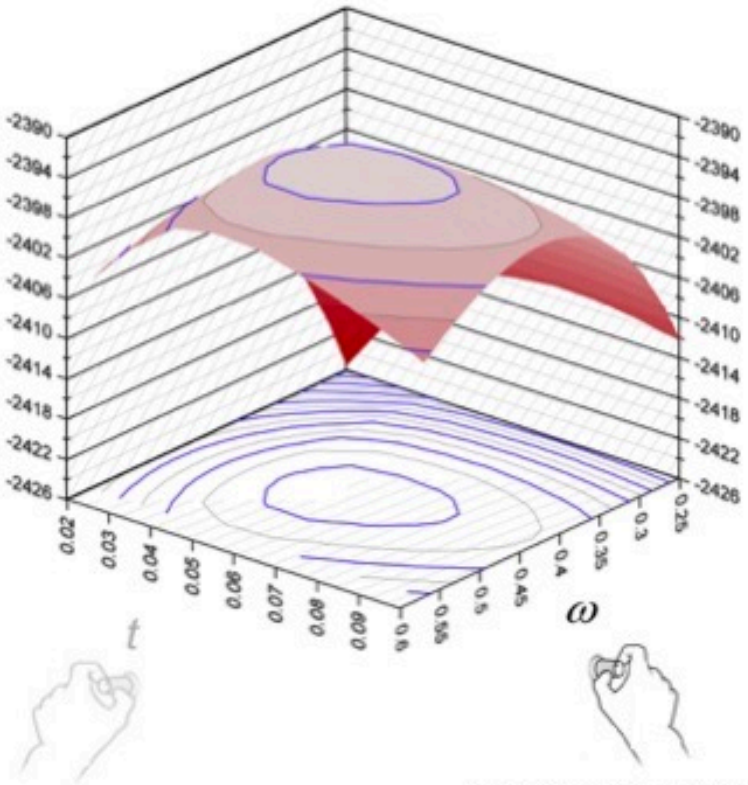
This line gives the log likelihood (ln L) of the pair of sequences

This is the value of ω. In this case it was fixed = 0.001

Exercise 1:

ML estimation of the d_N/d_S (ω) "by hand" for *GstD1*

exercise 1:



Parameters: t and ω

Gene: *GstD*

You will work with THIS "knob"

exercise 1:

```
  seqfile = seqfile.txt          * sequence data filename
  outfile = results_0.001.txt    * main result file name [CHANGE THIS]

    noisy = 9       * 0,1,2,3,9: how much rubbish on the screen
  verbose = 1       * 1:detailed output
  runmode = -2      * -2:pairwise

  seqtype = 1       * 1:codons
CodonFreq = 3       * 0:equal, 1:F1X4, 2:F3X4, 3:F61
    model = 0       *
  NSsites = 0       *
    icode = 0       * 0:universal code

fix_kappa = 0       * 1:kappa fixed, 0:kappa to be estimated
    kappa = 2       * initial or fixed kappa

fix_omega = 1       * 1:omega fixed, 0:omega to be estimated
    omega = 0.001   * 1st fixed omega value  [CHANGE THIS]

   *NOTEs: alternate fixed omega values
   *omega = 0.005   * 2nd fixed value
   *omega = 0.01    * 3rd fixed value
   *omega = 0.05    * 4th fixed value
   *omega = 0.10    * 5th fixed value
   *omega = 0.20    * 6th fixed value
   *omega = 0.40    * 7th fixed value
   *omega = 0.80    * 8th fixed value
   *omega = 1.60    * 9th fixed value
   *omega = 2.00    * 10th fixed value
```

When you are done...

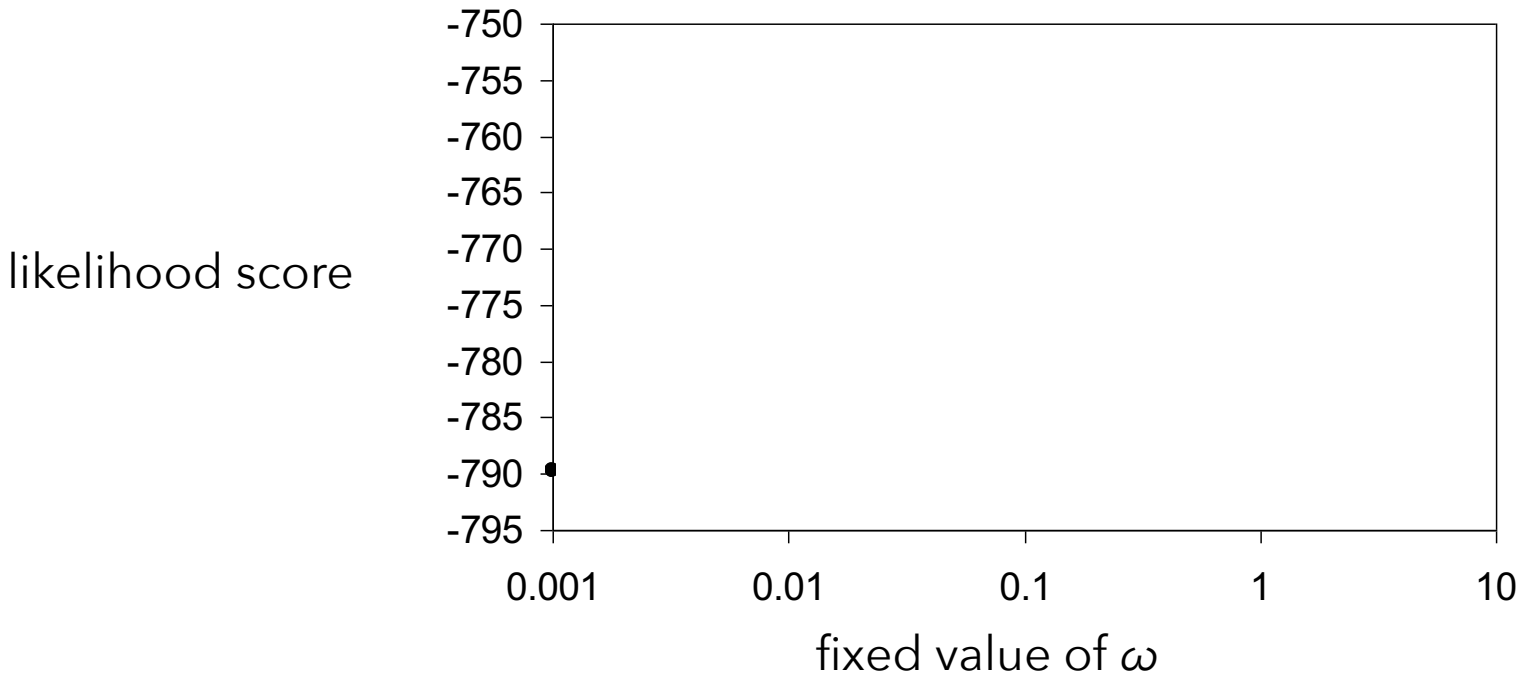
set...

```
fix_omega = 0
omega = 10
```

... now codeml will estimate the MLE for omega

exercise 1:

plot: likelihood score vs. omega (log scale)

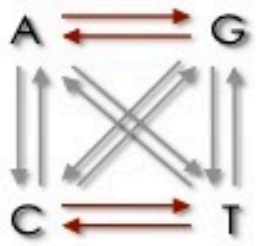


Exercise 2:

Investigating the sensitivity of the d_N/d_S ratio to assumptions in the *GstD1* gene

exercise 2:

transitions vs. **transversions**:



Kappa (**ts**/tv) = 2.71

preferred vs. **un-preferred** codons:

partial codon usage table for the ***GstD* gene of *Drosophila***

Phe F	TTT	0	Ser S	TCT	0	Tyr Y	TAT	1	Cys C	TGT	0
	TTC	27		TCC	15		TAC	22		TGC	6
Leu L	TTA	0		TCA	0	*** *	TAA	0	*** *	TGA	0
	TTG	1		TCG	1		TAG	0	Trp W	TGG	8
Leu L	CTT	2	Pro P	CCT	1	His H	CAT	0	Arg R	CGT	1
	CTC	2		CCC	15		CAC	4		CGC	7
	CTA	0		CCA	3	Gln Q	CAA	0		CGA	0
	CTG	29		CCG	1		CAG	14		CGG	0

exercise 2:

How to model frequencies?

example: A → C

AAA → CAA

AAA → ACA

AAA → AAC

	Δ at codon position		
	1st	2nd	3rd
GY (F61)	π_{CAA}	π_{ACA}	π_{AAC}
MG	π_c^1	π_c^2	π_c^3

Either way, these are **empirically estimated.**

exercise 2:

Example: A → C

AAA → CAA

AAA → ACA

AAA → AAC

	Target codon (nucleotide)			
	CAA	**ACA**	**AAC**	**NP**
No bias	1/61	1/61	1/61	0
F3×4 (GY)	$\pi_C^1 \pi_A^2 \pi_A^3$	$\pi_A^1 \pi_C^2 \pi_A^3$	$\pi_A^1 \pi_A^2 \pi_C^3$	9
F61 (GY)	π_{CAA}	π_{ACA}	π_{AAC}	61

NOTE: There are **even more ways** to model frequencies; but these are the only one we will deal with in this lab.

exercise 2:

```
  seqfile = seqfile.txt    * sequence data filename
  outfile = results.txt    * main result file name

    noisy = 9       * 0,1,2,3,9: how much rubbish on the screen
  verbose = 1       * 1:detailed output
  runmode = -2      * -2:pairwise

  seqtype = 1       * 1:codons
CodonFreq = 0       * 0:equal, 1:F1X4, 2:F3X4, 3:F61 [CHANGE THIS]
    model = 0       *
  NSsites = 0       *
    icode = 0       * 0:universal code

fix_kappa = 1       * 1:kappa fixed, 0:kappa to be estimated [CHANGE THIS]
    kappa = 1       * fixed or initial value

fix_omega = 0       * 1:omega fixed, 0:omega to be estimated
    omega = 0.5     * initial omega value
```

exercise 2:

You will evaluate 6 sets of assumptions:

Assumption set 1:	**Codon bias = none;**	**Ts/Tv bias = none**
`Control file..`	`CodonFreq=0;`	`kappa=1;  fix_kappa=1`
Assumption set 2:	**Codon bias = none;**	**Ts/Tv bias = Yes**
`Control file..`	`CodonFreq=0;`	`kappa=1;  fix_kappa=0`
Assumption set 3:	**Codon bias = yes [F3x4];**	**Ts/Tv bias = none**
`Control file..`	`CodonFreq=2;`	`kappa=1;  fix_kappa=1`
Assumption set 4:	**Codon bias = yes [F3x4];**	**Ts/Tv bias = Yes**
`Control file..`	`CodonFreq=2;`	`kappa=1;  fix_kappa=0`
Assumption set 5:	**Codon bias = yes [F61];**	**Ts/Tv bias = none**
`Control file..`	`CodonFreq=3;`	`kappa=1;  fix_kappa=1`
Assumption set 6:	**Codon bias = yes [F61];**	**Ts/Tv bias = Yes**
`Control file..`	`CodonFreq=3;`	`kappa=1;  fix_kappa=0`

exercise 2:

Complete this table **AND Interpret your findings**

Table E2: Estimation of d_S and d_N between *Drosophila melanogaster* and *D. simulans GstD1* genes

Assumptions	κ	S	N	d_S	d_N	ω	ℓ
Fequal + $\kappa = 1$	1.0	?	?	?	?	?	?
Fequal + κ = estimated	?	?	?	?	?	?	?
F3×4 + $\kappa = 1$	1.0	?	?	?	?	?	?
F3×4 + κ = estimated	?	?	?	?	?	?	?
F61 + $\kappa = 1$	1.0	?	?	?	?	?	?
F61 + κ = estimated	?	?	?	?	?	?	?

κ = transition/transversion rate ratio

S = number of synonymous sites

N = number of nonsynonymous sites

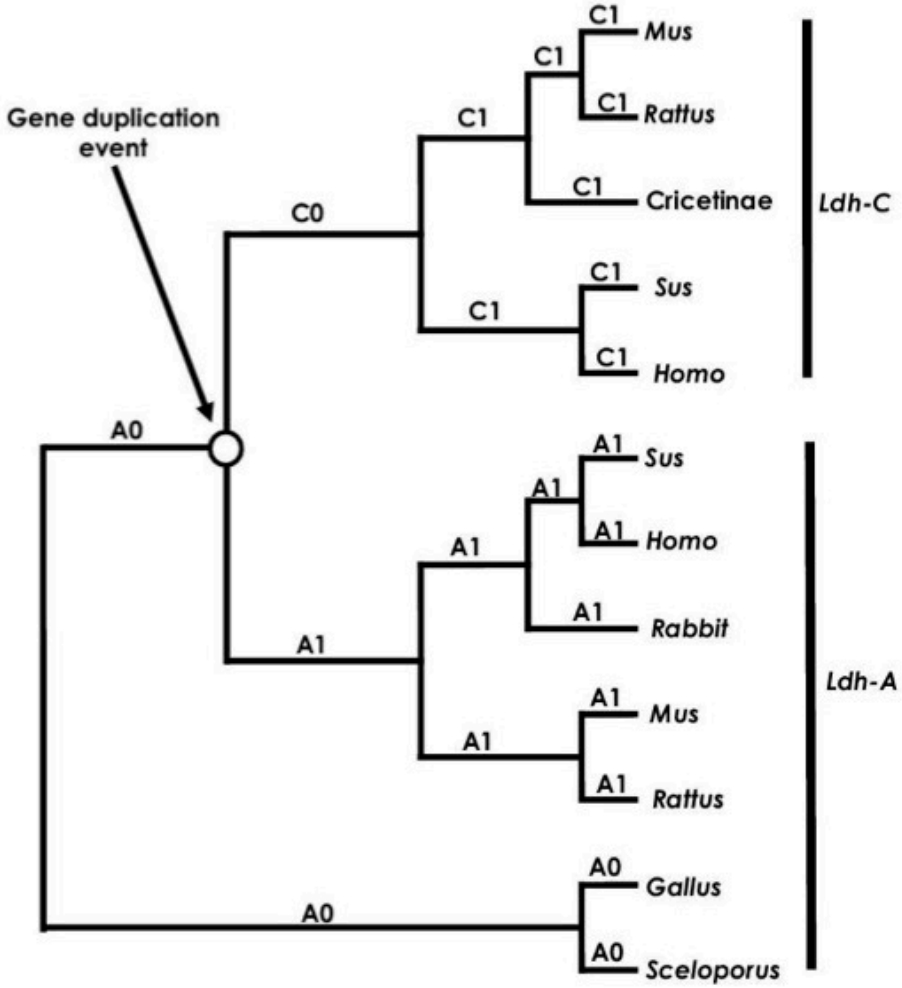
$\omega = d_N/d_S$

ℓ = log likelihood score

Exercise 3:

Test hypotheses about molecular evolution of *Ldh* gene family

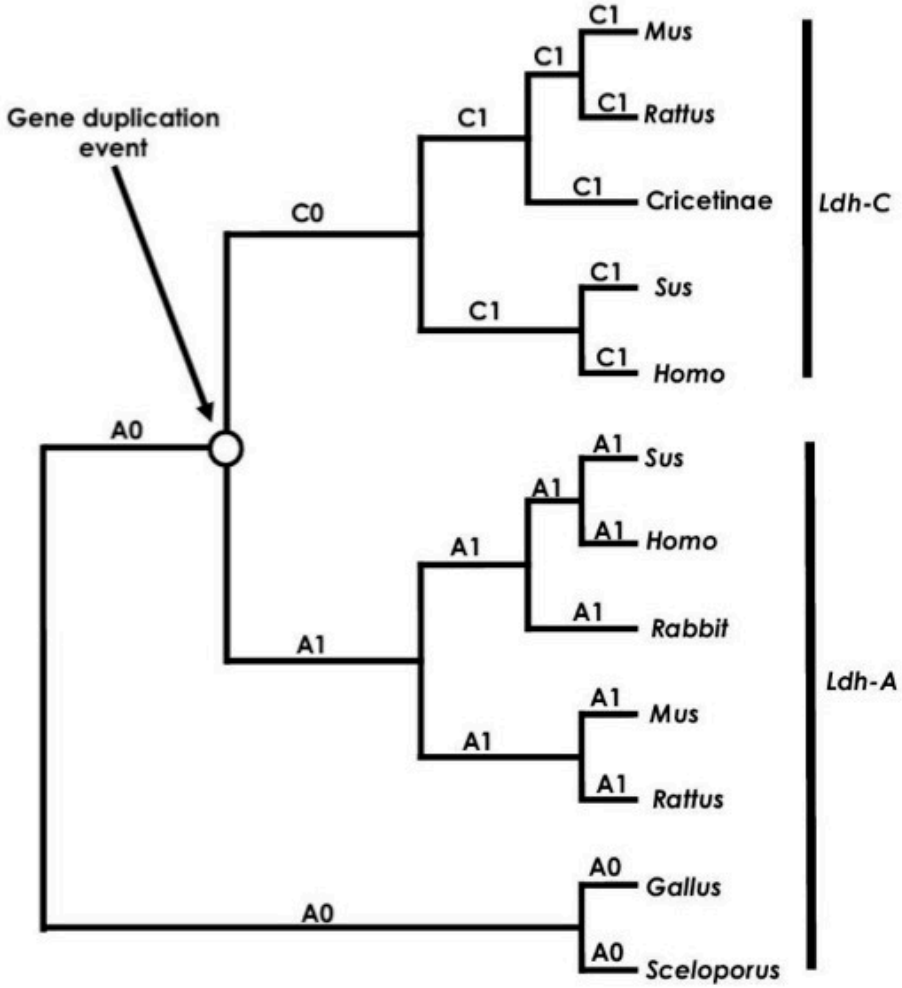
exercise 3:



Each one represents a different "branch model"

$$\begin{cases} H_0: \omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0} \\ H_1: \omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0} \\ H_2: \omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0} \\ H_3: \omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0} \end{cases}$$

exercise 3:



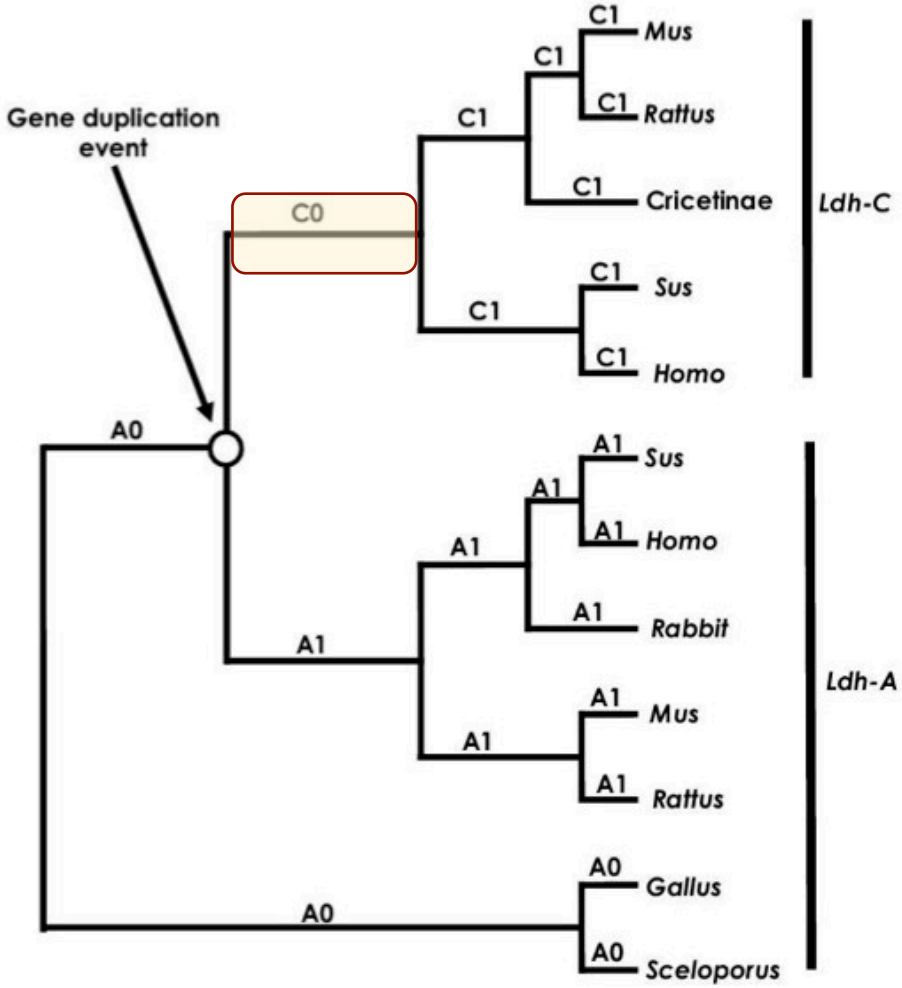
H_0: $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$ Null model

H_1: $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$

H_2: $\omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0}$

H_3: $\omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0}$

exercise 3:



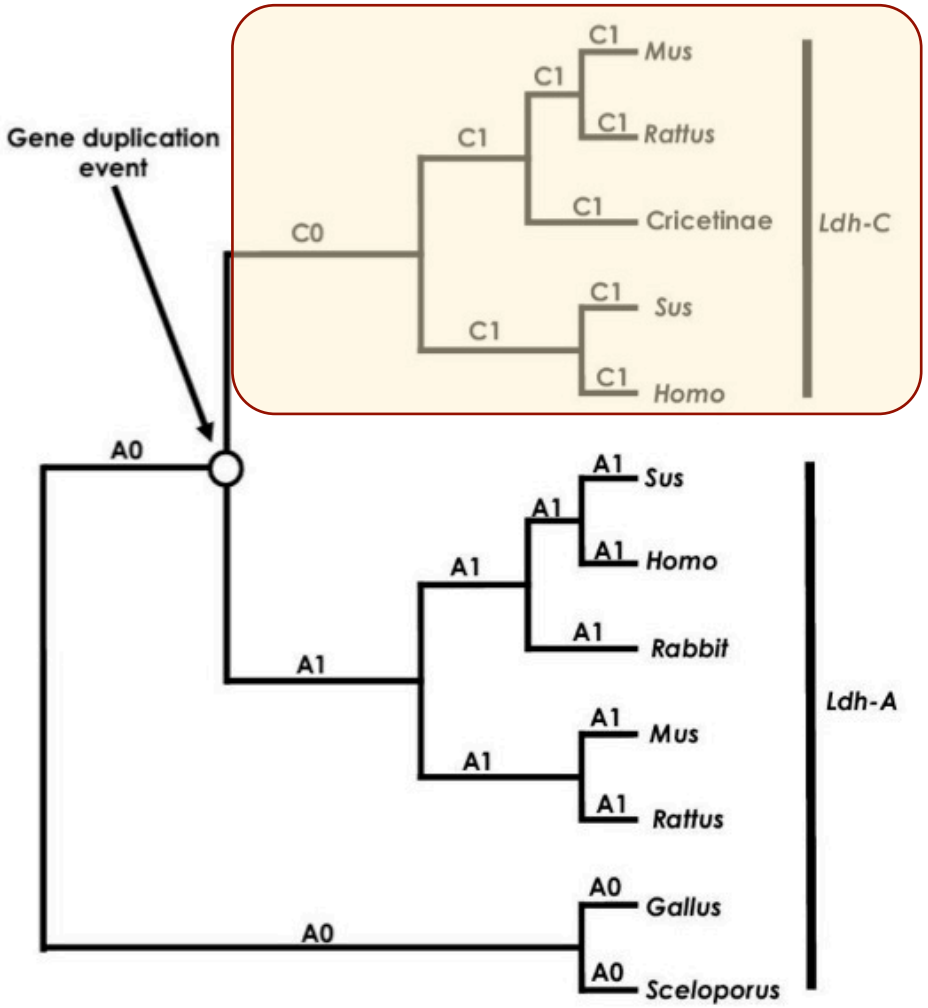
H_0: $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$

H_1: $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$ Episodic model

H_2: $\omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0}$

H_3: $\omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0}$

exercise 3:



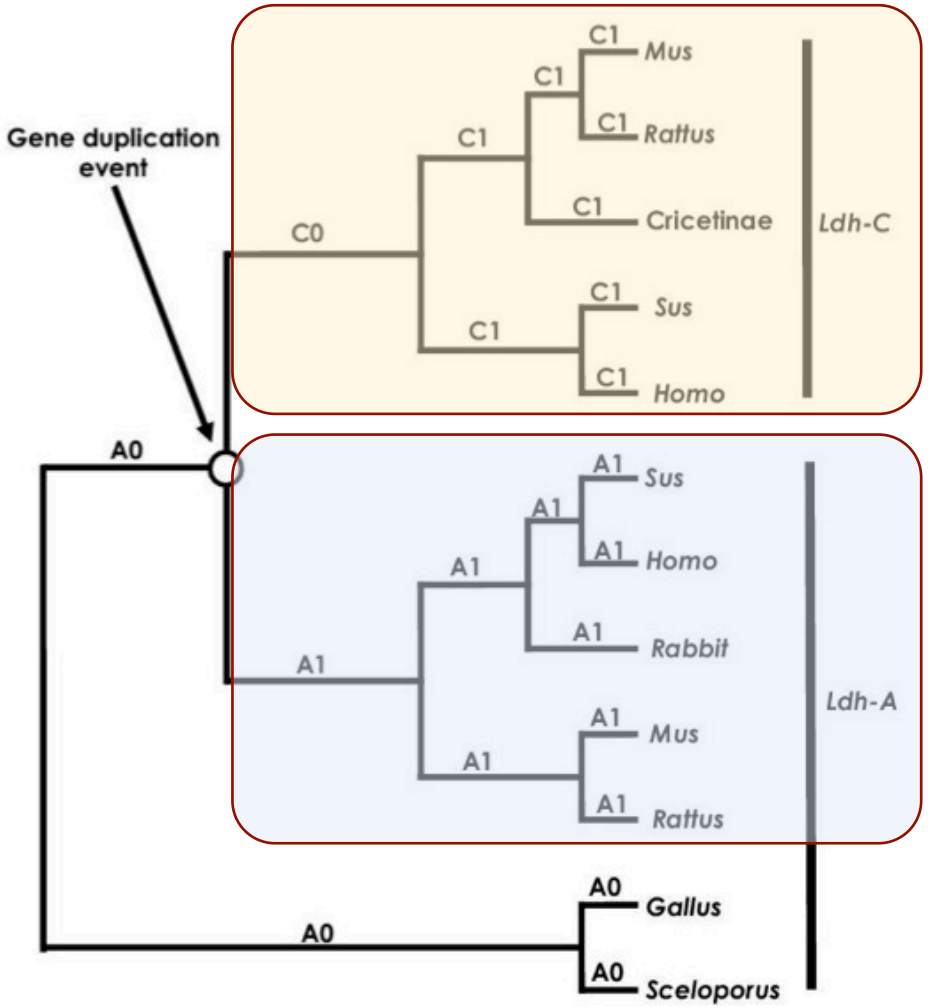
H_0: $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$

H_1: $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$

H_2: $\omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0}$ Long-term shift: 1-clade model

H_3: $\omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0}$

exercise 3:



H_0: $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$

H_1: $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$

H_2: $\omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0}$

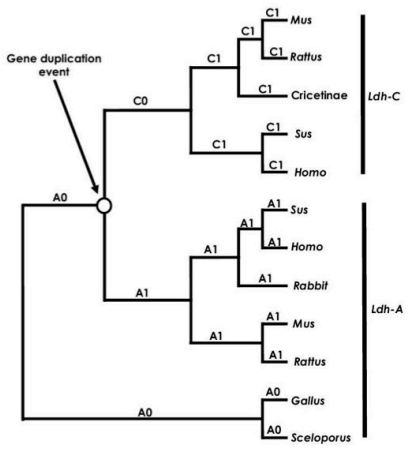
H_3: $\omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0}$

Long-term shift: 2-clade model

exercise 3:

```
  seqfile = seqfile.txt   * sequence data filename
 treefile = tree.H0.txt   * tree structure file name [CHANGE THIS]
  outfile = results.txt   * main result file name

    noisy = 9      * 0,1,2,3,9: how much rubbish on the screen
  verbose = 1      * 1:detailed output
  runmode = 0      * 0:user defined tree

  seqtype = 1      * 1:codons
CodonFreq = 2      * 0:equal, 1:F1X4, 2:F3X4, 3:F61

    model = 0      * 0:one omega ratio for all branches  [FOR MODEL H0]
                   * 1:separate omega for each branch
                   * 2:user specified dN/dS ratios for branches [FOR MODELS H1-H3]

  NSsites = 0      *

    icode = 0      * 0:universal code

fix_kappa = 0      * 1:kappa fixed, 0:kappa to be estimated
    kappa = 2      * initial or fixed kappa

fix_omega = 0      * 1:omega fixed, 0:omega to be estimated
    omega = 0.2    * initial omega
```

```
*H0 in Table 3:
*model = 0
*(X02152Hom,U07178Sus,(M22585rab,((NM017025Rat,U13687Mus),
*(((AF070995C,(X04752Mus,U07177Rat)),(U95378Sus,U13680Hom)),(X53828OG1,
* U28410OG2)))));

*H1 in Table 3:
*model = 2
*(X02152Hom,U07178Sus,(M22585rab,((NM017025Rat,U13687Mus),(((AF070995C,
*(X04752Mus,U07177Rat)),(U95378Sus,U13680Hom))#1,(X53828OG1,U28410OG2))
* )));

*H2 in Table 3:
*model = 2
* (X02152Hom,U07178Sus,(M22585rab,((NM017025Rat,U13687Mus),(((AF070995C
* #1,(X04752Mus #1,U07177Rat #1)#1)#1,(U95378Sus #1,U13680Hom #1)
* #1)#1,(X53828OG1,U28410OG2)))));

*H3 in Table 3:
*model = 2
* (X02152Hom,U07178Sus,(M22585rab,((NM017025Rat,U13687Mus),(((AF070995C
* #1,(X04752Mus #1,U07177Rat #1)#1)#1,(U95378Sus #1,U13680Hom #1)
* #1)#1,(X53828OG1 #2,U28410OG2 #2)#2))));
```



H_0: $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$
H_1: $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$
H_2: $\omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0}$
H_3: $\omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0}$

exercise 3:

Complete this table **AND Interpret your findings**

Table E3: Parameter estimates under models of variable ω ratios among lineages and LRTs of their fit to the *Ldh-A* and *Ldh-C* gene family.

Models	ω_{A0}	ω_{A1}	ω_{C1}	ω_{C0}	ℓ	LRT
H_0: $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$	?	$= \omega_{A.0}$	$= \omega_{A.0}$	$= \omega_{A.0}$	?	na
H_1: $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$	?	$= \omega_{A.0}$	$= \omega_{A.0}$	?	?	?
H_2: $\omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0}$	?	$= \omega_{A.0}$	?	$= \omega_{C.1}$	?	**?**
H_3: $\omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0}$	?	?	?	$= \omega_{C.1}$	?	?

The topology and branch specific ω ratios are presented in Figure 5.
H_0 v H_1: df = 1
H_0 v H_2: df = 1
H_2 v H_3: df = 1

When you interpret your results, THINK about why these involved models are nested.

Exercise 4:

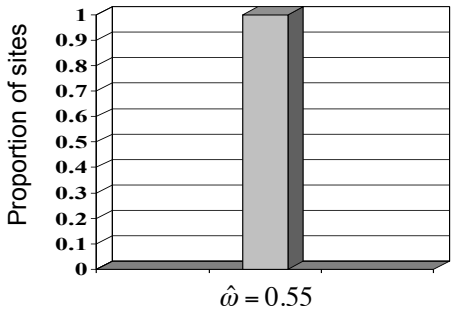
Testing for adaptive evolution in the *nef* gene of human HIV-2

exercise 4:

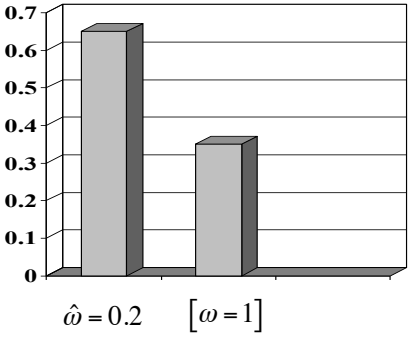
Now recommend a restricted version of M8 for the 3rd LRT (instead of M7)

H_0

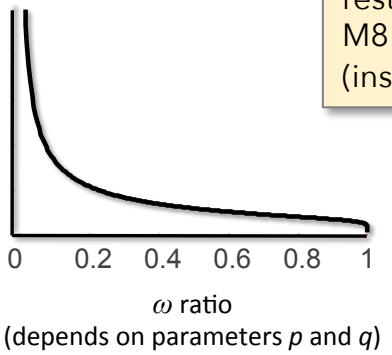
M0



M1a

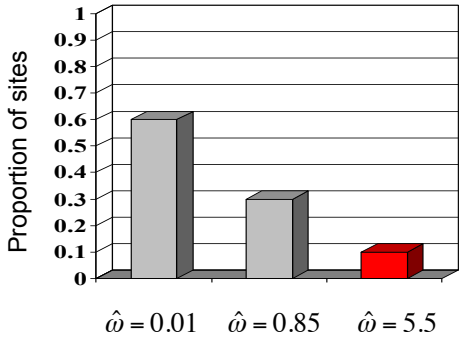


M7

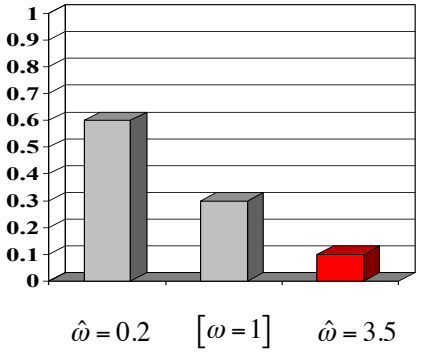


H_a

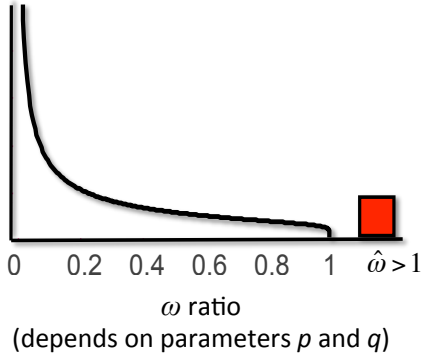
M3



M2a



M8



LRT

1: M0 vs. M3 test for variable selection pressure among sites

2: M1a vs. M2a tests for sites subject to positive selection

3: M7 vs. M8 tests for sites subject to positive selection

```
     seqfile = seqfile.txt          * sequence data filename

   * treefile = treefile_M0.txt     * SET THIS for tree file with ML branch lengths under M0
   * treefile = treefile_M1.txt     * SET THIS for tree file with ML branch lengths under M1
   * treefile = treefile_M2.txt     * SET THIS for tree file with ML branch lengths under M2
   * treefile = treefile_M3.txt     * SET THIS for tree file with ML branch lengths under M3
   * treefile = treefile_M7.txt     * SET THIS for tree file with ML branch lengths under M7
   * treefile = treefile_M8.txt     * SET THIS for tree file with ML branch lengths under M8

     outfile = results.txt          * main result file name
       noisy = 9                    * lots of rubbish on the screen
     verbose = 1                    * detailed output
     runmode = 0                    * user defined tree
     seqtype = 1                    * codons
   CodonFreq = 2                    * F3X4 for codon ferquencies
       model = 0                    * one omega ratio for all branches

   * NSsites = 0                    * SET THIS for M0
   * NSsites = 1                    * SET THIS for M1
   * NSsites = 2                    * SET THIS for M2
   * NSsites = 3                    * SET THIS for M3
   * NSsites = 7                    * SET THIS for M7
   * NSsites = 8                    * SET THIS for M8

       icode = 0                    * universal code
   fix_kappa = 1                    * kappa fixed
     * kappa = 4.43491               * SET THIS to fix kappa at MLE under M0
     * kappa = 4.39117               * SET THIS to fix kappa at MLE under M1
     * kappa = 5.08964               * SET THIS to fix kappa at MLE under M2
     * kappa = 4.89033               * SET THIS to fix kappa at MLE under M3
     * kappa = 4.22750               * SET THIS to fix kappa at MLE under M7
     * kappa = 4.87827               * SET THIS to fix kappa at MLE under M8

   fix_omega = 0                    * omega to be estimated
       omega = 5                    * initial omega

     * ncatG = 3                     * SET THIS for 3 site categories under M3
     * ncatG = 10                    * SET THIS for 10 of site categories under M7 and M8

 fix_blength = 2                    * fixed branch lengths from tree file
```

These trees contain **pre-computed MLEs for branch lengths** to speed the analyses.

You will want to estimate all the branch lengths via ML when you analyze your own data!

Be careful: there is a lot to change in this codeml.ctl file for each model.

It is very easy to miss something, or make a mistake

The models will run quick, so it is also easy to check/fix any mistakes.

exercise 4:

Complete this table **AND Interpret your findings**

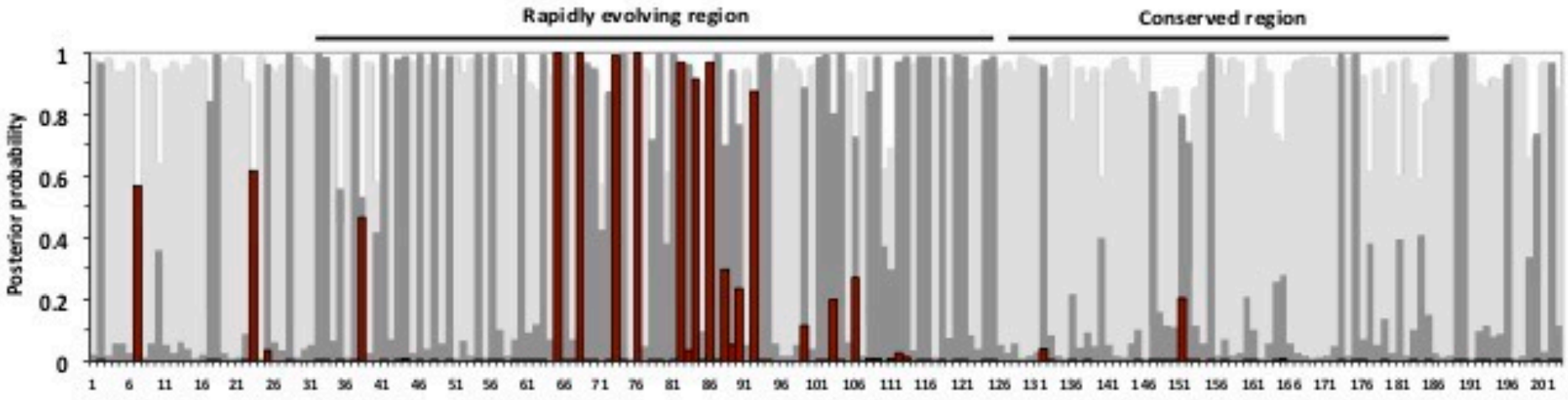
Table E4: Parameter estimates and likelihood scores under models of variable ω ratios among sites for HIV-2 *nef* genes.

Nested model pairs	d_N/d_S [b]	Parameter estimates [c]	PSS [d]	ℓ
M0: one-ratio (1) [a]	?	ω = ?	N.A.	?
M3: discrete (5)	?	p_0, = ?, p_1, = ?, (p_2 = ?) ω_0 = ?, ω_1 = ?, ω_2 = ?	? (?)	?
M1a: neutral (2)	?	p_0 = ?, (p_1 = ?) ω_0 = ?, (ω_1 = 1)	N.A.	?
M2a: selection (4)	?	p_0= ?, p_1 = ?, (p_2 = ?) ω_0 = ?, (ω_1 = 1), ω_2 = ?	? (?)	?
M7: beta (2)	?	p = ?, q = ?	N.A.	?
M8: beta&ω (4)	?	p_0 = ? (p_1 = ?) p = ?, q = ?, ω = ?	? (?)	?

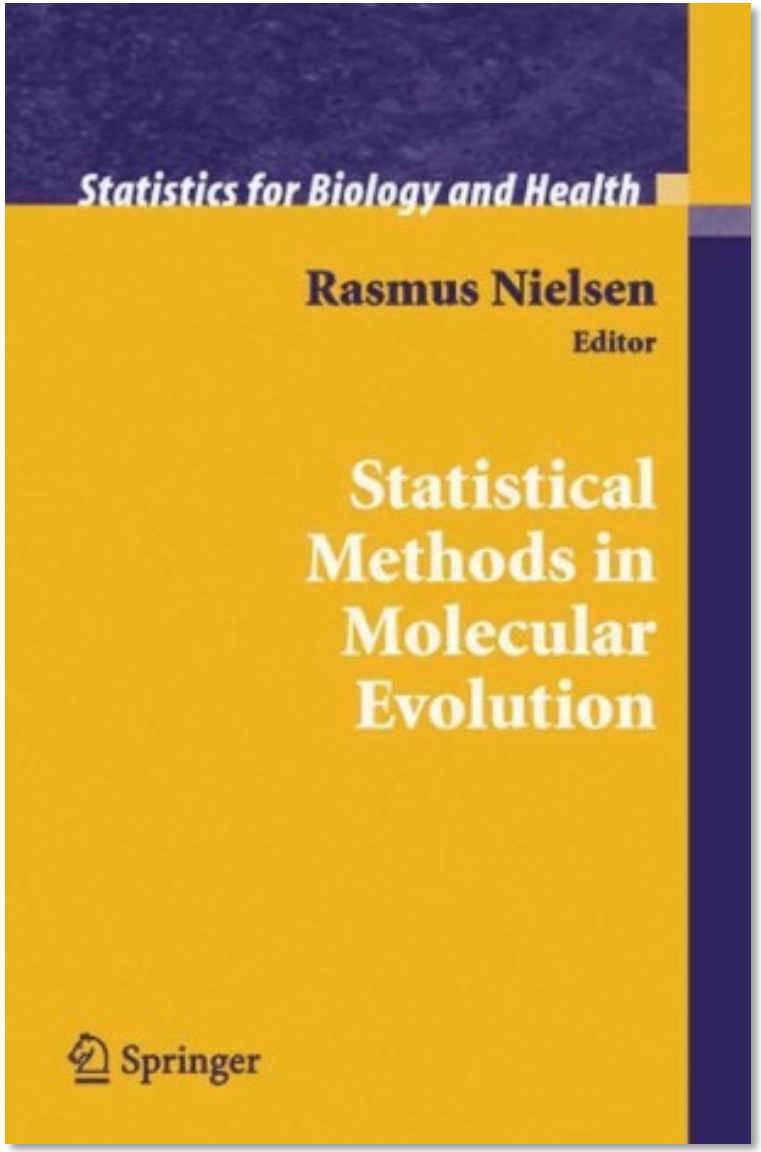
[a] The number after the model code, in parentheses, is the number of free parameters in the ω distribution.
[b] This d_N/d_S ratio is an average over all sites in the HIV-2 *nef* gene alignment.
[c] Parameters in parentheses are not free parameters.
[d] PSS is the number of positive selection sites (NEB). The first number is the PSS with posterior probabilities > 50%. The second number (in parentheses) is the PSS with posterior probabilities > 95%.

exercise 4:

Use the "**rst file**" for model **M3** to produce a plot like this for the *nef* gene



NOTE: This is **NOT** the distribution for the *nef* gene



5

Maximum Likelihood Methods for Detecting Adaptive Protein Evolution

Joseph P. Bielawski[1] and Ziheng Yang[2]

[1] Department of Biology, Dalhousie University, Halifax, Nova Scotia B3H 4J1, Canada, j.bielawski@dal.ca
[2] Department of Biology, University College London, Gower Street, London WC1E 6BT, United Kingdom, z.yang@ucl.ac.uk

5.1 Introduction

Proteins evolve; the genes encoding them undergo mutation, and the evolutionary fate of the new mutation is determined by random genetic drift as well as purifying or positive (Darwinian) selection. The ability to analyze this process was realized in the late 1970s when techniques to measure genetic variation at the sequence level were developed. The arrival of molecular sequence data also intensified the debate concerning the relative importance of neutral drift and positive selection to the process of molecular evolution [17]. Ever since, there has been considerable interest in documenting cases of molecular adaptation. Despite a spectacular increase in the amount of available nucleotide sequence data since the 1970s, the number of such well-established cases is still relatively small [9, 38]. This is largely due to the difficulty in developing powerful statistical tests for adaptive molecular evolution. Although several powerful tests for nonneutral evolution have been developed [33], significant results under such tests do not necessarily indicate evolution by positive selection.

A powerful approach to detecting molecular evolution by positive selection derives from comparison of the relative rates of synonymous and nonsynonymous substitutions [22]. Synonymous mutations do not change the amino acid sequence; hence their substitution rate (d_S) is neutral with respect to selective pressure on the protein product of a gene. Nonsynonymous mutations do change the amino acid sequence, so their substitution rate (d_N) is a function of selective pressure on the protein. The ratio of these rates ($\omega = d_N/d_S$) is a measure of selective pressure. For example, if nonsynonymous mutations are deleterious, purifying selection will reduce their fixation rate and d_N/d_S will be less than 1, whereas if nonsynonymous mutations are advantageous, they will be fixed at a higher rate than synonymous mutations, and d_N/d_S will be greater than 1. A d_N/d_S ratio equal to one is consistent with neutral evolution.

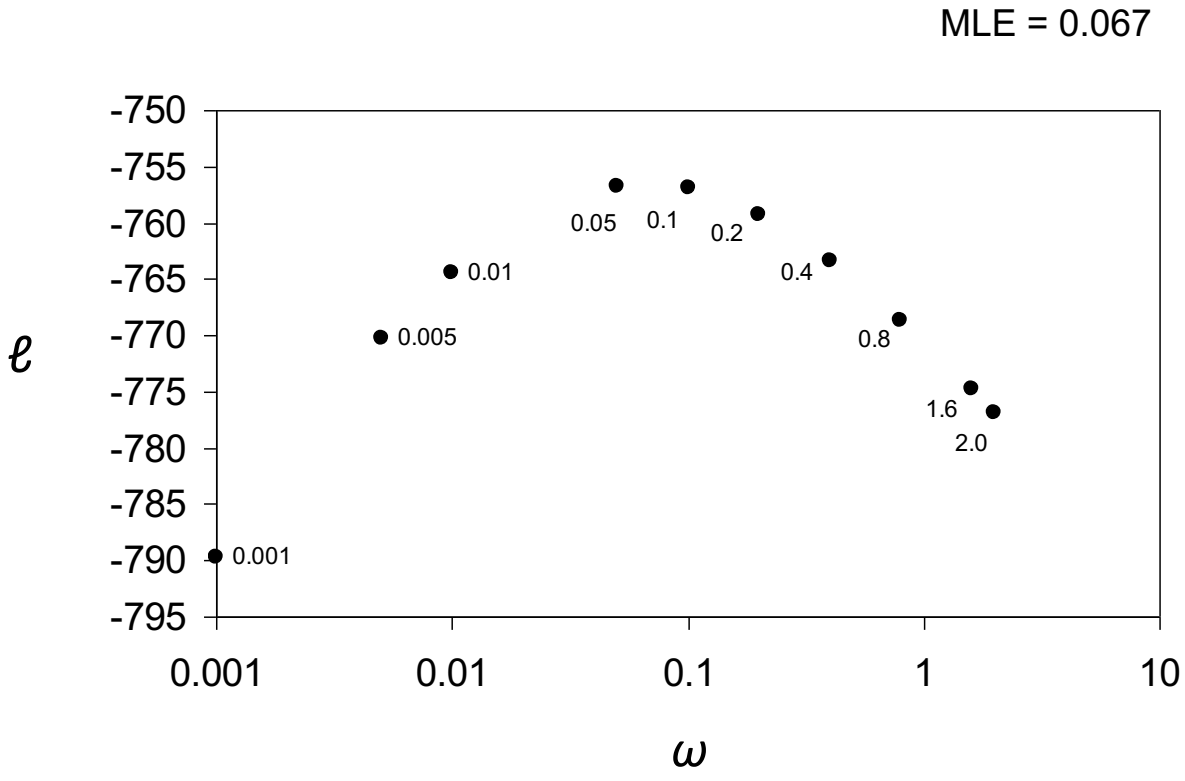
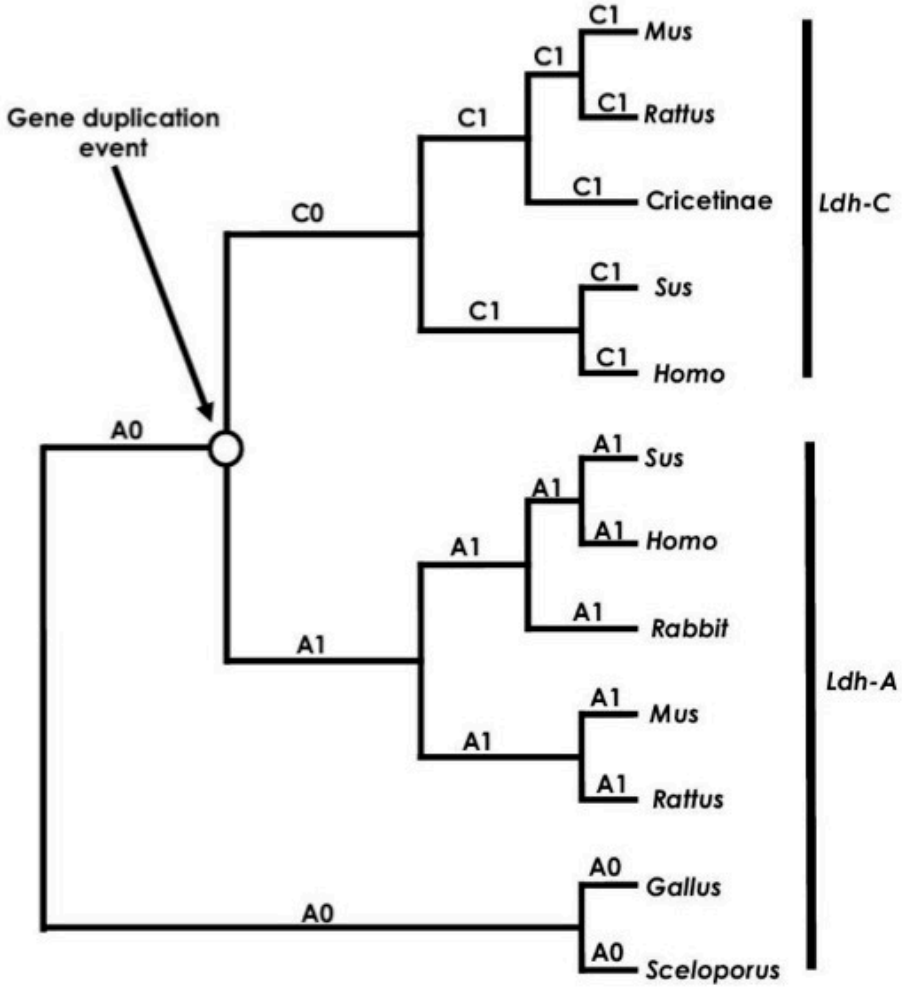

MLE = 0.067
-750
-755
-760
-765
-770
-775
-780
-785
-790
-795
ℓ
0.001
0.005
0.01
0.05
0.1
0.2
0.4
0.8
1.6
2.0
0.001
0.01
0.1
1
10
ω

Table 1. Estimation of d_S and d_N between *Drosophila melanogaster* and *D. simulans GstD1* genes

Asumptions	κ	S	N	d_S	d_N	ω	ℓ
Fequal, $\kappa = 1$	1.0	152.9	447.1	0.0776	0.0213	0.274	-927.18
Fequal, κ = estimated	1.88	165.8	434.2	0.0221	0.0691	0.320	-926.28
F3×4, $\kappa = 1$	1.0	70.6	529.4	0.1605	0.0189	0.118	-844.51
F3×4, κ = estimated	2.71	73.4	526.6	0.1526	0.0193	0.127	-842.21
F61, $\kappa = 1$	1.0	40.5	559.5	0.3198	0.0201	0.063	-758.55
F61, κ = estimated	2.53	45.2	554.8	0.3041	0.0204	0.067	-756.57



H_0: $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$ Null model

H_1: $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$ Episodic model

H_2: $\omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0}$ Long-term shift model (v1)

H_3: $\omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0}$ Long-term shift model (v2)

Parameter estimates under models of variable ω ratios among lineages and LRTs of their fit to the *Ldh-A* and *Ldh-C* gene family.

Models [a]	ω_{A0}	ω_{A1}	ω_{C1}	ω_{C0}	ℓ	LRT
H_0: $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$	0.14	$= \omega_{A.0}$	$= \omega_{A.0}$	$= \omega_{A.0}$	-6018.63	NA
H_1: $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$	0.13	$= \omega_{A.0}$	$= \omega_{A.0}$	0.19	-6017.57	$P = 0.14$ [b]
H_2: $\omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0}$	0.07	$= \omega_{A.0}$	0.24	$= \omega_{C.1}$	-5985.63	$\boldsymbol{P < 0.0001}$ [c]
H_3: $\omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0}$	0.09	0.06	0.24	$= \omega_{C.1}$	-5984.11	$P = 0.08$ [d]

[a] The topology and branch specific ω ratios are presented in Figure 5.

[b] H_0 v H_1: df = 1

[c] H_0 v H_2: df = 1

[d] H_2 v H_3: df = 1

Parameter estimates and likelihood scores under models of variable ω ratios among sites for HIV-2 *nef* genes.

Nested model pairs	d_N/d_S [b]	**Parameter estimates** [c]	**PSS** [d]	ℓ
M0: one-ratio (1) [a]	0.505	$\omega = 0.505$	none	-9775.77
M3: discrete (5)	0.629	$p_0 = 0.48$, $p_1 = 0.39$, ($p_2 = 0.13$) $\omega_0 = 0.03$, $\omega_1 = 0.74$, $\boldsymbol{\omega_2 = 2.50}$	31 (24)	-9232.18
M1: neutral (1)	0.63	$p_0 = 0.37$, ($p_1 = 0.63$) ($\omega_0 = 0$), ($\omega_1 = 1$)	not allowed	-9428.75
M2: selection (3)	0.93	$p_0 = 0.37$, $p_1 = 0.51$, ($p_2 = 0.12$) ($\omega_0 = 0$), ($\omega_1 = 1$), $\boldsymbol{\omega_2 = 3.48}$	30 (22)	-9392.96
M7: beta (2)	0.423	$P = 0.18$, $q = 0.25$	not allowed	-9292.53
M8: beta&ω (4)	0.623	$p_0 = 0.89$, ($p_1 = 0.11$) $p = 0.20$, $q = 0.33$, $\boldsymbol{\omega = 2.62}$	27 (15)	-9224.31

[a] The number after the model code, in parentheses, is the number of free parameters in the ω distribution.

[b] This d_N/d_S ratio is an average over all sites in the HIV-2 *nef* gene alignment.

[c] Parameters in parentheses are not free parameters.

[d] PSS is the number of positive selection sites. The first number is the PSS with posterior probabilities > 50%. The second number, in parentheses, is the PSS with posterior probabilities > 95%.