

PAML (Phylogenetic Analysis by Maximum Likelihood)

A program package by Ziheng Yang
(Demonstration by Joseph Bielawski)

1. Three inference tasks

(relevant to all approaches)

model based inference

3 analytical tasks

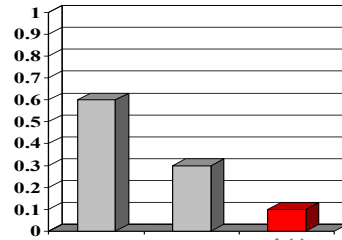
task 1. parameter estimation (e.g., ω)

task 2. hypothesis testing

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

Concept map for tasks 1-3...

model:
5% have $\omega > 1$



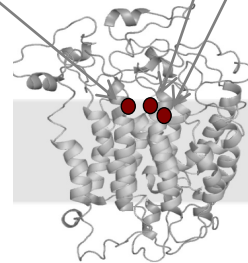
1. Fit model to data \rightarrow MLEs
2. Test hypotheses via Null and Alternative models for ω

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..	..TGC	A..
...C	..T	A..	...	A.TAA	...	A.C	...	AGC	...
...	..C	...	G.A	..ATA	A..	...	AA.	TG.G	...	A..	..T	..GC	..T
...	..C	..G	GA.	..TT	C..	..G	..A	...	AT.TG	..A	..GC	...

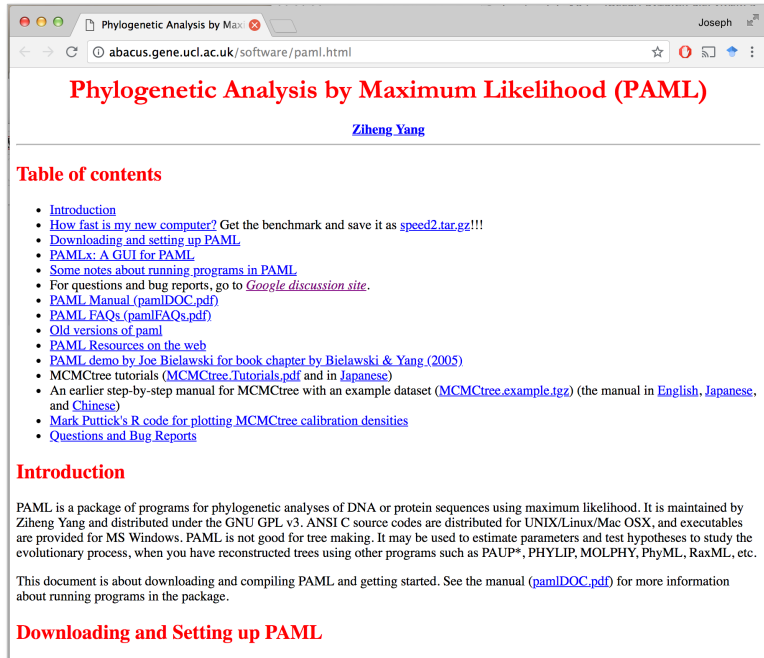
3. Predict which sites have $\omega > 1$

structure:
sites are in contact



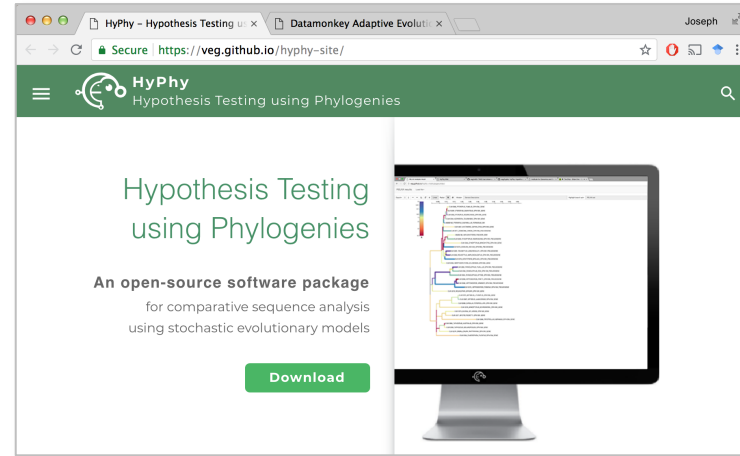
4. Interpret results in known biological context

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



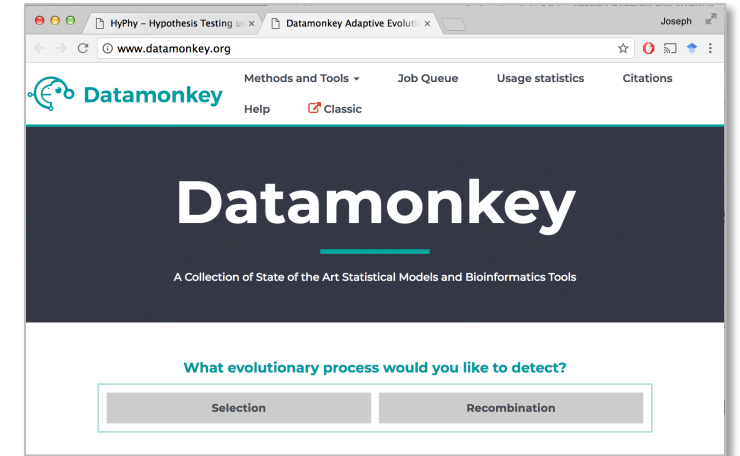
The screenshot shows the PAML website at <http://abacus.gene.ucl.ac.uk/software/paml.html>. The page title is "Phylogenetic Analysis by Maximum Likelihood (PAML)" by Ziheng Yang. It features a "Table of contents" section with links to various resources such as "Introduction", "How fast is my new computer?", "Downloading and setting up PAML", "PAMLx: A GUI for PAML", "Some notes about running programs in PAML", "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)", "Mark Puttick's R code for plotting MCMCtree calibration densities", and "Questions and Bug Reports". There is also an "Introduction" section and a "Downloading and Setting up PAML" section.

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



The screenshot shows the HyPhy website at <https://veg.github.io/hyphy-site/>. The page title is "HyPhy - Hypothesis Testing using Phylogenies". It features a "Download" button and a description: "An open-source software package for comparative sequence analysis using stochastic evolutionary models". The page also includes a search bar and a navigation menu.

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



The screenshot shows the Datamonkey website at <http://www.datamonkey.org/>. The page title is "Datamonkey" and it features a "Download" button. The page also includes a search bar and a navigation menu. Below the main heading, there is a section titled "What evolutionary process would you like to detect?" with two buttons: "Selection" and "Recombination".

<http://www.datamonkey.org/>

NOTE: We are actively developing codon models in IQTree (2026)

Objective: To gain a deeper understanding of the basic principles of *model-based inference* in general.

We are NOT trying to teach a particular software package.

Engage with the concept questions. It is more important to understand what you are doing (compared to knowing a particular software package).

YOU must attempt to understand the relationship between your model and your data.

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 — 98x39
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 +CCYCYCYCY

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)

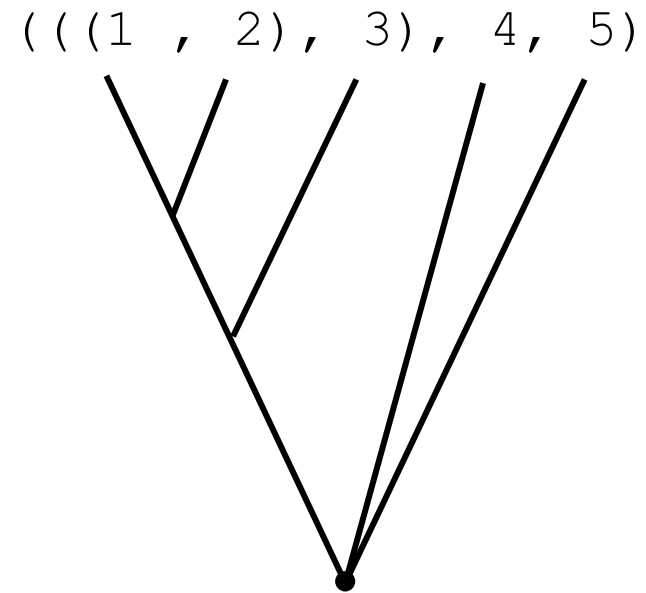
```
4 20
sequence_1 TCATT CTATC TATCG TGATG
sequence_2 TCATT CTATC TATCG TGATG
sequence_3 TCATT CTATC TATCG TGATG
sequence_4 TCATT CTATC TATCG TGATG
```



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



2. tree file ("Newick" format)



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

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

Statistics for Biology and Health

Rasmus Nielsen
Editor

Statistical Methods in Molecular Evolution

 Springer

5

Maximum Likelihood Methods for Detecting Adaptive Protein Evolution

Joseph P. Bielawski¹ and Ziheng Yang²

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.

The image shows a web browser window with the address bar displaying 'awarnach.mathstat.dal.ca'. The page content is as follows:

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 exercise:

```
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`

This is STEP 1

2. If you are doing the lab INDEPENDENTLY 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.

Re-naming files: 2 important points...

1. For each exercise you must **remove the "exN_" prefix** from the control files

for example: `cp ex1_codem1.ct1 codem1.ct1`

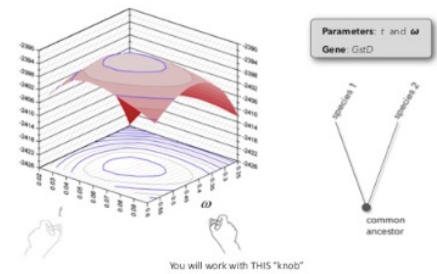
2. PAML will overwrite its own outfiles without warning you!!! **Rename any outfiles you want to save!!!**

Step-by-step protocols

results "help-files"

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.cti**, **seqfile.txt**) and familiarize yourself with them. Pay close attention to the contents of the modified control file called **ex1_codeml.cti**.
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.cti**). 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!

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

In 2026...

- **exercises 1 and 3** we will do together
- **exercise 4** we will be self-paced (step 8 is optional)
- we will SKIP exercise 2

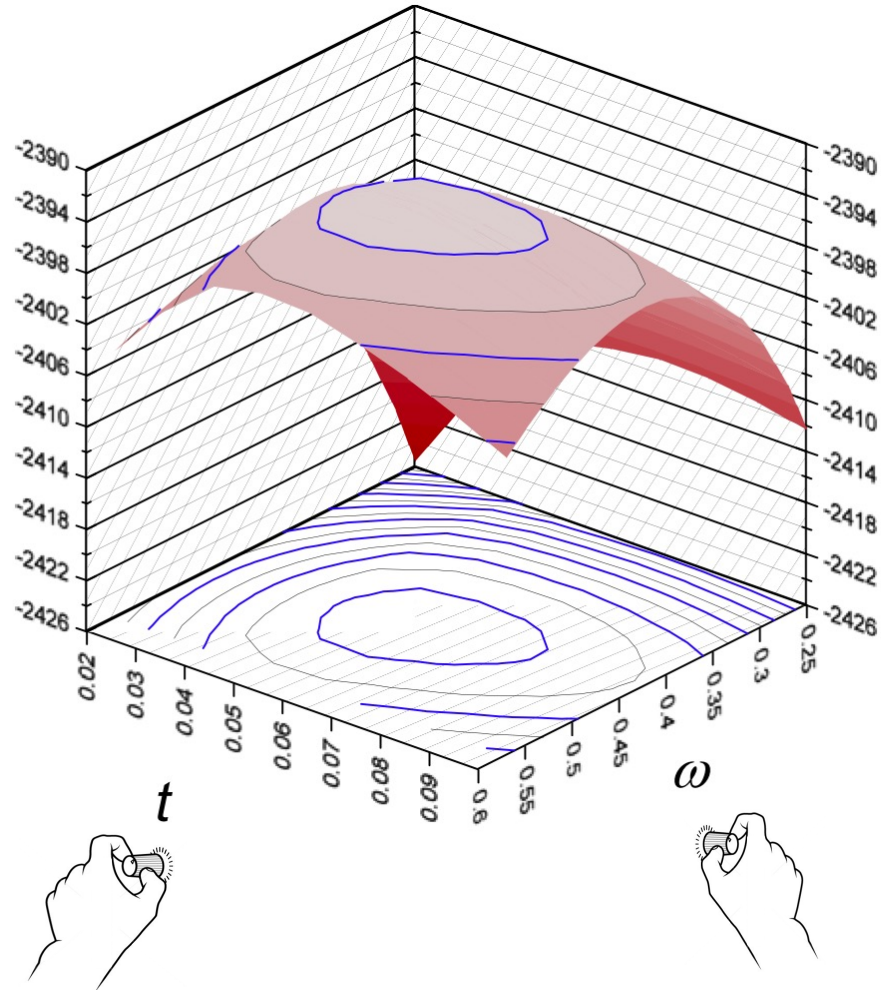


Work in teams and discuss your progress!!!

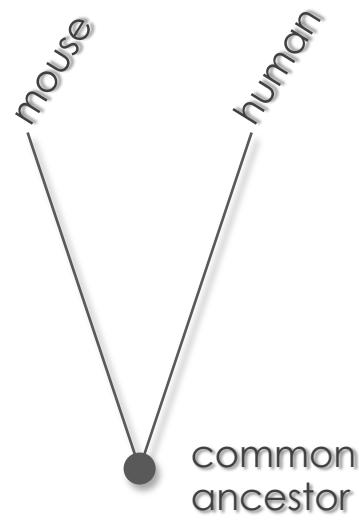
Exercise 1:

ML estimation of the $d_N/d_S(\omega)$ "by hand" for *GstD1*

exercise 1:



Parameters: t and ω
Gene: acetylcholine α receptor



$\ln L = -2399$

exercise 1:
you will work 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

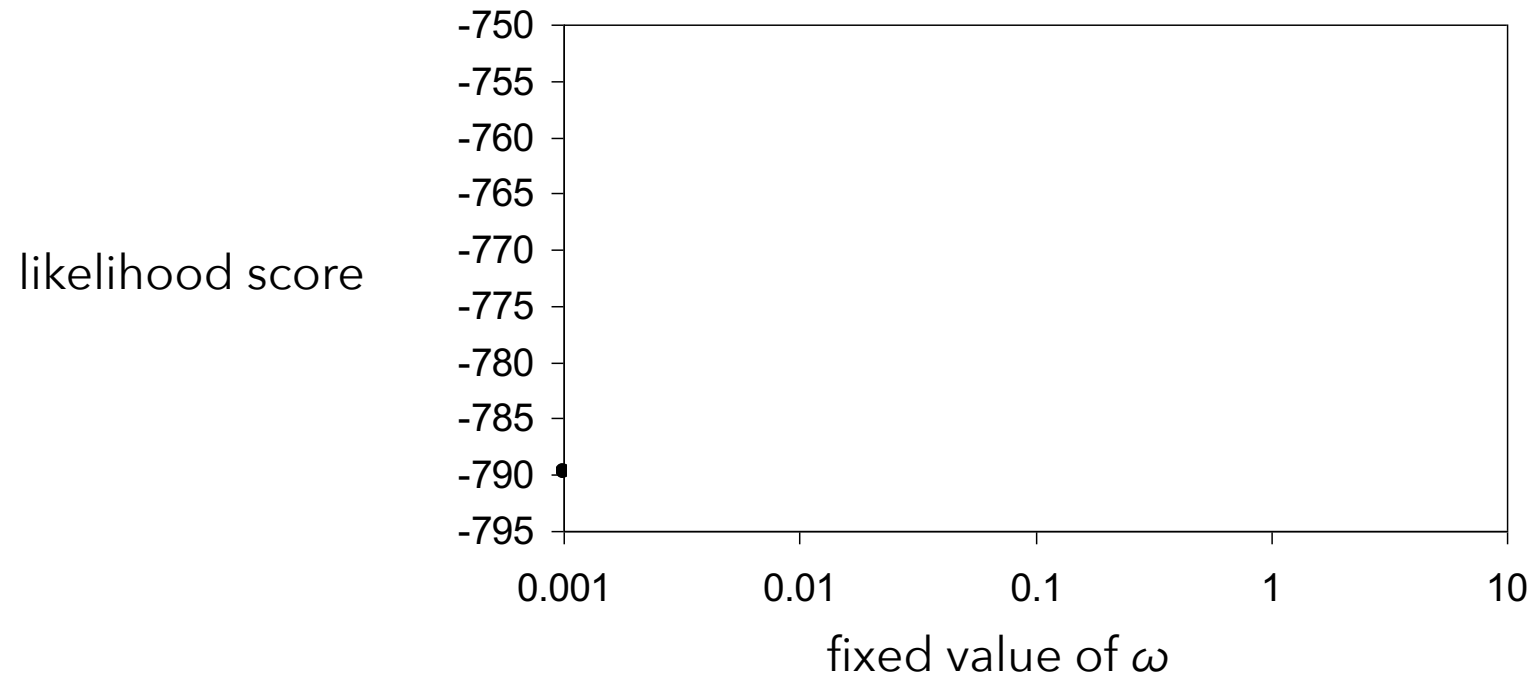
fix_kappa = 0  * 1:kappa fixed, 0:kappa to be estimated
kappa = 2     * initial or fixed kappa

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
```

exercise 1:

plot: likelihood score vs. omega (log scale)



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 concept questions:

1. How close was your “by-hand” estimate of the MLE compared to the one produced by the codeml optimization algorithm?
2. What does the MLE tell you about the evolution of this gene?
3. Does the area under the curve sum to 1.0?
4. Can you explain, *in non-technical language*, what the MLE represents and why you would want to estimate it?

Exercise 3:

Test hypotheses about molecular evolution of *Ldh* gene family

Likelihood Ratio Test (LRT)

Does the more complex model explain the data *significantly better* than the simpler model?

1

fit two nested models



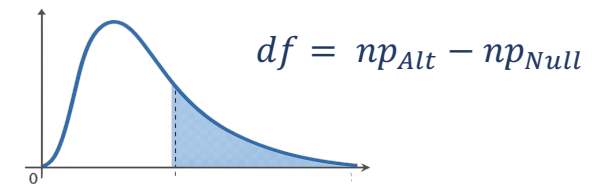
2

compare likelihoods \rightarrow compute test statistic

$$\text{LRT} = 2(\ln L_{\text{Alt}} - \ln L_{\text{Null}})$$

3

assess significance

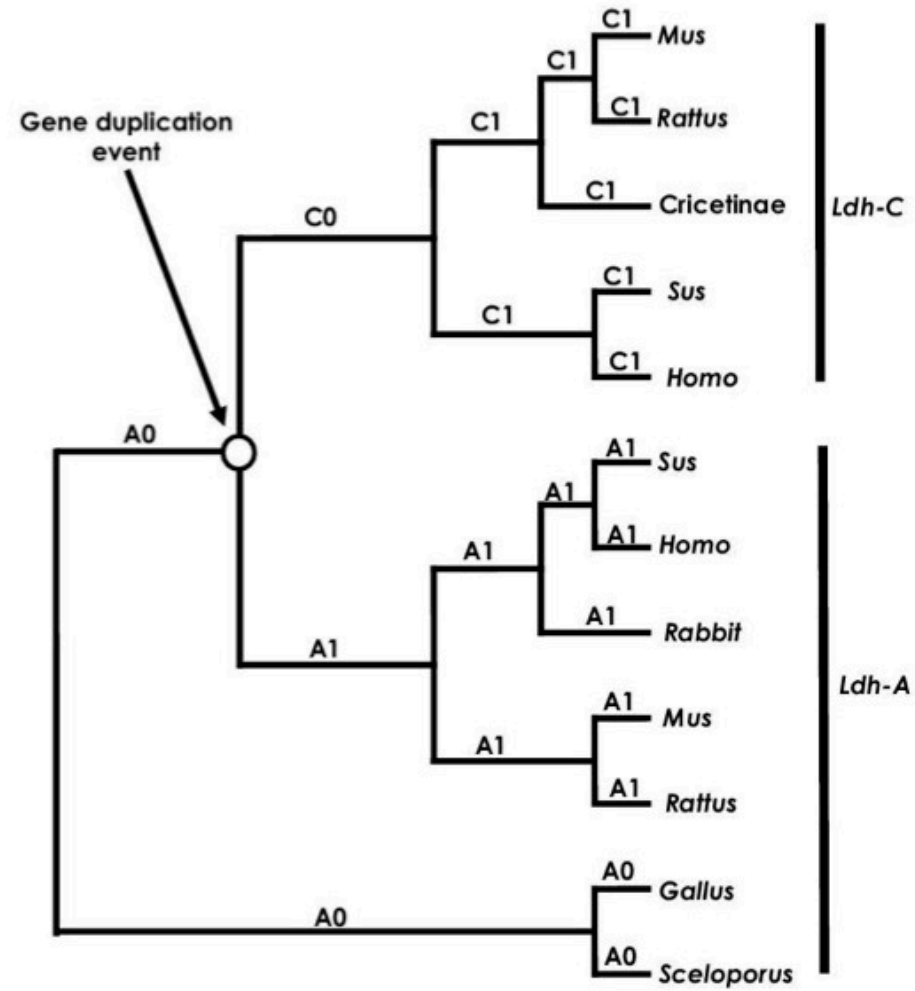


4

use your brain

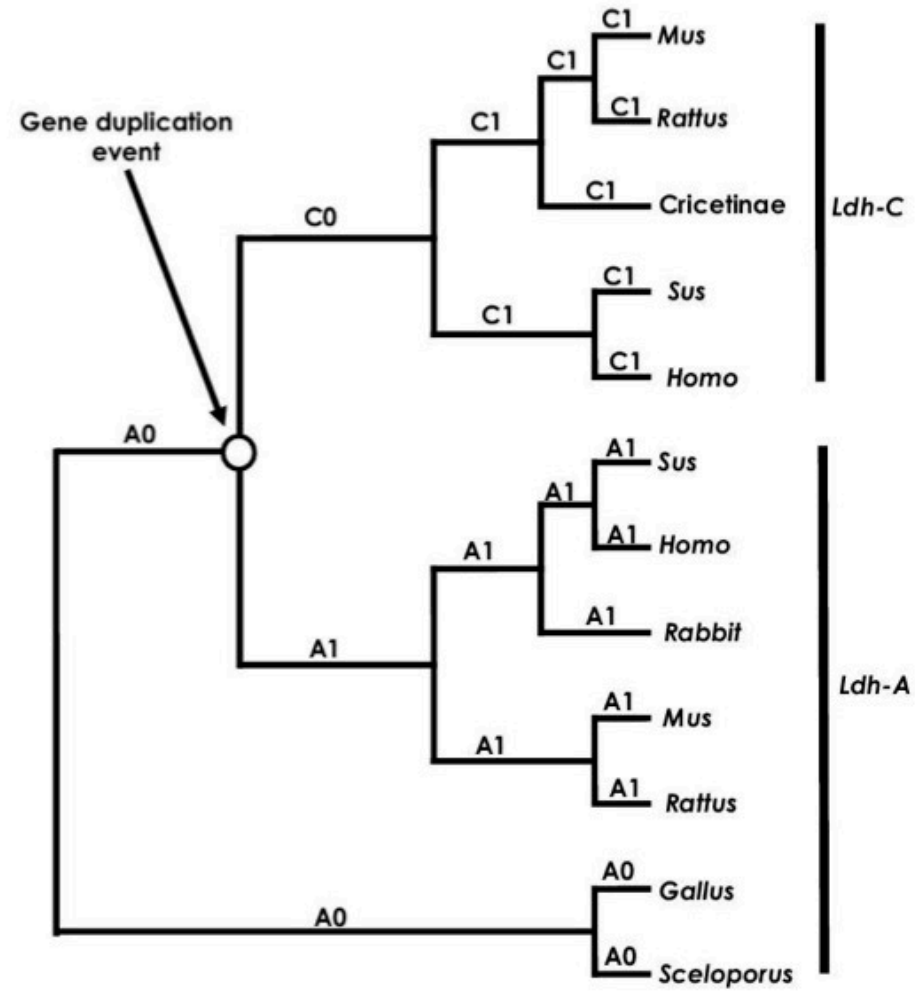


exercise 3:



- Each one represents a different "branch model"
- H₀:** $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$
 - H₁:** $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$
 - H₂:** $\omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0}$
 - H₃:** $\omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0}$

exercise 3:



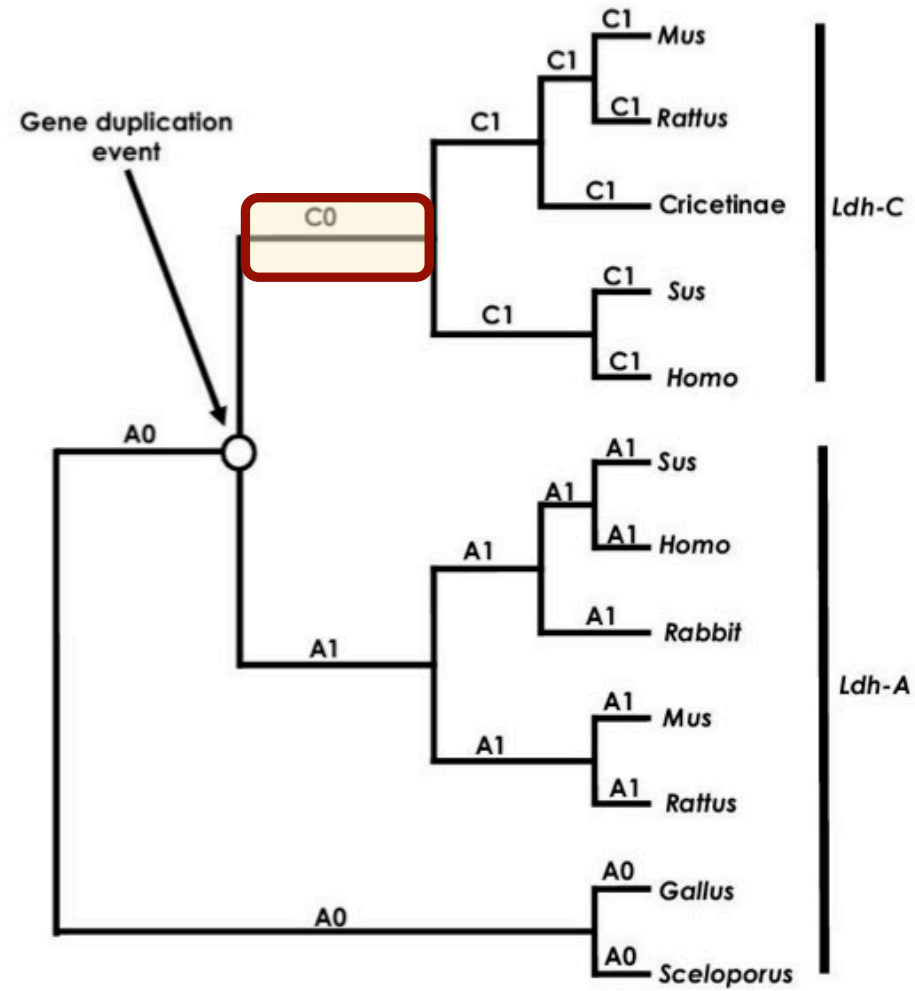
H₀: $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$ Null model

H₁: $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$

H₂: $\omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0}$

H₃: $\omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0}$

exercise 3:



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

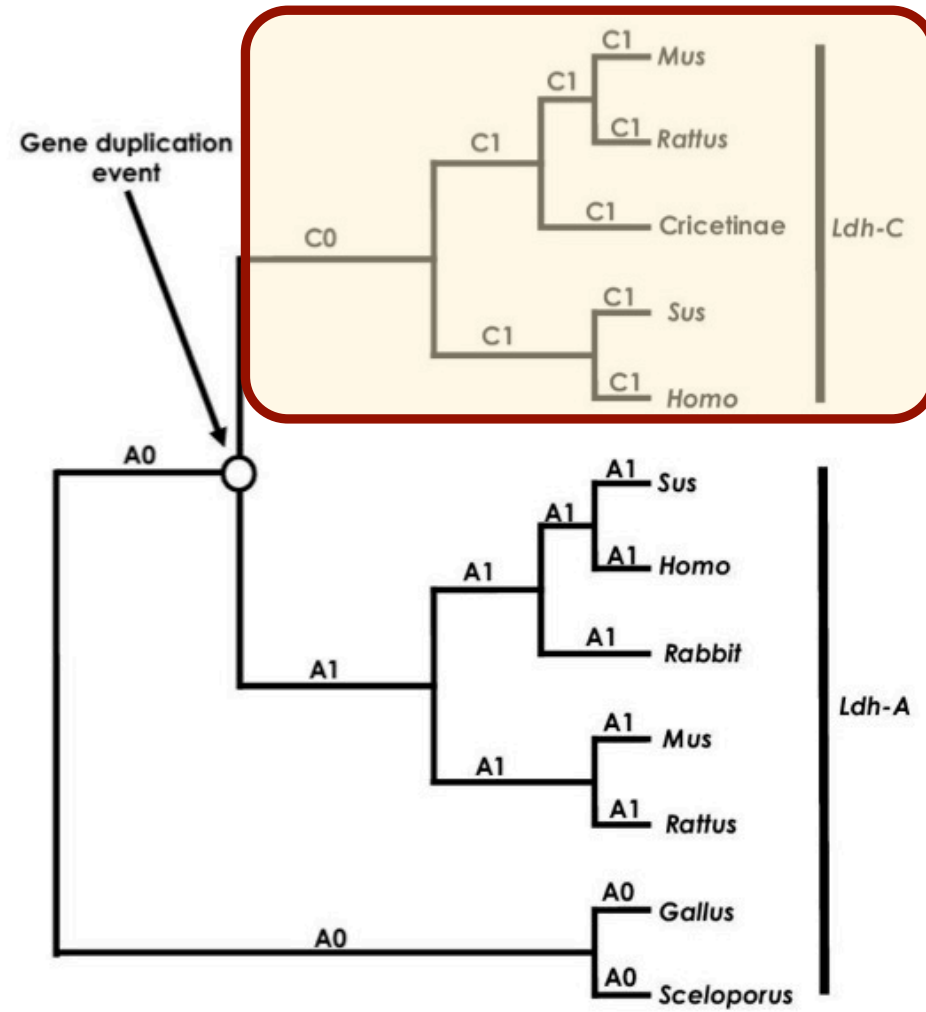
$$H_1: \omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$$

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

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

Episodic model

exercise 3:



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

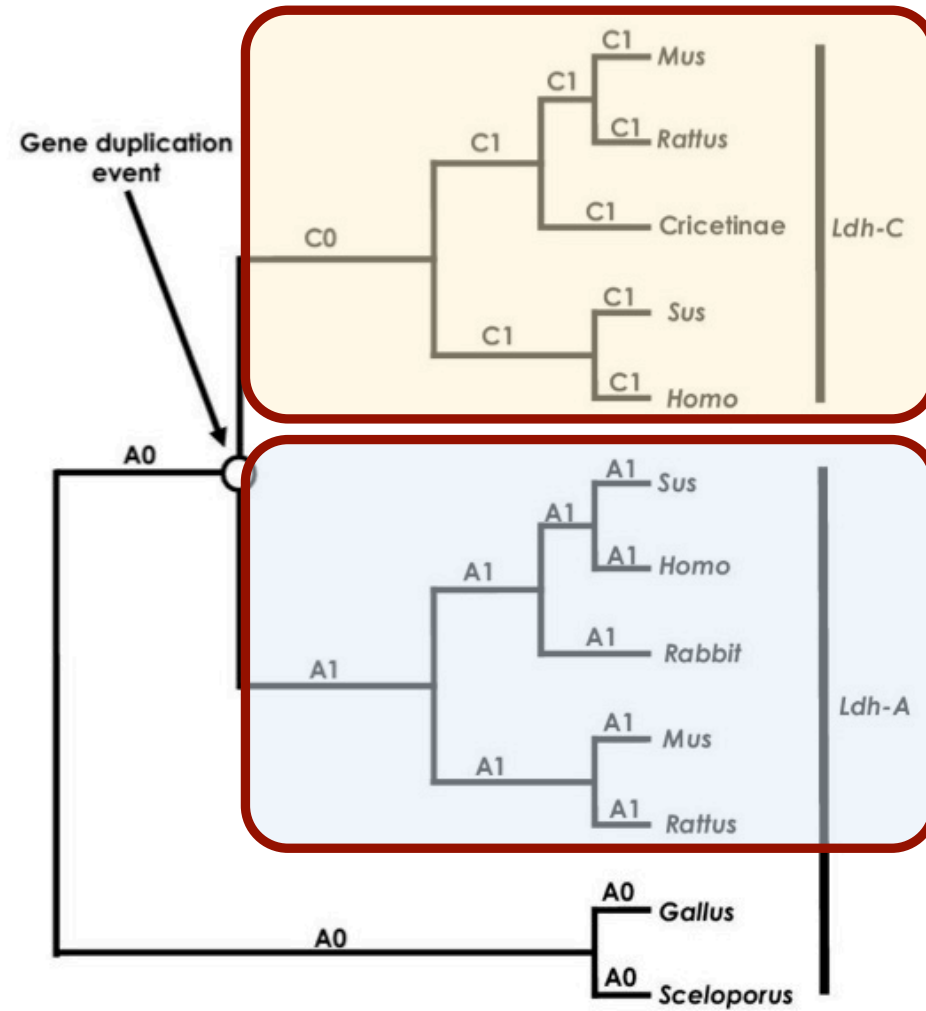
$$H_1: \omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$$

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

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

Long-term shift: 1-clade model

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)))));

```

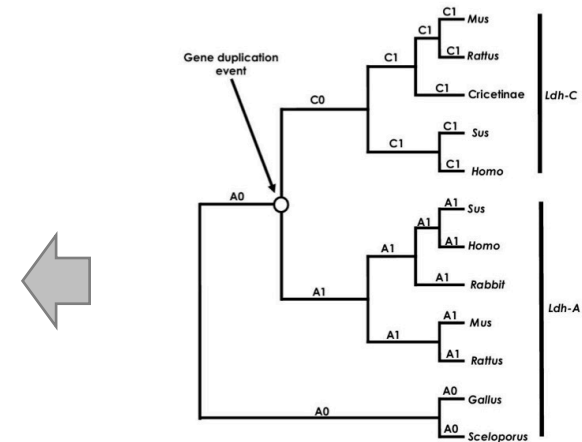
*H₃ in Table 3:

*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)))));

```



H₀: $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$
H₁: $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$
H₂: $\omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0}$
H₃: $\omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0}$

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
```

RENAME result files
to avoid overwriting
your results!

***H₀** in Table 3:

***model = 0**

```
* (X02152Hom,U07178Sus,(M22585rab,((NM017025Rat,U13687Mus),
*((AF070995C,(X04752Mus,U07177Rat)),(U95378Sus,U13680Hom)),(X53828OG1,
* U28410OG2)))));
```



Null model

***H₁** in Table 3:

***model = 2**

```
* (X02152Hom,U07178Sus,(M22585rab,((NM017025Rat,U13687Mus),((AF070995C,
*(X04752Mus,U07177Rat)),(U95378Sus,U13680Hom))#1,(X53828OG1,U28410OG2))
* ));
```



Episodic model

***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)))));
```



Long-term shift: 1-clade model

***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)))));
```



Long-term shift: 2-clade model

Likelihood Ratio Test (LRT)

Does the more complex model explain the data *significantly better* than the simpler model?

1

fit two nested models



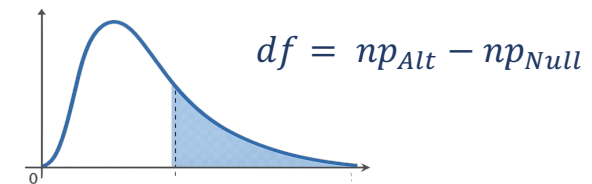
2

compare likelihoods \rightarrow compute test statistic

$$\text{LRT} = 2(\ln L_{\text{Alt}} - \ln L_{\text{Null}})$$

3

assess significance



4

use your brain



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.

H₀ v H₁: df = 1

H₀ v H₂: df = 1

H₂ v H₃: df = 1

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

exercise 3 concept questions:

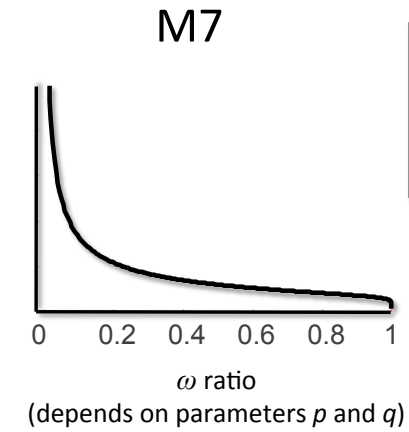
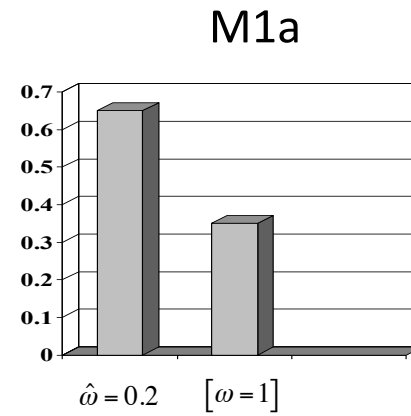
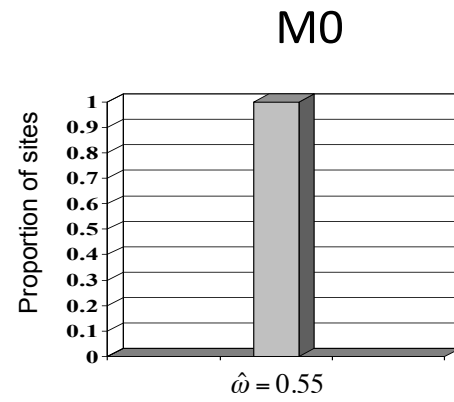
1. Can you explain the biological interpretation of all 4 models (hypotheses) of *Ldh* gene-family evolution?
2. Can you explain how these models are nested? Why is nesting a concern here? Do you understand the df for the relevant LRTs?
3. What evolutionary scenario is the “best” explanation of *Ldh* gene-family evolution?
4. Is there evidence of positive selection during the history of *Ldh* evolution? Are there any scenarios in which *Ldh* could have evolved by positive selection that would be undetectable by these LRTs?

Exercise 4:

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

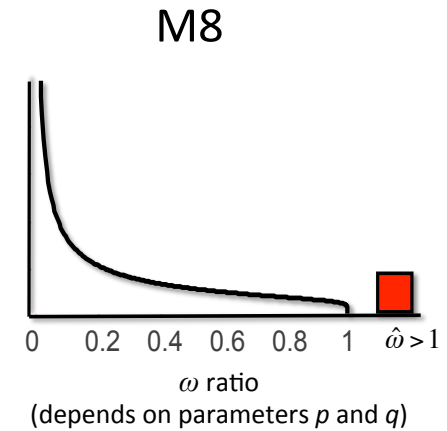
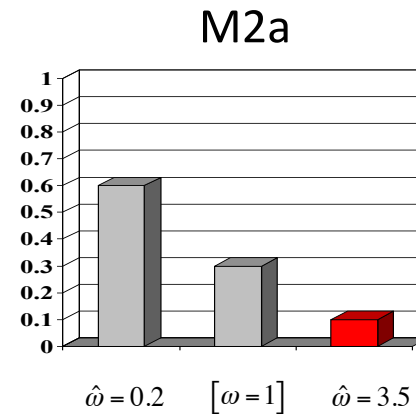
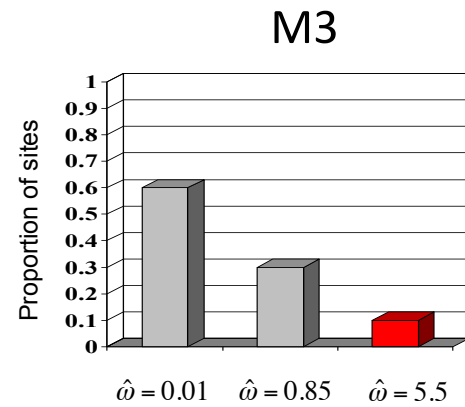
exercise 4:

H_0



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

H_a



LRT

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

df = 4

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

df = 2

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

df = 2

```

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 frequencies
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.

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	?	$\omega = ?$	N.A.	?
M3: discrete (5)	?	$p_0 = ?, p_1 = ?, (p_2 = ?)$ $\omega_0 = ?, \omega_1 = ?, \omega_2 = ?$? (?)	?
M1a: neutral (2)	?	$p_0 = ?, (p_1 = ?)$ $\omega_0 = ?, (\omega_1 = 1)$	N.A.	?
M2a: selection (4)	?	$p_0 = ?, p_1 = ?, (p_2 = ?)$ $\omega_0 = ?, (\omega_1 = 1), \omega_2 = ?$? (?)	?
M7: beta (2)	?	$p = ?, q = ?$	N.A.	?
M8: beta& ω (4)	?	$p_0 = ? (p_1 = ?)$ $p = ?, q = ?, \omega = ?$? (?)	?

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

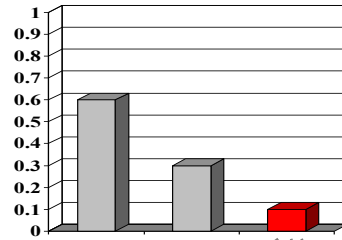
^b This d_N/d_S ratio is an average over all sites in the HIV-2 *nef* gene alignment.

^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%.

Concept map for tasks 1-3...

model:
5% have $\omega > 1$



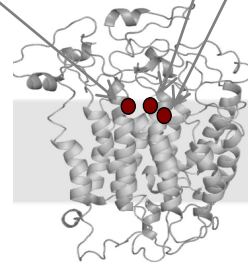
1. Fit model to data \rightarrow MLEs
2. Test hypotheses via Null and alternative models for ω

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..	..TGC	A..
...C	..T	A..	...	A.TAA	...	A.C	...	AGC	...
...	..C	...	G.A	..ATA	A..	...	AA.	TG.G	...	A..	..T	..GC	..T
...	..C	..G	GA.	..TT	C..	..G	..A	...	AT.TG	..A	..GC	...

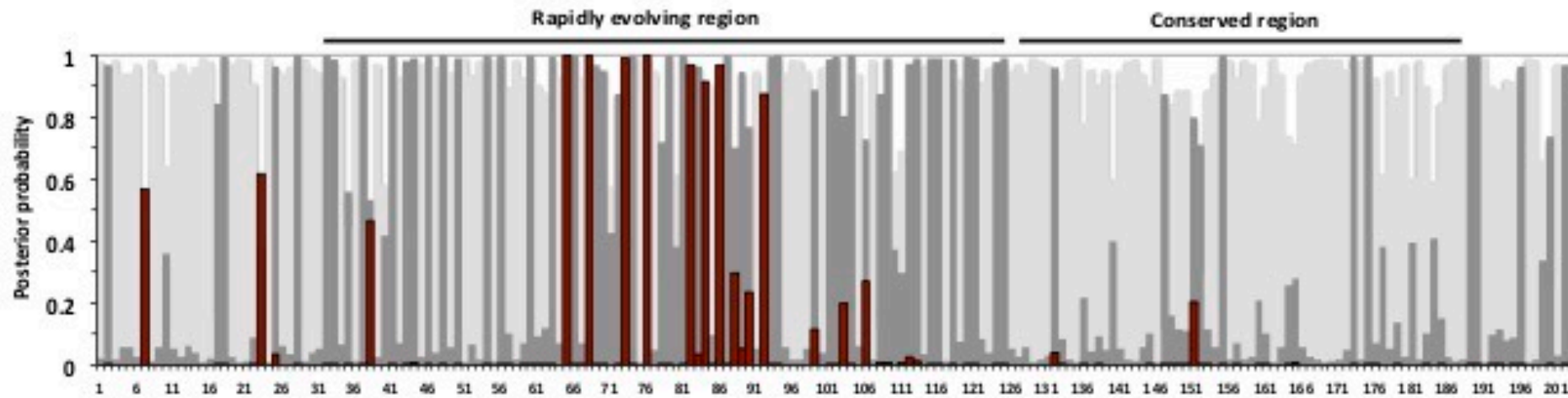
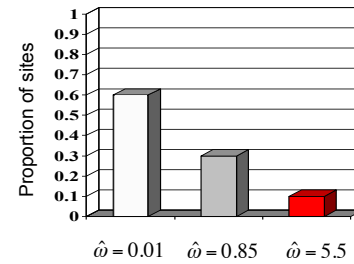
3. Predict which sites have $\omega > 1$

structure:
sites are in contact



4. Interpret results in known biological context

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

exercise 4 concept questions:

Try to synthesize all your results and attempt a biological interpretation of the sort that you would want to publish within an actual research paper. The following two general questions should help get you going. I strongly encourage you to do this last step in collaboration with other workshop students; talk it through!

1. What biological conclusions are well-supported by these data?
2. What aspects of the results can you interpret according your prior biological knowledge of this, or similar, systems?

exercise 4, step 8 (optional)...

1. re-run M0 (note time)
2. change .ctl file for M0: set **fix_blength = 0**
3. run M0 and estimate branch lengths (note time)

step 8 concept questions:

We want to assess how much extra computation is associated with different parts of these codon models.

1. *What is the effect of tree size on ML based hypothesis testing?*
2. *Do you think branch lengths have a big impact on the likelihood of the data?
How about hypothesis testing?*
3. *Can you think of a way to use **fix_blength = 2** to check?*