PAML (Phylogenetic Analysis by Maximum Likelihood)

A program package by Ziheng Yang

(Demonstration by Joseph Bielawski)

1. Three inference tasks

3 inference tasks

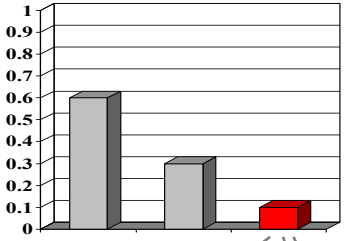
task 1. parameter estimation (e.g., ω)

task 2. hypothesis testing

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

Concept map for tasks 1-3…

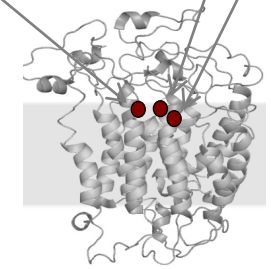
model:
5% have $\omega > 1$



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 ...
```

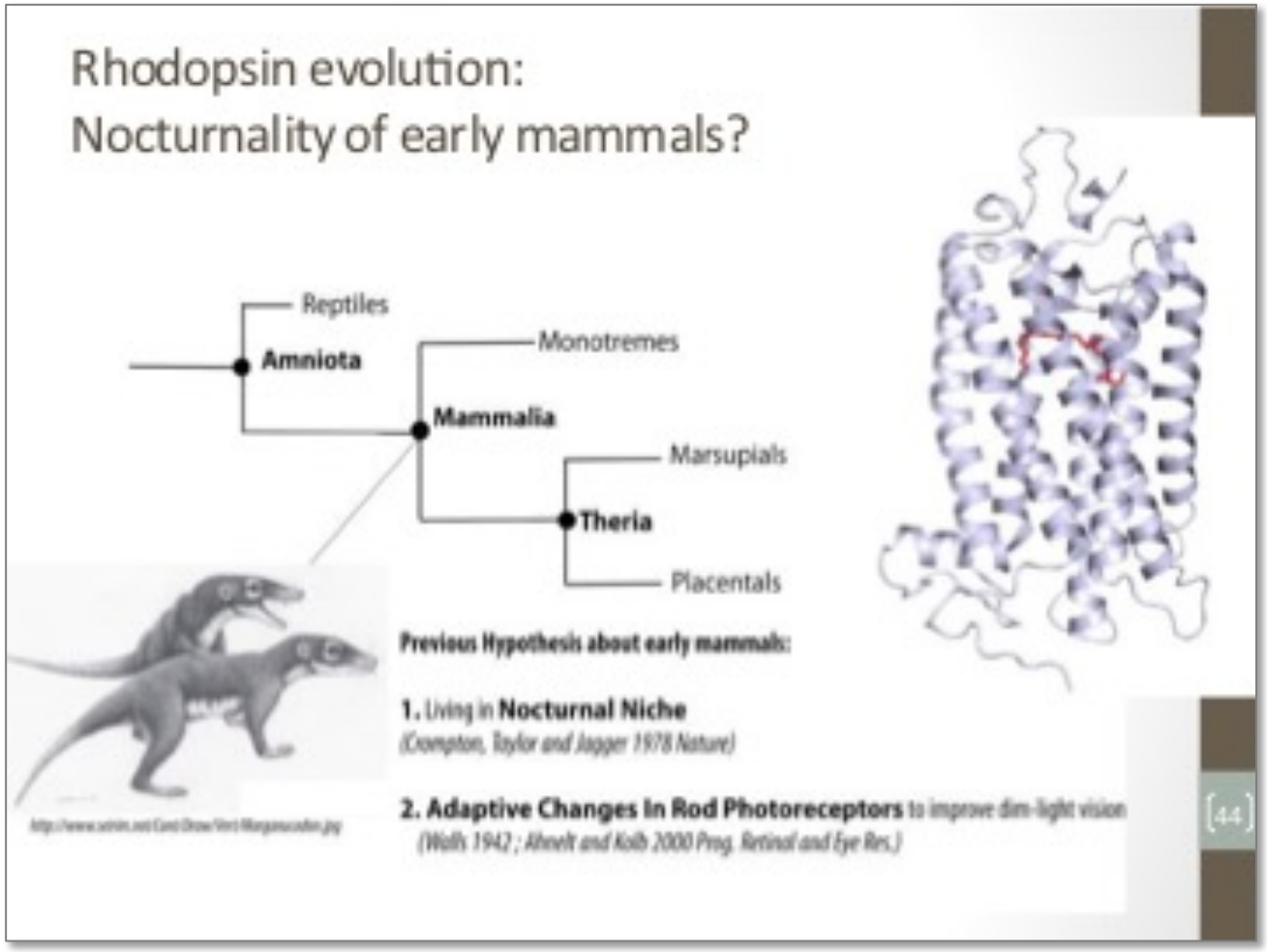
structure:
sites are in contact



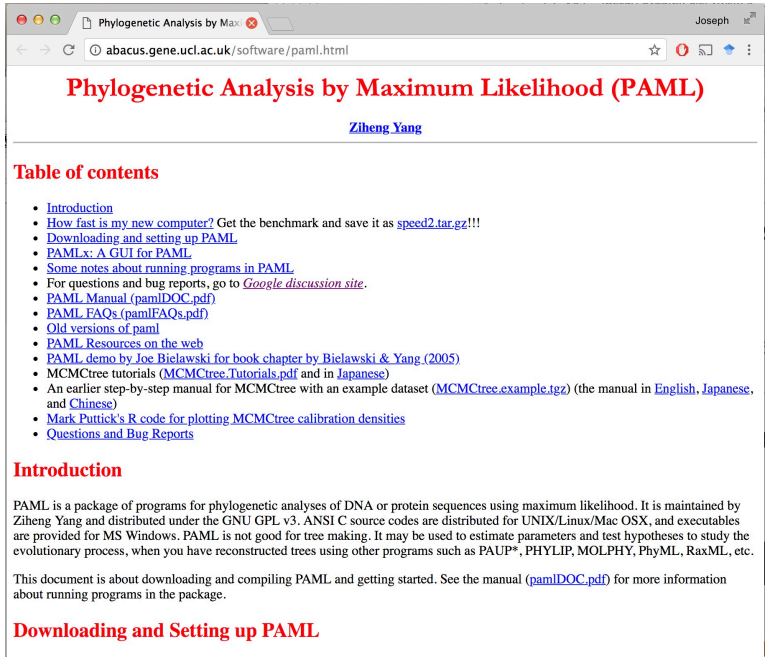
1. Fit model to data → MLEs
2. Test hypotheses via Null and Alternative models for ω
3. Predict which sites have $\omega>1$
4. Interpret results in known biological context

Gold Standard

Combine **evolutionary computation with experimental investigation**



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

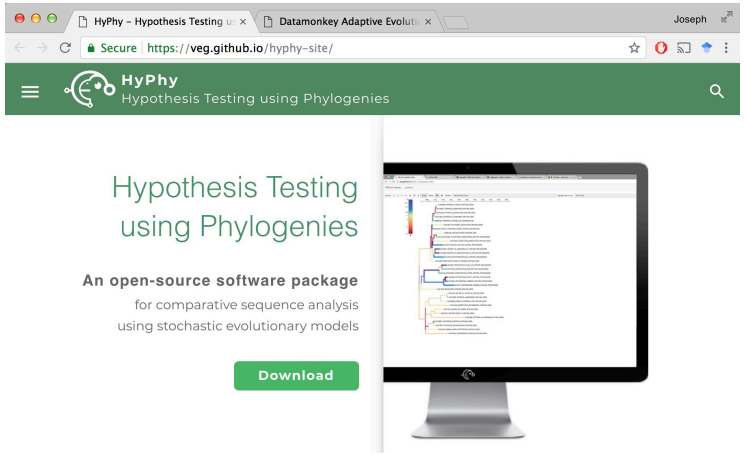
Introduction

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.

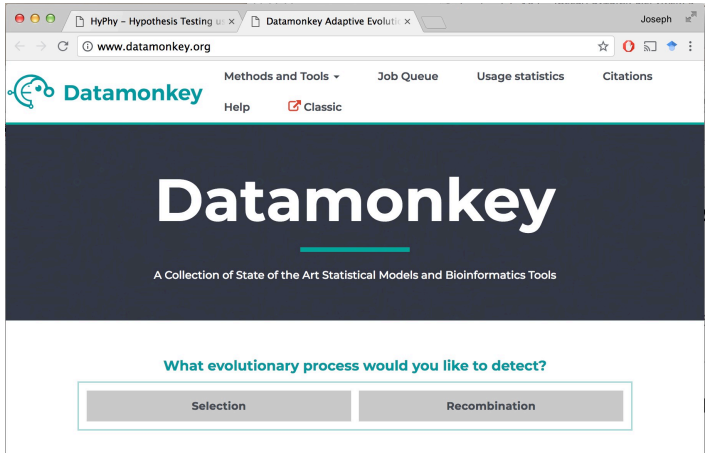
This document is about downloading and compiling PAML and getting started. See the manual (pamlDOC.pdf) for more information about running programs in the package.

Downloading and Setting up PAML

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



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



http://www.datamonkey.org/

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

We are NOT tyring to teach a particular software package.

Engauge 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.

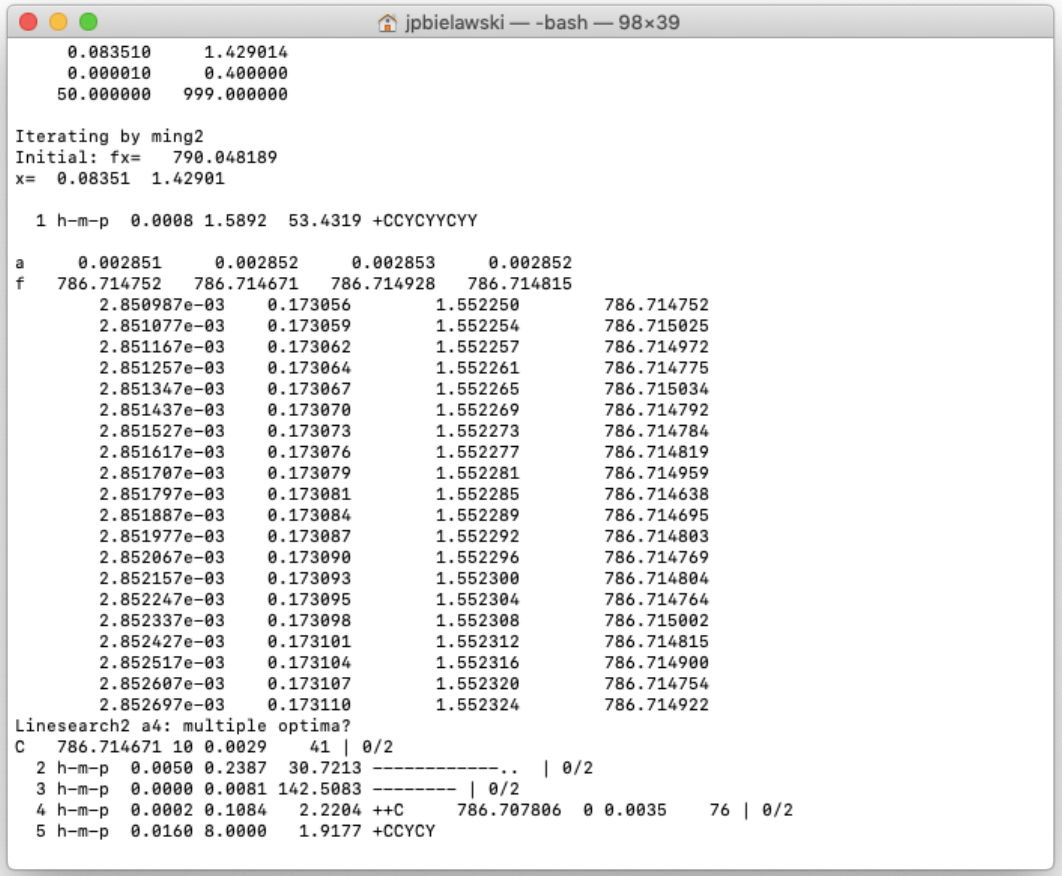
2. Brief introduction to PAML

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)

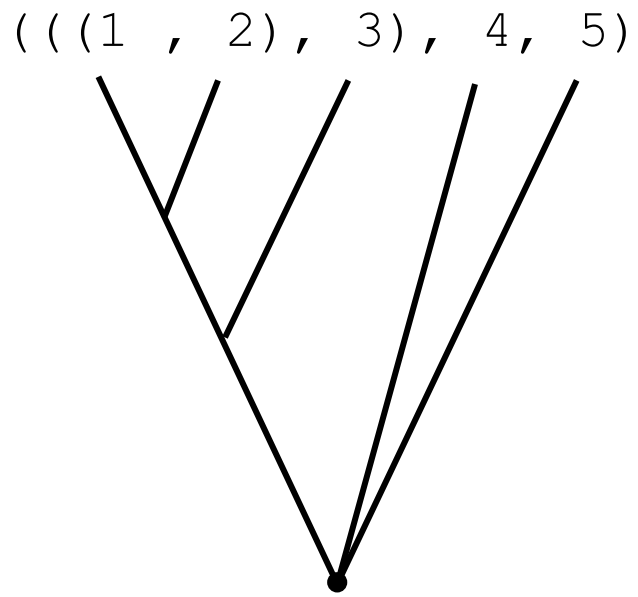


1. sequence file (modified “PHYLIP” format)

```
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
```

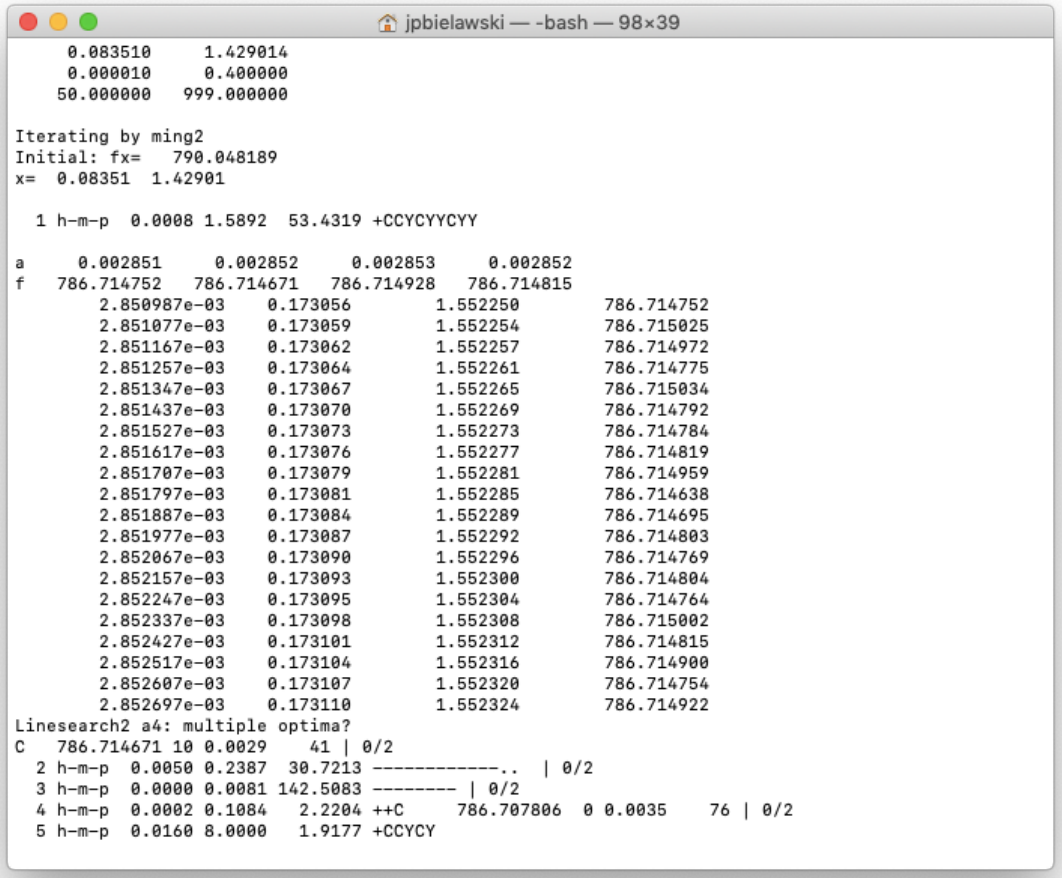
2. tree file (“Newick” format)



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

Running PAML programs

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



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

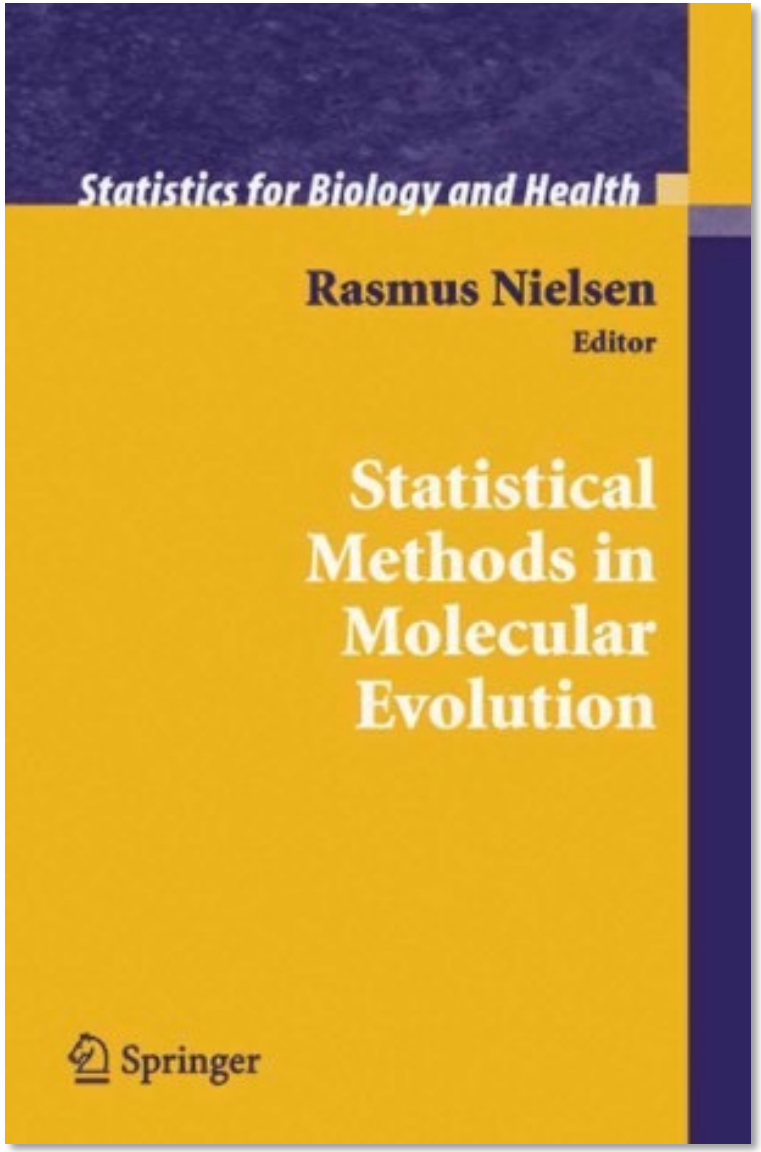
```
  seqfile = seqfile.txt   * sequence data filename
 treefile = tree.txt      * tree structure file name
  outfile = results.txt   * main result file name

    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.

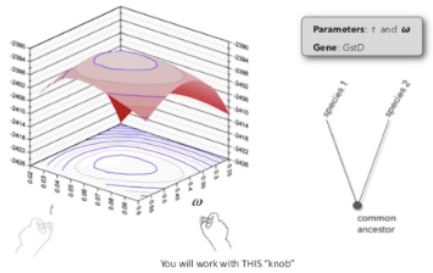
Either way, it is still *"best practice" to run each exercise in a separate working directory that you will create* (*e.g.*, `PAML_ex1`), *and work with copies of the required files within that directory*.

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

Let's try something a little different in 2023…

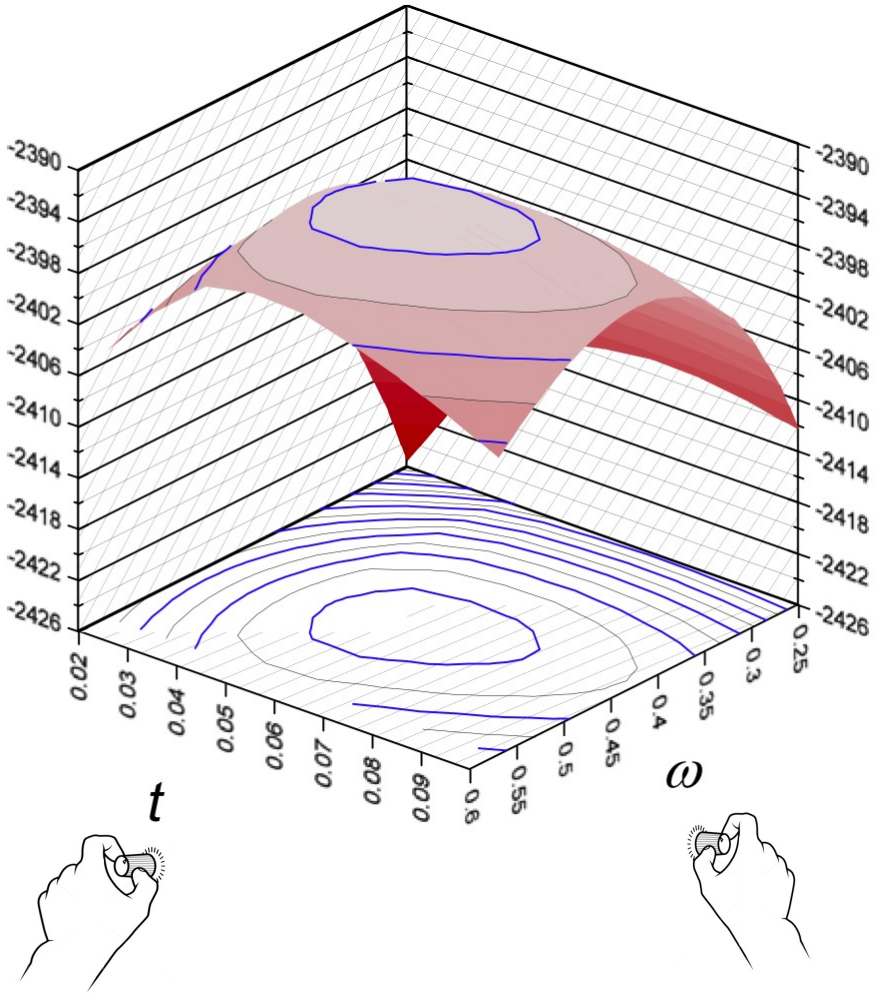
- exercises 1-2 we will do together

- exercises 3-4 you will do on your own

Exercise 1:

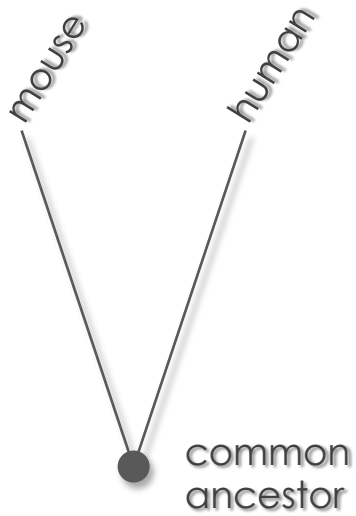
ML estimation of the d_N/d_S (ω) "by hand" for *GstD1*

exercise 1:



Parameters: t and ω

Gene: acetylcholine α receptor



lnL = -2399

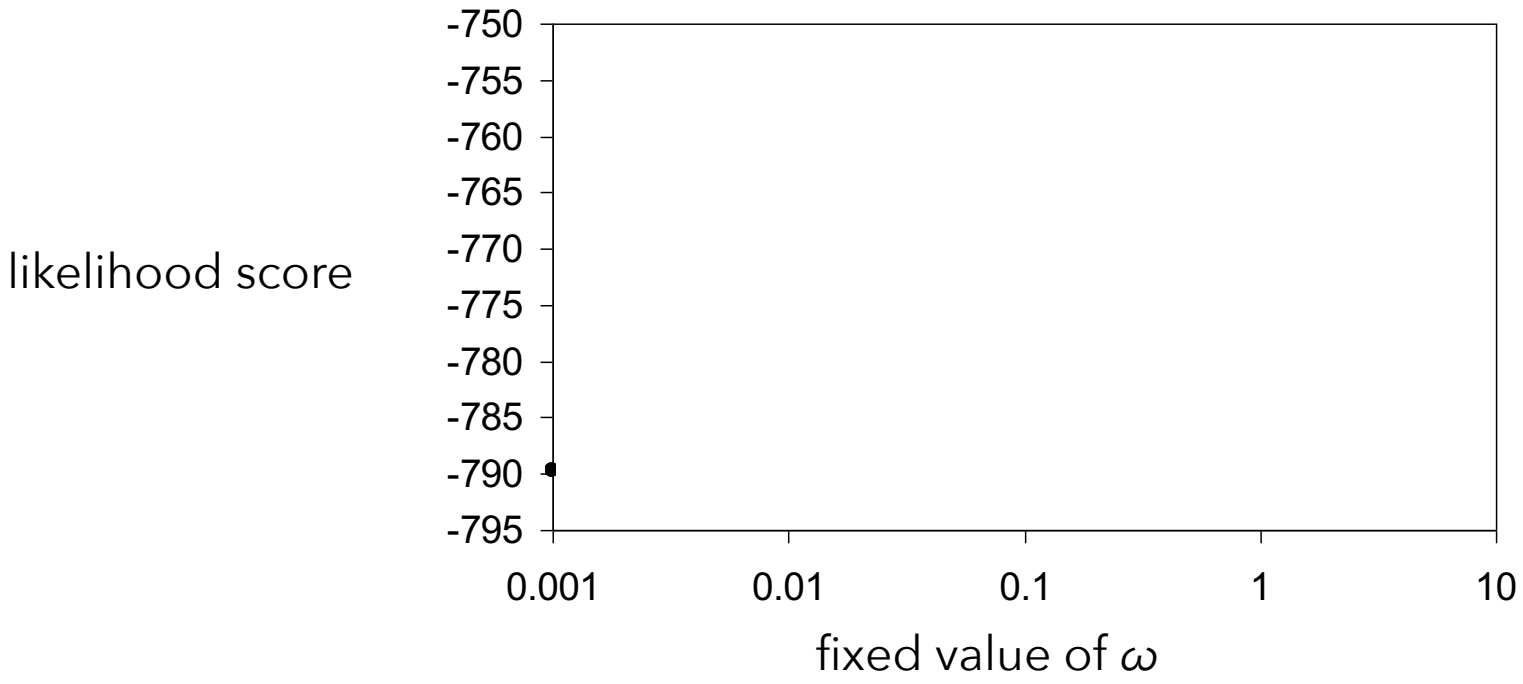
exercise 1:
you will work THIS "knob"

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. Does the area under your likelihood curve sum to 1.0?

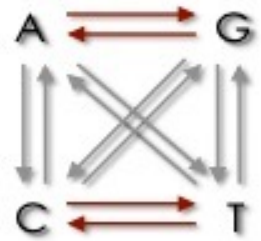
3. Can you explain*, in non-technical language*, what the MLE represents and why you would want to estimate it?

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}	60

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 2 concept questions:

1. What does the value of *S* represent?

2. Which model assumptions had the largest and smallest impact on *S*? Can you use your knowledge of this dataset to explain your observations?

3. Can you explain how the estimates of *S* impacted the estimates of the intensity of natural selection pressure?

4. What model of codon frequencies would you choose for these data and why?

Short aside to help with the interpretation of *S*…

Why use d_N and d_S? (Why not use raw counts?)

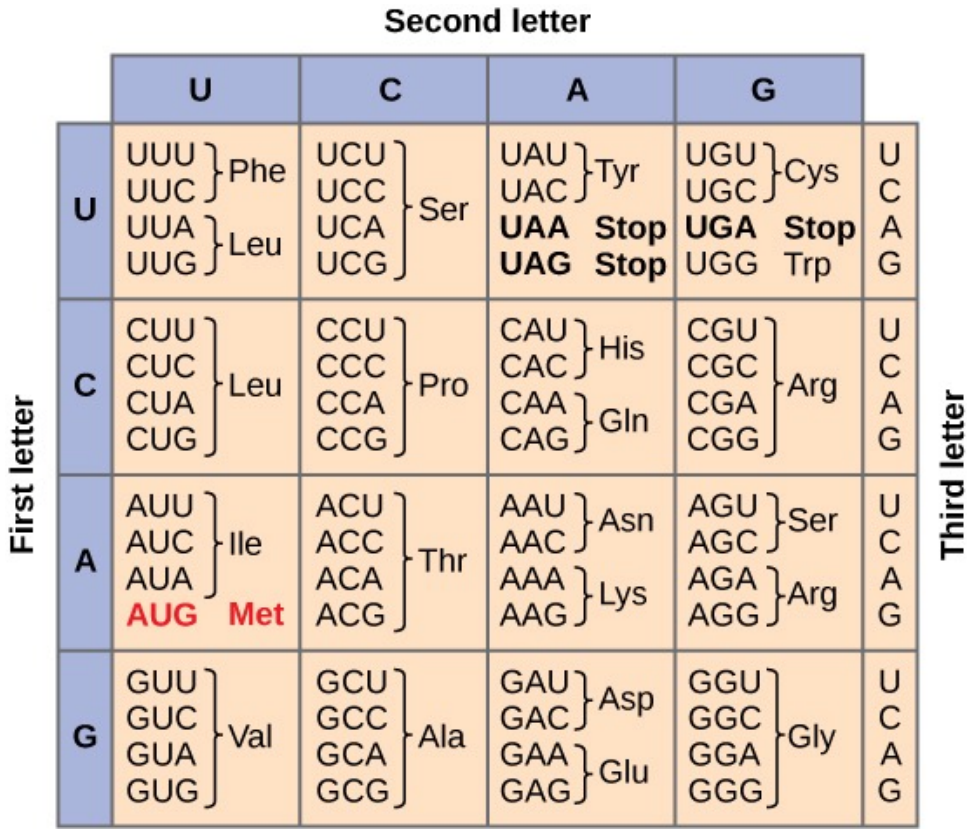
example of counts:

- 300 codon gene from a pair of species
- 5 synonymous differences
- 5 nonsynonymous differences

$$5/5 = 1$$

why don't we conclude that rates are equal (i.e., **neutral evolution**)**?**

Genetic code



all possible mutations → **two types**

synonymous (**S**): no change to protein

non-synonymous (**N**): changes the amino acid composition of protein

mutational opportunity vs. physical site

Relative proportion of different types of mutations in hypothetical protein coding sequence.				
Type	**Expected number of changes (proportion)**			
	All 3 Positions	**1st positions**	**2nd positions**	**3rd positions**
Total mutations	549 (100)	183 (100)	183 (100)	183 (100)
Synonymous	134 (25)	8 (4)	0 (0)	126 (69)
Nonsyonymous	392 (71)	166 (91)	176 (96)	57 (27)
nonsense	23 (4)	9 (5)	7 (4)	7 (4)

Note that by framing the counting of sites in this way we are using a "mutational opportunity" definition of the sites. Thus, a synonymous or non-synonymous site is not considered a physical entity!

Note that this is NOT a model-free exercise: for this to make sense, we assume a hypothetical model where all codons are used equally and that all types of point mutations are equally likely.

Why do we use d_N and d_S ?

same example, but using d_N and d_S:

Synonymous sites (**S**) = 25.5%

S = 300 × 3 × 25.5% = **229.5**

Nonsynonymous sites(**N**) = 74.5%

N = 300 × 3 × 74.5% = **670.5**

So, d_S = 5/**229.5** = 0.0218

d_N = 5/**670.5** = 0.0075

d_N/d_S (ω) = 0.34, **purifying selection** !!!

Now take another look at the table of codon counts for the *GstD1* gene and think about the meaning of *S*…

RECALL…

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

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	0.0691	0.0221	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

exercise 2 concept questions:

Work on these questions now…

1. Is the value of *S* meant to represent a count of physical sites or the concept of mutational opportunities?

2. Which model assumptions had the largest and smallest values of S? Can you use your knowledge of this dataset to explain your observations?

3. What model of codon frequencies would you choose for these data and why?

4. The biological conclusions for *GstD* are sensitive to model assumptions; will all genes be this sensitive to the codon frequency model? Why?

Some more information on exploring the relationship between the model and *your* data…

Chapter 13

Looking for Darwin in Genomic Sequences: Validity and Success Depends on the Relationship Between Model and Data

Christopher T. Jones, Edward Susko, and Joseph P. Bielawski

Abstract

Codon substitution models (CSMs) are commonly used to infer the history of natural section for a set of protein-coding sequences, often with the explicit goal of detecting the signature of positive Darwinian selection. However, the validity and success of CSMs used in conjunction with the maximum likelihood (ML) framework is sometimes challenged with claims that the approach might too often support false conclusions. In this chapter, we use a case study approach to identify four legitimate statistical difficulties associated with inference of evolutionary events using CSMs. These include: (1) model misspecification, (2) low information content, (3) the confounding of processes, and (4) phenomenological load, or PL. While past criticisms of CSMs can be connected to these issues, the historical critiques were often misdirected, or overstated, because they failed to recognize that the success of any model-based approach depends on the relationship between model and data. Here, we explore this relationship and provide a candid assessment of the limitations of CSMs to extract historical information from extant sequences. To aid in this assessment, we provide a brief overview of: (1) a more realistic way of thinking about the process of codon evolution framed in terms of population genetic parameters, and (2) a novel presentation of the ML statistical framework. We then divide the development of CSMs into two broad phases of scientific activity and show that the latter phase is characterized by increases in model complexity that can sometimes negatively impact inference of evolutionary mechanisms. Such problems are not yet widely appreciated by the users of CSMs. These problems can be avoided by using a model that is appropriate for the data; but, understanding the relationship between the data and a fitted model is a difficult task. We argue that the only way to properly understand that relationship is to perform in silico experiments using a generating process that can mimic the data as closely as possible. The mutation-selection modeling framework (MutSel) is presented as the basis of such a generating process. We contend that if complex CSMs continue to be developed for testing explicit mechanistic hypotheses, then additional analyses such as those described in here (e.g., penalized LRTs and estimation of PL) will need to be applied alongside the more traditional inferential methods.

Key words Codon substitution model, dN/dS, False positives, Maximum likelihood, Mechanistic model, Model misspecification, Mutation-selection model, Parameter confounding, Phenomenological load, Phenomenological model, Positive selection, Reliability, Statistical inference, Site-specific fitness landscape

Maria Anisimova (ed.), *Evolutionary Genomics: Statistical and Computational Methods*, Methods in Molecular Biology, vol. 1910, https://doi.org/10.1007/978-1-4939-9074-0_13,

Phenomenological Load on Model Parameters Can Lead to False Biological Conclusions

Christopher T. Jones,*[1] Noor Youssef,[2] Edward Susko,[1] and Joseph P. Bielawski[2]

[1]Department of Mathematics and Statistics, Dalhousie University, Halifax, NS, Canada

[2]Department of Biology, Dalhousie University, Halifax, NS, Canada

***Corresponding author:** E-mail: cjones2@dal.ca.*

Associate editor: Jeffrey Thorne

Abstract

When a substitution model is fitted to an alignment using maximum likelihood, its parameters are adjusted to account for as much site-pattern variation as possible. A parameter might therefore absorb a substantial quantity of the total variance in an alignment (or more formally, bring about a substantial reduction in the deviance of the fitted model) even if the process it represents played no role in the generation of the data. When this occurs, we say that the parameter estimate carries phenomenological load (PL). Large PL in a parameter estimate is a concern because it not only invalidates its mechanistic interpretation (if it has one) but also increases the likelihood that it will be found to be statistically significant. The problem of PL was not identified in the past because most off-the-shelf substitution models make simplifying assumptions that preclude the generation of realistic levels of variation. In this study, we use the more realistic mutation-selection framework as the basis of a generating model formulated to produce data that mimic an alignment of mammalian mitochondrial DNA. We show that a parameter estimate can carry PL when 1) the substitution model is underspecified and 2) the parameter represents a process that is confounded with other processes represented in the data-generating model. We then provide a method that can be used to identify signal for the process that a given parameter represents despite the existence of PL.

Key words: **codon substitution models, mechanistic versus phenomenological, phenomenological load, false positives, reliability.**

Introduction

There are in general two ways to quantitatively describe a natural process. The phenomenological approach is to summarize relationships between variables with little or no reference to causation. The alternative is to specify a model based on known or hypothetical mechanistic links between variables that explain their relationships. For example, although Newton's law of universal gravitation provides a highly accurate description of the apparent force of attraction between objects, it does so without explaining the cause. Newton's law is therefore phenomenological. Einstein, by contrast, described gravitation mechanistically as the result of mass generating curvature in space-time. Biology is replete with models of complex processes that cannot be placed into either of these neat categories. On the one hand, there is a natural desire to build mechanistic models that capture as much of the complexity and richness of a process as possible. On the other hand, limitations in information and computational resources often make simplifying assumptions unavoidable, thereby forcing a more phenomenological approach. The result of this tension is that models of biological processes often fall somewhere on a continuum between phenomenological and mechanistic.

A key feature of a model characterized as mechanistic is that its parameters are interpretatable with repsect to the real data-generating process (Liberles et al. 2013). This underlines the fact that the terms mechanistic and phenomenological are more aptly applied to individual parameters. Indeed, mechanistic and phenomenological parameters often appear together in the same model (Rodrigue and Philippe 2010). However, the distinction between the two is not always clear. Codon substitution models (CSMs), for example, consist of two submodels, one for the effect of selection at the amino acid level (the selection submodel) and the other for DNA-level substitution processes (the DNA submodel). The processes described by these submodels, the appearance of a new mutation in an individual and its eventual fixation or elimination in the population, are necessarily probabilistic (Moran 1958; Kimura 1962). In this context, we define a mechanistic parameter as one meant to provide an explanation for differences in probability distributions rather than in specific outcomes. For example, a probabilistic bias for or against replacement substitutions is represented in the selection submodel by a nonsynonymous-to-synonymous rate ratio ω. An estimate $\hat{\omega}$ of this rate ratio has traditionally been used to support one of three possible explanations: stringent selection by $\hat{\omega} < 1$; neutrality by $\hat{\omega} = 1$; and positive selection by $\hat{\omega} > 1$. However, ω can only be estimated by combining the information contained in a number of sites, at least when the number of taxa is limited (but see Rodrigue et al. 2010;

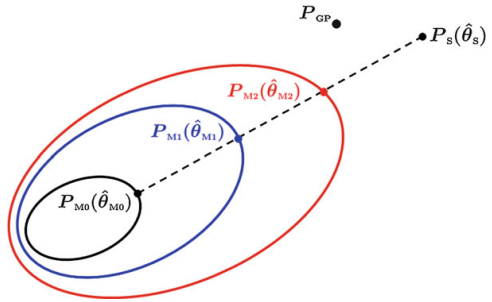
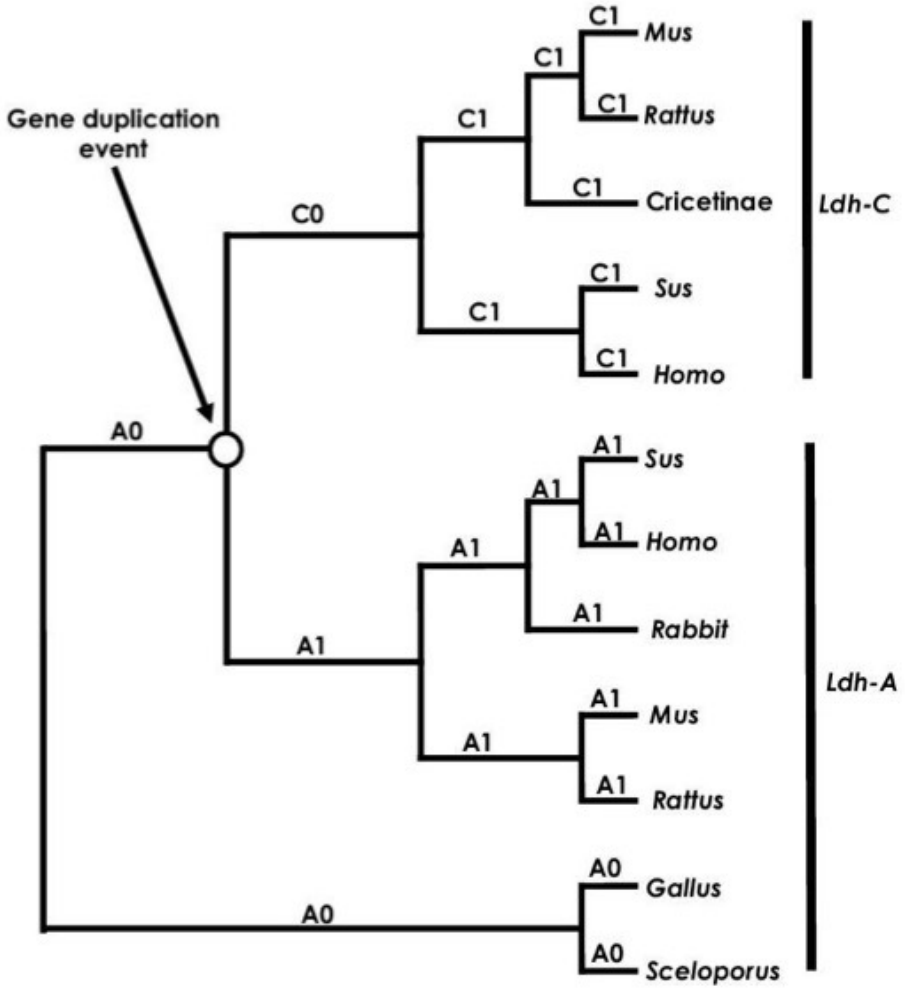


Fig. 2 The $(61^N - 1)$-dimensional simplex containing all possible site-pattern distributions for an N-taxon alignment is depicted. The innermost ellipse represents the subspace $\{P_{M0}(\theta_{M0})|\theta_{M0} \in \Omega_{M0}\}$ that is the family of distributions that can be specified using M0, the simplest of CSMs. This is nested in the family of distributions that can be specified using M1 (blue ellipse), a hypothetical model that has the same parameters as M0 plus some extra parameters. Similarly, M1 is nested in M2 (red ellipse). Whereas models are represented by subspaces of distributions, the true generating process is represented by a single point P_{GP}, the location of which is unknown. The empirical site-pattern distribution $P_S(\hat{\theta}_S)$ corresponds to the saturated model fitted to the alignment; with large samples, $P_S(\hat{\theta}_S) \approx P_{GP}$. For any other model M, the member $P_M(\hat{\theta}_M) \in \{P_M(\theta_M) \mid \theta_M \in \Omega_M\}$ most consistent with X is the one that minimizes deviance, which is twice the difference between the maximum log-likelihood of the data under the saturated model and the maximum log-likelihood of the data under M

Exercise 3:

Test hypotheses about molecular evolution of *Ldh* gene family

exercise 3:



Each one represents a different “branch model”

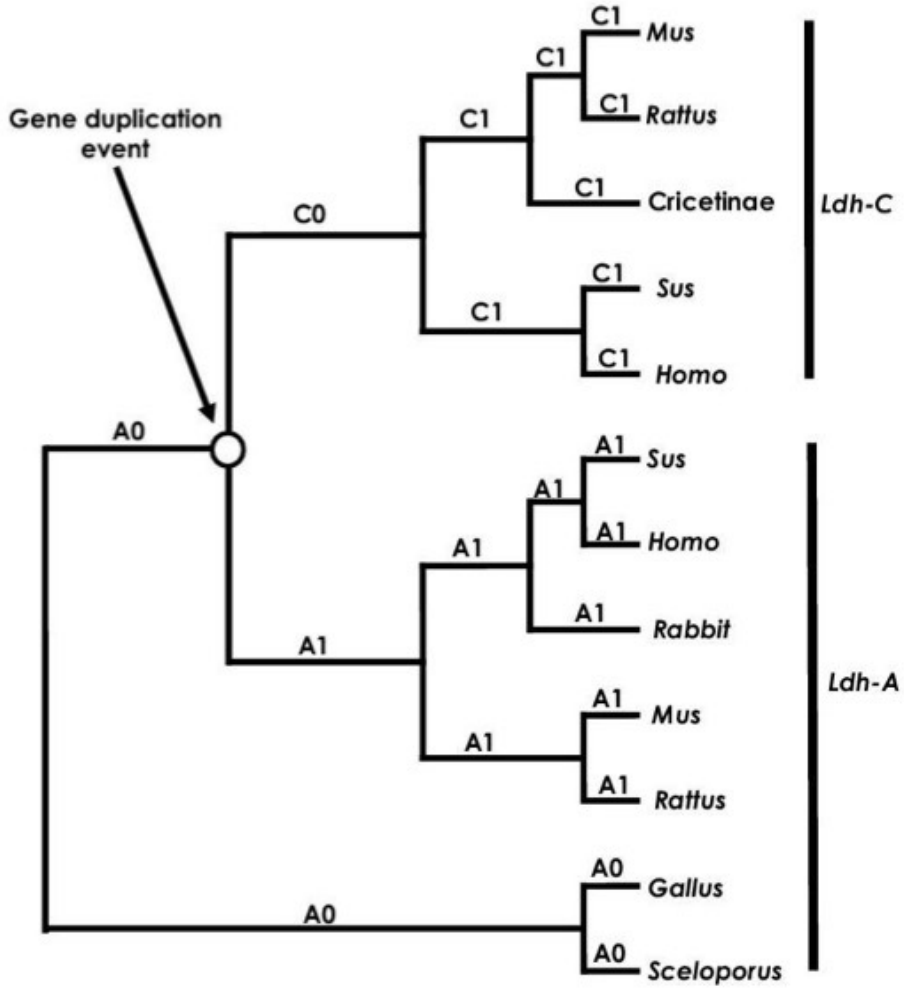
H_0: $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$

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:



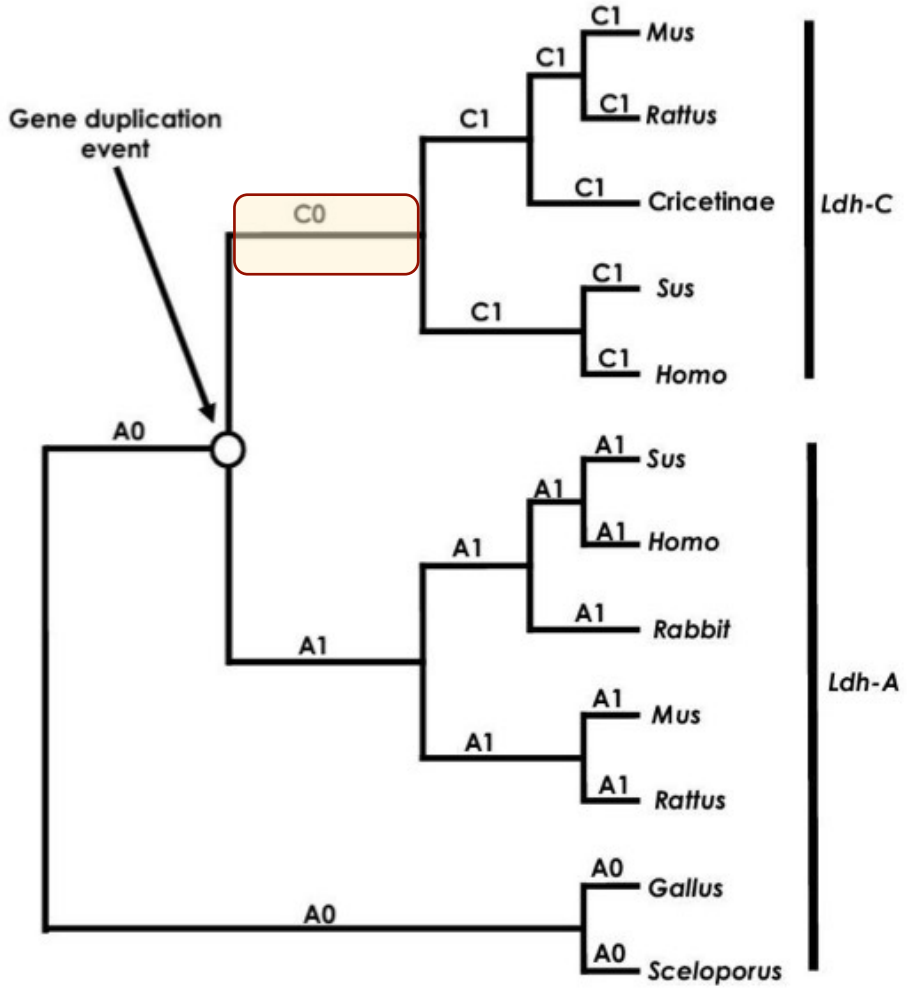
H_0: $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$ Null model

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

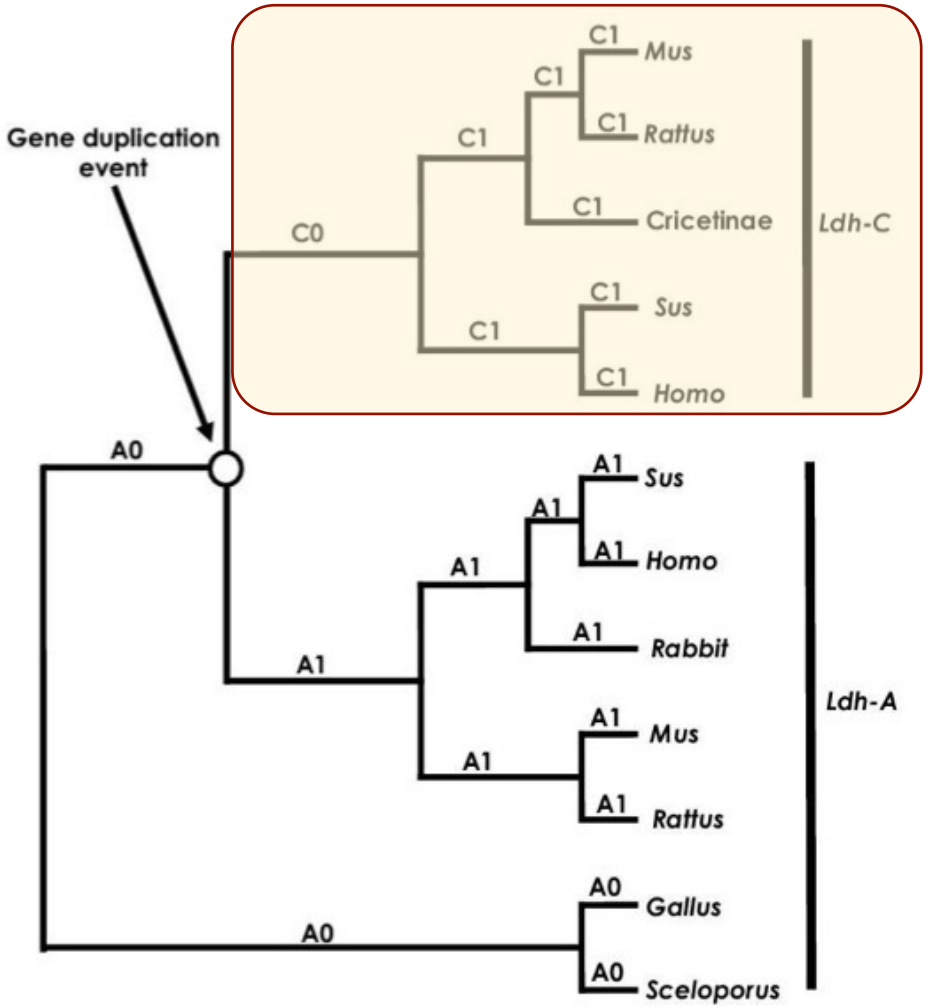
H_1: $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$

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

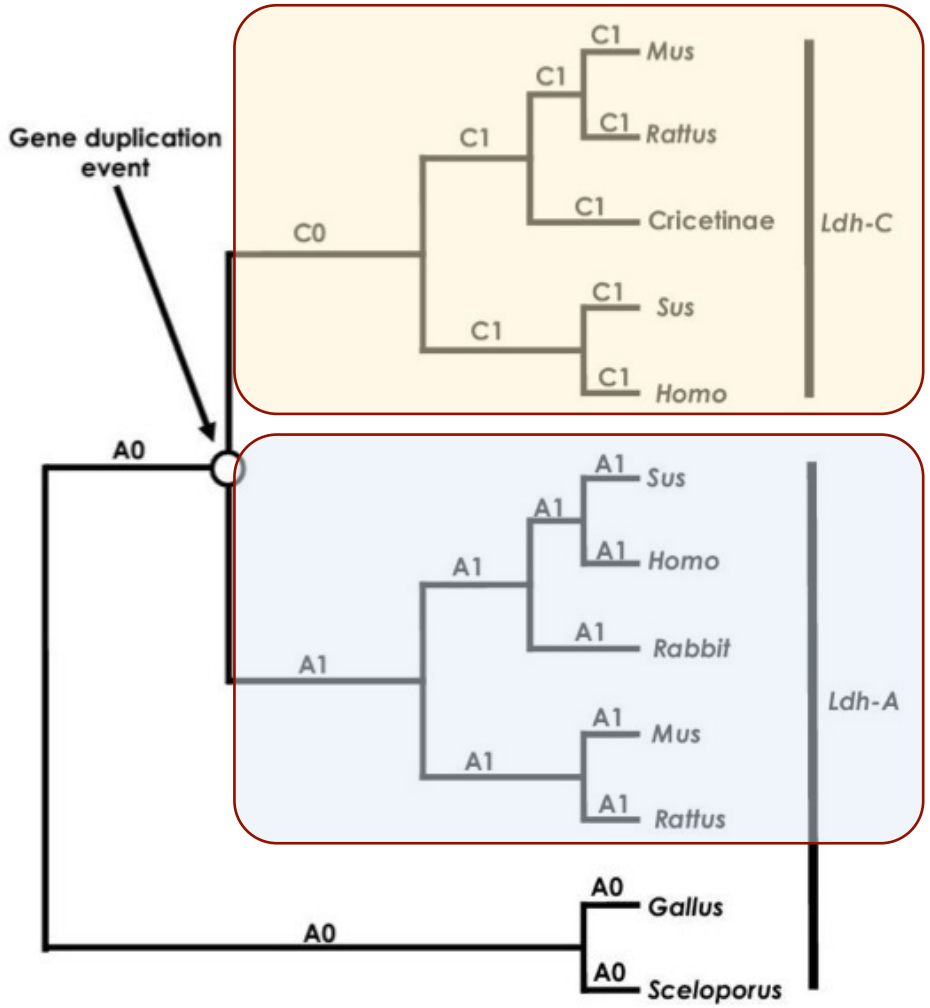
H_1: $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$

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:

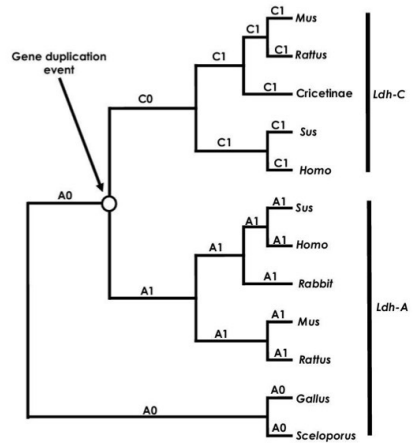
```
      seqfile = seqfile.txt   * sequence data filename
     treefile = tree.H0.txt   * tree structure file name [CHANGE THIS]
      outfile = results.txt   * main result file name

        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
```

*H_0 in Table 3:
*model = 0
*(X02152Hom,U07178Sus,(M22585rab,((NM017025Rat,U13687Mus),
*(((AF070995C,(X04752Mus,U07177Rat)),(U95378Sus,U13680Hom)),(X53828OG1,
* U28410OG2)))));

*H_1 in Table 3:
*model = 2
*(X02152Hom,U07178Sus,(M22585rab,((NM017025Rat,U13687Mus),(((AF070995C,
*(X04752Mus,U07177Rat)),(U95378Sus,U13680Hom))#1,(X53828OG1,U28410OG2))
*)));

*H_2 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)))));

*H_3 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 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?

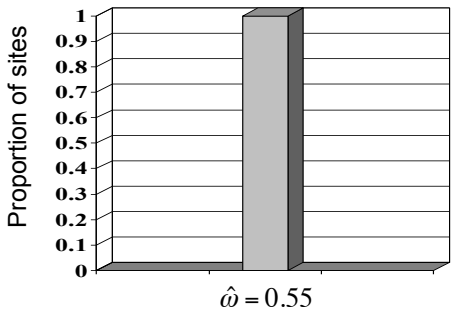
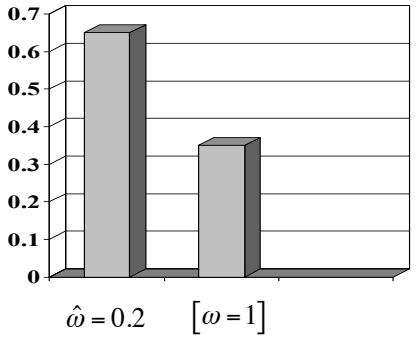
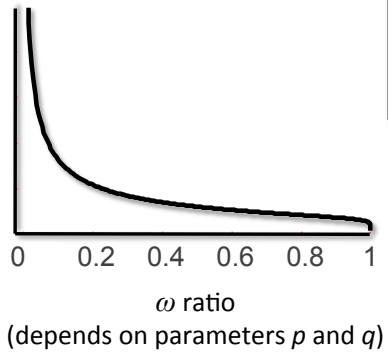
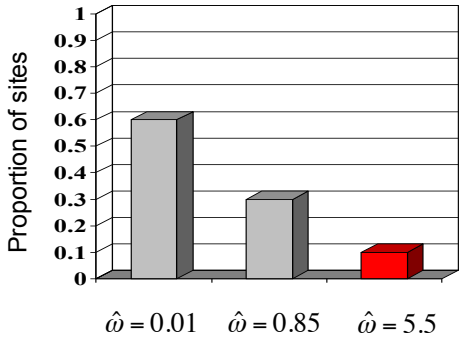
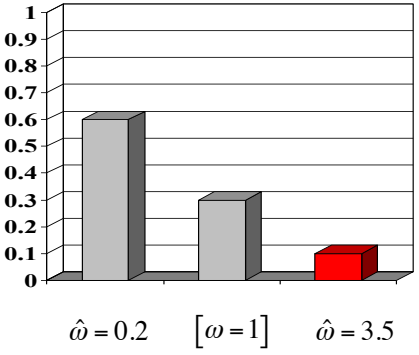
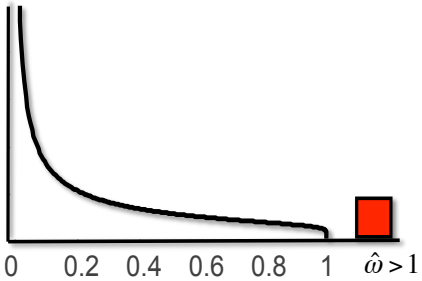
3. What evolutionary scenario is the best explanation of *Ldh* gene-family evolution?

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:

	M0	M1a	M7
H_0			
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

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

```
      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**

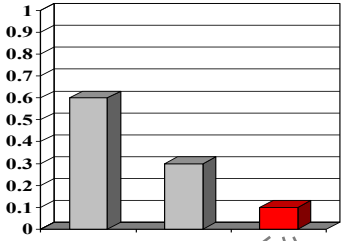
Table E4: Parameter estimates and likelihood scores under models of variable ω ratios among sites for HIV-2 *nef* genes.

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%.

Concept map for tasks 1-3…

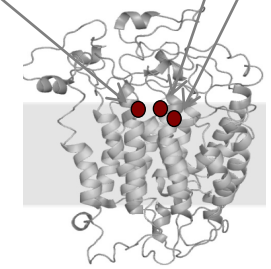
model:
5% have $\omega > 1$



Bayes' rule:
site 4, 12 & 13

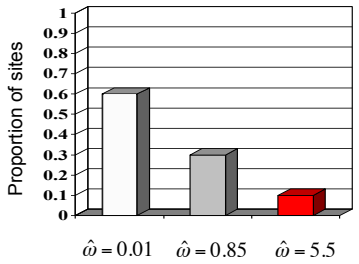
```
GTG CTG TCT CCT GCC GAC AAG ACC AAC GTC AAG GCC GCC TGG GGC AAG GTT GGC GCG CAC
... ... ... G.C ... ... ... T.. ..T ... ... ... ... ... ... ... ... .GC A..
... ... ... ..C ..T ... ... ... ... A.. ... A.T ... ... .AA ... A.C ... AGC ...
... ..C ... G.A .AT ... ..A ... ... A.. ... AA. TG. ... ..G ...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

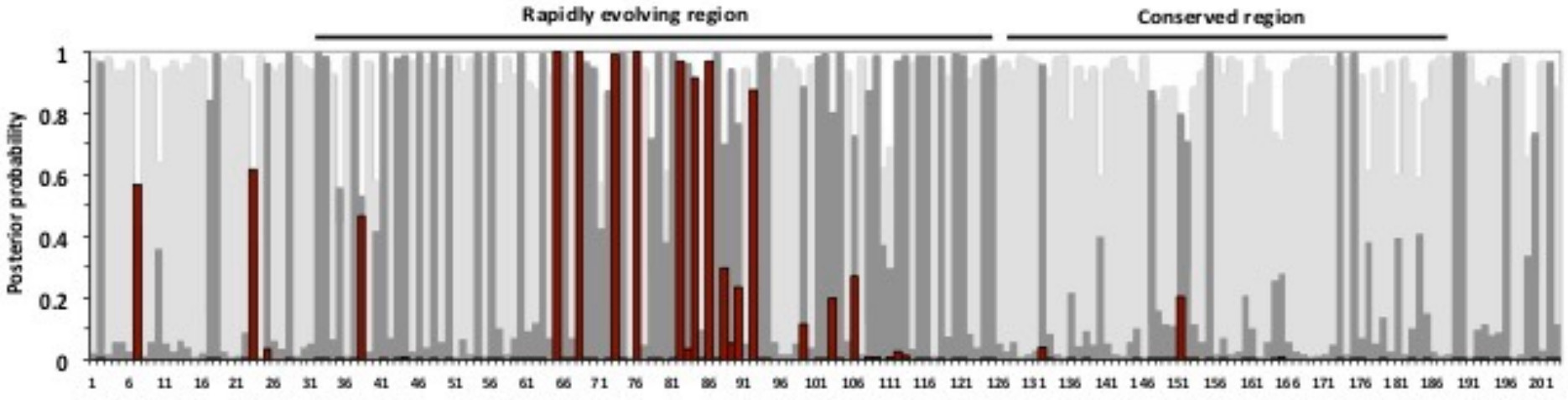
structure:
sites are in contact



1. Fit model to data → MLEs
2. Test hypotheses via Null and alternative models for ω
3. Predict which sites have $\omega>1$
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?