**I. Header**

**1.1 topic**

***Sub-topic***

* Notes
* notes

**I. Input Files**

**1.1 Sequence File**

**1.2 Tree File**

***Branch/Node Labels***

* Clock = 2 or 3 – allows 2 or 3 branch rate groups
* Model = 2 or 3 (codeml) – allows d/ gourps to have different ω
* 2-ratio
* 3-ratio
* Identify a branch w/ ‘#’
* #0 – ω0, default label for all branches
* #1 – ω1
* Can also label entire clades using ‘$’
* Rules:
* # > precedence over $
* Tips > precedence over clades
* ((((rabbit, rat) $2, human #3), goat\_cow) $1, marsupial)
* $1 is applied to entire clade, except humans (#>$)
* Check tree in TreeView or NJPlot to make sure labels are correct

***Fossil Calibration***

* @(date) in front of clade

**1.3 BaseML Control File**

***Option*** – description

* 0: changes
* 1: changes

***seqfile, outfile, and treefile***—names of the sequence data file, main

result file, and the tree structure file, respectively. **No spaces, weird characters**

***Noisy*** – (0-9) controls how much output you want on the screen

If the model being fitted involves much computation, you can choose a large number for noisy to

avoid loneliness.

***Verbose*** – (0-2) how much output in control file

***Runmode*** –

* **0**: evaluates tree topologies from tree file
* 1: heuristic tree search by star-decomposition algorithm, estimate tree from treefile
* 2: “ “ “ “ “ , starts from star tree
* 3: stepwise addition
* 4: NNI perturbation w/ starting tee f/ maximum parsimony
* 5: “ “ “ starting tree from treefile
* \*\*\*Use runmode =0, algorithms don’t work that well

***model*** – specifies model for nt subs

* 0-8: JC69, K80, F81, F84, HKY85, T92, TN93, REV (also known as GTR), and UNREST
* 9: REV special
* 10: special unrestricted

***Mgene*** – combined w/ option G in sequence data file – combined analysis of data f/ multiple genes/site partitions

* 0: if option G is not used in data file
* 1: if option G is used

***clock*** – molecular clock?

* 0: no clock, rates vary
* 1: global clock – all branches same rate
* 2: local clock – branches have several rate gourps
* Use (# & $) for program to estimate rates for those groups
* 3: multiple gene/partition data – must specifiy use option G & Mgene
* Note: 1-3 use rooted tree
* 5-6: use ad hoc rate smoothing procedure

***Fix\_kappa*** – is k at K80, F84, or HKY85 have fixed value

* 0: estimate kappa, initial value can be given under kappa
* 1: kappa (k) is fixed value
* **Kappa =** fixed or initial value

***Fix\_alpha*** – alpha refers to parameter ‘a’ of gamma dist. For variable subs rates across sites

* 0: infinity, estimates alpha
* 1: alpha is fixed
* **Alpha = 0:** single rate for all (infinity),
* Alpha = positive value: specifies value for discrete (gamma) model
* ncatG number of catergories for discrete-gamma model (baseml)

***Fix\_rho*** – independence/correlation of rates at adjacent sites, p (rho) is correlation parameter of auto-discrete gamma model

* 0: independent rates for all sites
* 1: independent rates for all sites, **p = 0**

***nparkK*** – specifies nonparametric models for variable and Markov-dependent rates across sites

* 0: changes
* 1: several (ncatG) categories of independent rates for sites, each category has = probability
* 2: several (ncatG) categories of independent rates for sites, unrestricted probability
* 3: rates are Markov-dependent at adjacent sites, each category has = probability
* 4: rates are Markov-dependent at adjacent sites, unrestricted probability

***nhomo*** – (baseml only) – frequency paramters in some subs models

* 0: estimates by avgs of observed freqs
* 1: homogeneous model, estimates frequency parameters
* 3-5: F84, HKY85, T92 – nonhomogeneous model

***getSE*** – estimates standard errors of estimated parameters (crude estimates)

* 0: ignores
* 1: estimates
* **Note:** LRT should be preferred

***RateAncestor*** – for variable rate models, calculates rates for sites along the sequence & performs marginal ancestral reconstruction

* 0: ignore
* 1: reconstruct ancestor sequences, **runmode = 0**

***Small\_Diff*** – small value used in difference approximation of derivatives

***cleandata*** – ambiguity characters?

* 0: uses all sites
* 1: removes ambiguity data (undetermined nt & gaps)

***method*** – iteration algorithm for estimating branch lengths

* 0: old algorithm 🡪 updates all parameters simultaneously
* 1: new algorithm 🡪 updates branch lengths one by one **(clock =0)**

***icode*** – genetic code for ancestral reconstruction of CDS sequences.

* ?

***readpattf*** – forces prog to read site pattern freqs instead of sequenced data (**rarely used**)

* 0: ignores
* 1: uses

***Fix\_blength*** – what to do with branch lengths

* 0: ignore branch lengths
* -1: start from random starting point (if multiple local optima)
* 1: use branch lengths as initial values for ML iteration
* **Avoid using branch lengths from parsimony analysis from PAUP**
* 2: branches fixed as given in treefile

***Output*** – The output should be self-explanatory. Descriptive statistics are always listed.

The observed site patterns and their frequencies are listed, together with the proportions

of constant patterns. Nucleotidefrequencies for each species (and for each gene in case

of multiple gene data) are counted and listed. lmax = ln(Lmax) is the upper limit of the

log likelihood and may be compared with the likelihood for the best (or true) tree under

the substitution model to test the model's goodness of fit to data (Goldman 1993; Yang,

Goldman, and Friday 1995). You can ignore itif you don’t know what it means. The

pairwise sequence distances are included in the output as well, and also in a separate file

called 2base.t. This is a lower-diagonal distance matrice, readable by the NEIGHBOR

program in Felesenstein's PHYLIP package (Felsenstein 2002). For models JC69, K80,

F81, F84, the appropriate distance formulas are used, while for more complex models,

the TN93 formula is used. baseml is mainly a maximum likelihood program, and the

distance matrix is printed out for convenience and really has nothing to do with the later

likelihood calculation.

**1.4 CodeML (Codons) Control File**

***codonFreq*** – equilibrium codon freqs in codon subs model.

* 0: (1/61 each for standard code), 0 parameters
* 1: (calculated from avg nt freqs), 3 parameters
* 2: (f/ avg nt freq at 3 codon positions), 9 parameters
* 3: (free parameters), 60 parameters

***aaDist***– are AA seq assumed?

* 0: assumed
* +: geometric
* - : linear
* 1 – 6: Graman’s, Miyata, c,p,v,a

***runmode*** – ML estimates of ds & dn in pairwise

* -2: estimates dn & ds in pairwise comparison

***Model*** – assumptions about ω ratios among branches

* 0: one ω for all lineages (branches)
* 1: one ratio for each branch (free-ratio)
* 2: arbitrary number of ratios – **specify NSsites#**
* Must group branches using symbols # & $ on treefile
* Fix\_omega fizes last ratio (ωk-1, for k ratios) at value of ω specified

***NSsites*** – specifies model that allow ω ratio to vary

* m: model Mm
* variable ncatG – specifies # of categories in ω distribution
* can specify several models at once: **NSsites = 0 1 2 3 7 8**
* 0: model M0 (one ω ratio)
* 1: model M1a (neutral ratio) 0 < ω0 <1
* 2: model M2a (positive selection) 0 < ω0 <1
* 3: model M3 (discrete)
* 4: model M4 (frequency)
* 5: model M5 (gamma)
* 6: model M6 (gamma2)
* 7: model M7 (beta)
* 8: model M8 (beta&ω)
* 9: model M9 (beta&gamma)
* 10: model M10 (beta&gamma+1)
* 11: model M11 (beta&normal>1)
* 12: model M12 (0&2normal>1)
* 13: model M13 (3normal>0)

***icode*** – specifies genetic code

* 0: universal
* 1: mammalian mt.
* 2: yeast mt.
* 3: mold mt.
* 4: invert mt.
* 5: ciliate nuc
* 6: echinoderm mt.
* 7: euploid mt.
* 8: alt. yeast nuc
* 9: ascidian mt.
* 10: blepharisma nuc
* 11: Yang’s regularized code

***RateAncestor*** – reconstructs ancestor seqs

* 0: ignore
* 1: reconstruct ancestor sequences

***Output*** for codon sequences (seqtype= 1): The codon frequencies in each sequence are counted and listed in a genetic code table, together with their sums across species. Each table contains six or fewer species. For data of multiple genes (option G in the sequence file), codon frequencies in each gene (summed over species) are also listed. The nucleotide distributions at the three codon positions are also listed. The method of Nei and Gojobori (1986) is used to calculate the number of synonymous substitutions per synonymous site (dS) and the number of nonsynonymous substitutions per nonsynonymous site (dN) and their ratio (dN/dS). These are used to construct initial estimates of branch lengths for the likelihood analysis but are not MLEs themselves.

**1.5 CodeML (Amino Acids) Control File**

**II. Models $ Analyses**

* 1. **General Info**

ƒ ℓ θ ω ρ π

***Maximum likelihood estimate* (MLE) –** prob of observing the data (X) when viewed as a function of the unknown parameters (θ)

* L(θ, X) = ƒ(θ | X)
* Likelihood function has all info in data about parameters (θ)
* Best MLE of θ, is given by the θ that maximizes the Likelihood (L) or log likelihood
* ℓ(θ, X) = log[L(θ, X)]
* top of the curve
* curve also provides uncertainty info

***Likelihood Ratio Test (LRT)***

* Null model (H0) has parameters (ρ0), with optimal (log likelihood) values (ℓ0)
* Alternate model (HA) has parameters (ρ1) with optimal (log likelihood) values (ℓ1)
* Null model: Twice the log likelihood difference has a chi (χ2) distribution, asymptotically
* 2Δℓ = 2(ℓ1 - ℓ0), with df = ρ1 - ρ0
* So, the test statistic 2Δℓ can be compared to χ2 distribution. If significantly different, reject Ho.

***Assumptions of the models/programs***

* Substitutions occur independently in d lineages
* Subs occur independently among sites (save for gamma-discrete)
* Process of subs is described by time-homogenous Markov process
* Process of subs is assumed stationary – freq of nt (baseml), codons (codeml) or amino acids (aaml) remain constant over time period covered by data
* Existence of molecular clock (rate constancy among lineages) not necessary, but can be imposed.
* Variation (and dependence) of rates at sites is allowed by discrete-gamma models

**2.2 Nucleotide Sub Models (baseml)**

***General*** –

* JC69 (Juckes and Cantor 1969)
* K80 (Kimura 1980)
* F81 (Gelsenstein 1981)
* F84 (Felsenstein/DNAML prog. 1984)
* HKY85 (Hasegawa, Yano, Kishino 1984)
* Tamura 1992
* Tamura and Nei 1993
* REV/GTR (General Time Reversible) (Yang 1994b, Zharkikh 1994)

***Transitions/Transversion Ratios*** – what

* Kappa (k) = average transition/transversion ratio
* Calculations change based on substitution model

**2.3 Codon Substitution Models (Codeml)**

***Basic Model*** – each codon is unit of evolution

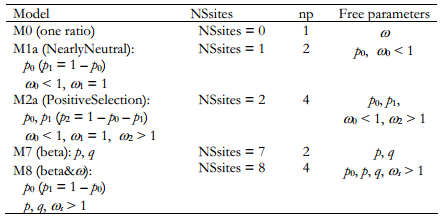
* Model = 0
* qij = Codon (i) vs. codon (j) =
* 0 – 2 codons differ at >1 position
* πj – S transversions
* kπj – S transition
* ωπj – NS transversion
* ωkπj – NS transition
* can be calculated from nt frequencies at 3 codon position (control variable CodonFreq)
* ω = dN/dS – NS to Syn sub rates
* ω < 1 – Negative (Purifying) Selection
* ω = 1 – Neutral Selection/Evolution
* ω > 1 – Positive (Directional) Selection
* **Note:** average ratio over all sites & all lineages is never >1, positive selection does not affect all sites for a prolonged time
* M=0 (one ratio), NSSites = 0

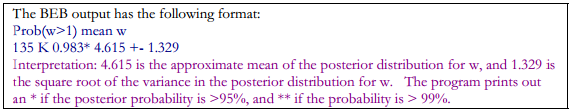
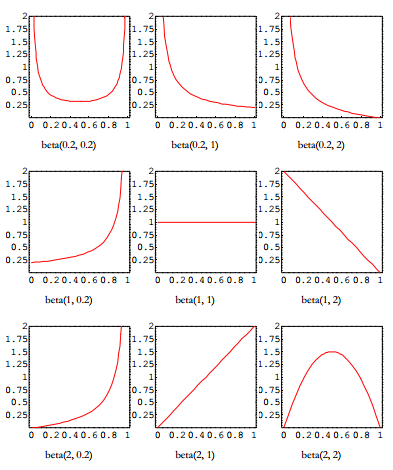
***Branch Model*** – allows ω ratio to vary among branches/lineages

* Model = 1, Variable model
* Independent ω for each branch
* Parameter rich & discouraged
* Model =2
* Specify how many ratios, and which branches have them (up to 4)
* #1 (leaves/tips, dominant), $1 (clades, subordinate)

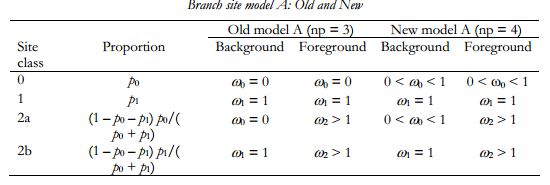
***Site Model*** – Allows ω to vary among sites

* Use variable: NSsites



* Old models M1, M2:
* ω0 = 1­
* ω1 = 1
* New models M1a (nearly neutral) & M2a (Positive Selection)
* 0 < ω0 < 1­
* ω1 = 1 – avoid false positives. Sites under weak selection will be absorbed into this neutral class rather than being claimed under positive selection
* performs better than M3
* NSsites specifies site models (use model = 0)
* Several models of NSSites can be specified using space delimitations
* Ex: NSsites = 0 1 2 7 8 will run 5 models
* Run **2 LRT for positive selection**:
* M1a (nearly neutral) vs. M2a, chi2 with df =2
* M7 vs. M8, chi2 with df = 2
* **LRT significant**:
* Empirical Bayes will calculate Posterior Probabilities
* Old: Naïve empirical Bayes (NEB),
* Uses MLR of parameters, ignores sampling error
* New: Bayes empirical Bayes **(BEB),** i
* Uses M2a and M8
* **Output**:
* Beta distribution – beta (p, q) is flexible distribution for range (0,1)
* x-axis (ω)
* y-axis (number/proportion of sites with that ratio, ω)
* 9 probabilites according to parameters (p,q)
* 

***Branch-Site Model*** – Yang, Nielson (2002), models that let ω ratio vary along sites & lineages. Attempts to detect positive selection that affects only a few sites along a few lineages.

* Model A: model =2, NSsites =2
* 
* Model B: model = 2, NSsites =3

***Clade Model*** – what

* Model C: model =3, NSsites = 2
* Model D: model =3, NSsites = 3
* ncatG: specify number of site classes
* Clade model D can work with ncatG = 3 or 2, but branch-site model A, B, and clade model C works only with ncatG = 3.
* The BEB procedure is implemented for clade model C but not for model D. You should use model C in combination with the BEB procedure. Ignore the NEB output.

**2.4 Nucleotide Sub Models**

***Sub-header***– what

* notes