

LIKELIHOOD IN MOLECULAR PHYLOGENETICS

Peter G. Foster
The Natural History Museum, London

Lausanne, September 2003

The likelihood supplies a natural order of preferences among the possibilities under consideration.

-R.A. Fisher, 1956

Likelihood in molecular phylogenetics

- Why use likelihood?
- Simple likelihood calculations
- Choosing a model
- Practicals using PAUP

Reference 0

Swofford, Olsen, Waddell, and Hillis, 1996. *in Hillis et al, Molecular Systematics.*

Why use likelihood?

- Models take into account branch lengths
 - Accurate branch lengths even if there are superimposed hits (*ie* more than one mutation at the same site)
- models are explicit
 - assumptions are stated, not hidden
- You can make the model fit the data
- likelihood is efficient and powerful
 - it uses all the data

You can make the model fit the data

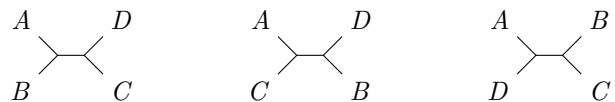
If the data ...

- have an unusual composition
 - have a transition/transversion ratio different from 1
 - have both quickly and slowly evolving sites
- ...you can use that information in your model.

Likelihood uses all the data

A acgcaa
B acataa
C atgtca
D gcgtta

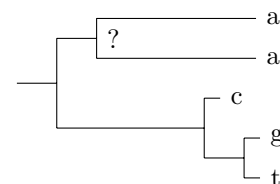
- There are no parsimony informative sites in these data
- Under unweighted parsimony, all three possible trees have zero length



- Although there are no parsimony informative sites, there appear to have been several evolutionary events, which should provide useful phylogenetic information if we could use it.
- It appears that transitions are more common than transversions
- The constant site provides useful information regarding the tendency of the a to stay the same.
- If we use this information, then one tree is more optimal than the other two.

What is the ancestral state?

Consider one site on this tree, with these character states—



- Ancestral state “a” is most parsimonious
- Wide character state variation together with short branches tells us that this is a fast site—so we should expect a large amount of change over the long branches.
- Likelihood is equivocal about the ancestral state (it could be anything)

Branch lengths under parsimony and likelihood

- Parsimony considers that you would have the same expectation that a character would change along both long and short branches.
- Likelihood and distance methods, using models, consider that change is more probable along long branches than along short branches.

Molecules do not evolve like morphological characters

- Molecular sequences appear to evolve mostly by random change, with a small amount of selection.
- This behavior can be described well by stochastic models which incorporate among-site rate variation.
- This allows us to use probabilistic methods in our analyses
 - and puts our analyses on a sound statistical footing.

Likelihood is appropriate for data generated by a random process

These data were probably *not* generated by a random process

```
000000000000
010301001000
222022100100
131130010011
```

- There are no constant sites.
- There is an obvious ancestral taxon.
- Some characters are binary, some are multi-state

Simple likelihood calculations

Likelihood

In general...

The likelihood is the probability of the data given the model.

In phylogenetics, we can say (loosely) that the tree is part of the model

The likelihood is the probability of the data given the tree and the model.

Flip a coin– get a “head”

What is the likelihood of that data?

- The likelihood depends on the model
- If you think its a fair coin, the likelihood of the data is 0.5
- If you think it is a two-headed coin, the likelihood of the data is 1.0
- ...So the model that you use can have a big effect on the likelihood

Likelihood calculations

- In molecular phylogenetics, the data are an alignment of sequences
- We optimize parameters and branch lengths to get the maximum likelihood
- Each site has a likelihood
 - this differs depending on the model and tree
- The total likelihood is the product of the site likelihoods
 - or the sum of the log of the site likelihoods
- The maximum likelihood tree is the tree topology that gives the highest (optimized) likelihood under the given model.
- We use reversible models, so the position of the root does not matter.

Reference 1

P. G. Foster 2001. “The Idiot’s Guide to the Zen of Likelihood in a Nutshell in Seven Days for Dummies, Unleashed”

- Read this only if you want to know where the numbers come from
- Elementary likelihood calculations and definitions
- Probability and rate matrices
- Finding the maximum likelihood branch length
- Calculating likelihood values on a tree
- Checking that PAUP* gets the correct likelihood values

Choosing a model

- Don’t “assume” a model
- Rather, find a model that fits your data.

Models are described in terms of...

- the tendency of one base to change to another
- the composition
- site-to-site rate variation

Models often have “free” parameters. These can be fixed to a reasonable value, or estimated by ML.

Tendency of one base to change to another

- This can be described by a rate matrix
- The most complex in PAUP is the GTR, general time-reversible
- Other models are simplifications of this
 - HKY, F81, K2P, JC, *etc* ...

GTR: General time-reversible model

$$\mathbf{R} = \begin{bmatrix} - & a & b & c \\ a & - & d & e \\ b & d & - & f \\ c & e & f & - \end{bmatrix}$$

- Symmetrical, so time-reversible
- There are 6 substitution types (`lset nst=6`), so 5 free parameters
- You can restrict these using the `rclass` subcommand in `lset`
 - eg `lset rclass=(a b c c b a)` to make a subset with only 3 substitution types
 - The program `modeltest` uses `rclass` a lot, see the file `modelblock3`

Base frequencies (composition)

- equal
- specified
- empirical
 - often a good approximation to ML-estimated, and much faster
- estimated by ML
- For DNA, there are 4 compositions, so 3 free parameters

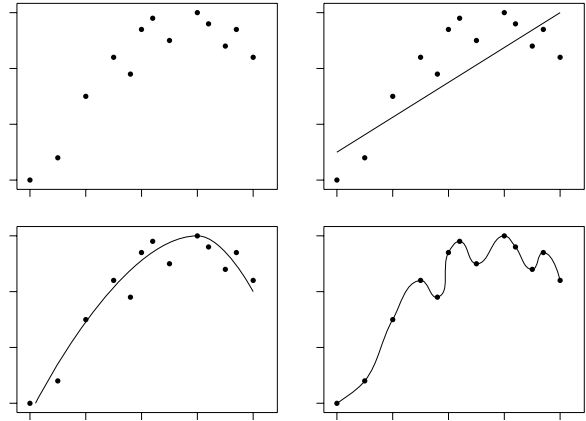
Among-site rate heterogeneity

- pInvar
- gamma-distributed variable sites
 - has an average rate of 1.0
 - shape can change greatly with only one parameter (α , `shape` in PAUP)
 - approximated with a discrete gamma distribution with `nCat` divisions
- pInvar + gamma
- site-specific
 - good for codons

Parameters

- Models differ in their free, *ie* adjustable, parameters
- More parameters are often necessary to better approximate the reality of evolution
- The more free parameters, the better the fit (higher the likelihood) of the model to the data. (Good!)
- The more free parameters, the higher the variance, and the less power to discriminate among competing hypotheses. (Bad!)
- We do not want to “over-fit” the model to the data

What is the best way to fit a line (a model) through these points?



How to tell if adding (or removing) a certain parameter is a good idea?

- Use statistics
- The null hypothesis is that the presence or absence of the parameter makes no difference
- In order to assess significance you need a null distribution

Is it worth adding a parameter? — An example

We have some DNA data, and a tree. Evaluate the data with 3 different models.

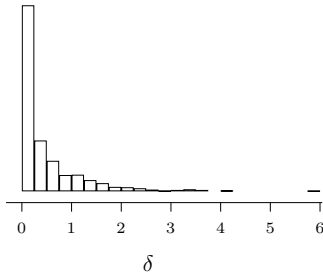
model	ln likelihood	Δ
JC	-2348.68	
K2P	-2256.73	91.95
GTR	-2254.94	1.79

- Evaluations with more complex models have higher likelihoods
- The K2P model has 1 more parameter than the JC model, the `tRatio`.
- The GTR model has 4 more parameters than the K2P model
- Are the extra parameters worth adding?

Is the K2P model better than the JC model for these data?

- Null hypothesis (generally): the extra parameter does not make any difference
- Null hypothesis (specifically): the tree and the JC model
- We need to know how much of an improvement in likelihood we can expect from true null hypothesis data when we add the `tratio` parameter
 - The increase in likelihood will be due to noise, only
- We need a null distribution, which we can get by simulating data many times under the null hypothesis

- Evaluate the likelihood of each simulated data set with both the JC and the K2P models
- Keep the log likelihood differences—they are the null distribution

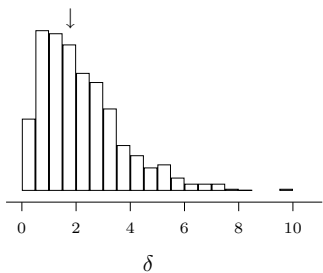


We have generated many true null hypothesis data sets and evaluated them under the JC model and the K2P model. 95% of the differences are under 2. The statistic for our original data set was 91.95, and so it is highly significant. In this case it is worthwhile to add the extra parameter (**tRatio**).

Is the GTR model better than the K2P model?

model	ln likelihood	Δ
JC	-2348.68	
K2P	-2256.73	91.95
GTR	-2254.94	1.79

- Null hypothesis: the tree and the K2P model
- Evaluate the likelihood of each simulated data set with both the K2P and the GTR models
- Keep the log likelihood differences—they are the null distribution



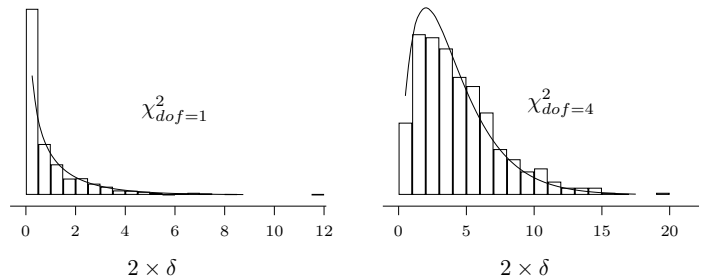
We have generated many true null hypothesis data sets and evaluated them under the K2P model and the GTR model. The statistic for our original data set was 1.79, and so it is not significant. In this case it is not worthwhile to add the extra parameters.

You can use the χ^2 approximation to assess significance of adding parameters

- When comparing nested models, twice the difference in log likelihoods is approximately χ^2 distributed
- The 95% point for JC vs K2P
 - by the simulation: 3.73
 - by $\chi^2_{df=1}$: 3.84

- The 95% point for K2P vs GTR
 - by the simulation: 10.44
 - by $\chi^2_{df=4}$: 9.49

You can use the χ^2 approximation to assess significance of adding parameters



Note that the χ^2 rule only works for nested models

AIC—Akaike Information Criterion

- Another approach to asking whether the log likelihood ratio is significant
- It applies to non-nested models, as well as to nested models
- The AIC says that adding a useless parameter generally increases the log likelihood by about 1 unit.
 - So if adding a parameter increases the log likelihood by more than 1, it is not useless.

AIC

model	ln L	nParams	ln L – nParams
JC	-2348.68	0	-2348.68
K2P	-2256.73	1	-2257.73 ←
GTR	-2254.94	5	-2259.94

- We penalize each likelihood by the number of free parameters
- By the AIC, the K2P model is the best model

Reference 2

D. Posada and K. A. Crandall 1998. “MODELTEST: testing the model of DNA substitution” *Bioinformatics* 14: 817-818.

- Automates the process of choosing a model
- Uses PAUP to do the likelihood calculations
- Uses the AIC for comparison of non-nested models
- Incorporates some corrections to using χ^2 significance of the likelihood ratio
- site-specific rate variation is *not* covered

Choosing a model— “maxing out”

- Often you will need the most complex model, GTR+I+G, or GTR+SS.
- It makes you think that you could improve the model by adding other parameters
- ... and you might be right.

Among-site rate variation

- It is usually important that among-site rate variation be modeled
- Failure to do so will lead to underestimated distances and long-branch attraction
- Use pInvar or gamma-distributed variable rates, or a combination of the two
- ... or site-specific (ss) rates

Comparing tree topologies using likelihood

- When we compare trees, we can't use the same strategy that we used to compare nested models
- The Kishino-Hasegawa test to compare trees has been in use for a dozen years.
- However, the KH test has problems
- A similar but better test is the recent Shimodaira-Hasegawa test

Reference 3

Goldman, Anderson, and Rodrigo 2000. Likelihood-based tests of topologies in phylogenetics. *Syst. Biol* 49: 652–670.

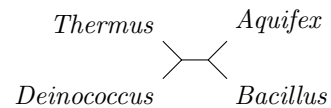
- A critique of the Kishino-Hasegawa test
- An explanation of various ways of comparing tree topologies with likelihood

Comparing tree topologies using the SH test

- The Shimodaira-Hasegawa test can tell you whether sub-optimal trees are significantly worse than the ML tree.
- If sub-optimal trees are *not* significantly worse than the ML tree (often the case!) then the ML tree is not a strong hypothesis, perhaps because the data are weak.
- You often can get reasonable sub-optimal trees using topologically-constrained searches.

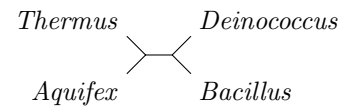
Convergent composition

- *Deinococcus* are radiation resistant bacteria.
- *Thermus* are thermophilic
 - There is good evidence for a close phylogenetic relationship between *Deinococcus* and *Thermus*
- *Aquifex* is another thermophile, and *Bacillus* is a mesophile
 - Neither is closely related to either *Deinococcus* or *Thermus*



- We can take this as the “true” topology.

- However, the two thermophiles share a compositional bias, and group together, giving the wrong tree with many phylogenetic methods.



The shared compositional bias of *Aquifex* and *Thermus* is so strong that the true phylogenetic signal is masked, and the two taxa “attract” each other in the tree

Shimodaira-Hasegawa test of the 3 possible trees with these 4 taxa

SH test using RELL bootstrap (one-tailed test)
Number of bootstrap replicates = 1000

Tree	-ln L	Diff -ln L	P
attract	3983.00041	(best)	
true	3985.30568	2.30526	0.465
other	3995.26719	12.26677	0.027*

- Maximum likelihood with the GTR+G model erroneously finds the “attract” tree as the best tree
- However, the true tree cannot be rejected under this model

Likelihood practicals

- Use PAUP blocks
 - PAUP references
- Choosing a model
- Fast heuristic search
- Quickies ...
 - ModelTest
 - Site-specific rates
 - SH test
 - Constrained searches

No more Mr Nice GUI

- Use a PAUP block
- It is often easier and faster to use commands that you have already typed in
- It is easy to repeat exactly, or with a small variation
- It is a record of what you did
- Often you will run long analyses in the background, where the results of one analysis are fed to the next analysis

PAUP block

You will never use something this simple for a search, but here is an example that shows a few elements of a PAUP block.

```
#NEXUS

begin paup;
  log start file=myPaupOutLog;
  execute yourDataFile.nex;
  set criterion=likelihood;
  hsearch;
  log stop;
  savetrees file=mlTree.nex;
  quit;
end;
```

LSET

This PAUP block reads in a data file, finds the MP tree(s), and uses the first tree from that search to compare the JC+I model with the JC+I+G model.

```
begin paup;
  execute yourDataFile.nex;
  hsearch;
  set criterion=likelihood;
  lset nst=1 basefreq=equal pInvar=estimate;
  lscore 1;
  lset rates=gamma shape=estimate;
  lscore 1;
end;
```

Get the current PAUP settings...

In interactive PAUP, not in a paup block

```
lset ?
lscore ?
hsearch ?
describetrees ?
```

LSET ?

See Figure 1.

Read the manuals

- PAUP comes with ...
 - command reference (pdf)
 - tutorial (pdf)

- In addition, see ...
 - FAQ, inside <http://paup.csit.fsu.edu/>
 - PAUP Forum, same place

Choosing a model for bacterial 16S data

- Start with the nj tree
- Choose among JC, F81, HKY, and GTR models
- Choose among-site rate variation

Preliminaries...

```
log start file=paupOut replace;
execute bacterial_16S.nex;
nj;
set criterion=likelihood;
```

This starts a log file, reads in the data, does a quick neighbor-joining tree with p-distances, and then sets the optimality criterion to likelihood.

Try the Jukes-Cantor model

```
lset nst=1 basefreq=equal;
lscore 1;
```

This sets the number of substitution types to 1, and the composition to 25% each nucleotide. This is the JC model. Then the nj tree is evaluated and branch lengths optimized under this model. The log likelihood is -5211.7.

Is the composition not equal?

```
lset nst=1 basefreq=estimate;
lscore 1;
```

This is the F81 model (Felsenstein, 1981), the same as the JC model except that the composition parameters are allowed to be free. There are 4 nucleotides, so there are 3 parameters and 3 degrees of freedom. The log likelihood is -5166.6

	$\ln L$	Δ
JC	-5211.7	
F81	-5166.6	45.2

Is there a transition/transversion ratio bias?

```
lset nst=2 tratio=estimate basefreq=estimate;
lscore 1;
```

This is the HKY85 (Hasegawa, Kishino, Yano, 1985) model. Now $nst=2$, with one additional parameter.

	$\ln L$	Δ
JC	-5211.7	
F81	-5166.6	45.2
HKY85	-5125.0	41.6

paup> lset ?

Usage: LSet [options...] ;

Available options:

```

Keyword ---- Option type ----- Current default setting --
NST          1|2|6                    2
TRatio       <real-value>|Estimate|Previous  2
RMatrix      (<rAC><rAG><rAT><rCG><rCT>)|
              Estimate|Previous            (1 1 1 1 1)
RClass       (<cAC><cAG><cAT><cCG><cCT><cGT>)    (a b c d e f)
Variant      HKY|F84                    HKY
BaseFreq     Empirical|Equal|Estimate|Previous|
              (<frqA><frqC><frqG>)          Empirical
...<more>...

```

Figure 1: How to get the current likelihood settings.

Now try the GTR model

```

lset nst=6 rmatrix=estimate basefreq=estimate;
lscore 1;

```

The GTR (nst=6) model has 5 more parameters than the F81 model, and 4 more than the HKY85 model. It is the most parameter-rich rate matrix that PAUP has to offer, and is the one chosen for our data.

	ln L	Δ	
JC	-5211.7		
F81	-5166.6	45.2	
HKY85	-5125.0	41.6	
GTR	-5092.5	32.4	⇐ choose this

Among-site rate variation I

```

lset nst=6 rmatrix=estimate
      basefreq=empirical pinvar=estimate;
lscore 1;

```

Allowing a proportion of sites to be invariant (pinvar=estimate) adds one free parameter.

	ln L	Δ
GTR	-5092.5	
GTR +I	-4946.1	146.4

Among-site rate variation II

```

lset nst=6 rmatrix=estimate
      basefreq=empirical pinvar=0.0
      rates=gamma shape=estimate;
lscore 1;

```

Allowing Γ -distributed rates adds one free parameter over the GTR model.

	ln L	Δ
GTR	-5092.5	
GTR +I	-4946.1	146.4
GTR + Γ	-4937.8	154.7

Among-site rate variation: III

```

lset nst=6 rmatrix=estimate
      basefreq=empirical pinvar=estimate
      rates=gamma shape=estimate;
lscore 1;

```

	ln L	Δ	
GTR	-5092.5		
GTR +I	-4946.1	146.4	
GTR + Γ	-4937.8	154.7	⇐ choose this
GTR +I + Γ	-4937.2	0.6	not significant

Assessing models with the AIC

See Table 1.

Heuristic search strategies

- It is too time-consuming to estimate parameters (other than branch lengths) while searching.
- Parameters should be fixed to reasonable values before searching.

Don't do this...

```

lset pinvar = estimate
      shape = estimate;
hsearch;

```

Do this ...

```

lset pinvar = 0.213
      shape = 0.679;
hsearch;

```

Or this ...

```

lset pinvar = previous
      shape = previous;
hsearch;

```

Table 1. Choosing a model with the AIC

	$\ln L$	n	$-\ln L + n$	$2(-\ln L + n)$	
JC	-5211.7	0	5211.7	10423.4	
F81	-5166.6	3	5169.6	10339.2	
HKY85	-5125.0	4	5129.0	10258.0	
GTR	-5092.5	8	5100.5	10201.0	
GTR +I	-4946.1	9	4955.1	9910.2	
GTR + Γ	-4937.8	9	4946.8	9893.6	\Leftarrow choose lowest
GTR +I + Γ	-4937.2	10	4947.2	9894.4	

n is the number of free parameters.

A problem

- You search with parameters optimized on a tree
- On searching, you may find a different tree
- There is a possibility that a search based on parameters optimized on the new tree will find a better tree
 - so one search is not good enough
- If we had very fast computers, we would search and optimize at the same time, and avoid this problem.

Search by successive iteration

- Optimize parameters (eg `shape=estimate`) to reasonable values on a single tree.
 - Fix parameters (eg `shape=previous`), and search.
 - Re-optimize parameters based on the best tree from the search.
 - Repeat ...
 - fix parameters
 - search
 - re-optimize parameters
- ...until things don't improve anymore.

How to get the initial parameters?

- Rule of thumb: Parameters for reasonably good trees do not differ much.
- Start with a reasonably good tree, eg a NJ or MP tree, and use that to get valid initial parameters.

Branch swapping

- NNI is fastest, but least complete
- SPR is intermediate
- TBR is best, most complete, but slowest

A fast heuristic search strategy

- Successive iteration
- Quickly get the model parameters close to their final values using inexpensive NNI and SPR branch swapping

- Then one round of TBR branch swapping, with `lset approxlim=2`
 - The default is `approxlim=5`. When searching, PAUP will only fully evaluate candidate trees where an approximate likelihood calculation places the candidate within the `approxlim` of the best tree found so far.
- Finally, all that is needed is to finish with one round of expensive TBR branch swapping, with `approxlim=5`.
- This strategy can be much faster than successive iteration using only TBR branch swapping.

Fast heuristic search

Replace the content <between brackets> as appropriate ...

```
begin paup;
  nj;
  set crit=like;
  lset <xxx=estimate>;
  lsc 1;
  lset <xxx=previous>;
  hsearch start=1 swap=nni;
  <repeat with swap=spr (2x),
    tbr (lset approxlim=2),
    and tbr (lset approxlim=5)>
  [finish with one last optimization, to be sure
   that the successive iteration is finished]
  lset <xxx=estimate>;
  lsc 1;
end paup;
```

Quickies

ModelTest

- Execute your data file in PAUP
- default `lscores longfmt=yes`;
- Execute `modelblock3`, which makes
 - `model.scores`
 - `modelfit.log`
- Run `model.scores` through `modeltest`, which assesses the likelihood scores by the LRT and AIC and suggests a model for each

- Does not do site-specific rates, nor the molecular clock.
 - If you want to test these models “by hand” and compare to the ModelTest results, save the tree that ModelTest uses

Site-specific rates (one way)

```
begin sets;
  charPartition p0 = first: 1-.\3,
                    second: 2-.\3,
                    third: 3-.\3;
end;

begin paup;
  set crit=like;
  lset rates=sitespec siterates=partition:p0;
  lscore 1;
end;
```

Shimodaira-Hasegawa test

```
begin paup;
  gettrees file=myTrees.nex;
  set crit=like;
  lscore all / shtest=rell;
end;
```

Topologically constrained searches

Lets say we have several taxa: A1-A4, B1, C1 ... We want to do an heuristic search but only consider trees where all the “A” taxa go together.

```
begin paup;
  constraints monoA = ((A1, A2, A3, A4));
  hsearch enforce constraints=monoA;
end paup;
```

A Bayesian Approach to Phylogenetics

A Bayesian approach compared to ML

- In ML, we choose the hypothesis that gives the highest (maximized) likelihood to the data
- The likelihood is the probability of the data given the hypothesis.
- A Bayesian analysis expresses its results as the probability of the hypothesis given the data.
 - this may be a more desirable way to express the result
- Both ML and Bayesian methods use the likelihood function
 - In ML, free parameters are optimized, maximizing the likelihood
 - In a Bayesian MCMC approach, free parameters are probability distributions, which are sampled.

Bayesian analysis in phylogenetics is recent

- The first papers and demonstration programs for phylogenetics were in the mid-1990’s
- The first practical program was BAMBE, by Larget and Simon, in 1999
- In 2000 MrBayes was released, a program by John Huelsenbeck and Fredrik Ronquist

Posterior and prior probabilities

- Bayesian analysis expresses its result as the *posterior probability density* of the tree topologies and model parameters
- The posterior probability is proportional to the likelihood, and also proportional to the *prior probability*
 - In our analyses we usually do not have a strong prior opinion, and so the prior probability has no influence on the result.
 - Rather, the result is driven by the likelihood

$$P(t_i|X) = \frac{P(X|t_i)P(t_i)}{\sum_{j=1}^{nTrees} P(X|t_j)P(t_j)}$$

Bayesian analysis in practice

- It is practically impossible to do Bayesian calculations analytically
- Rather, the posterior probability density is approximated by the *Markov chain Monte Carlo* (MCMC) method

Markov Chain Monte Carlo

- After the chain equilibrates, it visits tree space and parameter space in proportion to the *posterior probability* of the hypotheses, *ie* the tree and parameters.
- We let the chain run for many thousands of cycles so that it builds up a picture of the most probable trees and parameters
- We sample the chain as it runs and save the tree and parameters to a file
- The result of the MCMC is a sampled representation of the parameters and tree topologies
- The samples mostly come from regions of highest posterior probability

How the MCMC works

- Start somewhere
 - That “somewhere” will have a likelihood associated with it
 - Not the optimized, maximum likelihood

- Randomly propose a new state
 - If the new state has a better likelihood, the chain goes there

If the proposed state has a worse likelihood

- Choose a random number between 0 and 1. If the random number is less than the the likelihood ratio of the two states, then the proposed state is accepted.
- If the likelihood of the proposed state is only a little worse, it will sometimes be accepted
- This means that the chain can cross likelihood valleys

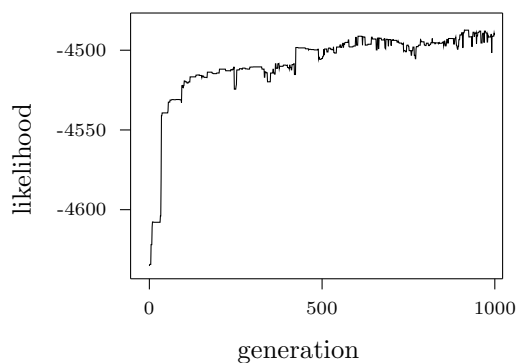
Thinning the chain

- Often proposed states are not accepted, so the chain does not move
- This is not good for getting a good picture of the uncertainty
- Rather than sampling the chain at every generation, the chain is sampled more rarely, *eg* every 10 or every 100 generations.
- These sampled states will more likely be different from each other, and so be more useful.

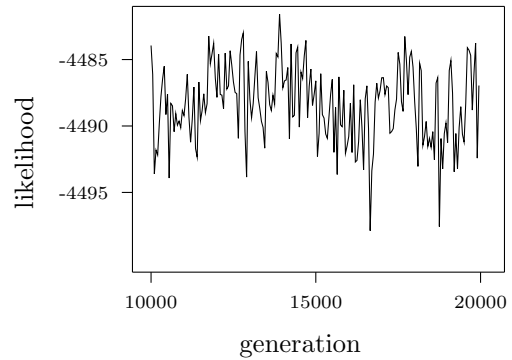
MCMC burn-in

- The chain only works properly after it has equilibrated
- The first samples (100? 1000?) are discarded as “burn-in”
- You can plot the likelihood of the chain to see if it has reached a plateau
 - only use the samples from the plateau
 - this is a widely-used but unreliable way of assessing convergence

MCMC before convergence



MCMC, after convergence



MCMCMC

- MrBayes introduced Metropolis-coupled MCMC
- Several chains run in parallel
- All but one is “heated”
 - increases the acceptance probabilities
 - allows easier crossing of likelihood valleys
- Chains are allowed to swap with the cold chain
- Only the cold chain is sampled

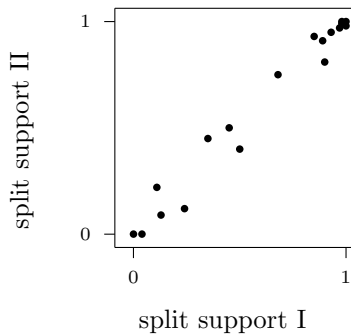
MCMC result

- Sampled trees are written to a file during the chain
- We can summarize those samples
 - The proportion of a given tree topology (after burn-in) in these trees is the posterior probability of that tree topology
 - Trees in the file can be analyzed for tree partitions, from which a consensus tree can be made
 - * The proportion of a given tree partition in the trees is the posterior probability of that partition
- Other parameters are written to a different file
- These continuous parameters may be averaged, and the variance calculated

Assessing convergence

- Commonly:
 - plot the likelihood
 - plot other parameters
 - Can be an unreliable indicator of convergence
- Plot node support?
- Do multiple runs, starting from different random trees
- Convergence can be affected by tuning parameters (props)

Two MCMC runs



Prset

- The prior probability should usually be a distribution
 - uniform
 - exponential
- Can also be fixed to single values
 - generally not a good idea
 - generally we fix the rate matrix for proteins

Problems with Bayesian analysis

- High posterior probability can be given to poorly-supported splits
 - Eg a Bayesian analysis can give you well-supported resolution from a true star tree
- Can be sensitive to taxon sampling
- Can be difficult to be sure it has converged
 - occasionally we see jumps in the likelihood long after apparent convergence
 - tree space coverage problems with large trees

New research in statistical phylogenetics

- identification of selection
- codon models
- heterogeneous models
 - heterogeneous over the data
 - heterogeneous over the tree
 - mixed models
 - heterogeneous models
- likelihood of morphological data
- covarion model
- modeling indels
- correlated sites
- assessment of model fit

Reference 4

Paul Lewis 2001. Phylogenetic systematics turns over a new leaf. *TREE* 16: 30–37.

- Very short history of models
- Codon and secondary structure models
- Likelihood of morphological data
- Bayesian methods, MCMC

Elementary UNIX

GUI and CLI

- The most reliable and powerful way to interact with a Unix machine is with the command line at the terminal.
 - Reliable because it generally works—GUI's may not work over networks.
 - Powerful because you can do more
 - * more options
 - * scripting and automation
 - Often there is no GUI version of a program.
- The CLI is therefore more user-friendly
- You may be working at remote computers
 - Often you do not have a GUI or mouse (you may have X-windows)
- You have to log in
- You get a prompt, which might be [you@yourMachine you]\$
 - the prompt can be customized
- You type commands (programs), at the command line. Often you follow commands with arguments, that modify the command.

```
ls -l  
paup data.nex
```

- You can sometimes use the GUI and CLI together
- Usually you can start up a web browser (eg Mozilla) by double-clicking it in a Unix GUI.
 - You can also start it up at the command line.

```
mozilla &
```
 - Following the command with an & puts it in the background, so that you get your prompt back.

Hierarchical files

- Files and directories (folders) are arranged in a hierarchy or tree, starting with `/`, the root directory
- eg your home directory is often `/home/you`
- You may have other directories in your home directory
 - `/home/you/Documents`
 - `/home/you/bin`
- You can ask where you are in the directory hierarchy with the `pwd` command.

Elementary commands

- You can change directory with the `cd` command.
 - You can use absolute (starting with `/`) or relative paths
 - Change to a remote directory by eg `cd /usr/local/share`
 - Change to a higher directory with `cd ..`
 - Change to your home directory with `cd`
 - `~` means your home directory, eg `cd ~/bin`
- list files with the `ls` command.
- move a file, or change its name with `mv`
- copy a file with `cp`
- delete a file with `rm`
- delete a directory (with content) with `rm -r`
- make a directory with `mkdir`, and delete an empty directory with `rmdir`

Reading files at the terminal

- You can read text files at the terminal with `cat`, `more`, or `less`
- The `cat` command will write out the entire file to the screen at once, so if it is a large file the top might scroll off the screen.
- You can “page” through a large file with `more` or `less`. Within `more` or `less`,
 - `<space>` goes to the next page
 - `b` goes backward
 - `<return>` goes forward one line
 - `q` quits
 - `h` tells you about other commands
- If you come upon a file called `README`, you can read it by saying
 - `more README`
- You can get the beginning of files with `head`
- You can get the end of files with `tail`
 - `head` and `tail` give 10 lines by default, but that can be changed

Edit files with a command line editor

- Perhaps the easiest one to use is `pico`
 - and is recommended for beginners.
- The editors `vi` (or better, `vim`) and `emacs` (or `xemacs`) are much more powerful
 - but require a bit of learning
 - both have tutorials
 - both excellent and highly recommended
 - both have GUI versions

Completion

- The shell will complete
 - commands
 - file and directory names
- Complete with a `<tab>`
 - sometimes with `cntrl-d`
- Completion goes to the point of ambiguity, and then lists alternative completions for you to choose

Backgrounding

- You can put processes in the background by following the command with a `&`
 - eg `yourcommand > out &` (note output redirection with `>`)
 - doesn't work for interactive processes
- You can put interactive processes in the background with `screen`
 - start by eg `screen yourCommand`, then do your interactive stuff
 - put it in the background with `ctrl-a ctrl-d`
 - reconnect with `ctrl-r`
- See running jobs with `top`
- Kill jobs with `kill <pid>`, or using `k` in `top`

Customization

- You can customize
 - your shell
 - many programs
- Usually customizations are stored in “dot” files or directories in your home directory
 - eg `~/.cshrc` customizes the `cs`h and `tc`sh shells
- You can also customize by setting environment variables
 - eg `PATH` sets locations for known usable commands
 - see all your current environment variables by `printenv`

Scripts are programs

- You can write a series of shell commands in a text file
 - that text file becomes a program
- There are other scripting languages, especially
 - Perl
 - Python

Line endings

- The end of a line of text is a special character
 - `\n` for Unix and Mac OSX
 - `\r` for older macs
 - DOS/Windows uses `\r\n`
- You can see these characters with `od -c`, eg
 - `od -c <yourFile> | more`

A perl script mac2unix

```
#!/usr/bin/perl -i
while(<>){
    s/\r/\n/g;
    print;
}
```

Tarring and compressing

- You can archive directories and files into one tar file eg
 - `tar cf yourArchiveName.tar YourDirectory`
- You can expand (untar) by eg
 - `tar xf yourArchiveName.tar`
- You can compress files with `gzip`, and uncompress with `gunzip`
- You can tar and compress at the same time by eg
 - `tar czf foo.tgz Foo`
 - `c` to create an archive
 - `x` to expand
 - `z` to compress/uncompress
 - `f` is followed by the archive file name

Networking

- You can work on remote machines with `ssh`, eg to log in to your account `me` on the machine `mothership`,
 - `ssh me@mothership`
 - Use the `-X` flag to enable X-windows
 - The older `telnet` is not secure, and is deprecated
- You can transfer files (often tar files) with `scp` or `sftp`, eg
 - `scp foo.tgz me@mothership:`
 - `scp me@mothership:foo.tgz .`
(the dot means “where you are”)
 - The older `ftp` is still in wide use. Try `ncftp` for a nice ftp client.
- You can set up `ssh et al.` so that you do not need to give your password, yet have full security

Practicals

The practicals are in numbered directories; do them in turn. The data sets are small due to time constraints. Read and try to understand both the files that are given and the files that are produced during the analyses.

```
1_ChooseModel
2_ModelTest
3_Hsearch
4_SH_test
5_p4_hetero_ml
6_MrBayes
7_p4_hetero_mcmc
```

1 Choosing a model

In this exercise, we will evaluate the likelihood of some data using a single tree but with 4 different models. That way we can compare the models by the LRT and the AIC.

Open the file `executeThisWithPaup.nex` This file has the following contents.

```
#nexus

begin paup;
  execute ../bacterial_16S.nex;
  nj;
  set crit=like;
  set warntsave=no;
  log start file=paupLog replace;
  lset basefreq=estimate;
  lset nst=2 tratio=estimate;
  lscore 1;
  lset nst=6 rmatrix=estimate;
  lscore 1;
  lset rates=gamma shape=estimate;
  lscore 1;
  lset rates=gamma shape=estimate
    pinvar=estimate;
  lscore 1;
  log stop;
  quit;
end;
```

Execute it by saying, to your command line,

```
paup executeThisWithPaup.nex
```

- All nexus files must start with `#NEXUS`
- The data are contained in the file `bacterial_16S.nex`, which is located one level up.
- `nj` tells PAUP to make a neighbor-joining tree, which will be used to evaluate the 4 different models.
- The optimality criterion is set to likelihood, and PAUP is told to quit without asking whether the tree should be saved
- The log file `paupLog` is started. If a file by that name exists, it is overwritten.
- These four models are evaluated. The composition is free in all cases, which accounts for 3 parameters.

- HKY model. It allows a different rate for transitions and transversions, which is described by the parameter `tratio`. 4 free parameters.
- GTR model. It allows a different rate among all combinations of bases. 8 free parameters.
- GTR+G model. It incorporates gamma-distributed among-site rate variation. 9 free parameters.
- GTR+GI model. It incorporates a combination of gamma-distributed among-site rate variation and a proportion of invariant sites. 10 free parameters.

- At the end, the log file is closed, and PAUP quits.

Using the information in the log file, evaluate the 4 models by the LRT and the AIC. To do that, here are some 95% critical points for χ^2 .

dof	
1	3.8
2	6.0
3	7.8
4	9.5
5	11.1

Which of the models has the highest likelihood? Which of the models should we use? Do the LRT and AIC agree?

2 ModelTest and SS ASRV

The program ModelTest can help remove some of the tedium of choosing a model. It uses PAUP to evaluate many different models using the PAUP block `modelblock3`. Read it and try to make sense out of it. The results of the likelihood evaluations are fed to the program ModelTest, which assesses the likelihood values and suggests a model. So it is a two step process—first generating the likelihood scores using PAUP, and second assessing those scores with the ModelTest program proper.

The dataset that we use is simulated. Execute `firstExecuteThisWithPaup.nex`, which contains

```
begin paup;
  default lscores longfmt=yes;
  execute data.nex;
  execute modelblock3;
  savetrees file=theModelTestTree.nex brlens=yes;
  quit;
end;
```

- ModelTest (the program) depends on a certain output format from PAUP. That default output format changed with the newest release of PAUP. A workaround is to issue the command `default lscores longfmt=yes;` which makes a format which is readable by ModelTest.
- The data is read in, followed by `modelblock3`.
- `modelblock3` makes a NJ tree with JC distances. We will need that tree later, so it is saved.

When `firstExecuteThisWithPaup.nex` is executed, it makes 3 files:

1. `theModelTestTree.nex`, the NJ tree
2. `model.scores`, a concise output from PAUP
3. `modelfit.log`, a verbose output from PAUP

Process the file `model.scores` with the program ModelTest. It will suggest two models, one from the LRT, and one from the AIC. What are those models? Do they agree?

Note that ModelTest does not consider site-specific among-site rate variation (SS ASRV), so you need to do that “by hand”. To do that, execute the file `secondExecuteThisWithPaup.nex`, which contains

```
begin paup;
  execute data.nex;
end;

begin sets;
  charPartition by_codon = pos1: 1-.\3,
                          pos2: 2-.\3,
                          pos3: 3-.\3;
end;

begin paup;
  gettrees file=theModelTestTree.nex;
  set crit=like;
  lset nst=2 rates=gamma shape=estimate
      basefreq=estimate tratio=estimate;
  lscores 1;
  lset nst=2 rates=sitespec
      siterates=partition:by_codon
      basefreq=estimate tratio=estimate;
  lscores 1;
  quit;
end;
```

We use the same tree that ModelTest used, so that we can compare values. First, we (redundantly) evaluate the data with the HKY+G model, and then with the HKY+SS model, with data partitions based on codon position. Is the likelihood for the HKY+G model the same as that obtained by ModelTest? Does the HKY+SS model have a higher likelihood? To assess whether it is significantly higher, you may need to know that in data partitioned into n parts there are $n - 1$ free parameters due to the among site rate variation. Is the HKY+SS model significantly better?

3 An heuristic search

See the file `executeThisWithPaup.nex`.

```
begin paup;
  execute ../bacterial_16S.nex;
  log start file=paupLog replace;
  set crit=like;
  set autoclose=yes;
  Lset Base=(0.2287 0.2621 0.3426) Nst=6
      Rmat=(0.5147 1.3958 0.9436 1.4810 3.3455)
      Rates=gamma Shape=0.3731 Pinvar=0;
  hsearch swap=tbr;
  lset basefreq=est rmatrix=est shape=est;
  lscores 1;
  savetrees file=bestTree.nex format=altnex brlens=yes replace;
  log stop;
end;
```

In this part, we search for the best tree with the `hsearch` command. We start with the model and parameters values suggested by ModelTest, and search with those parameters. After the search, we make the parameters free again (`xxx=estimate`) and optimize on the resulting tree. We do this because we want to reassure ourselves that we have made the (last) search based on fully optimized parameters. Is this so? Is the tree that is obtained biologically reasonable? If not, why not? How could you analyze these data that might give you the correct tree?

4 SH test

Here we demonstrate successive iteration in the `hsearch` strategy. We alternate between fixing parameter values and searching, and freeing parameter values and re-optimizing on the tree resulting from the search. This is continued long enough that we know that the last search used fully optimized parameters.

This exercise also demonstrates searching with a topological constraint. We search with and without a constraint for monophyly of a group of taxa, and obtain 2 different trees. We then ask if those trees are significantly different by the SH test.

I have already chosen a model to use; it is the TVMef+I model, chosen by the AIC in ModelTest. Read and execute `pConstrHS.nex`.

When we do the heuristic search for the `ml` tree, we notice that all the 'A' taxa (A1, A2, and so on) are not in the same split. It is not possible that the A-taxa are monophyletic in the `ml` tree. But let us say that we expected them to be related, and we expected the 'B' taxa to be all together in their own split. So we are a little surprised by our `ml` tree. We ask whether we can really reject monophyly of the A-taxa. The way we do that is to find the best tree where the A-taxa are constrained to be in the same group, and ask, using the Shimodaira-Hasegawa test, whether that constrained tree is significantly worse than the `ml` tree.

So after finding the `ml` tree, we start over again, but this time with a topological constraint

```
constraints monoA = ((A1, A2, A3, A4, A5, A6, A7, A8));
```

When we do the `hsearch` with this constraint, we find a best tree, which is of course slightly less likely than the `ml` tree. Is this difference significant?

We assess significance using the SH test. We read in both trees (having saved them before) and ask for the `lscore` under the current model, and also we ask for an `shtest`:

```
gettrees file=ml_tree.nex;
gettrees file=constrained_tree.nex mode=7;
lsc 1-2/ shtest=rell;
```

In the `gettrees` command, the default mode is 3, which replaces trees in memory with the trees from the file. So here the first `gettrees` command wipes out any trees in memory. Of course we do not want to do that when we do the second `gettrees` command, so we use mode 7, which adds the trees in the file to the trees in memory.

At the end of the second `gettrees` command we have two trees in memory.

In the completed SH test, the P value that we get tells us that the constrained tree is not significantly worse than the `ml` tree, and so we do not have enough evidence, using this test and using these data, to reject monophyly of the A-taxa. However, note that the SH test depends on the number of trees compared and does not work well if there are only two trees; it should compare more plausible trees.

When the PAUP job is finished, look at the log file, and try to relate the contents of the log file with the commands in the file `pConstrHS.nex`. Were all of the steps of the successive iteration necessary?

5 ML with a heterogeneous model using p4

Here we re-analyse the bacterial 16S data with a heterogeneous model that allows a separate composition for the thermophiles, and a separate composition for the mesophiles. The file `mt.nex` specifies the model and the trees. We only compare 2 trees, the true tree and the attract tree. Command comments (eg `[&&p4 model (mesophiles)]`) in the tree description specifies which branches get which models.

To analyse the data, say

```
p4 s.py
```

(For a description of the program, type in `p4` with no arguments. Quit with `ctrl-d`.)

The `s.py` python script reads in the data, and the tree and model. Then, for each tree, it calculates the maximum likelihood of the data under the model. How the results are written out can be specified by the user; here we draw the two trees, and write out their ML values. Did use of a heterogeneous model allow recovery of the true tree?

6 Bayesian analysis with MrBayes

The executable for MrBayes is `mb`. Start the program and read in the data by saying

```
mb bacterial_16S.nex
```

and then, at the MrBayes > prompt,

```
execute executeThisWithMrBayes.nex
```

which contains

```
begin mrbayes;
  log start filename=mbLog replace;
  set autoclose=yes;
  lset nst=6 rates=gamma;
  mcmc ngen=20000 printfreq=500 samplefreq=100
      nchains=4 savebrlens=yes filename=mbout;
  plot filename=mbout.p;
  sumt filename=mbout.t burnin=101;
  log stop;
end;
```

- The current version of `MrBayes` requires that the data be in the same directory as the analysis.
- At the end of a run, by default, `MrBayes` asks you if you want to continue the chain. `autoclose` turns this off.
- Setting the model is similar to PAUP, but is different. Here we set the GTR+G model.
- We ask that the chain run for 20000 generations, sampling every 100, for a total of 200+1 samples. We run 4 parallel chains in the MCMCMC.
- We start with a random tree topology by default.
- At the end of the MCMC, after a burnin of 101 samples (not generations), the trees in the file `mbout.t` are summarized, making a consensus tree `mbout.t.con` and a list of tree bipartitions in `mbout.t.parts`. Note that high confidence is given to the wrong tree.

7 Bayesian analysis with p4

To analyse the bacterial 16S data using a heterogeneous model, say

```
p4 s.py
```

This script reads in the data, and the model description in the file `m.nex`. A random tree is made, and the 2 models are assigned to branches randomly. A Bayesian MCMC run is set up, and some adjustments are made to the proposal probabilities and to the tuning parameters. A short run of 10000 generations is done, and a consensus tree made from the sampled trees. Although the 2 models were assigned to branches randomly, the branches are allowed to choose either model as part of the MCMC, allowing the 2 models to move around the tree while simultaneously adjusting their parameter values. What consensus tree is found? Is it biologically correct? How well supported is it?