Handout for the Phylogenetics Lectures

Dirk Metzler

February 6, 2026

Contents

1 Intro: Outline and Tree Notation

Tentative plan for phylogenetics part

- Maximum Likelihood v. Maximum Parsimony vs. Distance-based Phylogeny Inferrence
- Sequence Evolution Models (JC, F81, HKY, F84, GTR, PAM and Γ-distributed rates)
- Bootstrap
- MCMC and Bayesian Inferrence
- Calculations with sequence evolution models (and other stochastic processes)
- How to select a model
- Relaxed Molecular Clock and Time Calibration
- Independent Contrasts for Quantitative Traits

- Tests for trees and branches
- Statistical Alignment (TKF91, TKF92, pairHMMs, multiple HMMs)

Aims

- Understand princples and rationales underlying the methods
- Explore available software
- What is efficiently doable, what is difficult?
- What are the strengths and weaknesses of the methods?
- Which method is appropriate for which dataset?
- Learn what is necessary to read papers about new computational methods
- Future directions of phylogenetics

Recommended Books

References

[Fel04] J. Felsenstein (2004) *Inferring Phylogenies*

[Yang06] Z. Yang (2006) *Computational Molecular Evolution*

[Niel05] R. Nielsen, [Ed.] (2005) *Statistical Methods in Molecular Evolution*

[DEKM98] R. Durbin, S. Eddy, A. Krogh, G. Mitchison (1998) *Biological Sequence Analysis*

[EG05] W. Ewens, G. ¸Grant (2005) *Statistical Methods in Bioinformatics*

ECTS and work load per week

For *Computational Methods in Evolutionary Biology*, 9 ECTS $\approx$ 0.6 per week, 18 hours per week:

- 4 hours lecture (each 45 min + break)
- 3 hours exercise sessions
- 6 hours homework (exercises)
- 5 hours study lecture contents

For *Phylogentics*, 6 ECTS = 0.8 per week, 24 hours per week:

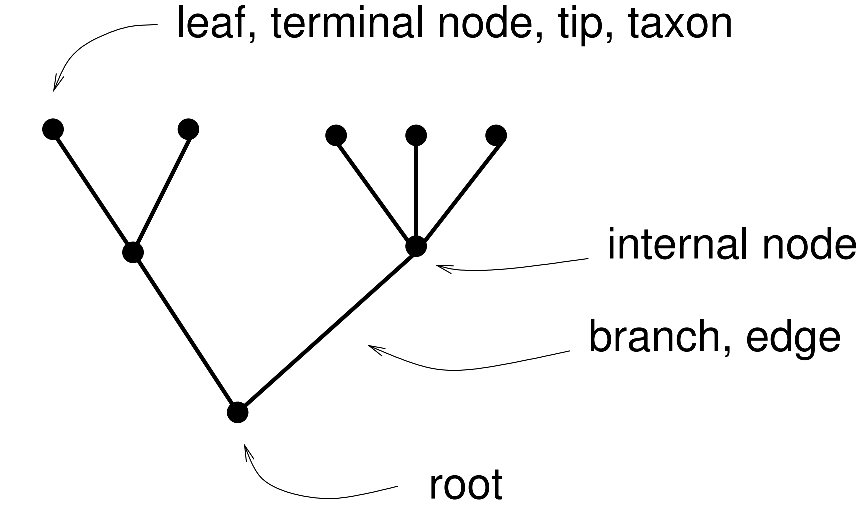
- all as above plus
- 2 hours of practicals and additional exercise session
- 2 hours learn software, apply to data, prepare presentation
- 2 more hours to learn algorithms and maths

How to study the content of the lecture

For the case that you are overwhelmed by the contents of this course, and if you don't have a good strategy to study, here is my recommendation:

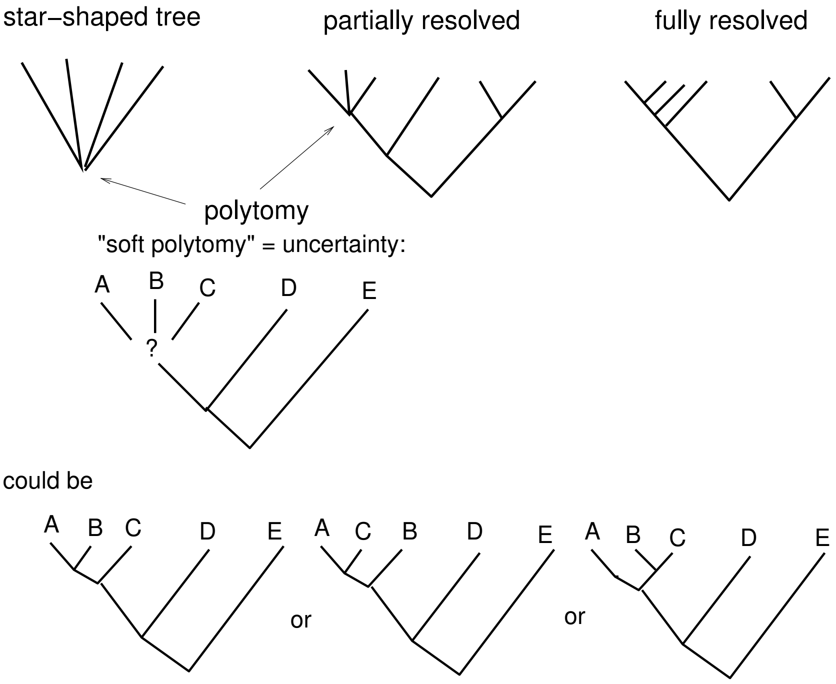
1. Try to explain the items under "Some of the things you should be able to explain"
2. Discuss these explanations with your fellow students
3. Do this before the next lecture, such that you can ask questions if things don't become clear
4. Do the exercises (at least some of them) in time
5. Study all the rest from the handout, your notes during the lecture, and in books

Terminology for trees

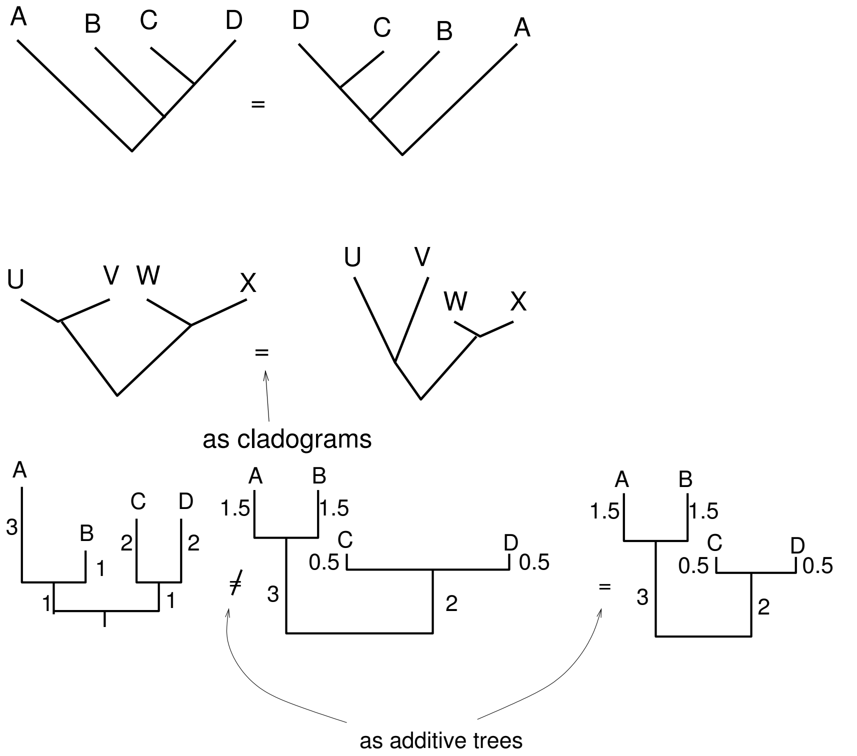


degree of a node = number of edges adjacent to the node

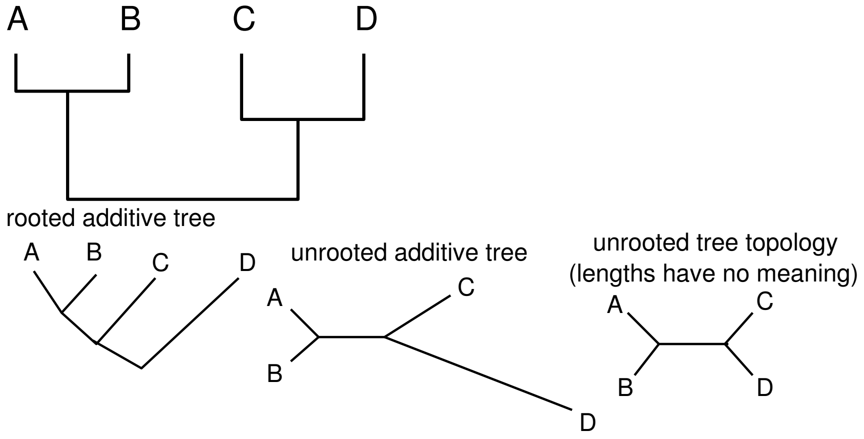
binary tree = fully resolved tree: root has degree two, all other nodes have degree 3



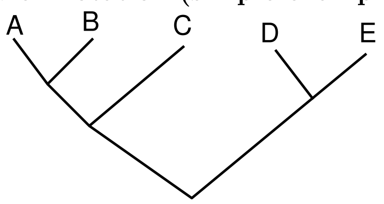
cladogram: branch lengths not meaningful



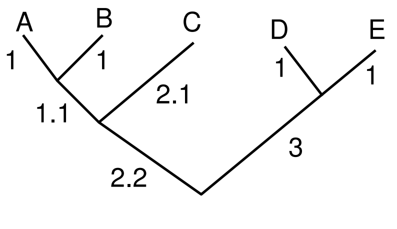
dendrogam = chronogram = ultrametric tree = additive trees that are compatible with molecular-clock assumption, i.e. all tips have the same distance to the root



Newick notation (simple examples)



(((A,B),C),(D,E));



(((A:1,B:1):1.1,C:2.1):2.2,(D:1,E:1):3);

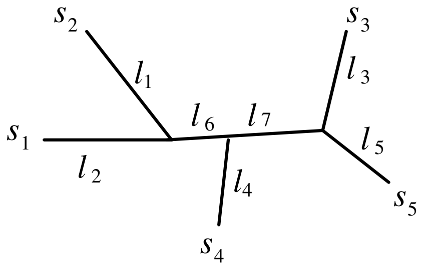
Some of the things you should be able to explain

- Basic terminology of rooted and unrooted trees
- basics of Newick notation

2 Distance-Based Phylogeny Reconstruction

2.1 What is a distance?

Given a set of taxa $S = \{s_1, s_2, \ldots, s_n\}$ and matrix of distances $(d_{ij})_{ij \leq n}$, where d_{ij} is the (estimated) distance between s_i and s_j we search for a tree whose tips are labeled with S and whose edges are labeled with lengths, such that the distances between tips labeled with s_i and s_j should be (approximately) d_{ij} for all i, j.



For example, $l_2 + l_6 + l_7 + l_3$ should be (as close as possible to) $d_{1,3}$

Distances should be *additive*. E.g. the *Hamming distance* (number of observed differences) between DNA sequences is in general not be additive if back-mutations or double-hits happened. A more useful distance is the (*expected*) number of mutations according to a sequence evolution model; more about this later.

To be a proper *distance matrix*, $(d_{ij})_{ij \leq n}$ must fulfill the following requirements for all i, j, k:

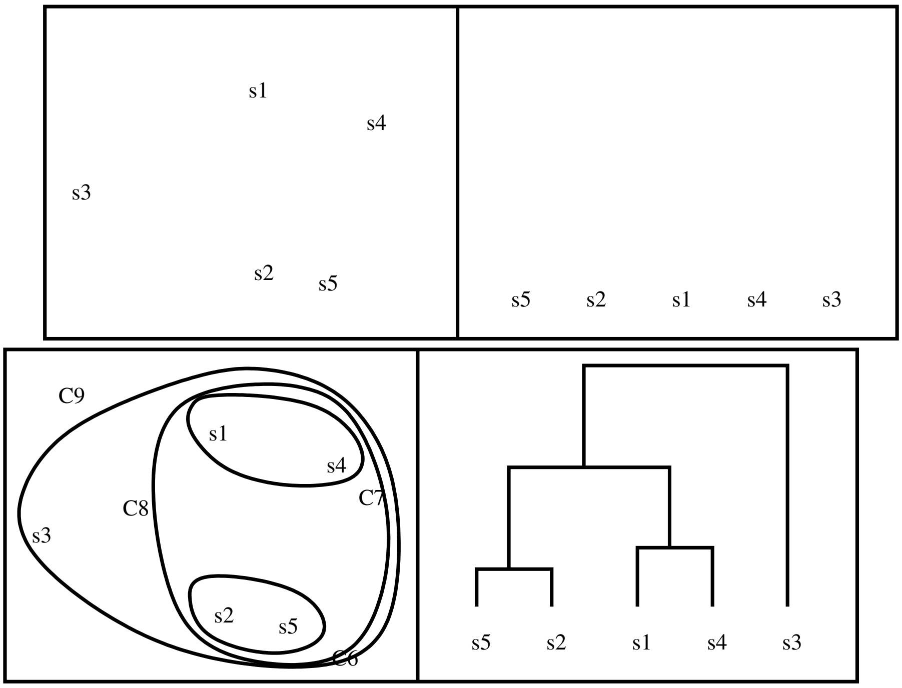
- $d_{ij} = d_{ji}$
- $d_{ij} = 0 \Leftrightarrow i = j$
- $d_{ij} + d_{jk} \geq d_{ik}$ (triangle inequality)

2.2 UPGMA

UPGMA (**U**nweighted **P**airwise **G**rouping **M**ethod with **A**rithmetic mean, Sokal & Michener, 1985) is hierarchical cluster method using means:

```
for $i \leq n$ set $C_i := \{s_i\}$
$\mathcal{C} := \{C_1, \ldots, C_n\}$ is the current set of clusters
$m := n$
repeat ...
    • $m := m + 1$
    • find $C_i, C_j \in \mathcal{C}$ with minimum $d_{ij} > 0$
    • $C_m := C_i \cup C_j$
    • $\mathcal{C} := \mathcal{C} \cup \{C_m\} \setminus \{C_i, C_j\}$
    • For all $C_k \in \mathcal{C}$ set
          $d_{km} := d_{mk} := \frac{1}{|C_k| \cdot |C_m|} \sum_{s_x \in C_k, s_y \in C_m} d_{xy}$
```

… until $C_m = \{s_1, \dots, s_n\}$ and $\mathcal{C} = \{C_m\}$.



common ways to define the distance between clusters C and C' cluster algorithms:

single linkage: $d(C, C') = \min_{s_i \in C, s_j \in C'} d_{ij}$

complete linkage: $d(C, C') = \max_{s_i \in C, s_j \in C'} d_{ij}$

means (like in UPGMA): $d(C, C') = \frac{1}{|C| \cdot |C'|} \sum_{i \in C, j \in C'} d_{ij}$

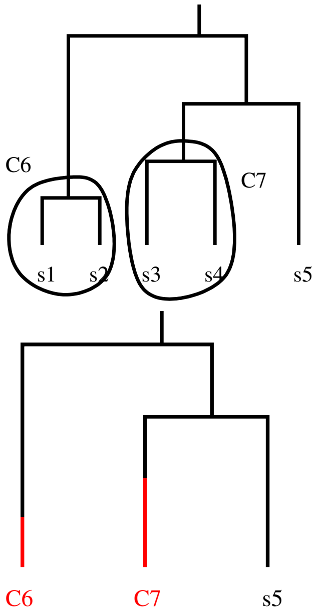
UPGMA works under ideal conditions

Assume the the there is an **ultrametric** tree (i.e. molecular-clock) in which the tips have **exactly** the given distances d_{ij}. Then, UPGMA will find this tree.

Reason: in the first step UPGMA will correctly join the closest relatives.

As a consequence of the molecular clock assumption, UPGMA will define reasonable distances between the clusters.

Example:



From

$$d_{13} = d_{14} = d_{23} = d_{24}$$

follows

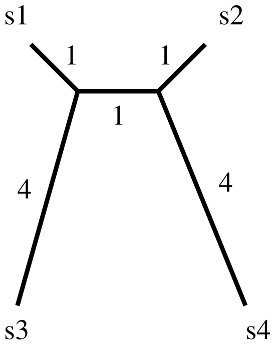
$$d_{67} = \frac{1}{2 \cdot 2} \cdot (d_{13} + d_{14} + d_{23} + d_{24}) = d_{13}$$

This means that we are in the same situation as in the first step: The clusters are tips of an ultrametric tree, and the distances for the clusters are just like the distances of any taxa in the clusters.

Thus, UPGMA will not only get the first step right but also any other step.

When UPGMA fails

If the tree is not compatible with molecular-clock assumptions, UPGMA may fail even if the precise distances are known.



In this example, UPGMA will first join s_1 and s_2 and will not have a chance to correct this in any later step.

Ultrametric distances

Theorem 1 *Let $D = (d_{ij})_{ij}$ be a distance matrix for $(s_1, \ldots, s_n)$. The following two properties are equivalent:*

(a) A binary tree exists that fulfills the molecular-clock assumption and the tips of this tree have the distances given in D. (The distance between two tips is the sum of the lengths of the edges between them.)

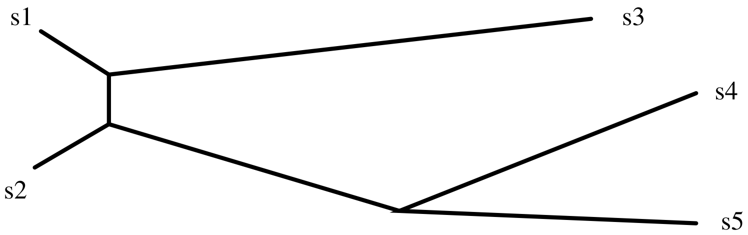
(b) D is ultrametric, i.e.

$$\forall_{\textit{sets of three indices } \{i,j,k\}} \exists_{i \in \{i,j,k\}} : d_{jk} < d_{ij} = d_{ik}$$

2.3 Neighbor Joining

Idea: use modified distances that take into account how far a taxon is to all other taxa

$$D_{ij} := d_{ij} - (r_i + r_j), \qquad \text{where} \quad r_i \ = \ \frac{1}{n-2} \sum_k d_{ik} \ = \ \frac{n-1}{n-2} \cdot \overline{d_{i.}}$$

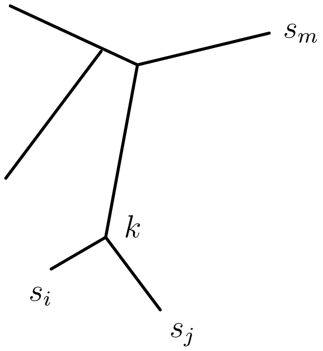


Neighbor Joining algorithm (Saitou, Nei, 1987)

Input $T = \{s_1, \ldots, s_n\}$ with distance matrix $(d_{ij})_{i,j \leq n}$

NeighborJoining(T):

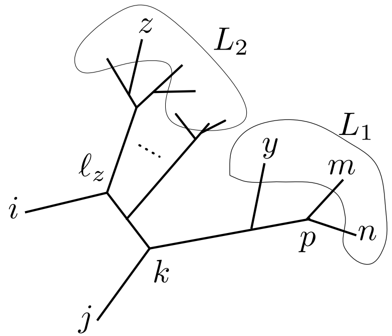
- done if $n \leq 2$
- compute all D_{ij}
- find taxa s_i and s_j in T with minimum D_{ij}
- define internal node k with distances $\forall_m : d_{km} := \frac{1}{2}(d_{im} + d_{jm} - d_{ij})$
- NeighbourJoining($\{k\} \cup T \setminus \{s_i, s_j\}$)



Theorem 2 (Neighbor-Joining theorem, Studier & Keppler, 1988) *If a tree exists whose tips have precisely the distances given by $(d_{ij})_{ij}$, then Neighbor-Joining will find this tree.*

Sketch of proof: assume that i and j are not neighbors and show that D_{ij} can then not be minimal. Let set of tips L_1 and L_2 be defined as below and assume w.l.o.g. $|L_1| \leq |L_2|$. By definition,

$$D_{ij} - D_{mn} = d_{ij} - d_{mn} - \frac{1}{n-2}\left(\sum_u d_{iu} + d_{ju} - d_{mu} - d_{nu}\right).$$



Using additivity, we can show that

$$d_{iy} + d_{jy} - d_{my} - d_{ny} = d_{ij} + 2d_{ky} - 2d_{py} - d_{mn}$$

and $\quad d_{iz} + d_{jz} - d_{mz} - d_{nz} = d_{ij} - d_{mn} - 2d_{pk} - 2d_{\ell_z k}$

hold for all tips $y \in L_1 \setminus \{n, m\}$ and $z \in L_2$.

And: $d_{ii} + d_{ji} - d_{m_i} - d_{ni} + d_{ij} + d_{jj} - d_{mj} - d_{nj} \;=\; -4d_{kp} - 2d_{mn}$

Further, we obtain $d_{in} + d_{jn} - d_{mn} - d_{nn} = d_{ij} + 2d_{kp} + 2d_{pn} - d_{mn}$ and $d_{im} + d_{jm} - d_{mm} - d_{mn} = d_{ij} + 2d_{kp} + 2d_{pm} - d_{mn}$

With the equations above follows that

$$\begin{aligned} D_{ij} - D_{mn} &= \frac{\left(\sum_{y \in L_1 \setminus \{m,n\}} 2d_{py} - 2d_{ky}\right) + \left(\sum_{z \in L_2} 2d_{pk} + 2d_{\ell_z k}\right)}{n-2} \\ &> 2d_{pk}(|L_2| - |L_1|)/(n-2) \qquad \text{(because of } d_{py} - d_{ky} > -d_{pk}\text{)} \\ &\geq 0 \end{aligned}$$

and thus $D_{ij} > D_{mn}$, q.e.d.

Some of the things you should be able to explain

- basic properties of distance measures
- how does UPGMA work
- what is different in the approach of neighbor joining (NJ)?
- under what conditions will UPGMA and/or NJ find the right tree?
- example when UPGMA and NJ lead to a different result
- when is a distance ultrametric and what does this mean for the tree?

3 Parsimony in phylogeny reconstruction

3.1 Parsimony of a tree

Given n homologous DNA or protein sequences

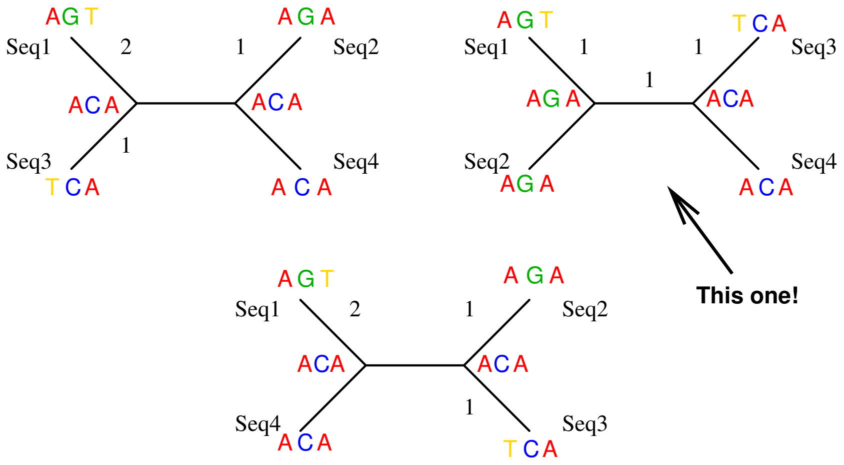
$$\begin{array}{ccc} x^1 & = & x^1_1, x^1_2, \ldots, x^1_m \\ x^2 & = & x^2_1, x^2_2, \ldots, x^2_m \\ \vdots & \vdots & \ddots \quad \vdots \\ x^n & = & x^n_1, x^n_2, \ldots, x^n_m \end{array}$$

e.g. $n = 4$:

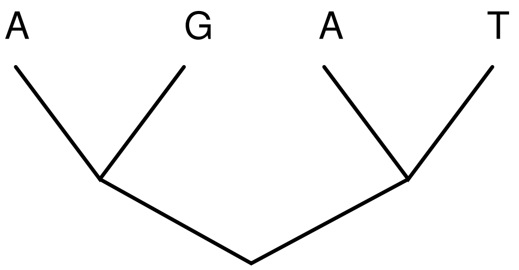
Seq1	GCAGGGTAC
Seq2	GCAGGGAAC
Seq3	GCTGGCAAC
Seq4	GCAGGCAAC

Which tree is *most parsimonious*, i.e. explains the data with the least number of mutations?

For this question we can neglect all non-polymorphic sites.
Which tree is most parsimonious?

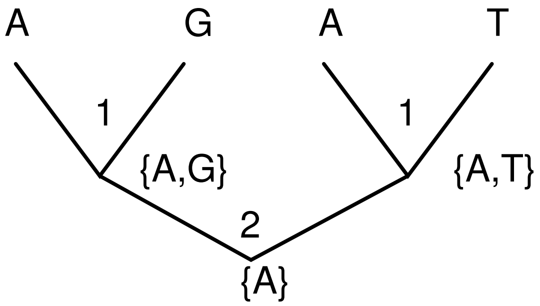


Given a tree whose tips are labeled with sequences, how can we efficiently compute the minimal number of mutations?



ideas:

1. Do separately for each alignment column
2. label each inner node with the optimal states for the tips above it and with the least number of mutations
3. go from tips to root by dynamic programming



Fitch algorithm

C is a counter of mutations, and M_k is the set of optimal states in node k.
Do for all sites s:

1. $C_s := 0$ will be the counter of mutations at that site
2. for all tips b with label x set $M_b = \{x\}$.
3. Moving from tips to root do for all nodes k with daughter nodes i and j:

   if $M_i \cap M_j = \emptyset$**:** set $M_k = M_i \cup M_j$ and $C_s := C_s + 1$

   else: set $M_k = M_i \cap M_j$

output $\sum_s C_s$

weighted parsimony

It is possible to take into account that different types of mutations (e.g. transitions and transversions) differ in the frequency by defining a cost $S(a,b)$ for a mutation $a \to b$.

A variant of the Fitch algorithm calculates the minimal cost of a given tree to generate given sequences at the tips. ($\rightsquigarrow$ exercise)

3.2 Finding parsimonious trees for given data

Given a large number n taxa, it is not feasible to consider all trees, because the number of unrooted bifurcating trees with n taxa is

$$3 \times 5 \times 7 \times \cdots \times (2n-5)$$

n	$3 \times 5 \times 7 \times \cdots \times (2n-5)$
5	15
7	945
10	2,027,025
12	654,729,075
20	$2.2 \cdot 10^{20}$
50	$2.8 \cdot 10^{74}$
100	$1.7 \cdot 10^{182}$

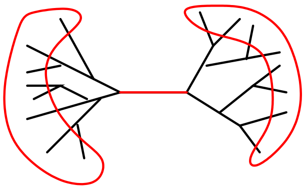
for comparison:

(estimated number of atoms in the observable universe) × (number of second since big bang) $\approx 5 \cdot 10^{97}$

problem of perfect parsimony

Given n sequences of length m with up to 2 different states per position (alignment column). Is there a perfectly parsimonious tree, i.e. one that never has more than one mutation at the same position?

Idea: each polymorphism defines a split of the set of taxa $L = A \cup B$, $A \cap B = \emptyset$.
A branch of a tree also defines a split of L



Go through the alignment from left to right and further subdivide L until there is a contradiction or you reach the end of the alignment.

Theorem 3 (Four-gamete condition) *A contradiction will occur if and only if there are two polymorphisms that lead to two splits $L = A \cup B = C \cup D$ such that the four intersections $A \cap C, A \cap D, B \cap C, B \cap D$ are all non-empty.*

This gives us an efficient solution for the problem of perfect parsimony. How about a slight generalization?

Given n homologous sequences of length m with up to r different states in each column.
Is there a perfectly parsimonious tree, i.e. one without back-mutations and without more than one mutation into the same state in the same position?

complexity: NP-complete for unbounded r and polynomial for any fixed $r \in \mathbb{N}$.

The problem of maximum parsimony

Given n homologous sequences of length m with up to 2 different states in each column, find the tree that needs the minimum number of mutations to explain the tree.

complexity: NP-complete

There is a method that can guarantee to find a tree that needs at most twice as many mutations as needed by the most parsimonious tree. However, in practice heuristic search algorithms are more relevant.

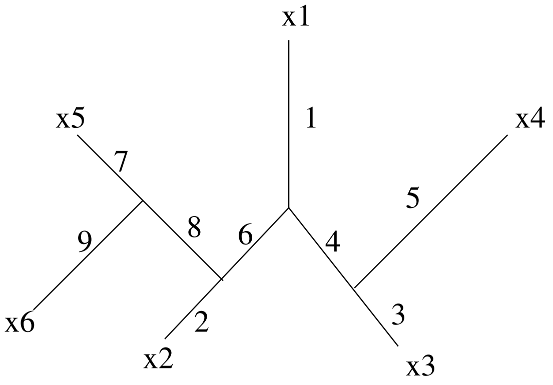
enumerating all tree topologies

The sequence of numbers $[i_3][i_5][i_7]\dots[i_{2n-5}]$ with $i_k \in \{1,\dots,k\}$ represents a tree topology with n labeled leaves. It can be decoded as follows.

- Start with a 3-leaved tree whose leaves are labeled with x_1, x_2, x_3 and whose edges are labeled accordingly with 1,2,3.
- repeat for $j = 4, \dots, n$:
  1. $k := 2j - 5$
  2. Add an edge to the new leaf x_j to edge i_k
  3. Call the new edge $k + 2$.
  4. In the subdivided edge i_k, give the part that is closer to x_1 the label $k + 1$. The other part keeps the label i_k.

enumerating all tree topologies

Example:



This tree can be represented by [3][2][7]

enumerating all labeled tree topologies

Enumerate leaves-labeled topologies by iterating $[a][b][c]....[x]$ like a mileage counter for all allowed values ($a \le 3, b \le 5, c \le 7, \dots$):

[1]	[1]	[1]	...	[1]	[1]	[1]
[1]	[1]	[1]	...	[1]	[1]	[2]
[1]	[1]	[1]	...	[1]	[1]	[3]
			⋮			
[1]	[1]	[1]	...	[1]	[1]	$[2n-5]$
[1]	[1]	[1]	...	[1]	[2]	[1]
[1]	[1]	[1]	...	[1]	[2]	[2]
[1]	[1]	[1]	...	[1]	[2]	[3]
			⋮			

Branch and Bound

Let

[3][4][2]....[19][0][0][0]

denote the tree in which the last three taxa are not yet inserted. (zeros are only allowed at the end of a series).

Now we also iterate over these trees. If, e.g. u,v,w,x,y are the maxima of the last five positions:

$$\begin{array}{c}
[a][b][c]...[m-1][u][v][w][x][y] \\
[a][b][c]\ldots[\ m\ ][0][0][0][0][0] \\
[a][b][c]\ldots[\ m\ ][1][0][0][0][0] \\
[a][b][c]\ldots[\ m\ ][1][1][0][0][0] \\
[a][b][c]\ldots[\ m\ ][1][1][1][0][0]
\end{array}$$

If the tree corresponding to $[a][b][c]...[m][1][1][1][0][0]$ already needs more mutations than the best tree found so far, go directly to

$$[a][b][c]...[m][1][1][2][0][0] \qquad\qquad (\text{“Bound”})$$

“Branch and Bound” saves time and can be used in practice for up to about 11 taxa.

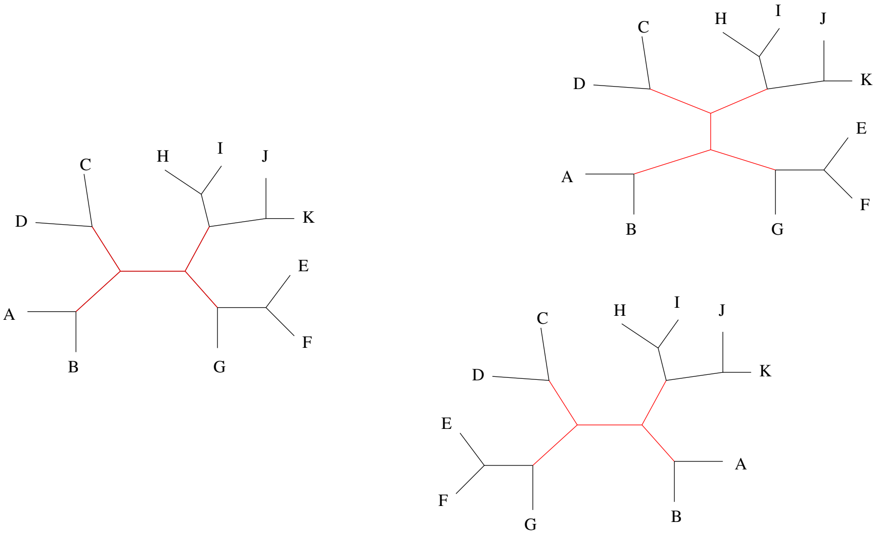
For larger numbers of taxa we need to move around in tree space and try to optimize the tree topology. Possible steps are

NNI: nearest neighbor interchange

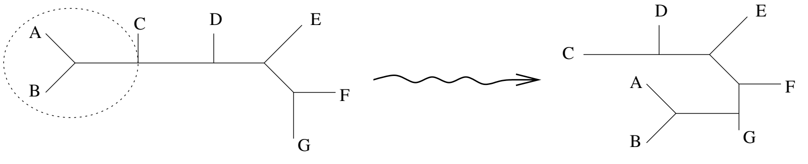
SPR: subtree pruning and regrafting

TBR: tree bisection and reconnection

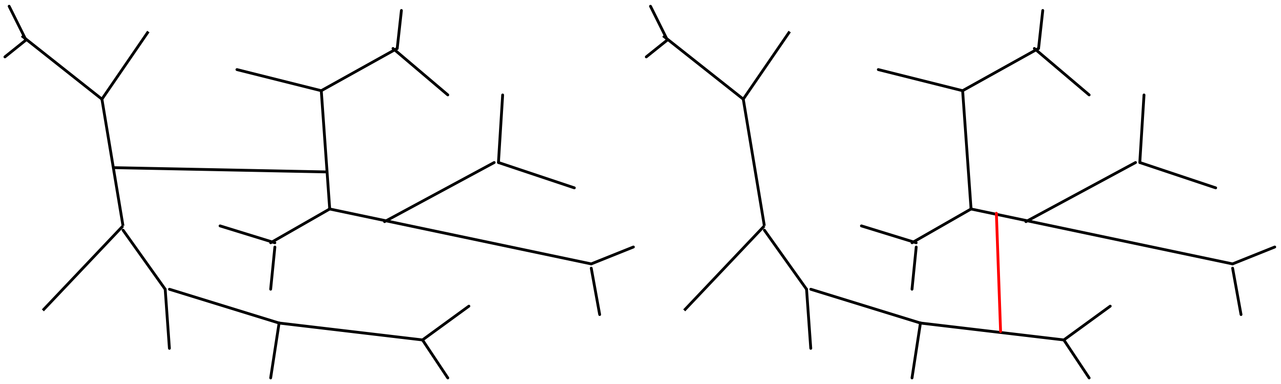
Nearest Neighbor Interchange



Subtree Pruning and Regrafting



Tree Bisection and Reconnection



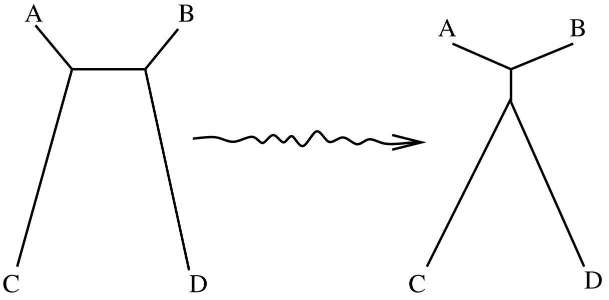
Note that each NNI move is special type of SPR move where the pruned subtree is regrafted in an edge neighboring the original edge.

Each SPR move is a special TBR where one of the nodes of the new edge is the old node.

3.3 Limitations of the parsimony principle

limitations of parsimony

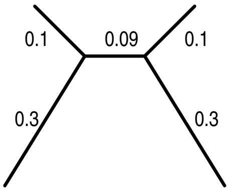
Parsimonious phylogeny reconstruction methods do not take back mutations and double-hits into account in a proper way. This can lead to problems when there are long branches with many mutations.



“long-branch attraction”

Comparison of phylogeny estimation methods

Durbin et al. (1998) simulated for several sequence lengths 1000 quartets of sequences along this tree to compare the accuracy of phylogeny reconstruction methods. Branch lengths are mean frequencies of transitions per position. (no transversions)



Proportion of correctly estimated trees

Seq.length	Max.Pars.	Neigh.Join.	ML
20	39.6%	47.7%	41.9%
100	40.5%	63.5%	63.8%
500	40.4%	89.6%	90.4%
2000	35.3%	99.5%	99.7%

Some of the things you should be able to explain

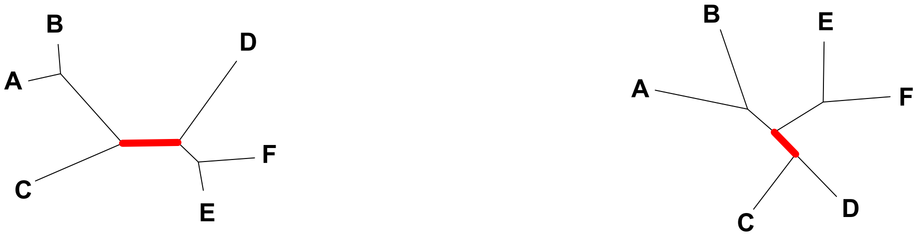
- What is the Fitch algorithm and how does it work?
- the four-gamete condition and how it helps to solve the perfect-parsimony problem
- NP-complete problems in parsimonious phylogeny reconstruction

- how to enumerate all trees and how to branch and bound
- how NNI, SPR and TBR are related to each other
- when will parsimony fail even if for very long sequences?

4 Measures for how different two trees are

The symmetric difference (aka "partition metric")

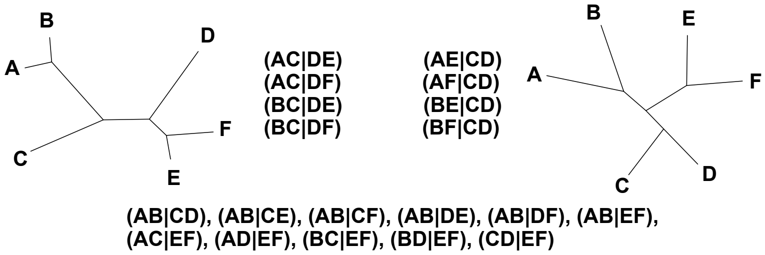
Bourque (1978), Robinson and Foulds (1981)



Each edge in the tree is a partition of the set of taxa. The *symmetric difference* is the number of edges that exist in one tree but not in the other.

quartet distance

for fully resolved trees of n taxa.


B
D
A
F
C
E
(AC|DE)
(AC|DF)
(BC|DE)
(BC|DF)
(AE|CD)
(AF|CD)
(BE|CD)
(BF|CD)
B
E
A
F
C
D
(AB|CD), (AB|CE), (AB|CF), (AB|DE), (AB|DF), (AB|EF),
(AC|EF), (AD|EF), (BC|EF), (BD|EF), (CD|EF)

Each of the $\binom{n}{4}$ quartets of taxa have a tree topology in each tree. The *quartet distance* is the relative frequency of quartets for which the topologies do not coincide.

NNI distance

Waterman, Smith (1978)

The *NNI distance* is the number of NNI moves needed to change the one tree topology into the other.

Problem: It has been shown that the computation of the NNI distance is NP-hard.

Allen and Steel (2001) showed that the TBR distance is easier to compute.

Path-length difference metric

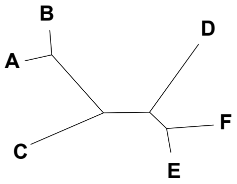
Penny, Watson, Steel 1993

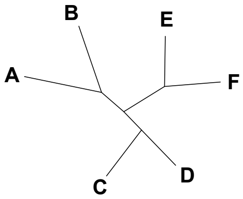
Let n_{ab}^T be the number of edges that separate taxa a and b in tree T. Then, the path-length difference metric between the trees T and T' is defined as

$$\sqrt{\sum_{a,b}\left(n_{ab}^T - n_{ab}^{T'}\right)^2}$$

taking branch lengths into account

For each partition P of the taxa set let $f_T(P)$ be the length of the corresponding edge in tree T if such an edge exists. Otherwise set $f_T(P) = 0$





branch score distance (Kuhner, Felsenstein, 1994)

$$\sum_P (f_T(P) - f_{T'}(P))^2$$

Robinson-Foulds distance

$$\sum_P |f_T(P) - f_{T'}(P)|$$

Some of the things you should be able to explain

- symmetric difference
- branch score distance
- Robinson-Foulds distance
- Why some other distances are hard to compute

5 Maximum-Likelihood (ML) in phylogeny estimation

5.1 What is a likelihood?

Frequentistic parameter estimation

- Assume that we observe some data D.
- D is influenced by random effects but a parameter p plays a role.
- We are interested in the value of p
- D is random but observed
- p is unknown but not random
- A model describes how the probability of D depends on p

Maximum-Likelihood principle: estimate p by

$$\widehat{p} = \arg\max_p \Pr_p(D)$$

To describe how $\Pr_p(D)$ depends on p we define the likelihood function:

$$L_D(p) := \Pr_p(D)$$

The ML estimator $\widehat{p}$ is the parameter value that maximizes the probability of the observed data.

simple example

If you toss a thumbtack, what is the probability p that the sting touches the ground?
Assume you made and experiment. In 1000 tosses, the sting touched the ground 567 times.

$$\widehat{p} \;=\; \arg\max_p \Pr_p(567) \;=\; \arg\max_p \binom{1000}{567} p^{567} \cdot (1-p)^{1000-567}$$

We calculate the derivative with the product rule and set it to 0 to look for the maximum:

$$\frac{\partial}{\partial p}\left(p^{567} \cdot (1-p)^{433}\right) = 567 \cdot p^{566} \cdot (1-p)^{433} - p^{567} \cdot 433 \cdot (1-p)^{432}$$

$$0 = 567 \cdot \widehat{p}^{566} \cdot (1-\widehat{p})^{433} - \widehat{p}^{567} \cdot 433 \cdot (1-\widehat{p})^{432}$$

As it is clear that $0 < \widehat{p} < 1$, we can divide both sides of the equation by $\widehat{p}^{566} \cdot (1-\widehat{p})^{432}$ and obtain

$$0 = 567 \cdot (1-\widehat{p}) - \widehat{p} \cdot 433 \qquad \Rightarrow \widehat{p} = 0.567.$$

Another approach to solve this uses the (natural) logarithm:

$$\begin{aligned}
\widehat{p} \;&=\; \arg\max_p \Pr_p(567) \;=\; \arg\max_p \binom{1000}{567} p^{567} \cdot (1-p)^{1000-567} \\
&=\; \arg\max_p \quad \log\left(p^{567} \cdot (1-p)^{433}\right) \\
&=\; \arg\max_p \quad 567 \cdot \log(p) + 433 \cdot \log(1-p) \\
&=\; \arg\max_p \quad \log\left(p^{567} \cdot (1-p)^{433}\right) \\
&=\; \arg\max_p \quad 567 \cdot \log(p) + 433 \cdot \log(1-p)
\end{aligned}$$

$$\frac{\partial}{\partial p}\left(567 \log(p) + 433 \log(1-p)\right) = \frac{567}{p} - \frac{433}{1-p}$$

$$\Rightarrow 0 = \frac{567}{\widehat{p}} - \frac{433}{1-\widehat{p}} \qquad \Rightarrow \widehat{p} = 0.567$$

Important: the parameter p is *not* a random object. Thus it does not make sense to ask for the *probability* that it takes some particular value p_0. However, the *likelihood* of p_0 is defined. It is the probability of the observed data if $p = p_0$.

5.2 How to compute the likelihood of a tree

ML estimation of phylogenetic trees: Given an alignment D, find the tree T that maximizes

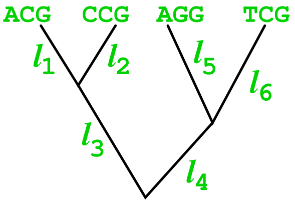
$$\Pr(D|T) =: L_D(T), \qquad \widehat{T} := \arg\max_T L_D(T)$$

What is $\Pr(D|T)$ and how can we compute it?

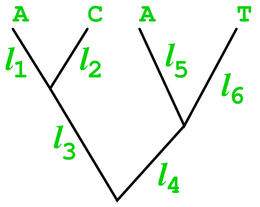
We assume that all alignment columns evolve independently of each other. Then

$$\Pr(D|T) = \prod_i \Pr(d_i|T),$$

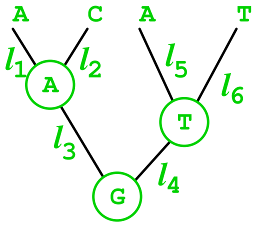
where d_i is the sequence data in the i-th alignment column.



But how can we compute $\Pr(d_i|T)$?



How to compute $\Pr(d_i|T) = L_{d_i}(T)$?
Let's first assume that d_i also contains labels of the inner nodes. Assume that for all nucleotide x, y and all $\ell \in \mathbb{R}_{>0}$ we can compute the frequency p_x of x and the probability $P_{x\to y}(\ell)$ that an x is replaced by a y along a branch of length ℓ.



Then, we get for the example tree

$$\begin{aligned}\Pr(d_i|T) &= p_G \cdot P_{G\to A}(\ell_3) \cdot P_{G\to T}(\ell_4) \cdot \\ &\cdot P_{A\to A}(\ell_1) \cdot P_{A\to C}(\ell_2) \cdot \\ &\cdot P_{T\to A}(\ell_5) \cdot P_{T\to T}(\ell_6).\end{aligned}$$

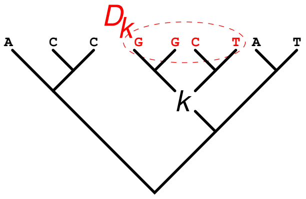
But usually, inner nodes are not labeled. What to do then?

Felsenstein's pruning algorithm

For each node k let D_k be the part of the data d_i that are labeled to tips that stem from k and define

$$w_k(x) = \Pr(D_k|k \text{ has an } x \text{ at this site })$$

for every nucleotide x.



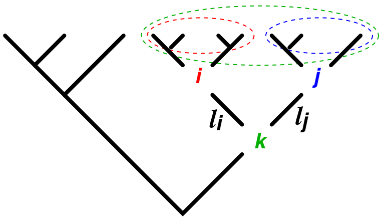
Idea: compute $w_k(x)$ for all k and all x. Then you know it also for the root r and can compute

$$L(T) = \Pr(D|T) = \Pr(D_r|T) = \sum_{x\in\{A,C,G,T\}} p_x \cdot w_r(x).$$

Compute all $w_k(x)$ from the tips to the root by dynamic programming.

For any leave b with nucleotide y we have

$$w_b(x) = \begin{cases} 0 & \text{if} \quad x \neq y \\ 1 & \text{if} \quad x = y \end{cases}$$



If k is a node with child nodes i and j and corresponding branch lengths ℓ_i and ℓ_j, then

$$w_k(x) = \left(\sum_{y\in\{A,C,G,T\}} P_{x\to y}(\ell_i) \cdot w_i(y)\right) \cdot \left(\sum_{z\in\{A,C,G,T\}} P_{x\to z}(\ell_j) \cdot w_j(z)\right)$$

5.3 Jukes-Cantor model of sequence evolution

How to compute $P_{x\to y}(\ell)$?

You need a model for sequence evolution.The simples one is the Jukes-Cantor model:

- all sites independent of each other (given the tree)
- all p_x equal
- "mutations" appear at rate λ
- a "mutation" lets the site forget its state and sample the new one uniformly from $\{A, C, G, T\}$. (i.e. A can be replaced by another A)
- (in original paper for protein sequences)

What is a rate?

Let $M_{a,b}$ be the number of "mutations" in time interval $[a, b]$.

- Rate λ means that the expected number of "mutations" in a time interval of length t is λt:
$$\mathbb{E}M_{0,t} = \lambda t$$
- If $\varepsilon > 0$ is extremly small, then the we may neglect the probability of more than one "mutation" in a time interval of length ε.
- Then, $\lambda\varepsilon$ is not only the expected number of mutations but also the probability that there ist one in that time interval:
$$\Pr(M_{0,\varepsilon} > 0) \approx \Pr(M_{0,\varepsilon} = 1) \approx \mathbb{E}M_{0,\varepsilon} = \lambda\varepsilon$$
- numbers of "mutations" on disjoint intervals are stochastically independent

For longer time intervals $[0, t]$ we choose a large $n \in \mathbb{N}$ and argue:

$$\begin{aligned}
\Pr(M_{0,t} = 0) &= \Pr(M_{0,t/n} = 0, M_{t/n,\ 2t/n} = 0, \ldots, M_{(n-1)t/n,\ t} = 0) \\
&= \Pr(M_{0,t/n} = 0) \cdot \Pr(M_{t/n,2t/n} = 0) \cdots \Pr(M_{(n-1)t/n,t} = 0) \\
&\approx \left(1 - \lambda\frac{t}{n}\right)^n \overset{n\to\infty}{\longrightarrow} e^{-\lambda t}
\end{aligned}$$

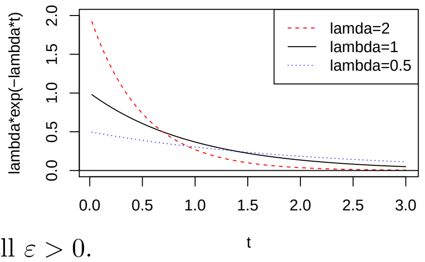
This means: the waiting time τ for the first mutation is exponentially distributed with rate λ. This means it has

$\Pr(\tau > t)\ = e^{-\lambda t}$

expectation value $\mathbb{E}\tau\ = 1/\lambda$

standard deviation $\sigma_\tau\ = 1/\lambda$

density $f(t) = \lambda \cdot e^{-\lambda t}$



This means $\Pr(\tau \in [t, t + \varepsilon]) \approx f(t) \cdot \epsilon$ for small $\varepsilon > 0$.

After this preparation we can finally compute $P_{x\to y}(t)$, first for $y \neq x$:

$$\begin{aligned}
P_{x\to y}(t) &= \Pr(M_{0,t} > 0) \cdot \Pr(\text{last "mutation" leads to } y) \\
&= \left(1 - e^{-\lambda t}\right) \cdot \frac{1}{4}
\end{aligned}$$

and

$$\begin{aligned}
P_{x\to x}(t) &= \Pr(M_{0,t} = 0) + \Pr(M_{0,t} > 0) \cdot \Pr(\text{last "mutation" leads to } x) \\
&= e^{-\lambda t} + \left(1 - e^{-\lambda t}\right) \cdot \frac{1}{4} \\
&= \frac{1}{4} + \frac{3}{4}e^{-\lambda t}
\end{aligned}$$

Overview of DNA substitution models

Jukes-Cantor-Modell (JC): nucleotide type not considered.

from\ to	A	C	G	T
A	—	α	α	α
C	α	—	α	α
G	α	α	—	α
T	α	α	α	—

Kimura's 2 Parameter Model (K2) transitions more frequent than transversions.

from\ to	A	C	G	T
A	—	α	β	α
C	α	—	α	β
G	β	α	—	α
T	α	β	α	—

Felsenstein (1981) **(F81)** takes nucleotide frequencies $(\pi_A, \pi_C, \pi_G, \pi_T)$ into account.

from\ to	A	C	G	T
A	—	$\alpha\pi_C$	$\alpha\pi_G$	$\alpha\pi_T$
C	$\alpha\pi_A$	—	$\alpha\pi_G$	$\alpha\pi_T$
G	$\alpha\pi_A$	$\alpha\pi_C$	—	$\alpha\pi_T$
T	$\alpha\pi_A$	$\alpha\pi_C$	$\alpha\pi_G$	—

Hasegawa, Kishino und Yano (HKY) regards nucleotide frequencies as well as differences between transitions and transversions.

from \ to	A	C	G	T
A	—	$\alpha\pi_C$	$\beta\pi_G$	$\alpha\pi_T$
C	$\alpha\pi_A$	—	$\alpha\pi_G$	$\beta\pi_T$
G	$\beta\pi_A$	$\alpha\pi_C$	—	$\alpha\pi_T$
T	$\alpha\pi_A$	$\beta\pi_C$	$\alpha\pi_G$	—

Felsenstein (1984) **(F84)** also regards nucleotide frequencies and differences between transitions and transversions. No matrix algebra is needed to compute transition probabilities,

from\ to	A	C	G	T
A	—	$\lambda\pi_C$	$\lambda\pi_G + \frac{\mu\pi_G}{\pi_A+\pi_G}$	$\lambda\pi_T$
C	$\lambda\pi_A$	—	$\lambda\pi_G$	$\lambda\pi_T + \frac{\mu\pi_T}{\pi_C+\pi_T}$
G	$\lambda\pi_A + \frac{\mu\pi_A}{\pi_A+\pi_G}$	$\lambda\pi_C$	—	$\lambda\pi_T$
T	$\lambda\pi_A$	$\lambda\pi_C + \frac{\mu\pi_C}{\pi_C+\pi_T}$	$\lambda\pi_G$	—

The **General Time-Reversible Model (GTR)** considers differences between pairs of nucleotide types.

von\ nach	A	C	G	T
A	—	$\alpha\pi_C$	$\beta\pi_G$	$\gamma\pi_T$
C	$\alpha\pi_A$	—	$\delta\pi_G$	$\epsilon\pi_T$
G	$\beta\pi_A$	$\delta\pi_C$	—	$\eta\pi_T$
T	$\gamma\pi_A$	$\epsilon\pi_C$	$\eta\pi_G$	—

In the models F81, F84, HKY and GTR, $(\pi_A, \pi_C, \pi_G, \pi_T)$ is the stationary distribution, in JC and K2 $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$. All these models are reversible. We will discuss later how to calculate transition probabilities $P_{x\to y}(t)$ for all these models.

Some of the things you should be able to explain

- What is a likelihood and why don't we just say "probability"?
- How to calculate the likelihood of a tree with Felsenstein's pruning algorithm

- What exactly is the meaning of $w_k(A)$ in Felsentein's pruning algorithm?
- How does the runtime of Felsenstein's pruning algorithm scale with the size of the tree?
- What is a mutation rate and what is the probability distribution is the time until a mutation occurs?

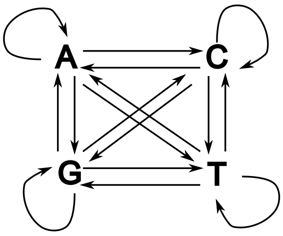
5.4 Reversibility and convergence into equilibrium

Markov chain

A (discrete-time) Markov chain is a sequence of random variables $X_1, X_2, X_3, \dots$ on a state space $\mathcal{Z}$ such that for every "time point" n the next state X_{n+1} depends only on the present X_n but not additionally on the previous states.As formula:

$$\forall_{n\in\mathbb{N},a,b,c,d,\dots\in\mathcal{Z}} \Pr(X_{n+1}=a \mid X_n=b) = \Pr(X_{n+1}=a \mid X_n=b, X_{n-1}=c, X_{n-2}=d, \dots)$$

Example: X_n is the nucleotide at some position in generation n.
(The Markov assumption is a simplification.)



Markov jump process

For each time point $t > 0$ there is a random variable X_t on the state space $\mathcal{Z}$ such that if $X_t = z$ then a "jump" to a different state $y \in \mathcal{Z}$ occurrs a rate λ that only depends on X_t but in addition to this not on any X_s with $s < t$.

Examples: Jukes-Cantor, K2, HKY, F84, GTR,...

Let $X = (X_1, X_2, \dots)$ or $(X_t)_{t\in\mathbb{R}_{\leq 0}}$ be a Markov chain with finite state space $\mathcal{Z}$ and transition probabilities $P_{x\to y}(t)$ for $t \in \mathbb{N}$ or $t \in \mathbb{R}_{\geq 0}$.

The transition dynamics P is **irreducible**, if

$$\forall_{x,y\in\mathcal{Z}} \exists_t : \ P_{x\to y}(t) > 0.$$

In the discrete-time case, P is **periodic**, if

$$\exists_{z\in\mathcal{Z},k>1} \forall_{n\in\mathbb{N}\setminus\{k,2k,3k,\dots\}} P_{z\to z}(n) = 0$$

Otherwise, P is called **aperiodic**.

Note that the models JC, K2, F81, F84, HKY and GTR are

irreducible because mutations from each state to each other state have positive probability and

aperiodic because the possibility to stay in a state for arbitrary time already destroys all periodicities. (That is, continuous-time Markovian jump processes are always aperiodic.)

Theorem 4 *Each aperiodic irreducible transition dynamic (or rate matrix) P on a finite state space $\mathcal{Z}$ has one and only one* ***stationary distribution*** $(\pi_z)_{z\in\mathcal{Z}}$*, i.e.*

$$\forall_{z\in\mathcal{Z}} \qquad \pi_z = \sum_{x\in\mathcal{Z}} \pi_x \cdot P_{x\to z},$$

and converges against this distribution in the sense that

$$\forall_{x,z} \lim_{t\to\infty} P_{x\to z}(t) = \pi_z,$$

where $P_{x\to z}(t)$ is the transition probability from x to z for time span t.

An equivalent expression for **stationary distribution** is **equilibrium distribution**.

Sketch of proof of convergence: Start two Markov chains X and Y with transition matrix P, one with X_1 in x and one with Y_1 taken from the stationary distribution. When they meet in some step k, i.e. if $X_k = Y_k$, couple them: $X_j = Y_j$ for all $j > k$. If P is irreducible and aperiodic, and the probability q_k that X and Y do not meet before step k converges to 0, and

$$\begin{aligned}
|\Pr(X_j = z) - \pi_z| &= |\Pr(Y_j = z) - \Pr(X_j = z)| \\
&= |\Pr(Y_j = z, X_j = Y_j) + \Pr(Y_j = z, X_j \neq Y_j) \\
&\quad - \Pr(X_j = z, X_j = Y_j) - \Pr(X_j = z, X_j \neq Y_j)| \\
&= |\Pr(Y_j = z, X_j \neq Y_j) - \Pr(X_j = z, X_j \neq Y_j)| \\
&\leq \max\{\Pr(Y_j = z, X_j \neq Y_j)\ ,\ \Pr(X_j = z, X_j \neq Y_j)\} \\
&\leq q_j \longrightarrow 0.
\end{aligned}$$

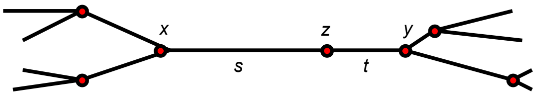
A Markov chain with transition matrix P (or rate matrix P in the continuous-time case) and stationary distribution $(\pi_z)_{z\in\mathcal{Z}}$ is **reversible** if

$$\forall_{z,y\in\mathcal{Z}}:\ \pi_z \cdot P_{z\to y} = \pi_y \cdot P_{y\to z}.$$

("detailed-balance condition")

Note: the detailed-balance condition already implies that $(\pi_z)_{z\in\mathcal{Z}}$ is a stationary distribution of P.

The evolutionary dynamics described by Jukes-Cantor, F81, F84, HKY, GTR or PAM matrices are reversible. If we assume reversibility and no molecular clock, the likelihood does not depend on the position of the root in the tree topology.



If the root divides a branch of length $s + t$ into sections of length s and t, reversibility implies that the probability stays the same if we move the root into one of the nodes:

$$\begin{aligned}
\sum_z \pi_z \cdot P_{z\to x}(s) \cdot P_{z\to y}(t) &= \sum_z \pi_x \cdot P_{x\to z}(s) \cdot P_{z\to y}(t) \\
&= \pi_x \cdot P_{x\to y}(s+t) \\
&= \pi_y \cdot P_{y\to x}(s+t)
\end{aligned}$$

Some of the things you should be able to explain

- JC, K2, F81, HKY, GTR
- What is a Markov chain and what is convergence of a Markov chain?
- What is a stationary distribution and under what conditions will a Markov chain on a finite state space converge against it?
- What do "equilibrium", "stationary distribution", "reversibility" and "detailed balance" have to do with each other?
- consequences of reversibility of substitution models for the placement of the root in ML trees
- examples in which reversibility does not hold for in sequence evolution and why it is still common as a model assumption

5.5 How to search for the ML tree

Given a large number n of taxa (i.e. sequences), it is difficult to find the ML phylogeny. Two partial problems have to be solved:

1. Given the tree topology, find the optimal branch lengths
2. Find the tree topology for which your solution of problem 1 leads to the highest likelihood value.

We first turn to problem 1.

Tree length optimization in the very first version of PHYLIP dnaml

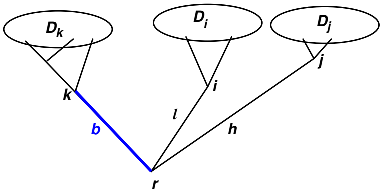
Expectation-Maximization (EM) algorithm: Iterate the following steps:

E step given the current branch lengths and rates, compute the expected number of mutations for each branch

M step optimize branch lengths for the expected numbers of mutations computed in the E step

More common: use the derivative of the likelihood with respect to the branch length

To optimize the length b of some branch, first rotate it, such that one of its adjacent nodes is the root.



First we assume that the alignment D has only one column. Then, $L_D(T)$ is $\sum_x p_x \cdot \sum_y P_{x\to y}(b) \cdot w_k(y) \cdot (\sum_z P_{x\to z} w_i(z)) \cdot (\sum_{z'} P_{x\to z'} w_j(z'))$.

$\Rightarrow$

$\frac{\partial L_D(T)}{\partial b} = \sum_x p_x \cdot \sum_y \frac{\partial P_{x\to y}(b)}{\partial b} \cdot w_k(y) \cdot (\sum_z P_{x\to z} w_i(z)) \cdot (\sum_{z'} P_{x\to z'} w_j(z'))$

and $\frac{\partial^2 L_D(T)}{\partial b^2} = \sum_x p_x \cdot \sum_y \frac{\partial^2 P_{x\to y}(b)}{\partial b^2} \cdot w_k(y) \cdot (\sum_z P_{x\to z} w_i(z)) \cdot (\sum_{z'} P_{x\to z'} w_j(z'))$

In the Jukes-Cantor model we can compute for example for $x \neq y$:

$$\begin{aligned}
\frac{\partial}{\partial b} P_{x\to y}(b) &= \frac{\partial}{\partial b}\left(1 - e^{-\lambda b}\right) \cdot \frac{1}{4} = \frac{1}{4}\lambda e^{-\lambda b} \\
\frac{\partial^2}{\partial b^2} P_{x\to y}(b) &= -\frac{1}{4}\lambda^2 e^{-\lambda b}
\end{aligned}$$

For alignments D with columns $D_1 \ldots D_m$ we can compute all $L'_h := \frac{\partial}{\partial b} L_{D_h}(T)$ and $L''_h := \frac{\partial^2}{\partial b^2} L_{D_h}(T)$ as explained above, and then compute the first two derivatives of $L_D(T) = \prod_h L_{D_h}(T)$ by applying the product rule for derivatives:

$$\frac{\partial}{\partial b} L_D(T) = L_D(T) \cdot \sum_h \frac{L'_h}{L_{D_h}(T)}$$

and

$$\frac{\partial^2}{\partial b^2} L_D(T) = L_D(T) \cdot \sum_h \left(\frac{L''_h}{L_{D_h}(T)} + \sum_{\ell \neq h} \frac{L'_h \cdot L'_\ell}{L_{D_h}(T) \cdot L_{D_\ell}(T)} \right)$$

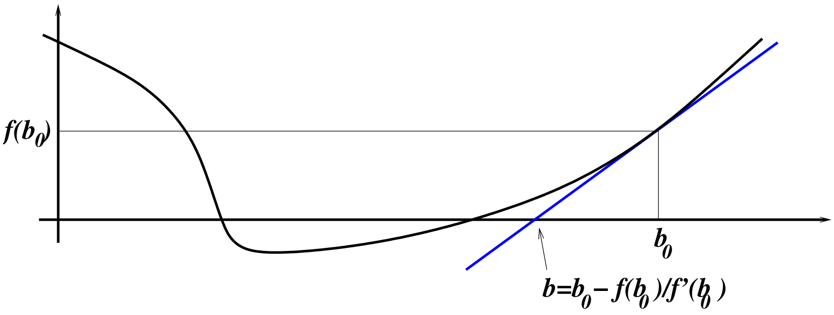
To optimize b, solve

$$f(b) := \frac{\partial L_D(T)}{\partial b} = 0.$$

This is done numerically with a Newton-Raphson scheme using $f'(b) = \frac{\partial^2 L_D(T)}{\partial b^2}$.

Newton-Raphson scheme to solve $f(b) = 0$

1. Start with some initial value b_0
2. as long as $f(b_0)$ is not close enough to 0, replace b_0 by $b_0 - f(b_0)/f'(b_0)$ and try again.



Optimizing the topology
Now that we know how to search for the optimal tree, given the topology, how do we search for the best topology?

stepwise addition (default of DNAML):

- start with the only possible tree of the first three taxa
- stepwise add one taxon
- to do this when k taxa are already added, try all $2k - 5$ possible branches to add the next taxon, optimize branch lengths
- when all are added, optimize with NNI steps
- repeat whole procedure with different input orders

branch and bound if only few taxa
Start with NeighborJoining and continue with SPR is nowadays most common
Supertree methods like TREE-PUZZLE: ML for all quartets, then build tree that respects most of them.

Some of the things you should be able to explain

- What is the Newton-Raphson scheme and how is it used in ML phylogeny methods
- How to modify the Felsenstein pruning algorithm to calculate derivatives of the likelihood
- strategies of DNAML and more recent ML programs to optimize the topology

5.6 Maximum Parsimony from a probabilistic perspective

If we assume a probabilistic substitution model, we can set $s(a, b) = -\log P_{a\to b}(1)$ and use the values $s(a, b)$ as costs in weighted parsimony. Thus, maximum parsimony can be considered as an approximation for the case that

1. all edges in the tree have the same length
2. double-hits and back-mutations are negligible

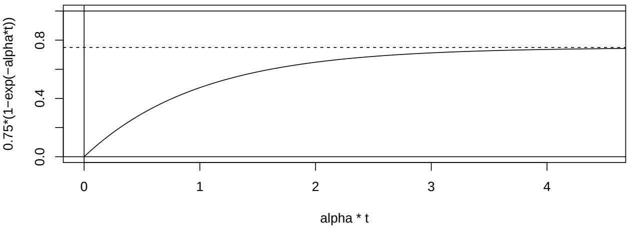
5.7 Maximum likelihood for pairwise distances

Jukes-Cantor model

Let the rate of “mutations” in which nucleotides “forget” their type be α.

Probability of segregating site with branch length t:

$$\frac{3}{4} \cdot \left(1 - e^{-\alpha \cdot t}\right)$$



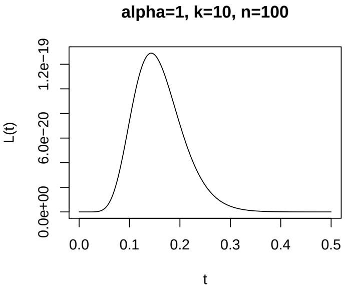
This is also the expectation value for the fraction of sites that are segregating.

Given a substitution model with known parameters we can compute the ML distance d_{xy}^{ML} between sequence $x = (x_1, x_2, \ldots, x_n)$ and sequence $y = (y_1, \ldots, y_n)$ by

$$\begin{aligned} d_{xy}^{\mathrm{ML}} &= \arg\max_t \left\{ \prod_i \pi_{x_i} \cdot P_{x_i \to y_i}(t) \right\} \\ &= \arg\max_t \left\{ \prod_i P_{x_i \to y_i}(t) \right\} \end{aligned}$$

E.g. for the Jukes-Cantor Model with rate α we get in the case of k mismatches:

$$\prod_i P_{x_i \to y_i}(t) = \left(\frac{1}{4}(1 - e^{-t\alpha})\right)^k \left(\frac{1}{4}(1 + 3e^{-t\alpha})\right)^{n-k}$$



Optimizing this with the usual procedure we get:

$$d_{xy}^{\mathrm{ML}} = -\frac{1}{\alpha} \cdot \ln\left(1 - \frac{4k}{3n}\right)$$

This ML estimator is consistent, i.e. will give us the true distances in the limit of long sequences. This implies that applying NeighborJoining to the ML distances is also consistent.

If the sequecens are not extremely long, direct ML methods may tend to give more reliable results (as long as they are computationally tractable.)

5.8 Consistency of the Maximum-Likelihood method

Theorem 5 *The ML estimator for phylogenetic trees is **consistent**. This means, if the model assumptions are fulfilled and you add more and more data (i.e. make the sequences longer) for a fixed set of taxa, the probability that the ML tree will converge against the true tree is 1.*

Note:

1. the ML tree is the tree with the highest likelihood. ML tree estimation programs do not always find the ML tree

2. the model assumptions include a model for the substitution process and that all sequence positions are independent and correctly aligned

Sketch of proof for the consistency of the ML tree:

Let $a_1, \ldots, a_m$ be the different alignment columns and let $n_1, \ldots, n_m$ be their numbers of occurrences in the data D. The likelihood of a tree T is then

$$L(T) = \Pr(D \mid T) = \prod_{i=1}^{m} \Pr(a_i \mid T)^{n_i}$$

Idea of the proof: the probabilities $\Pr(a_i \mid T)$ are characteristic for T, and those of the true tree will be reflected in the relative frequencies $R_i = n_i/n$ with $n = n_1 + \cdots + n_m$.

The log likelihood is

$$\ln L(T) = \sum_{i=1}^{m} n_i \ln \Pr(a_i \mid T) = n \cdot \sum_{i=1}^{m} R_i \ln \Pr(a_i \mid T)$$

For long sequences we get $R_i \to p_i$, where $p_1, \ldots, p_m$ are the (unknown) probabilities of $a_1, \ldots, a_m$ for the true tree T^*. Let $q_1, \ldots, q_m$ be those probabilities for some other tree T.Then we obtain

$$\frac{1}{n} \ln L(T) \overset{n\to\infty}{\longrightarrow} \sum_{i=1}^{m} p_i \ln q_i \qquad \text{and} \qquad \frac{1}{n} \ln L(T^*) \overset{n\to\infty}{\longrightarrow} \sum_{i=1}^{m} p_i \ln p_i.$$

For $p \neq q$ we get

$$\sum_{i=1}^{m} p_i \ln q_i < \sum_{i=1}^{m} p_i \ln p_i,$$

because

$$\sum_{i=1}^{m} p_i \ln p_i - \sum_{i=1}^{m} p_i \ln q_i = \sum_{i=1}^{m} p_i \ln \frac{p_i}{q_i} > 0,$$

and the last inequation follows since $\sum_{i=1}^{m} p_i \ln \frac{p_i}{q_i}$ is the relative entropy, also called Kullback-Leibler-Information, which is positive for $p \neq q$. $\sum_{i=1}^{m} p_i \ln \frac{p_i}{q_i} = -\sum_{i=1}^{m} p_i \ln \frac{q_i}{p_i} > -\sum_{i=1}^{m} p_i \left(\frac{q_i}{p_i} - 1\right) = -\sum_{i=1}^{m} q_i + \sum_{i=1}^{m} p_i = -1 + 1 = 0$

Some of the things you should be able to explain

- What does consistency of ML tree reconstruction mean?
- implicit model assumptions in parsimony (from a frequentist perspective)
- How to estimate evolutionary distances of sequences accounting for back-mutations and double hits

6 Bootstrapping

6.1 The concept of bootstrapping

Assume a panmictic Hardy-Weinberg population and a locus in equilibrium with genotypes MM, MN, and NN.This means, the frequencies of these genotypes are $(1-\theta)^2$, $2\theta(1-\theta)$, and θ^2, where θ is the frequency of allele N.

Assume the following observations:

MM	MN	NN	total
342	500	187	1029
X	Y	Z	

(Example taken from Rice (1995) *Mathematical Statistics and Data Analysis.* Duxbury press.)

We estimate θ by $\widehat{\theta} = \frac{2Z+Y}{2(X+Y+Z)} = 0.4247$. How accurate is this estimation?

Simulate 1000 datasets, each consisting of 1029 individuals drawn from a Hardy-Weinberg population with frequency 0.4247 of allele N.

Let $\theta_1^*, \theta_2^*, \ldots, \theta_{1000}^*$ be the estimates of θ from the 1000 datasets. We can then estimate the standard deviation of our estimator $\widehat{\theta}$ by

$$\sigma_{\widehat{\theta}} \approx \sqrt{\frac{\sum_i \left(\theta_i^* - \widehat{\theta}\right)^2}{1000}}$$

Bootstrapping is a general approach in statistics that is often used to assess the accuracy of an estimator.

It is based on the following idea: If we estimate a parameter θ by $\widehat{\theta}$, we can check the accuracy of the estimation method with simulated data.

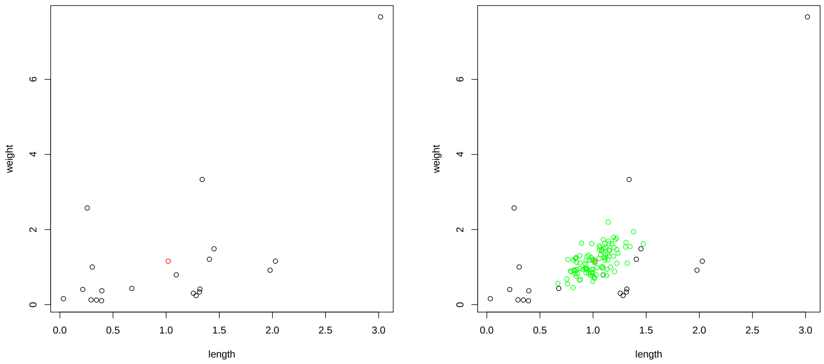
Problem: We do not know the true value of θ but need a value for the simulations.

idea: We pull ourselves up by our own bootstraps by using $\widehat{\theta}$ for the simulations and assume that the difference $\widehat{\theta} - \theta^*$, where θ^* is the estimation from the simulated data, has a similar distribution as $\theta - \widehat{\theta}$:

$$\mathcal{L}(\theta - \widehat{\theta}) \approx \mathcal{L}(\widehat{\theta} - \theta^*)$$

Since wie use the parameter and assumptions about its distribution, this is called *paramteric bootstrap.* In the next example we use *non-parametric bootstrapping*, which means that we just the original data to simulate new data.

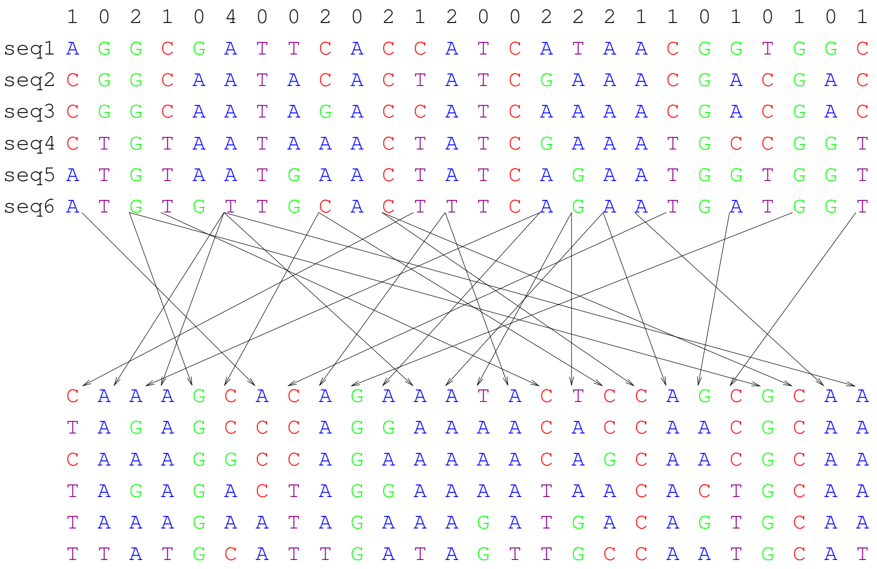
We have caught 20 fishes from a lake and want to estimate the distribution of size and weight in the population by the sample means. How accurate is this estimation? Idea: simulate sampling from a population by putting the 20 fishes into a pond and take a sample of size 20. To avoid getting precisely the same sample, *sample with replacement.* Compute the mean length and weight from the "bootstrap sample". Repeat this procedure 1000 times. The 1000 pairs of means can be used for bias correction and to estimate the variance of the estimator.



Bias correction: $\widehat{\theta} - (\overline{\theta^*} - \widehat{\theta}) = 2\widehat{\theta} - \overline{\theta^*}$

6.2 Bootstrap for phylogenetic trees

non-parametric bootstrap of an alignment



```
     1 0 2 1 0 4 0 0 2 0 2 1 2 0 0 2 2 2 1 1 0 1 0 1 0 1
seq1 A G G C G A T T C A C C A T C A T A A C G G T G G C
seq2 C G G C A A T A C A C T A T C G A A A C G A C G A C
seq3 C G G C A A T A G A C C A T C A A A A C G A C G A C
seq4 C T G T A A T A A A C T A T C G A A A T G C C G G T
seq5 A T G T A A T G A A C T A T C A G A A T G G T G G T
seq6 A T G T G T T G C A C T T T C A G A A T G A T G G T

     C A A A G C A C A G A A A T A C T C C A G C G C A A
     T A G A G C C C A G G A A A A C A C C A A C G C A A
     C A A A G G C C A G A A A A A C A G C A A C G C A A
     T A G A G A C T A G G A A A A T A A C A C T G C A A
     T A A A G A A T A G A A A G A T G A C A G T G C A A
     T T A T G C A T T G A T A G T T G C C A A T G C A T
```

A bootstrap alignment has the same length as the original alignment. It consists of columns that were randomly drawn from the original alignment with replacement. To the bootstrap alignment we apply the same phylogeny reconstruction method as for the original alignment.

We repeat this many times and thus get many bootstrap trees. We label each branch of our originally reconstruted tree by the percentages of bootstrap trees that have this branch. These bootstrap values are supposed to give an impression of how reliable the branches are.

Alternatives to non-parametric bootstrap:

Jackknife: Create shorter alignments, e.g. 90%, by sampling without replacement. Like in non-parametric bootstrapping, the bootstrap dataset is slightly less informative than the original data.

Parametric Bootstrap: Use the estimated tree and substitution rates estimated along with the tree to simulate new data. (Disadvantage: does not take uncertainty about the substitution model into account.)

6.3 How can we interprete the bootstrap values?

There are at least three different interpretations of the bootstrap values of tree branches:

1. posterior probability of the branch
2. measures of repeatability
3. confidence levels for the existence of the branch

None of these interpretations is perfect.

Are bootstrap values posterior probabilities?

Rather not, because posterior probabilities depend on the prior, and the bootstrap values do not (at least if a non-Bayesian method was used for tree reconstruction).

Do bootstrap values measure repeatability?

This is the original interpretation of Felsenstein, who first proposed bootstrapping for phylogenetic trees. However, the bootstrap value can only be an approximative measure because the bootstrap sample is slightly less informative than the original sample. The question is also what repeatability would actually mean? If the analysis is repeated with different data, varitions between loci may play a role, which is not incorporated in bootstrapping.

Are bootstrap values confidence levels?

If a branch has a bootstrap value 97% and this is interpreted as confidence level, then this means the following: Under the null hypothesis that the branch is actually not there or has length 0, the probability of getting a bootstrap support of 97% is 100%-97%=3%. This means: Among all branches that appear in the estimated trees but are actually wrong, only 3% get such a high bootstrap level.

It has been conjecured that bootstrap values underestimate confidence because bootstrap datasets are less informative than the original dataset. However, this argument disregards that the bootstrap result θ^* does not need to be an approximation for θ, but $\theta^* - \widehat{\theta}$ should be an approximation for $\widehat{\theta} - \theta$.

References

[EHH96] B. Efron, E. Halloran, S. Holmes (1996) Bootstrap confidence levels for phylogenetic trees. *Proc. Nat. Acad. Sci. U.S.A.* **93(13)**:429–434

show that bootstrap values can either over- or underestimate confidence, but are at least first-order approximations of confidence values. They propose a meta-bootstrap procedure to correct the over- or underestimation for each branch.

Some of the things you should be able to explain

- difference between parameteric and non-parameteric bootstrap
- how is bootstrap applied in phylogenetics
- Basic assumption of bootstrapping and what it means, e.g. for bias correction
- possible interpretations of bootstrap values on branches and why none of these interpretations is perfect

7 Bayesian phylogeny reconstruction and MCMC

7.1 Principles of Bayesian statistics

In Bayesian statistics, also model parameters are random variables and thus have probability distributions.

E.g. for a phylogenetic tree T:

prior probability distribution: $P(T)$ is the probability density of the tree T disregarding the data, e.g. we could a priori assume a uniform probability density for all trees up to a certain total branch length.

posterior probability distribution: $P(T|D)$ is the conditional probability density of the tree T, given the data D.

Bayes-Formula:

$$P(T|D) = \frac{P(T,D)}{\Pr(D)} = \frac{\Pr(D|T) \cdot P(T)}{\int_{T'} \Pr(D|T') \cdot P(T')\ dT'}$$

Computing

$$P(T|D) = \frac{\Pr(D|T) \cdot P(T)}{\int_{T'} \Pr(D|T') \cdot P(T')\ dT'}$$

is not trivial. We can compute $\Pr(D|T) = \Pr_T(D) = L_D(T)$ by Felsenstein pruning and $P(T)$ is defined by our prior distribution, but integrating over all trees is difficult.

What we can compute is the ratio of the probabilities of two candidate trees T_A and T_B:

$$\frac{P(T_A|D)}{P(T_B|D)} = \frac{\frac{\Pr(D|T_A) \cdot P(T_A)}{\int_{T'} \Pr(D|T') \cdot P(T')\ dT'}}{\frac{\Pr(D|T_B) \cdot P(T_B)}{\int_{T'} \Pr(D|T') \cdot P(T')\ dT'}} = \frac{\Pr(D|T_A) \cdot P(T_A)}{\Pr(D|T_B) \cdot P(T_B)}$$

7.2 MCMC sampling

We are not just interested in finding the *maximum a-posteriori* (MAP) tree

$$\text{arg max}_T P(T|D),$$

but, very much in the spirit of Bayesian statistics, to sample trees from the posterior distribution, that is, to generate a set of (approximately) independent random trees $T_1, T_2, \ldots, T_n$ according to the probability distribution given by $P(T|D)$. This will allow us not only to infer the phylogeny but also to assess the uncertainty of this inferrence.

Idea: Simulate a Markov chain on the space of trees with stationary distribution $P(T|D)$ and let it converge.

How can we do that if we can only compute ratios $\frac{P(T_A|D)}{P(T_B|D)}$ for given trees T_A and T_B?
Given the probability distribution $\Pr(.|D)$, how can we construct a Markov chain that converges against it?

One possibility: **Metropolis-Hastings**
Given current state $X_i = x$ propose y with Prob. $Q(x \to y)$
Accept proposal $X_{i+1} := y$ with probability

$$\min\left\{1, \frac{Q(y \to x) \cdot \Pr(y \mid D)}{Q(x \to y) \cdot \Pr(x \mid D)}\right\}$$

otherwise $X_{i+1} := X_i$

(All this also works with continuous state space, with some probabilities replaced by densities.)

Why Metropolis-Hastings works

Let's assume that $\frac{Q(y \to x) \cdot \Pr(y \mid D)}{Q(x \to y) \cdot \Pr(x \mid D)} \leq 1$. (Otherwise swap x and y in the following argument).Then, if we start in x, the probability $\Pr(x \to y)$ to move to y (i.e. first propose and then accept this) is

$$Q(x \to y) \cdot \frac{Q(y \to x) \cdot \Pr(y \mid D)}{Q(x \to y) \cdot \Pr(x \mid D)} = Q(y \to x)\frac{\Pr(y \mid D)}{\Pr(x \mid D)}$$

If we start in y, the probability $\Pr(y \to x)$ to move to x is

$$Q(y \to x) \cdot 1,$$

since our assumption implies $\frac{Q(x \to y) \cdot \Pr(x \mid D)}{Q(y \to x) \cdot \Pr(y \mid D)} \geq 1$.
This implies that the reversibility condition

$$\Pr(x \mid D) \cdot \Pr(x \to y) = \Pr(y \mid D) \cdot \Pr(y \to x)$$

is fulfilled.This implies that $\Pr(. \mid D)$ is an equilibrium of the Markov chain that we have just constructed, and the latter will converge against it.(let's watch a simulation in R)

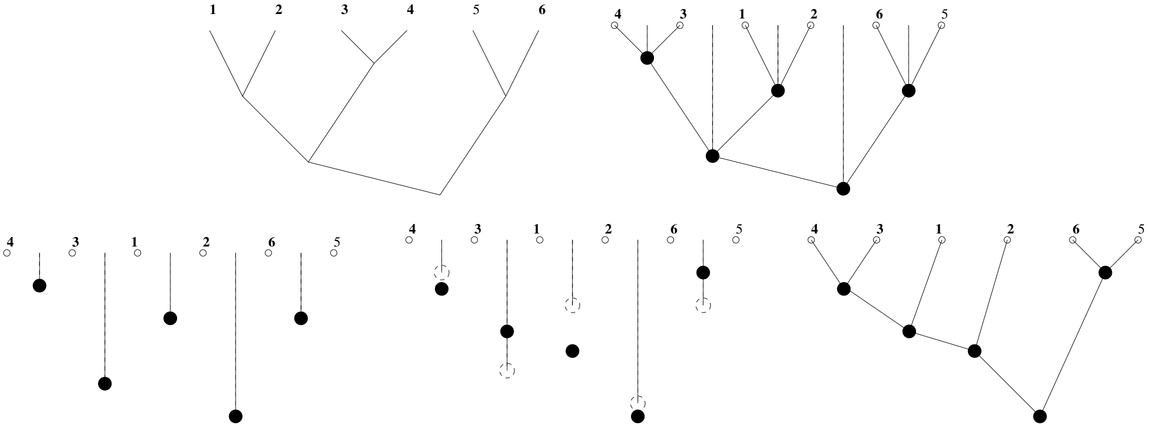
Applying Metropolis-Hastings

- You are never in equilibrium (your target distribution), but you can get close if you run enough steps.
- You can take more than one sample from the same chain, but you should run enough steps between the sampling steps to make the sampled objects only weakly dependent.
- Your initial state may be "far from equilibrium" (i.e. very improbable). So you should run the chain long enough before you start sampling ("burn-in").
- Launch many independent MCMC runs with different starting points and check whether they lead to the same results.

Mau, Newton, Larget 1996

Seminal paper on MCMC for phylogenies; propose a propsal chain for ultrametric trees.

1. Draw the tree in the plane.
2. In each internal node rotate subtrees with probability 1/2.
3. Remove edges from drawing.
4. Shift each internal node in time by a random amount.
5. Reconstruct edges from modified time points of nodes.



Most programs for Bayesian phylogeny inferrence can also estimate parameters of the substitution model. Combine the estimation of trees with the estimation of divergence times or even alignments.

Gibbs sampling is applied to combine Bayesian estimations for different kinds of parameters.

Gibbs samping

Assume we want to sample from a joint distribution $\Pr(A = a, B = b)$ of two random variables, and for each pair of possible values (a, b) for (A, B) we have Markov chains with transition probabilities $P^{(A=a)}_{b\to b'}$ and $P^{(B=b)}_{a\to a'}$ that converge against $\Pr(B = b|A = a)$ and $\Pr(A = a|B = b)$.

Then, any Markov chain with transition law

$$P_{(a,b)\to(a',b')} = \begin{cases} \frac{1}{2}P^{(B=b)}_{a\to a} + \frac{1}{2}P^{(A=a)}_{b\to b} & \text{if} \quad a = a' \quad \text{and} \quad b = b' \\ \frac{1}{2}P^{(B=b)}_{a\to a'} & \text{if} \quad a \neq a' \quad \text{and} \quad b = b' \\ \frac{1}{2}P^{(A=a)}_{b\to b'} & \text{if} \quad a = a' \quad \text{and} \quad b \neq b' \\ 0 & \text{else} \end{cases}$$

Most software packages use more common tree modifications like NNI, SPR and TBR.

Examples of software for Bayesian sampling:

MrBayes http://mrbayes.csit.fsu.edu/

RevBayes https://revbayes.github.io/

BEAST http://beast.bio.ed.ac.uk/Main_Page

BEAST2 http://www.beast2.org/

PhyloBayes http://www.atgc-montpellier.fr/phylobayes/binaries.php

BAli-Phy http://www.bali-phy.org/

TreeTime http://evol.bio.lmu.de/_statgen/software/treetime/

(MC)^3=**MCMCMC**

=Metropolis-Coupled MCMC= MCMC with "heated chains".

If $\beta_i \in (0,1]$, where $T_i = 1/\beta_i$ can be considered as "temperature" for chain i, then chain i samples from distribution p_i with $p_i(x) = p^{\beta_i}(x)\cdot$const. (For the unheated chain we have $\beta_1 = 1$ and thus $p_1 = p$.)

The usual MH acceptance prob. for chain i is

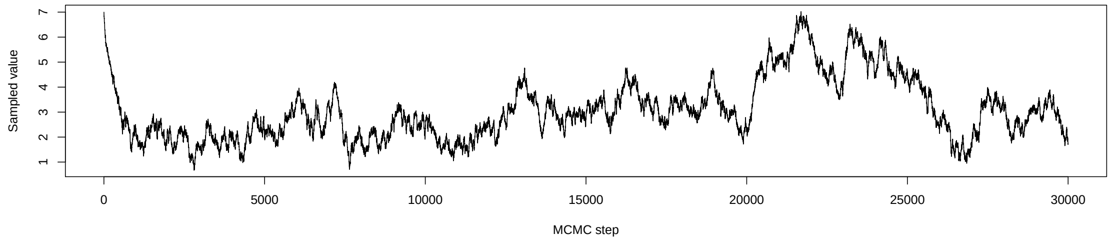
$$\min\left\{1, \frac{p_i(y)}{p_i(x)} \cdot \frac{Q_{y\to x}}{Q_{x\to y}}\right\} = \min\left\{1, \frac{p(y)^{\beta_i}}{p(x)^{\beta_i}} \cdot \frac{Q_{y\to x}}{Q_{x\to y}}\right\}.$$

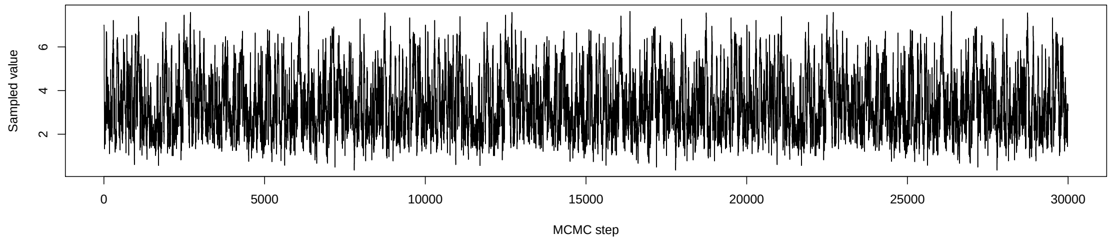
Sometimes a swap between the current state x_i of chain i and the current state x_j of chain j is proposed. The acceptance with probability

$$\min\left\{1, \frac{p(x_j)^{\beta_i}}{p(x_i)^{\beta_i}} \cdot \frac{p(x_i)^{\beta_j}}{p(x_j)^{\beta_j}}\right\}$$

fulfills the requirements of both chaines (check this!).

7.3 Checking convergence of MCMC





Effective Sampling Size (ESS)

Assume that we want to estimate the expectation value μ of a distribution by taking the mean $\overline{X}$ of n independent draws $X_1, X_2, \ldots, X_n$ from the distribution with variance σ^2. Then,

$$\begin{aligned}
\mathbb{E}\overline{X} &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}X_i = \mu \\
\mathrm{var}(\overline{X}) &= \mathrm{var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n^2}\mathrm{var}\left(\sum_{i=1}^{n} X_i\right) = \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{var}\left(X_i\right) = \frac{1}{n}\sigma^2.
\end{aligned}$$

If we instead use m *correlated* draws $Y_1, Y_2, \ldots, Y_m$ from the same distribution, then

$$\begin{aligned}
\mathbb{E}\overline{Y} &= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}Y_i = \mu \\
\mathrm{var}(\overline{Y}) &= \mathrm{var}\left(\frac{1}{m}\sum_{i=1}^{m} Y_i\right) = \frac{1}{m}\sigma^2 + \frac{2}{m^2}\sum_{i=1}^{m}\sum_{j=i+1}^{m}\mathrm{cov}(Y_i, Y_j).
\end{aligned}$$

Effective Sampling Size (ESS)

$$\frac{1}{n}\sigma^2 = \frac{1}{m}\sigma^2 + \frac{2}{m^2}\sum_{i=1}^{m}\sum_{j=i+1}^{m}\operatorname{cov}(Y_i, Y_j)$$

With the autocorrelation $\rho_k = \operatorname{cor}(Y_i, Y_{i-k}) = \operatorname{cov}(Y_i, Y_{i-k})/\sigma^2$, $\overline{Y}$ has (approximately) the same variance as $\overline{X}$, if

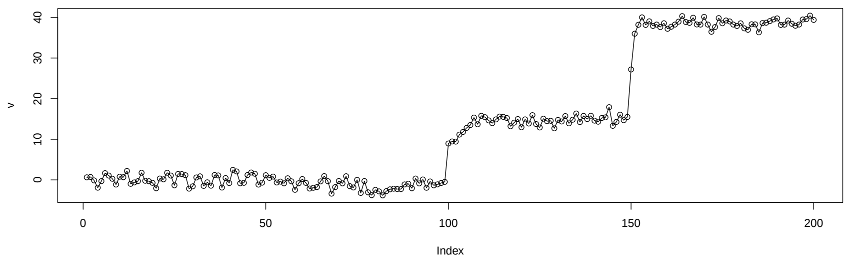
$$n = \frac{m}{1 + 2\cdot\sum_{k=1}^{\infty}\rho_k}.$$

Therefore, we estimate the Effective Sample Size by

$$ESS = \frac{m}{1 + 2\cdot\sum_{k=1}^{\infty}\widehat{\rho}_k},$$

where $\widehat{\rho}_k$ is an estimation of the autocorrelation $\rho_k := \operatorname{cor}(Y_i, Y_{i-k})$.

Problem: ESS may be too optimistic because correlation may be underestimated.



estimated effective sample sizes:

range	:	ess
1-90	:	7.88
110-140	:	31.00
160-200	:	28.77
1-200	:	1.53

Ways to check convergence of MCMC

- ESS
- visually inspect paths of log likelihood and parameter estimates
- start many MCMC runs with different start values and check whether they appear to converge against the same distribution

7.4 Interpretation of posterior probabilities and robustness

If the prior is correctly chosen and the model assumptions are fulfilled, the posterior probability of a tree topology should be the probability that the topology is correct.This is confirmed for trees with six taxa in a simulation study in:

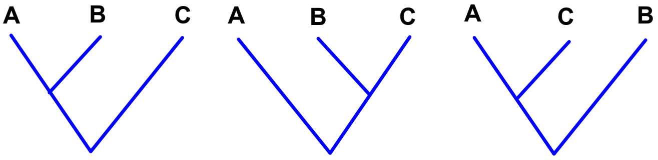
References

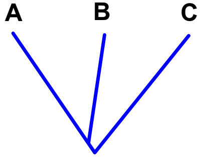
[HR04] J.P. Huelsenbeck, B. Rannala (2004) Frequentist Properties of Bayesian Posterior Probabilities of Phylogenetic Trees Under Simple and Complex Substitution Models. *Syst. Biol.* **53(6)**:904–913.

However, when a the model chosen for the substitution process is too simple (e.g. neglecting rate heterogeneity), the estimated posterior probabilities can be over-optimistic.Using a model that is more complex than necessary, may lead to just slightly conservative estimates of posterior probabilities. Recommendation: If you are not sure, rather use the more complex substitution model.

References

[YR05] Z. Yang, B. Rannala (2005) Branch-Length Prior Influences Bayesian Posterior Probability of Phylogeny *Syst. Biol.* *54(3)*:455–470





simulate rooted ultrametric trees with three tips and different priors for lengths of inner and outer branches. Compute posterior probabilities for the three possible topologies with various priors for tree lengths.

- MAP estimates are robust against misspecification of prior.
- High posteriors are underestimated and low posteriors are overestimated if prior favors very short internal edges.
- High posteriors are overestimated and low posteriors are underestimated if priors for internal edge lengths are flat.

Note: flat priors are sometimes called "uninformative", but this is misleading, and in Yang and Rannala's study these priors were most problematic!

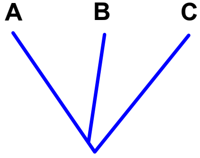
To decrease the risk of too optimistic posteriors for tree topologies when the substitution process is inappropriate,

References

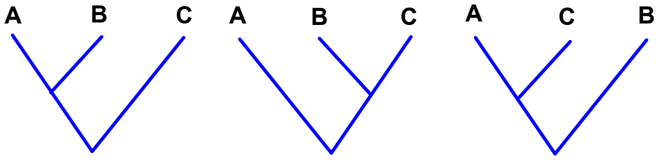
[Y08] Z. Yang (2008) Empirical evaluation of a prior for Bayesian phylogenetic inference *Phil. Trans. R. Soc. B* **363**: 4031–4039

recommends using priors favoring shorter internal branch lengths if the input alignment is long.

Star-tree paradox



If the inner branch of a rooted 3-taxa tree is extremely short, or even non-existing, and the Bayesian method takes only binary trees into account with "liberal" priors for the branch lenghts, it will often assign a high posterior probability to one of the three tree topologies, and with probability $\approx 2/3$ it will be a wrong one.



This is related to the *fair-coin paradox* and *Lindley's paradox*, which we will discuss in the context of Bayesian model selection.

Some of the things you should be able to explain

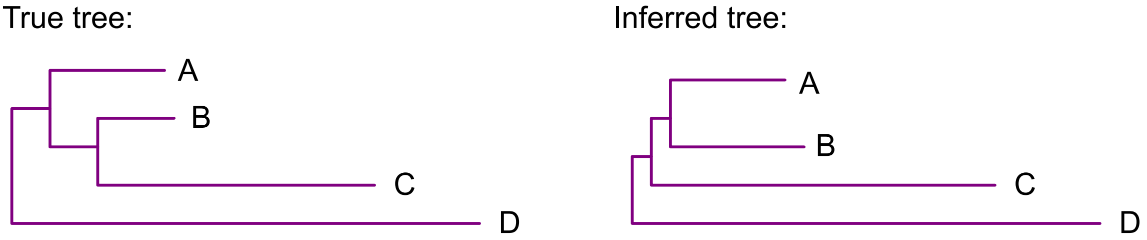
- differences between Bayesian and frequentistic stats
- role of priors in Bayesian stats
- idea of MCMC
- Metropolis-Hastings:

  - how it works
  - why it does not need the integral in the denominator of the posterior
  - why it converges to the target distribution (in our case the posterior)

- MCMCMC
- main idea of Gibbs sampling
- idea of effective sample size (ESS)
- why high ESS do not guarantee that MCMC ran long enough
- good practice of applying MCMC
- possible problems with inappropriate priors

8 Common problems in phylogenetics and consequences for phylogenomics

8.1 Long-branch attraction (LBA)



- appears as systematic error in parsimony based methods ("Felsenstein zone")
- can also occur in NJ, ML and Bayesian methods if
  - sequence evolution model is to simple (mixing models may help)
  - overoptimized alignment

8.2 Alignment

Alignment

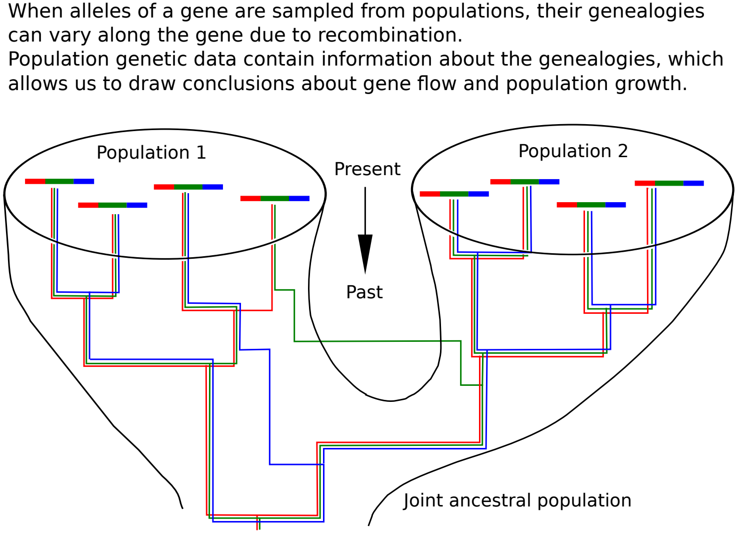
- For distantly related species only genes or even only gene domains may be alignable
- Thorough, model-bases alignment methods like BAli-Phy and StatAlign might be to slow for large datasets
- still somewhat model-based but faster: PRANK (Löytynoja, Goldman, 2008)
- Vialle, Tamuri and Goldmann (2018, *Mol. Biol. Evol.* **35**(7):1783–1797):
  - variants of MAFFT also show good accuracy in reconstructing ancestral states
  - Systematic bias of over-alignment or under-alignment in most methods
  - essentially unbiased: PRANK (but not PRANK+F), PAGAN (Löytynoja, Vilella, Goldman, 2012)

8.3 Gene trees and gene families

Major systematic error:
to assume that all gene trees are equal

8.3.1 Why gene trees differ from each other

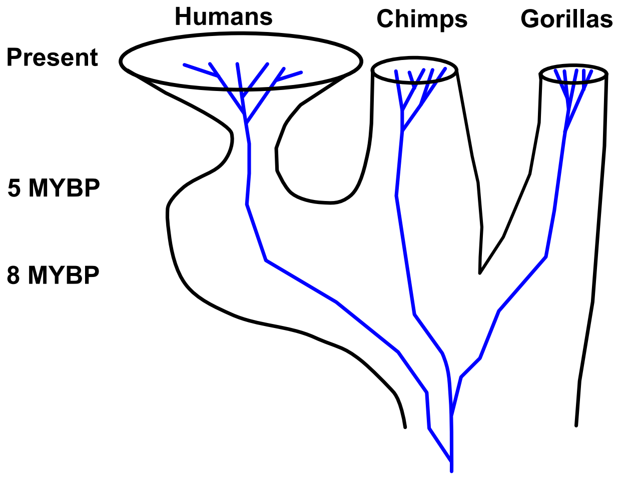
Recombination and Gene flow



When alleles of a gene are sampled from populations, their genealogies can vary along the gene due to recombination.

Population genetic data contain information about the genealogies, which allows us to draw conclusions about gene flow and population growth.

Incomplete Lineage Sorting (ILS)



References

[MV05] E. Mossel, E. Vigoda (2005) Phylogenetic MCMC Algorithms Are Misleading on Mixtures of Trees. *Science* **309**: 2207–2209

point out that, when the data is a mixture of data from two different trees, MCMC convergence can be slow and assign a high posterior probability to a tree that is different from both. See also

References

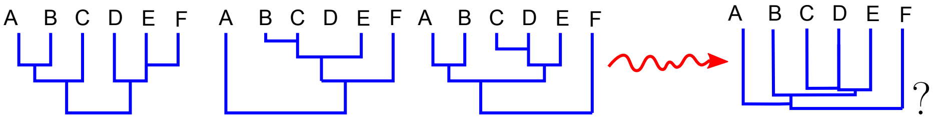
[RLH+06] F. Ronquist, B. Larget, J.P. Huelsenbeck, J.B. Kadane, D. Simon, P. van der Mark (2006) Comment on “Phylogenetic MCMC Algorithms Are Misleading on Mixtures of Trees” *Science* **312**:367a

[MV06] E. Mossel, E. Vigoda (2006) Response to Comment on “Phylogenetic MCMC Algorithms Are Misleading on Mixtures of Trees” *Science* **312**:367b

The problem is not restricted to Bayesian approaches:

References

[SH00] M.H. Schierup, J. Hein (2000) Consequences of recombination on traditional phylogenetic analysis *Genetics* **156**(2): 879–891



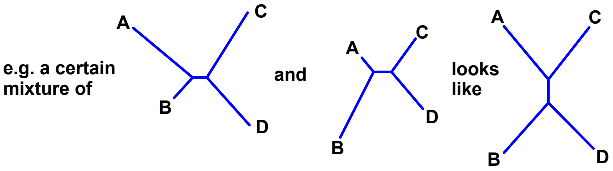
Neglecting recombination leads to more star-shaped phylogenies with short internal and too long external branches (perhaps falsely suggesting fast radiation or, in the case of population genetics, population growth).

The problem is even worse:

References

[RLH+06] F.A. Matsen, M. Steel (2007) Phylogenetic Mixtures on a Single Tree Can Mimic a Tree of Another Topology *Systematic Biology*, 56(5): 767–775

Mixtures of data from the same topology but different branch lengths can lead to the same site pattern frequency spectrum (that is, distribution of alignment columns when neglecting where they appear) as a tree of a different topology.



Possible solution: multi-species coalecent

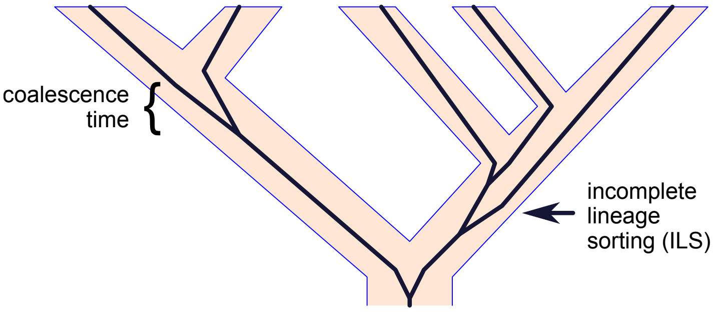
For example:

References

[LP07] L. Liu, D.K. Pearl (2007) Species trees from gene trees: reconstructing Bayesian posterior distributions of a species phylogeny using estimated gene tree distributions *Systematic Biology* **56**(3): 504–514 Software: BEST

[HD09] J. Heled, A.J. Drummond (2010) Bayesian Inference of Species Trees from Multilocus Data *Molecular Biology and Evolution* **27**(3): 570–580 Software: *BEAST (“star beast”)

(open problem: gene flow)

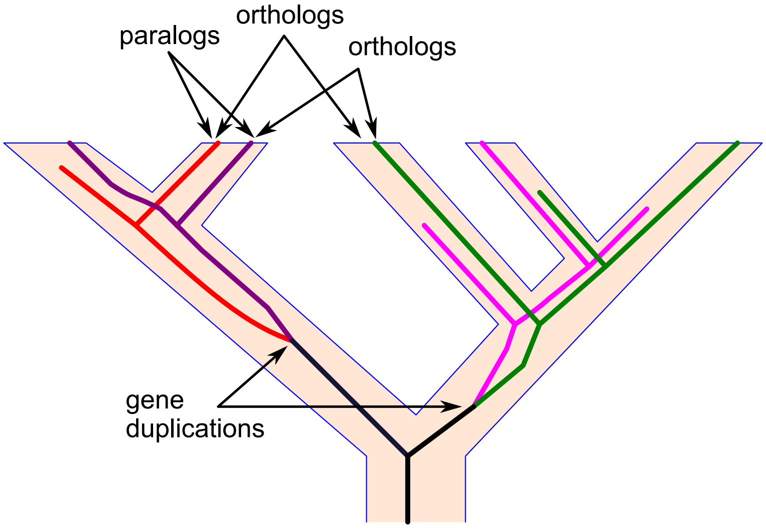


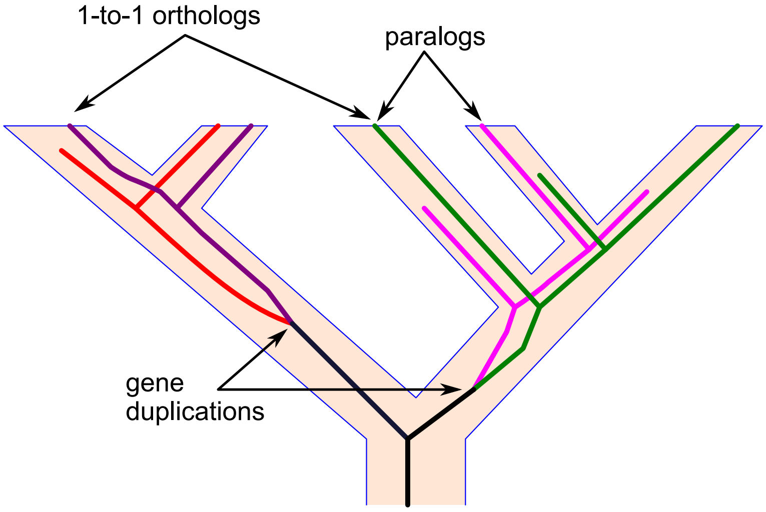
(depends on effective population sizes)

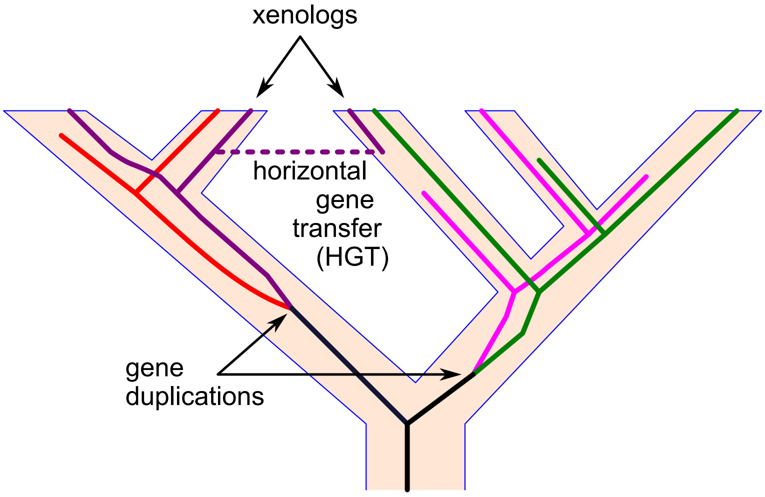
Furthermore:

- substitution models can differ between genes
- for some genes, selection can change evolution process on certain branches

8.3.2 Avoiding paralogues







Methods for gene tree reconstruction for **given** species tree:

References

[ASA+09] Ö. Åkerborg, B. Sennblad, L. Arvestad, and J. Lagergren (2009) Simultaneous Bayesian gene tree reconstruction and reconciliation analysis *PNAS* **106**(14): 5714–5719 https://doi.org/10.1073/pnas.0806251106

[MKSS20] B. Morel, A.M. Kozlov, A. Stamatakis, G.J. Szöllősi (2020) GeneRax: A Tool for Species-Tree-Aware Maximum Likelihood-Based Gene Family Tree Inference under Gene Duplication, Transfer, and Loss *Molecular Biology and Evolution* **37**(9): 2763–2774 https://doi.org/10.1093/molbev/msaa141

References in Kapli et al. (2020) for methods for the joint inference of gene trees and species trees:

References

[STDB15] Szöllősi, G.J., Tannier, E., Daubin, V., Boussau, B. (2015) The inference of gene trees with species treesSyst. Biol. 64, e42–e62, https://academic.oup.com/sysbio/article/64/1/e42/1634124

[BSD+13] Boussau, B., Szöllősi, G.J., Duret, L., Gouy, M., Tannler, E., Daubin, V. (2013) Genome-scale coestimation of species and gene trees *Genome Res.* https://doi.org/10.1101/gr.141978.112

[WBB+08] Wehe, A., Bansal, M. S., Burleigh, J. G., Eulenstein, O. (2008) DupTree: a program for large-scale phylogenetic analyses using gene tree parsimony *Bioinformatics* **24**: 1540–1541

[BBE10] Bansal, M.S., Burleigh, J. G., Eulenstein, O. (2010) Efficient genome-scale phylogenetic analysis under the duplication-loss and deep coalescence cost models. *BMC Bioinformatics* **11**(Suppl. 1): S42.

[CBF13] Chaudhary, R., Burleigh, J.G., Fernández-Baca, D. (2013) Inferring species trees from incongruent multi-copy gene trees using the Robinson-Foulds distance. *Algorithms Mol. Biol.* **28**: 8

[CBBF15] Chaudhary, R., Boussau, B., Burleigh, J. G., Fernández-Baca, D. (2015) Assessing approaches for inferring species trees from multi-copy genes. *Syst. Biol.* **64**: 325–339.

8.4 Consequences for phylogenomics

General remarks and application examples

- large datasets should in principle allow for accurate inference, even with fast neighbor joining
- many data pre-processing steps (alignment, finding orthologues, ...) take time for each locus and thus do not fit well in high-thoughput data analysis pipelines
- for large data sets model bias (e.g. due to long-branch attraction, pooling data from different trees,...) can make support values way too optimistic

Over-optimistic support values

Two sources of estimation error:

- random variation due to limited data
- systematic bias due to simplifying model assumptions

Statistical tools like

- testing (p values)
- posterior probabilities
- bootstrap values

estimate random variation but **rely on model assumptions**.

For very large data sets all error comes from model bias and support values typically indicate 100 % support and statistical tests reject all null hypotheses. This may however be an artifact of model assumptions combined with big data.

Literature

References

[KYT20] P. Kapli, Z. Yang, M.J. Telford (2020) Phylogenetic tree building in the genomic age *Nat. Rev. Genet.* **21**: 428–444 https://doi.org/10.1038/s41576-020-0233-0

[MLM+14] B. Misof, S. Liu, K. Meusemann et al. (2014) Phylogenomics resolves the timing and pattern of insect evolution *Science* **346**(6210): 763–767 https://science.sciencemag.org/content/346/6210/763

[QMP+19] Q. Zhu, U. Mai, W. Pfeiffer et al. (2019) Phylogenomics of 10,575 genomes reveals evolutionary proximity between domains Bacteria and Archaea *Nat. Commun.* **10**: 5477 https://doi.org/10.1038/s41467-019-13443-4

8.5 Possible approaches

- Multi-Species Coalescent (MSC):
  - account for coalescence, ILS, introgression and differences between gene trees
  - Software: e.g. in *BEAST, BPP, IMa2
  - more about coalescence in lectures on computational population genetics
- Full-Bayesian Gibbs sampling of
  - gene-duplication and gene loss
  - HGT
  - coalescence and ILS
  - alignment
  - sequence evolution models
  - and species trees

  seems computationally too demanding for large genomic data sets, but maybe iterative optimization of prelimnary reconstruction?
- massive parallelization of software, also using GPUs
- to assess reliability of tree reconstruction account check robustness against model assumptions, alignment errors, errors with paralogous genes etc…

Some of what you should be able to explain

- long-branch attraction and under what conditions it may happen
- how recombination an gene flow leads to different trees at different loci
- incomplete lineage sorting
- effects of pooling data from different tree topologies in a phylogenetic analysis
- gene duplications, paralogs, ortholog, 1-to-1-orthologs and xenologs
- misleading support values with large data sets and consequences for phylogenomics
- possible solutions

9 Modelling the substitution process on sequences

The methods of stochastic modelling that we discuss here in the context of substitution models apply for many other stochastic models, in biology e.g. for

- biochemical reactions
- ecological or behavioral interactions,
- speciation processes
- population genetics
- ...

9.1 Transition matrix and rate matrix

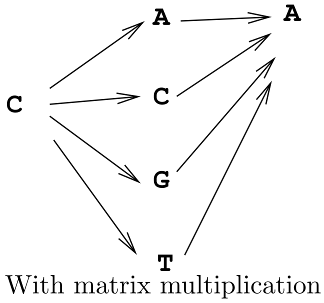
Let $P_{a \to b}(t)$ be the probability that a nucleotide a is a nucleotide b after time (i.e. branch length) t.

$$S(t) := \begin{pmatrix} P_{A\to A}(t) & P_{A\to C}(t) & P_{A\to G}(t) & P_{A\to T}(t) \\ P_{C\to A}(t) & P_{C\to C}(t) & P_{C\to G}(t) & P_{C\to T}(t) \\ P_{G\to A}(t) & P_{G\to C}(t) & P_{G\to G}(t) & P_{G\to T}(t) \\ P_{T\to A}(t) & P_{T\to C}(t) & P_{T\to G}(t) & P_{T\to T}(t) \end{pmatrix}$$

Each row has sum 1.

How can we compute $S(2)$ from $S(1)$?

For example: $P_{C\to A}(2)$



$$\begin{aligned} P_{C\to A}(2) \quad = \quad & P_{C\to A}(1) \cdot P_{A\to A}(1) + \\ & P_{C\to C}(1) \cdot P_{C\to A}(1) + \\ & P_{C\to G}(1) \cdot P_{G\to A}(1) + \\ & P_{C\to T}(1) \cdot P_{T\to A}(1) \end{aligned}$$

With matrix multiplication we can write this as

$$S(2) = S(1) \cdot S(1).$$

More generally:

$$S(t+s) = S(t) \cdot S(s)$$

Matrix multiplication $A \cdot B = C$

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \ddots & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{im} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1j} & \cdots & b_{1k} \\ b_{21} & b_{22} & \cdots & b_{2j} & \cdots & b_{2k} \\ \vdots & \ddots & & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mj} & \cdots & b_{mk} \end{pmatrix}$$

$$= \begin{pmatrix} c_{11} & c_{12} & \cdots & & & \cdots & c_{1k} \\ c_{21} & c_{22} & \cdots & & & \cdots & c_{2k} \\ & & & & c_{ij} & & \\ \vdots & \ddots & & \vdots & & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & & & \cdots & c_{nk} \end{pmatrix}, \qquad c_{ij} = \sum_{h=1}^{m} a_{ih} \cdot b_{hj}$$

Matrix times column vector $A \cdot v$ is column vector

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix} = \begin{pmatrix} a_{11} \cdot v_1 + a_{12} \cdot v_2 + \cdots + a_{1m} \cdot v_m \\ a_{21} \cdot v_1 + a_{22} \cdot v_2 + \cdots + a_{2m} \cdot v_m \\ \vdots \\ a_{n1} \cdot v_1 + a_{n2} \cdot v_2 + \cdots + a_{nm} \cdot v_m \end{pmatrix}$$

Row vector times matrix $v \cdot A$ is row vector

$$\begin{pmatrix} v_1, & v_2, & \cdots, & v_m \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \ddots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mk} \end{pmatrix}$$

$$= \begin{pmatrix} v_1 a_{11} + \cdots + v_m a_{m1}, & v_1 a_{12} + \cdots + v_m a_{m2}, & \cdots, & v_1 a_{1k} + \cdots + v_k a_{mk} \end{pmatrix}$$

Matrix addition $A + B = C$

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \ddots & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1m} + b_{1m} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2m} + b_{2m} \\ \vdots & \ddots & & \vdots \\ a_{n1} + b_{n1} & a_{n2} + b_{n2} & \cdots & a_{nm} + b_{nm} \end{pmatrix}$$

Rules:

$$A + B = B + A, \qquad (A + B) + C = A + (B + C), \qquad (A \cdot B) \cdot C = A \cdot (B \cdot C)$$

$$A \cdot (B + C) = (A \cdot B) + (A \cdot C), \qquad \text{but in general } A \cdot B \neq B \cdot A$$

Matrix multiplied by number r

$$r \cdot \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} = \begin{pmatrix} r \cdot a_{11} & r \cdot a_{12} & \cdots & r \cdot a_{1m} \\ r \cdot a_{21} & r \cdot a_{22} & \cdots & r \cdot a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r \cdot a_{n1} & r \cdot a_{n2} & \cdots & r \cdot a_{nm} \end{pmatrix}$$

Rules:

$$r \cdot (A + B) = r \cdot B + r \cdot A, \qquad r \cdot A = A \cdot r, \qquad (A \cdot r) \cdot B = A \cdot (r \cdot B)$$

Entrywise product $A \circ B$ (also known as Hadamard product)

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \circ \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \ddots & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11} \cdot b_{11} & a_{12} \cdot b_{12} & \cdots & a_{1m} \cdot b_{1m} \\ a_{21} \cdot b_{21} & a_{22} \cdot b_{22} & \cdots & a_{2m} \cdot b_{2m} \\ \vdots & \ddots & & \vdots \\ a_{n1} \cdot b_{n1} & a_{n2} \cdot b_{n2} & \cdots & a_{nm} \cdot b_{nm} \end{pmatrix}$$

In more compact notation:

$$(a_{ij})_{i \le n, j \le m} \circ (b_{ij})_{i \le n, j \le m} = (a_{ij} \cdot b_{ij})_{i \le n, j \le m}$$

Felsenstein's pruning recursion in matrix notation

$$w_{k,p}(x) = \left(\sum_{y \in \{A,C,G,T\}} P_{x \to y}(\ell_i) \cdot w_{i,p}(y) \right) \cdot \left(\sum_{z \in \{A,C,G,T\}} P_{x \to z}(\ell_j) \cdot w_{j,p}(z) \right)$$

$w_{k,p}(x)$ partial likelihood for node k, sequence position p and nucleotide x.

$$W_k = (P(\ell_i) \cdot W_i) \circ (P(\ell_j) \cdot W_j)$$

$$P(\ell_i) = (P_{x \to y}(\ell_i))_{x,y \in \{A,C,G,T\}}, \qquad W_k = \begin{pmatrix} w_{k,1}(A) & w_{k,2}(A) & \dots & w_{k,n}(A) \\ w_{k,1}(C) & w_{k,2}(C) & \dots & w_{k,n}(C) \\ w_{k,1}(G) & w_{k,2}(G) & \dots & w_{k,n}(G) \\ w_{k,1}(T) & w_{k,2}(T) & \dots & w_{k,n}(T) \end{pmatrix}$$

Exercise: Check whether this is really true!

Advantages of matrix notation

- Compact mathematical notation of equation systems
- Matrix algebra
- In programs shorter source code
- In R and python/numpy: Matrix operations more efficient than loops

Example in R with `w[x, p, k]` being the partial likelihood for nucleotide `x`, position `p` and node `k`.
With loops:

```
for(p in 1:n) {
   for(x in c("A", "C", "G", "T")) {
       L <- 0
       for(y in c("A", "C", "G", "T")) {
           L <- L + P[x, y, "i"] *  w[y, p, "i"]
       }
       R <- 0
       for(z in c("A", "C", "G", "T")) {
           R <- R + P[x, z, "j"] *  w[z, p, "j"]
       }
       w[x, p, "k"] <- L*R
   }
}
```

With matrix operation:

```
w[,,"k"] <- (P[,,"i"] %*% w[,,"i"]) * (P[,,"j"] %*% w[,,"j"])
```

In a test run with n=100,000 the code with the matrix operations was more than 500 times faster than the code with loops.

We can use matrix notations for mutation rates. To see how, let $\varepsilon > 0$ be a **very short** time span, such that we get for the Jukes-Cantor model:

$$P_{x \to x}(\varepsilon) \;=\; \frac{1}{4} + \frac{3}{4} \cdot e^{-\lambda\varepsilon} \;\approx\; \frac{1}{4} + \frac{3}{4}(1 - \lambda\varepsilon) \;=\; 1 - \frac{3}{4}\lambda\varepsilon$$

and for $y \neq x$:

$$P_{x \to y}(\varepsilon) \;=\; \frac{1}{4} \cdot \left(1 - e^{-\lambda\varepsilon}\right) \;\approx\; \frac{1}{4}\left(1 - (1 - \lambda\varepsilon)\right) \;=\; \frac{1}{4}\lambda\varepsilon$$

The same in matrix notation

$$\begin{aligned}
S(\varepsilon) &\approx \begin{pmatrix} 1-\frac{3}{4}\lambda\varepsilon & \frac{1}{4}\lambda\varepsilon & \frac{1}{4}\lambda\varepsilon & \frac{1}{4}\lambda\varepsilon \\ \frac{1}{4}\lambda\varepsilon & 1-\frac{3}{4}\lambda\varepsilon & \frac{1}{4}\lambda\varepsilon & \frac{1}{4}\lambda\varepsilon \\ \frac{1}{4}\lambda\varepsilon & \frac{1}{4}\lambda\varepsilon & 1-\frac{3}{4}\lambda\varepsilon & \frac{1}{4}\lambda\varepsilon \\ \frac{1}{4}\lambda\varepsilon & \frac{1}{4}\lambda\varepsilon & \frac{1}{4}\lambda\varepsilon & 1-\frac{3}{4}\lambda\varepsilon \end{pmatrix} \\
&= \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{I} + \varepsilon \cdot \underbrace{\begin{pmatrix} -\frac{3}{4}\lambda & \frac{1}{4}\lambda & \frac{1}{4}\lambda & \frac{1}{4}\lambda \\ \frac{1}{4}\lambda & -\frac{3}{4}\lambda & \frac{1}{4}\lambda & \frac{1}{4}\lambda \\ \frac{1}{4}\lambda & \frac{1}{4}\lambda & -\frac{3}{4}\lambda & \frac{1}{4}\lambda \\ \frac{1}{4}\lambda & \frac{1}{4}\lambda & \frac{1}{4}\lambda & -\frac{3}{4}\lambda \end{pmatrix}}_{R}
\end{aligned}$$

$S(\varepsilon) \approx I + \varepsilon \cdot R$ or, more precisely, $R = \lim_{\varepsilon \to 0} \frac{S(\varepsilon)-I}{\epsilon}$

Interpretation of rate matrix

$$R = \begin{pmatrix} -\frac{3}{4}\lambda & \frac{1}{4}\lambda & \frac{1}{4}\lambda & \frac{1}{4}\lambda \\ \frac{1}{4}\lambda & -\frac{3}{4}\lambda & \frac{1}{4}\lambda & \frac{1}{4}\lambda \\ \frac{1}{4}\lambda & \frac{1}{4}\lambda & -\frac{3}{4}\lambda & \frac{1}{4}\lambda \\ \frac{1}{4}\lambda & \frac{1}{4}\lambda & \frac{1}{4}\lambda & -\frac{3}{4}\lambda \end{pmatrix} = \begin{pmatrix} R_{AA} & R_{AC} & R_{AG} & R_{AT} \\ R_{CA} & R_{CC} & R_{CG} & R_{CT} \\ R_{GA} & R_{GC} & R_{GG} & R_{GT} \\ R_{TA} & R_{TC} & R_{TG} & R_{TT} \end{pmatrix}$$

Assume be start in x. Then R_{xy} is the increase (per time unit) in probability of being in y. (Where for $x = y$ the negative increase means a decrease).

For very small $\varepsilon > 0$, $P_{x \to x}(\varepsilon)$ is close to 1 and there is a matrix R, the so-called **rate matrix** (or Q-matrix), such that $S(\varepsilon) \approx (I + R \cdot \varepsilon)$, where I is the identity matrix (or unit matrix)

$$I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

with the property that $A \cdot I = A$ and $I \cdot B = B$ for all matrices A and B of suitable dimensions.

Thus, we obtain $S(t+\varepsilon) = S(t) \cdot S(\varepsilon) \approx S(t)(I + R\varepsilon) = S(t) + S(t)R\varepsilon$ and

$$\lim_{\varepsilon \to 0} \frac{S(t+\varepsilon) - S(t)}{\varepsilon} = S(t)R \qquad \text{and as } S(0) = I\text{:} \qquad \lim_{\varepsilon \to 0} \frac{S(\varepsilon) - I}{\varepsilon} = R$$

$S(t)R$ is like the derivative of the process, and R the derivative at $t = 0$. Note that the row sums in R are 0. The diagonal entries are negative. All other entries are the rates of the corresponding substitutions.

Rate matrix of the Jukes-Cantor-Model for DNA

$$\begin{pmatrix} -\frac{3}{4}\lambda & \frac{1}{4}\lambda & \frac{1}{4}\lambda & \frac{1}{4}\lambda \\ \frac{1}{4}\lambda & -\frac{3}{4}\lambda & \frac{1}{4}\lambda & \frac{1}{4}\lambda \\ \frac{1}{4}\lambda & \frac{1}{4}\lambda & -\frac{3}{4}\lambda & \frac{1}{4}\lambda \\ \frac{1}{4}\lambda & \frac{1}{4}\lambda & \frac{1}{4}\lambda & -\frac{3}{4}\lambda \end{pmatrix}.$$

The model F81 (Felsenstein, 1981) allows for unequal nucleotide frequencies $(\pi_A, \pi_C, \pi_G, \pi_T)$ and has the rate matrix

$$\begin{pmatrix} -\alpha + \alpha\pi_A & \alpha\pi_C & \alpha\pi_G & \alpha\pi_T \\ \alpha\pi_A & -\alpha + \alpha\pi_C & \alpha\pi_G & \alpha\pi_T \\ \alpha\pi_A & \alpha\pi_C & -\alpha + \alpha\pi_G & \alpha\pi_T \\ \alpha\pi_A & \alpha\pi_C & \alpha\pi_G & -\alpha + \alpha\pi_T \end{pmatrix}.$$

In addition, the HKY model (Hasegawa, Kishino, Yano, 1985) allows that transitions are more probable than transversions by using an additional parameter β. Its rate matrix is

$$R := \begin{pmatrix} -\alpha\pi_G - \beta(\pi_C + \pi_T) & \beta\pi_C & \alpha\pi_G & \beta\pi_T \\ \beta\pi_A & -\alpha\pi_T - \beta(\pi_A + \pi_G) & \beta\pi_G & \alpha\pi_T \\ \alpha\pi_A & \beta\pi_C & -\alpha\pi_A - \beta(\pi_C + \pi_T) & \beta\pi_T \\ \beta\pi_A & \alpha\pi_C & \beta\pi_G & -\alpha\pi_C - \beta(\pi_A + \pi_G) \end{pmatrix}.$$

$(\pi_A, \pi_C, \pi_G, \pi_T)$ is the **stationary Distribution** (or **equilibrium distribution**) for any of these rate matrices. This means $\forall_{x\in\{A,C,G,T\}} : \sum_{y\in\{A,C,G,T\}} \pi_y \cdot P_{y\to x}(t) = \pi_x$, or in matrix notation:

$$(\pi_A, \pi_C, \pi_G, \pi_T) \cdot S(t) = (\pi_A, \pi_C, \pi_G, \pi_T)$$

Equivalently, we can write this with the rate matrix R as

$$(\pi_A, \pi_C, \pi_G, \pi_T) \cdot R = (0, 0, 0, 0),$$

because

$$\begin{aligned} (\pi_A, \pi_C, \pi_G, \pi_T) \cdot R &= (\pi_A, \pi_C, \pi_G, \pi_T) \cdot \lim_{\varepsilon\to 0} \frac{S(\epsilon) - I}{\varepsilon} = \lim_{\varepsilon\to 0} \left((\pi_A, \pi_C, \pi_G, \pi_T) \cdot \frac{S(\epsilon) - I}{\varepsilon} \right) \\ &= \lim_{\varepsilon\to 0} \frac{(\pi_A, \pi_C, \pi_G, \pi_T) \cdot S(\epsilon) - (\pi_A, \pi_C, \pi_G, \pi_T) \cdot I}{\varepsilon} \\ &= \lim_{\varepsilon\to 0} \frac{(\pi_A, \pi_C, \pi_G, \pi_T) - (\pi_A, \pi_C, \pi_G, \pi_T)}{\varepsilon} \\ &= \lim_{\varepsilon\to 0} \frac{(0, 0, 0, 0)}{\varepsilon} = (0, 0, 0, 0). \end{aligned}$$

Some of the things you should be able to explain

- how matrix multiplication accounts for double-hits and back-mutation
- structure and properties of rate matrices
- how the equilibrium property of a distribution can be expressed with a rate matrix or a substitution matrix

9.2 Residence time

If we think of discrete generations and a per generation mutation probability of p, the probability of seeing the first mutation in generation k is $(1-p)^{k-1} \cdot p$.

A random variable X with values in $\{1, 2, \dots\}$ is **geometrically distributed** if $\Pr(X = k) = (1-p)^{k-1} \cdot p$.

Then,

$$\mathbb{E}X = \sum_{k=1}^{\infty} k \cdot (1-p)^{k-1} \cdot p = \frac{1}{p}$$

It is easy to check that this is the only possible value:

$$\begin{aligned} \mathbb{E}X &= \sum_{k=0}^{\infty} (k+1) \cdot (1-p)^k \cdot p \\ &= \sum_{k=1}^{\infty} k \cdot (1-p)^k \cdot p + \sum_{k=0}^{\infty} \cdot (1-p)^k \cdot p = (1-p) \cdot \mathbb{E}X + p \cdot \frac{1}{p} \\ \Rightarrow \qquad & \mathbb{E}X = \frac{1}{p} \end{aligned}$$

The geometric distribution is characterized by the no-memory condition:

$$\Pr(X = k + n \mid X > k) = \Pr(X = n)$$

The continuous analogon is the exponential distribution: A random variable Y with values in $\mathbb{R}_{\geq 0}$ is exponentially distributed with rate λ if

$$\Pr(Y > z) = e^{-\lambda z}.$$

In this case

$$\mathbb{E}Y = \int_0^\infty z\lambda e^{-\lambda z} dz = \frac{1}{\lambda}.$$

The exponential distribution approximates the geometric distriburion if p is small and k is large:

$$(1-p)^k \approx e^{-pk}.$$

In a continuous-time substitution model, the residence time in a state is exponential. For example, if a site has nucleotide `A`, and the HKY model applies, it stays an `A` for a exponentially distributed time with expectation value $1/(\alpha\pi_G + \beta(\pi_C + \pi_T))$. When it then mutates, it becomes a

$$\begin{array}{ll} \texttt{C} \;\; \text{with prob.} & \frac{\beta\pi_C}{\alpha\pi_G+\beta(\pi_C+\pi_T)} \\ \texttt{G} \;\; \text{with prob.} & \frac{\alpha\pi_G}{\alpha\pi_G+\beta(\pi_C+\pi_T)} \\ \texttt{T} \;\; \text{with prob.} & \frac{\beta\pi_T}{\alpha\pi_G+\beta(\pi_C+\pi_T)}. \end{array}$$

Using this approach to simulate stochastic processes is sometimes called Gillepie's algorithm. (It was presented by D. Gillespie in 1976 for simulations of chemical reactions, but in fact this approach was already used long before that.)

Some of the things you should be able to explain:

- In many models the time until the next event (e.g. mutation) is exponentially distributed (or geometrically distributed if time is discrete)
- no-memory condition of exponential and geometrical distribution
- other basic properties of these distributions
- how to simulate processes with exponential waiting times

9.3 Computing $S(t)$ from the rate matrix R

Any linear map $F : \mathbb{R}^n \to \mathbb{R}^m$ can be represented by a matrix

First some basics from linear algebra:

Linear means $\forall \mathbf{v}, \mathbf{w} \in \mathbb{R}^n, a \in \mathbb{R}$:

$$F(\mathbf{v} + \mathbf{w}) = F(\mathbf{v}) + F(\mathbf{w}) \qquad \text{and} \qquad F(a \cdot \mathbf{v}) = a \cdot F(\mathbf{v})$$

Note that $f(x) = 3 \cdot x + b$ is only linear if $b = 0$, e.g. $(3 \cdot 1 + 9) + (3 \cdot 2 + 9) \neq 3 \cdot (1 + 2) + 9$ If for row vectors $\mathbf{v}$, $\mathbf{w}$ and column vectors $\mathbf{x}$, $\mathbf{y}$, real numbers a and a matrix M of suitable dimensions, we have the linearity on both sides:

$$\begin{array}{rclcrcl} \mathbf{v} \cdot M + \mathbf{w} \cdot M &=& (\mathbf{v} + \mathbf{w}) \cdot M & \text{and} & (a \cdot \mathbf{v}) \cdot M &=& a \cdot (\mathbf{v} \cdot M) \\ M \cdot \mathbf{x} + M \cdot \mathbf{y} &=& M \cdot (\mathbf{y} + \mathbf{x}) & \text{and} & M \cdot (a \cdot \mathbf{x}) &=& a \cdot (M \cdot \mathbf{x}) \end{array}$$

A linear map $F : \mathbb{R}^n \to \mathbb{R}^m$ is fully determined by $F(\mathbf{e_1}), F(\mathbf{e_2}), \dots, F(\mathbf{e_n})$, where $\mathbf{e_1}, \dots, \mathbf{e_n}$ are the unit basis vectors of $\mathbb{R}^n$, e.g. as column vectors:

$$\mathbf{e_1} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \mathbf{e_2} = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \dots, \mathbf{e_n} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

If

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \quad = \quad v_1 \cdot \mathbf{e_1} + v_2 \cdot \mathbf{e_2} + \cdots + v_n \cdot \mathbf{e_n},$$

then $F(\mathbf{v}) = F(v_1 \cdot \mathbf{e_1} + v_2 \cdot \mathbf{e_2} + \cdots + v_n \cdot \mathbf{e_n}) = v_1 \cdot F(\mathbf{e_1}) + v_2 \cdot F(\mathbf{e_2}) + \cdots + v_n \cdot F(\mathbf{e_n})$.

In the matrix presentation of $F : \mathbb{R}^n \to \mathbb{R}^m$ as $F(\mathbf{v}) = M \cdot \mathbf{v}$, the columns of the matrix are $F(\mathbf{e_1}), F(\mathbf{e_2}), \dots, F(\mathbf{e_n})$.

The analogous statement applies for the rows of the matrix if the row vector notation is used.

Now back to transition matrices of substitution models:

If $S(1)$ is known, you can compute $S(n)$ by

$$S(n) = S(1)^n.$$

To do this efficiently, diagonalize $S(1)$. This means, find a matrix U and a diagonal matrix D (this means $D_{ij} = 0$ if $i \neq j$), such that

$$S(1) = U \cdot D \cdot U^{-1}.$$

For

$$D = \begin{pmatrix} \mu_1 & 0 & \dots & 0 \\ 0 & \mu_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mu_m \end{pmatrix}$$

we can use

$$D^n = \begin{pmatrix} \mu_1^n & 0 & \dots & 0 \\ 0 & \mu_2^n & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mu_m^n \end{pmatrix}$$

and

$$\begin{aligned} S(1)^n &= \left(U \cdot D \cdot U^{-1}\right)^n \\ &= U \cdot D \cdot U^{-1} \cdot U \cdot D \cdot U^{-1} \cdots U \cdot D \cdot U^{-1} \cdot U \cdot D \cdot U^{-1} \\ &= UD \cdot I \cdot D \cdot I \cdots D \cdot U^{-1} \; = \; UD^nU^{-1} \end{aligned}$$

But how to find a matrix U, such that

$$S(1) = U \cdot D \cdot U^{-1} \qquad \text{holds?}$$

The inverse U^{-1} of the matrix U is defined by $U^{-1} \cdot U = I = U \cdot U^{-1}$.

In this case, the diagonal entries $D_{ii} = \lambda_i$ of D are the **eigenvalues** of $S(1)$, the columns of U are corresponding **right eigenvectors** and the rows of U^{-1} (with entries u'_{ij}) are **left eigenvectors**, that is:

$$S(1) \cdot \begin{pmatrix} u_{1i} \\ u_{2i} \\ u_{3i} \\ u_{4i} \end{pmatrix} = \begin{pmatrix} u_{1i} \\ u_{2i} \\ u_{3i} \\ u_{4i} \end{pmatrix} \cdot \lambda_i, \qquad (u'_{i1}, \dots, u'_{i4}) \cdot S(1) = \lambda_i \cdot (u'_{i1}, \dots, u'_{i4})$$

Note that the eigenvectors in U and U^{-1} have to be scaled appropriately, to make sure that $u'_{.i} \cdot u_{i.} = 1$. Further, if $\lambda_k \neq \lambda_j$, then $u'_{.k} \cdot u_{j.} = 0$, but if $\lambda_k = \lambda_j$ for some $k \neq j$, we have some choice in the eigenspace and must make sure that $u'_{.k} \cdot u_{j.} = 0$.

(General explanations of eigenvectors and eigenvalues on whitebord and with R file.)

Why $\lambda_k \neq \lambda_j$ implies $u'_{.k} \cdot u_{j.} = 0$:

$$\lambda_k \cdot u'_{.k} \cdot u_{j.} = u'_{.k} \cdot S(1) \cdot u_{j.} = u'_{.k} \cdot \lambda_j \cdot u_{j.} = \lambda_j \cdot u'_{.k} \cdot u_{j.}$$

Note: $u'_{.k}$ and $u^t_{j.}$ (or $(u'_{.k})^t$ and $u_{j.}$) are orthogonal, as $u'_{.k} \cdot u_{j.}$ is their scalar product.

If $u'_{.j} \cdot u_{j.} = a \neq 1$, divide one of the two vectors by a or, alternatively, each by $\sqrt{a}$:

$$\frac{u'_{.j}}{\sqrt{a}} \cdot \frac{u_{j.}}{\sqrt{a}} = \frac{1}{\sqrt{a} \cdot \sqrt{a}} \cdot u'_{.j} \cdot u_{j.} = \frac{1}{a} \cdot a = 1$$

Note that the scaled eigenvectors $\frac{u'_{.j}}{\sqrt{a}}$ and $\frac{u_{.j}}{\sqrt{a}}$ are still eigenvectors with eigenvalue λ_j, e.g. for $\frac{u'_{.j}}{\sqrt{a}}$:

$$\frac{u'_{.j}}{\sqrt{a}} \cdot S(1) = \frac{1}{\sqrt{a}} \cdot u'_{.j} \cdot S(1) = \frac{1}{\sqrt{a}} \cdot \lambda_j \cdot u'_{.j} = \lambda_j \cdot \frac{u'_{.j}}{\sqrt{a}}$$

Calculating right eigenvectors in R

```
> (M <- matrix(c(0.8,-0.8,-0.5,1.2),ncol=2))
     [,1] [,2]
[1,]  0.8 -0.5
[2,] -0.8  1.2
> eigen(M)
$values
[1] 1.663325 0.336675

$vectors
           [,1]       [,2]
[1,]  0.5011716 -0.7334959
[2,] -0.8653479 -0.6796939
```

$\begin{pmatrix} 0.501 \\ -0.865 \end{pmatrix}$ is the right eigenvector of M with eigenvalue 1.663 and $\begin{pmatrix} -0.733 \\ -0.679 \end{pmatrix}$ is the right eigenvector of M with eigenvalue 0.336.

E.g.:

$$\begin{pmatrix} 0.8 & -0.5 \\ -0.8 & 1.2 \end{pmatrix} \cdot \begin{pmatrix} 0.501 \\ -0.865 \end{pmatrix} = 1.663 \cdot \begin{pmatrix} 0.501 \\ -0.865 \end{pmatrix} = \begin{pmatrix} 0.834 \\ -1.439 \end{pmatrix}$$

To calculate left eigenvectors with R, transepose the matrix with `t(M)` and calculate the right eigenvectors of the transposed matrix (and transpose them). Exercise: calculate the left eigenvectors for Matrix M, first without R, then with R.

Equilibrium distribution as eigenvector

Note that the equilibrium condition

$$(\pi_A, \pi_C, \pi_G, \pi_T) \cdot S(t) = (\pi_A, \pi_C, \pi_G, \pi_T)$$

means that the equilibrium distribution forms a left eigenvector with eigenvalue 1 for the transition matrix $S(t)$.

Thus, the equilibrium distribution can be found by calculating a left eigenvector for eigenvalue 1 and by scaling the eigenvector such that its entries sum up to 1.

Stochastic matrices, that is, matrices with non-negative entries and rows that add up to 1 always have 1 as their largest eigenvalue.

The situation is similar in the continuous case. For $t \in [0, \infty)$ we get $S(t) = U \cdot T^t \cdot U^{-1}$ with

$$T^t = \begin{pmatrix} e^{\lambda_1 t} & 0 & \dots & 0 \\ 0 & e^{\lambda_2 t} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & e^{\lambda_m t} \end{pmatrix},$$

where $\lambda_1, \lambda_2, \dots, \lambda_m$ are the eigenvalues of R (with $m = 4$ for nucleotides and $m = 20$ for amino acids).

Explanation: For very small $\varepsilon > 0$ we have

$$S(t) = S(\varepsilon)^{t/\varepsilon} \approx (I + R \cdot \varepsilon)^{t/\varepsilon} = U_\varepsilon \cdot D_\varepsilon^{t/\varepsilon} \cdot U_\varepsilon^{-1},$$

where D_ε is a diagonal matrix of the eigenvalues μ_i of $I + \varepsilon \cdot R$.

It is common to write this as $S(t) = e^{tR}$ and call it "Matrix exponential".

For the right eigenvectors $\mathbf{v}_i$ we have

$$(I + \varepsilon \cdot R) \cdot \mathbf{v}_i = \mu_i \cdot \mathbf{v}_i$$

and thus

$$R \cdot \mathbf{v}_i = \frac{\mu_i - 1}{\varepsilon} \cdot \mathbf{v}_i.$$

Therefore,

$$\lambda_i := \frac{\mu_i - 1}{\varepsilon}$$

is an eigenvalue of R (if $\mu_i \neq 1$) and we can write the diagonal entries of $D_\varepsilon^{t/\varepsilon}$ as

$$(1 + \varepsilon \lambda_i)^{t/\varepsilon},$$

which converges to $e^{\lambda_i t}$ for $\varepsilon \to 0$.

Calculation above also shows that columns of U are not only eigenvectors of $I + \varepsilon R$ but also of R. Note that transition matrices always have $\mu_1 = 1$ as greatest eigenvalue, which corresponds to the eigenvalue $\lambda_1 = \frac{\mu_1 - 1}{\varepsilon} = 0$ of the rate matrix, for which the diagonal entry in T^t is $e^{\lambda_1 t} = e^0 = 1$).

Further, note that the equilibrium distribution, e.g. $(\pi_A, \pi_C, \pi_G, \pi_T)$ is a left eigenvector of the rate matrix R for this eigenvalue 0, that is

$$(\pi_A, \pi_C, \pi_G, \pi_T) \cdot R = 0 \cdot (\pi_A, \pi_C, \pi_G, \pi_T) = (0, 0, 0, 0)$$

Efficient implementations functions for computing eigenvalues and eigenvectors are available for most programming languages and we can use them to calculate matrix exponentials. However, this is sometimes numerically unstable.

One alternative is to use the following alternative definition of the matrix exponential:

$$e^{tR} = \sum_{n=0}^{\infty} \frac{(tR)^n}{n!}$$

which can be made more stable by chosing $\beta > \max\{\lambda_1, \dots, \lambda_m\}$ and then using the variant

$$e^{tR} = e^{-\beta t} \cdot \sum_{n=0}^{\infty} \frac{(\beta t)^n \cdot (I + R/\beta)^n}{n!}.$$

Another approach is to use the limit

$$e^{tR} = \lim_{n\to\infty} \left(I + \frac{t}{n}R\right)^n$$

or its variant

$$e^{tR} = \lim_{n\to\infty} \left(\left(I - \frac{t}{n}R\right)^{-1}\right)^n$$

for the approximation

$$e^{tR} \approx \left(I + \frac{t}{n}R\right)^n$$

or

$$e^{tR} \approx \left(\left(I - \frac{t}{n}R\right)^{-1}\right)^n$$

with a large value of n.

Some of the things you should now be able to explain:

- how powers of matrices can be used to calculate transition probabilities
- how to calculate the powers of a diagonal matrices
- what are eigenvectors and eigenvalues and how do they help to
  - transform a matrix into a diagonal matrix and calculate a matrix power
  - find an equilibrium distribution for a transition matrix
- how matrix exponentials can be used to express transition matrices for a rate matrix
- one or two ways of calculating matrix exponentials

9.4 A model for transition probabilities in closed form

The F84 model (Felsenstein, 1984) is similar to the HKY model but allows the computation of transition probabilities without numerics by using similar ideas as in the Jukes-Cantor model.

F84 model: Pepper crosses and bullets into the ancestral lineages of the all positions that make them (partly) forget their former type.

crosses come rate λ. The new type is drawn according to $(\pi_A, \pi_C, \pi_G, \pi_T)$.

bullets come at rate μ. The lineage only remembers if it was a purine or a pyrimidine. If it was a purine, the new type is `A` or `G` with probability $\frac{\pi_A}{\pi_A+\pi_G}$ or $\frac{\pi_G}{\pi_A+\pi_G}$.If it was a pyrimidine, the new type is `C` or `T` with probability $\frac{\pi_C}{\pi_C+\pi_T}$ or $\frac{\pi_T}{\pi_C+\pi_T}$.

A transversion needs at least one cross. If we condition on having at least one cross but not on the nucleotide that was selected at the cross, then the last bullet or cross before time t draws a nucleotide according to the distribution $(\pi_A, \pi_C, \pi_G, \pi_T)$. Thus, we get, for example:

$$P_{A\to C}(t) = \left(1 - e^{-\lambda t}\right) \cdot \pi_C$$

A transition needs either at least one cross or no cross and at least one bullet. We get, for example:

$$P_{A\to G}(t) = \left(1 - e^{-\lambda t}\right) \cdot \pi_G + e^{-\lambda t} \left(1 - e^{-\mu t}\right) \cdot \pi_G / (\pi_A + \pi_G)$$

Even if we do not need it for computing the transition probabilities, we can write down the F84 rate matrix:

$$\begin{pmatrix} -\lambda(1-\pi_A) - \frac{\mu\pi_G}{\pi_A+\pi_G} & \lambda\pi_C & \lambda\pi_G + \frac{\mu\pi_G}{\pi_A+\pi_G} & \lambda\pi_T \\ \lambda\pi_A & -\lambda(1-\pi_C) - \frac{\mu\pi_T}{\pi_C+\pi_T} & \lambda\pi_G & \lambda\pi_T + \frac{\mu\pi_T}{\pi_C+\pi_T} \\ \lambda\pi_A + \frac{\mu\pi_A}{\pi_A+\pi_G} & \lambda\pi_C & -\lambda(1-\pi_G) - \frac{\mu\pi_A}{\pi_A+\pi_G} & \lambda\pi_T \\ \lambda\pi_A & \lambda\pi_C + \frac{\mu\pi_C}{\pi_C+\pi_T} & \lambda\pi_G & -\lambda(1-\pi_T) - \frac{\mu\pi_C}{\pi_C+\pi_T} \end{pmatrix}$$

9.5 Position-dependent mutation rates

Model for site-dependent rates

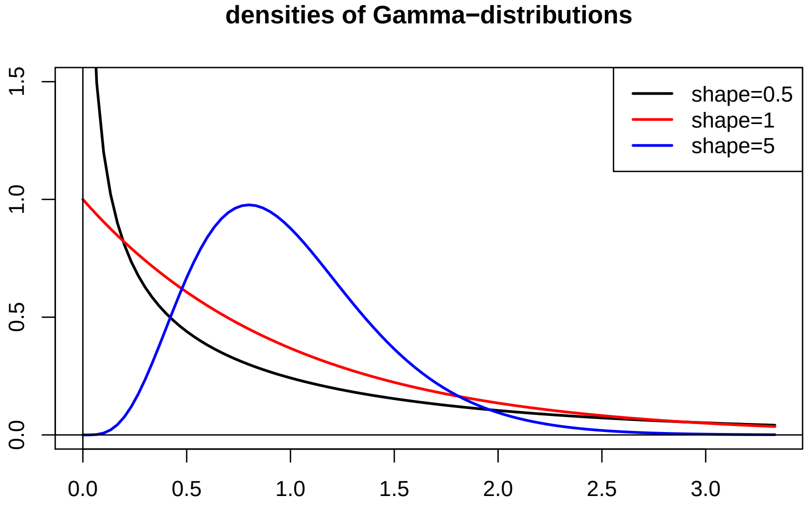
There is one rate matrix Q and for each site i there is a coefficient r_i, such that

$$R_i = r_i \cdot Q$$

is the substitution rate matrix for site i.

Estimating n additional parameters $r_1, \ldots, r_n$ is not feasible.

Instead estimate one meta-parameter α and assume Γ-prior with shape parameter α for all r_i.



The Γ distribution has another parameter, the scale parameter β. The expectation value of the Γ distribution is $\alpha \cdot \beta$.

We always assume $\beta = 1/\alpha$, such that

$$\mathbb{E}r_i = 1 \text{ and } \mathbb{E}Q = \mathbb{E}R_i$$

Density of the Γ-distribution:

$$g_{\alpha,\beta}(x) := \frac{x^{\alpha-1} \cdot e^{-x/\beta}}{\beta^\alpha \cdot \Gamma(\alpha)},$$

with $\Gamma(a) = \int_0^\infty x^{a-1} \cdot e^{-x} dx$

We use

$$g_\alpha(x) := g_{\alpha,1/\alpha}$$

To contribution of data column D_i to the Likelihood of a tree T is then

$$L_{D_i}(T) = \Pr_T(D_i) = \int_0^\infty \Pr(D_i \mid r_i = x) \cdot g_\alpha(x)\ dx.$$

For each fixed $r_i = x$ we can efficiently compute $\Pr(D_i \mid r_i = x)$ with the Felsenstein pruning algorithm. But not for all x from 0 to ∞.

Idea: compute $\Pr(D_i \mid r_i = x_j)$ for some x_j and approximate

$$\Pr(D_i) = \int_0^\infty \Pr(D_i \mid r_i = x) \cdot g_\alpha(x)\ dx \approx \sum_{j=1}^{k} w_j \cdot \Pr(D_i \mid r_i = x_j).$$

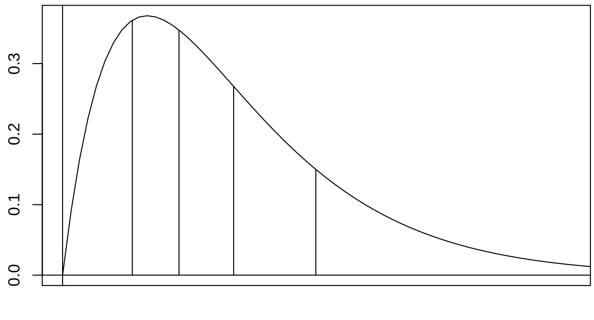
What are good choices for $w_1, \ldots, w_k$ and $x_1, \ldots, x_k$?

Method of Yang (1994)

Divide $[0, \infty]$ into k sections $[a, b]$ of equal probability

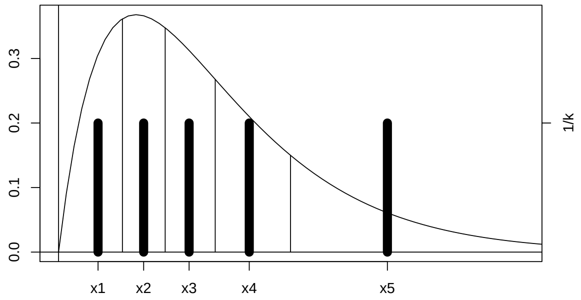
$$\int_a^b g_\alpha(x)dx = 1/k,$$

e.g. for $k = 5$:



Then, x_j is the expectation value of the Γ distribution conditioned of being in the jth section, i.e. the center of gravity of the area under the density.

All w_j are $1/k$.



Alternatitve to or (extension of) the Γ-model: A proprotion p of the sites in invariate ("+I").

Alexis Stamatakis' CAT approximation

The "CAT model" provided by RAxML can be seen as an approximation to the discretized Γ-model.

- sites belong to a few different categories
- each category has its own rate acceleration factor that must be estimated
- ML estimate for each site to which category it belongs
- instead of marginalizing over all categories only use ML categories for likelihood computation
- Assignments of positions to categories are part of the parameter space and must be updated during ML optimization
- recommended if more than 50 taxa

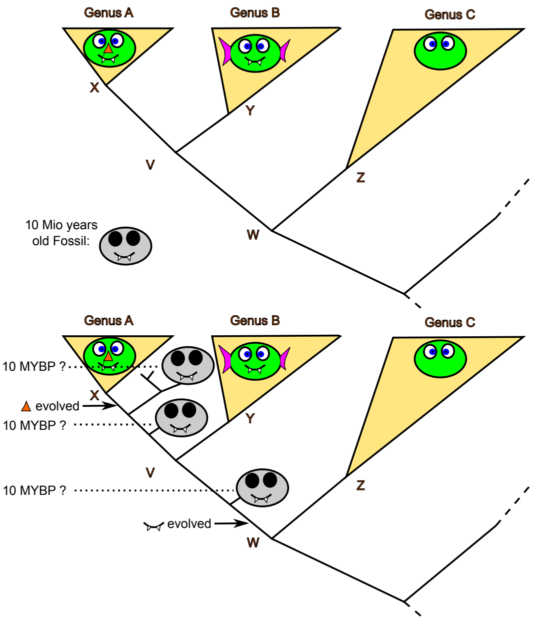
Note: There is a completely different substitution model also called CAT in Lartillot and Philippe's program PhyloBayes.

9.6 Time-calibration with fossils and relaxed molecular clocks

Drawing conclusions from fossils

Assume a 10 Mio year old fossil has features that both genera A **and** B have, but no feature that only A or only B has.

Further assume: such features would not evolve twice and not get lost.



Which conclusion can we draw?

- Must node V be 10 Mio years old? No!
- Must node W be older than 10 Mio years? Yes!
- Must node V be younger than 10 Mio years? No!
- Must node X (and Y) be younger than 10 Mio years? No!

We can only conlude that the **parent node** of the MRCA of A and B is older than the fossil.

Can we use fossil record to limit the age of a node? Not clear (to me)!

Maybe from the absence of fossils? But what if species lived where conditions did not lead to fossilation?

uncorrelated log-normal (ULN), uncorrelated exponential (UEX)

In ULN, UEX and DM each edge in the tree gets a rate randomly drawn from the distribution and uncorrelated to the neighboring branches.[2ex]

e.g. in the case of ULN, the logarithm of the rate on the current branch follows a normal distribution with mean $\log(\overline{r}) + \sigma^2/2$ and variance σ^2, which leads to an expectation value of $\overline{r}$ for the rate.

Compund Poisson Process (CPP)

- Rate change points are peppered randomly into the tree at rate λ.
- At each change point, the current rate is multiplied with r, which is drawn from a Γ-distribution.
- Problem: If $\mathbb{E}r = 1$ or $\mathbb{E}[\log r] < 0$, rates converge to 0, and if $\mathbb{E}[\log r] > 0$, rates converge to ∞ for long branches.
- Solution: Γ-parameters must lead to $\mathbb{E}[\log r] = 0$, and a prior on λ must limit the number of change-points.

Some of the things you should be able to explain:

- why we do not estimate mutation rates for each site
- how we can avoid this by estimating a meta-parameter
- properties of the Γ (Gamma) distribution and why it is appropriate to model rate heterogeneity (and many other things)
- why and how we need to discretize the Γ distribution
- how the runtime of your analysis depends on the number of “Gamma categories” and why

- What is the difference between the Gamma model and the CAT model in RAxML and when should you use which
- how to use fossil information in phylogenetic analyses
- some relaxed molecular-clock models, like ULN and UEX

10 Quantitative Characters and Independent Contrasts

10.1 Brownian motions along the branches of the tree

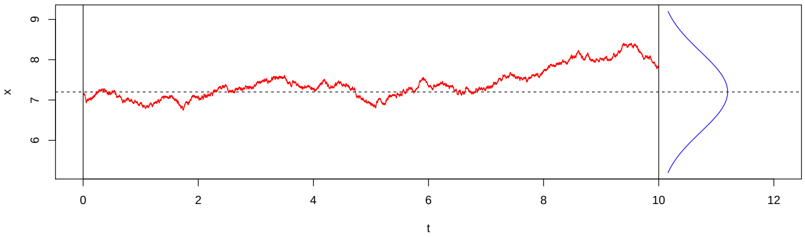
Type of questions to be answered

Quantitative traits like number of genes, mutation rates, or morphological traits like weight or body length differ for different species.

- Do two traits evolve in a correlated way or are their values just correlated because they evolved independently along the same tree?
- Is a trait significantly different for a certain group of species such that adaptation must have played a role?
- Can we use morphological traits for phylogeny reconstruction?

Model for the neutral evolution of a quantitative trait along the branches of a phylogenetic tree.

- Independent on different branches
- After an appropriate rescaling it changes randomly like a Brownian motion.

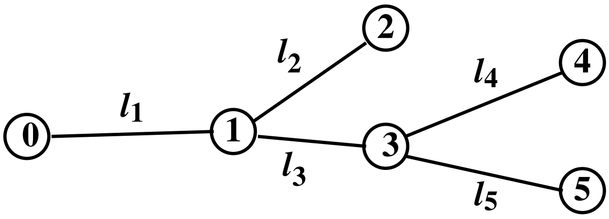


This is a Markov process with

$$X_{s+t} - X_s \sim \mathcal{N}(0, t),$$

where $\mathcal{N}(0, t)$ means normal distribution with mean 0 and variance $\sigma^2 = t$.

Example: Brownian motion starts in node 0 of this tree with a non-random value x_0:



Then, $\mathbb{E}X_i = x_0$ for all i, and the variance of any node is its distance to the root, e.g. $\mathrm{var}(X_5) = l_1 + l_3 + l_5$.

$$\begin{aligned}
\mathrm{cov}(X_5, X_4) &= \mathrm{cov}(X_5 - X_3 + X_3, X_4 - X_3 + X_3) \\
&= \mathrm{cov}(X_5 - X_3, X_4 - X_3) + \mathrm{cov}(X_5 - X_3, X_3) \; + \; \mathrm{cov}(X_3, X_4 - X_3) + \mathrm{cov}(X_3, X_3) \\
&= \mathrm{var}(X_3) \; = \; \ell_1 + \ell_3
\end{aligned}$$

In general: The covariance of the values X_k and X_ℓ at the nodes k and ℓ is the variance $\mathrm{var}(X_h)$ of the value at their most recent common ancestor h.

Let v_i be the parent node of node i, then the values $\left(\frac{X_i - X_{v_i}}{\sqrt{\ell_i}}\right)_{i=1,\ldots,n}$ are stochastically independent and standard-normally distributed. Together they are a standard-normally distributed random vector.

Moreover, the map

$$Y := \begin{pmatrix} \frac{X_1 - X_{v_1}}{\sqrt{\ell_1}} \\ \vdots \\ \frac{X_n - X_{v_n}}{\sqrt{\ell_n}} \end{pmatrix} \mapsto \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix} = X$$

is an affine transformation, i.e. can be represented as $Y \mapsto w + MY = X$ with appropriate vector w and matrix M. This implies that X is also normally distributed, and its distribution is determined by its expected value and its covariance matrix.

10.2 Excursus: Multidimensional Normal Distribution

- An d-dimensional random vector is a vector of d random elements
- The expectation of a random vector $X = (X_1, X_2, \ldots, X_d)^T$ is the vector of the expectations:

$$\mathbb{E}X = \mathbb{E}\begin{pmatrix} X_1 \\ \vdots \\ X_d \end{pmatrix} = \begin{pmatrix} \mathbb{E}X_1 \\ \vdots \\ \mathbb{E}X_d \end{pmatrix}$$

- The expectation of a random matrix $M = (M_{ij})_{i=1..n,j=1..d}$ is the matrix of the expectations:

$$\mathbb{E}\begin{pmatrix} M_{11} & M_{12} & \cdots & M_{1d} \\ \vdots & & \ddots & \vdots \\ M_{n1} & M_{n2} & \cdots & M_{nd} \end{pmatrix} = \begin{pmatrix} \mathbb{E}M_{11} & \mathbb{E}M_{12} & \cdots & \mathbb{E}M_{1d} \\ \vdots & \ddots & & \vdots \\ \mathbb{E}M_{n1} & \mathbb{E}M_{n2} & \cdots & \mathbb{E}M_{nd} \end{pmatrix}$$

- reminder: The variance of a univariate random variable X is $\mathrm{Var}(X) = \mathbb{E}\left[(X - \mathbb{E}X)^2\right] = \mathbb{E}\left[X^2\right] - (\mathbb{E}X)^2$.
- The analog in the multivariate case is the so called *covariance matrix* (or dispersion matrix or variance-covariance matrix). The covariance matrix $\mathrm{Var}(X) = \Sigma$ of $X = (X_1, \ldots, X_d)^T$ is

$$\begin{aligned} \Sigma &= \begin{pmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) & \cdots & \mathrm{Cov}(X_1, X_d) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) & \cdots & \mathrm{Cov}(X_2, X_d) \\ \vdots & & \ddots & \vdots \\ \mathrm{Cov}(X_d, X_1) & \mathrm{Cov}(X_d, X_2) & \cdots & \mathrm{Cov}(X_d, X_d) \end{pmatrix} \\ &= \mathbb{E}\left[\begin{pmatrix} X_1 - \mathbb{E}X_1 \\ \vdots \\ X_d - \mathbb{E}X_d \end{pmatrix} \cdot \Big(X_1 - \mathbb{E}X_1, \cdots, X_d - \mathbb{E}X_d\Big)\right] \\ &= \mathbb{E}\Big[(X - \mathbb{E}X) \cdot (X - \mathbb{E}X)^T\Big] \\ &= \mathbb{E}\Big[X \cdot X^T\Big] - \mathbb{E}X \cdot (\mathbb{E}X)^T \end{aligned}$$

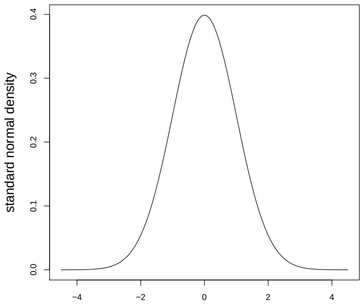
- Linearity of the expectation is analogous to the univarite case: Let $X = (X_1, \ldots, X_d)$ be a random vector and $C = (C_{ij})_{i=1..n,j=1..d}$ be a deterministic matrix. Then

$$\mathbb{E}(C \cdot X) = C \cdot \mathbb{E}(X)$$

- If $Y := X - \mathbb{E}(X)$, then

$$\begin{aligned} \mathrm{Var}(C \cdot X) &= \mathrm{Var}(C \cdot Y) \\ &= \mathbb{E}\left[C \cdot Y \cdot (C \cdot Y)^T\right] \\ &= \mathbb{E}\left[C \cdot Y \cdot Y^T \cdot C^T\right] \\ &= C \cdot \mathbb{E}\left[Y \cdot Y^T\right] \cdot C^T \\ &= C \cdot \mathrm{Var}(Y) \cdot C^T \\ &= C \cdot \mathrm{Var}(X) \cdot C^T \end{aligned}$$

- Reminder: Univariate normal distribution $\mathcal{N}(\mu, \sigma^2)$ with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 \in (0, \infty)$ has the density
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

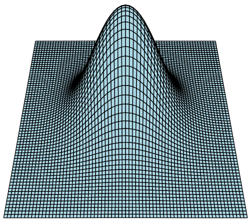


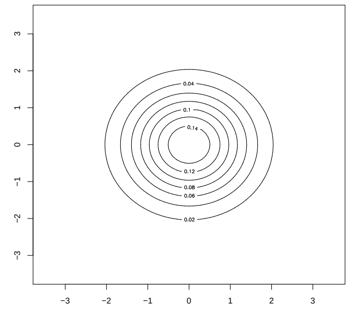
Remember: $\Pr(\mu - \sigma < X < \mu + \sigma) = 0.68$ and $\Pr(\mu - 1.96\sigma < X < \mu + 1.96\sigma) = 0.95$

- The density of the d-dimensional normal distribution with mean $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$ is analogous:
$$f(x) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left(-\frac{(x-\mu)^T \Sigma^{-1}(x-\mu)}{2}\right)$$
for $x \in \mathbb{R}^d$ where $\det(\Sigma)$ is the determinant of Σ, and Σ^{-1} is the inverse matrix. We write $\mathcal{N}_d(\mu, \Sigma)$ for this distribution.
- The *standard multivariate normal distribution* has mean $\mu = 0$ and the identity matrix $\Sigma = \mathbb{I}$ as covariance matrix.

Plots for $d = 2$

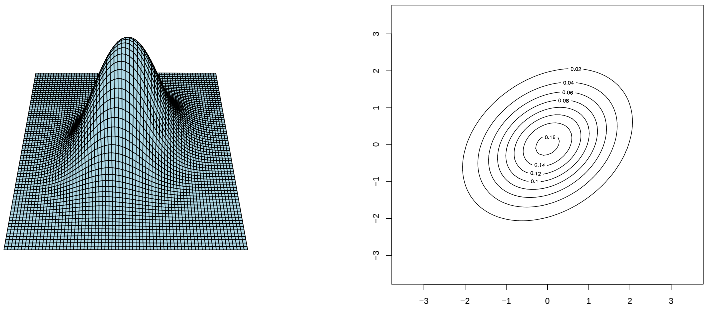
Correlation 0.0: $\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\mathrm{Var}(X_1) = 1 = \mathrm{Var}(X_2)$, $\mathrm{Cov}(X_1, X_2) = 0.0$





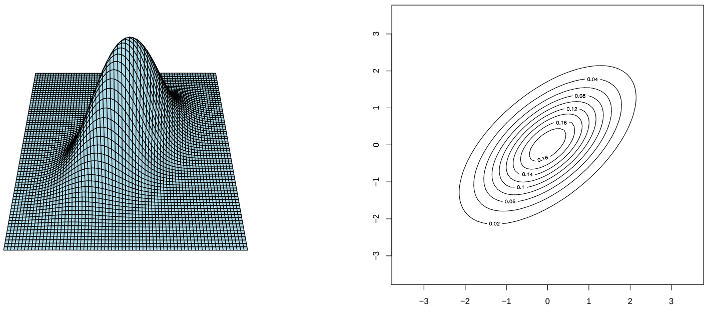
Plots for $d = 2$

Correlation 0.3: $\Sigma = \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}$, $\mathrm{Var}(X_1) = 1 = \mathrm{Var}(X_2)$, $\mathrm{Cov}(X_1, X_2) = 0.3$



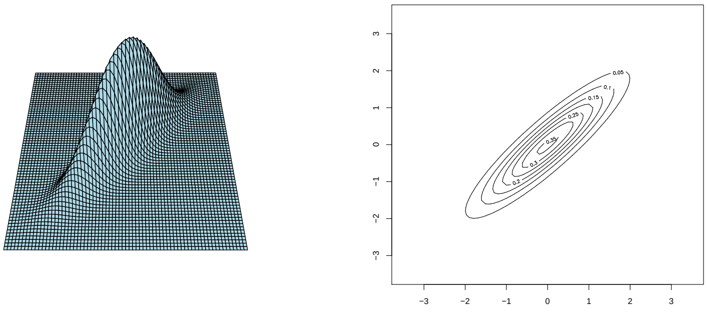
Plots for $d = 2$

Correlation 0.6: $\Sigma = \begin{pmatrix} 1 & 0.6 \\ 0.6 & 1 \end{pmatrix}$, $\mathrm{Var}(X_1) = 1 = \mathrm{Var}(X_2)$, $\mathrm{Cov}(X_1, X_2) = 0.6$



Plots for $d = 2$

Correlation 0.9: $\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$, $\mathrm{Var}(X_1) = 1 = \mathrm{Var}(X_2)$, $\mathrm{Cov}(X_1, X_2) = 0.9$



Properties of a $\mathcal{N}_d\left(\mu, \Sigma\right)$-distributed random vector X:

- Linear combinations are univariate normal distributed: $\langle c, X\rangle \sim \mathcal{N}\left(\langle c, \mu\rangle, c\Sigma c^T\right)$

Where $\langle .,. \rangle$ is the scalar product: $\langle v, w \rangle = \sum_{i=1}^{n} v_i \cdot w_i = (v_1, v_2, \ldots, v_n) \cdot \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \|v\| \cdot \|w\| \cdot \cos(\angle_{v,w})$

- X_i and X_j are independent $\iff$ $\mathrm{Cov}(X_i, X_j) = 0$
- The standardized normal distribution is standard normal distributed

$$\Sigma^{-\frac{1}{2}} \cdot (X - \mu) \sim \mathcal{N}_d(0, \mathbb{I})$$

where $M = \Sigma^{-\frac{1}{2}}$ is a matrix such that $M^T \cdot M \cdot \Sigma = \mathbb{I}$.

- The square of the standardized normal distribution is chi-squared distributed with d degrees of freedom:

$$(X - \mu)^T \, \Sigma^{-1} \, (X - \mu) \sim \chi^2_d.$$

- If $Y_1, Y_2, \ldots, Y_d$ are independent and standard normal distributed, then $(Y_1, \ldots, Y_d) \sim \mathcal{N}(0, \mathbb{I})$.
- If $M \in \mathbb{R}^{p \times d}$ is a non-random matrix, then $M \cdot X \ \sim \mathcal{N}_p \left(M \cdot \mu, M \Sigma M^T \right)$

10.3 Why to use REML

Assume now that the values of X_i in the tips of the tree are given and that the topology of the tree is known. How can we estimate the branch lengths? Let's apply ML!

Example: For a rooted tree with two tips, we measure the values x_{1i} and x_{2i} for $i = 1, \ldots, p$ of p different traits in the tips 1 and 2. The values x_{0i} in the root of the tree are unknown. For known values σ_i we assume that the value of trait x_{ji} for $j \in \{1, 2\}$ is normally distributed with mean x_{0i} and variance $\ell_j \sigma_i^2$, where ℓ_j is the unknown length of the branch to tip j.We have to maximize the likelihood

$$\begin{aligned} L(x_0, \ell_1, \ell_2) &= \prod_{i=1}^{p} \frac{1}{\sqrt{2\pi\sigma_i^2 \ell_1}} \cdot e^{-\frac{(x_{i1} - x_{i0})^2}{2\ell_1 \sigma_i^2}} \cdot \frac{1}{\sqrt{2\pi\sigma_i^2 \ell_2}} \cdot e^{-\frac{(x_{i2} - x_{i0})^2}{2\ell_2 \sigma_i^2}} \\ &= \prod_{i=1}^{p} \frac{1}{2\pi\sigma_i^2 \sqrt{\ell_1 \ell_2}} \cdot e^{-\frac{1}{2\sigma_i^2} \left(\frac{(x_{1i} - x_{0i})^2}{2\ell_1} + \frac{(x_{2i} - x_{0i})^2}{2\ell_2} \right)} \\ &= \frac{1}{\prod_{i=1}^{p} \sigma_i^2} \cdot \left(\frac{1}{2\pi \sqrt{\ell_1 \ell_2}} \right)^p \cdot e^{-\frac{1}{2} \cdot \left(\sum_{i=1}^{p} \frac{1}{\sigma_i^2} \cdot \left(\frac{(x_{1i} - x_{0i})^2}{\ell_1} + \frac{(x_{2i} - x_{0i})^2}{\ell_2} \right) \right)} \end{aligned}$$

To find values $x_{01}, \ldots, x_{0p}$ and ℓ_1 and ℓ_2 that maximize $L(x_0, \ell_1, \ell_2)$, we first note that for any ℓ_1 and ℓ_2, the x_{0i} that minimizes

$$\frac{(x_{1i} - x_{0i})^2}{\ell_1} + \frac{(x_{2i} - x_{0i})^2}{\ell_2}$$

is

$$\widehat{x_{0i}} = \frac{x_{1i} \cdot \ell_2 + x_{2i} \cdot \ell_1}{\ell_1 + \ell_2}$$

Then we search for ℓ_1 and ℓ_2 that minimize $\sqrt{\ell_1 \ell_2}$ and

$$\frac{(x_{1i} - \widehat{x_{0i}})^2}{\ell_1} + \frac{(x_{2i} - \widehat{x_{0i}})^2}{\ell_2} \ = \ \frac{\ell_1^2 \cdot (x_{1i} - x_{2i})^2}{\ell_2 \cdot (\ell_1 + \ell_2)^2} + \frac{\ell_2^2 \cdot (x_{2i} - x_{1i})^2}{\ell_1 \cdot (\ell_1 + \ell_2)^2} \ = \ \frac{(x_{1i} - x_{2i})^2}{\ell_1 + \ell_2}.$$

This means that $\ell_1 \ell_2$ should be small and $\ell_1 + \ell_2$ should be large, and we get that by setting $\ell_1 = 0$ and $\ell_2 \to \infty$ or vice versa.

This is perhaps not what we expected. What is the reason for this absurd result?

Heuristic explanation: We have one parameter per i too much in the model. This parameter vanishes for $\ell_1 = 0$ because this forces x_{0i} to be x_{1i}.

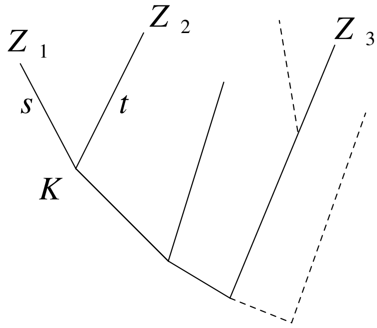
There are several ways to circumvent this problem, which also appears when for trees with more than two tips, e.g.:

- Assume a strict molecular clock such that all tips must have the same distance to the root (Thompson, 1975)
- Felsenstein's REML (REduced Maximum-Likelihood) approach is to avoid the root and consider only unrooted trees. For the example above this means that we only estimate $\ell_1 + \ell_2$ by the ML estimator
$$\widehat{\ell_1 + \ell_2} = \frac{1}{p} \sum_{i=1}^{p} \left(\frac{x_{1i} - x_{2i}}{\sigma_i} \right)^2 .$$

10.4 Computing Independent Contrasts by Pruning the Tree

Let $Z = (Z_1, \ldots, Z_m)^T$ be the vector of values for a quantitative character in the tips $b_1, \ldots, b_m$ of the tree. To compute the likelihood of the tree or correct correlations for phylogenetic relationship or to decide whether there is significant evidence for adaptation, we apply REML and transform the values in the tips back into a standard-normally distributed vector.

One way of doing this is a variant of Felsenstein's pruning algorithm. It leads to *independent* transformations – so-called *contrasts* – between the values in the tips that can be associated with the branches of the tree, which helps to interprete them.



We start with the contrast $Z_2 - Z_1$. Then we assign a value W to node k (the MRCA of nodes b_1 and b_2) that is a weighted average of Z_1 and Z_2 but independent of the contrast $Z_2 - Z_1$: Set

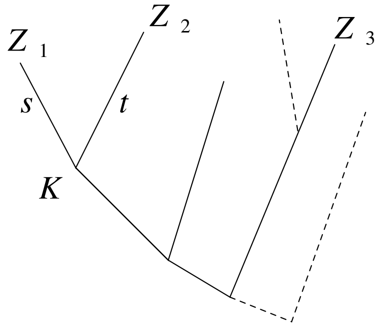
$$W := x \cdot Z_1 + (1 - x) \cdot Z_2$$

and search for x such that

$$0 = \mathrm{cov}(x \cdot Z_1 + (1 - x) \cdot Z_2, Z_1 - Z_2)$$

$$\begin{aligned} &= x \cdot \mathrm{var}(Z_1) - x \cdot \mathrm{cov}(Z_1, Z_2) + (1 - x) \cdot \mathrm{cov}(Z_2, Z_1) - (1 - x) \cdot \mathrm{var}(Z_2) \\ &= x \cdot \mathrm{var}(Z_1) - x \cdot \mathrm{var}(K) + (1 - x) \cdot \mathrm{var}(K) - (1 - x) \cdot \mathrm{var}(Z_2) \\ &= x \cdot \mathrm{var}(Z_1 - K) - (1 - x) \cdot \mathrm{var}(Z_2 - K) \\ &= x \cdot s - (1 - x) \cdot t \end{aligned}$$

$$\Rightarrow x = \frac{t}{s + t}$$



Hence, we set
$$W := \frac{t}{s + t} \cdot Z_1 + \frac{s}{s + t} \cdot Z_2.$$
If the distance between k and some tip with value Z_3 is r, then

$$\mathrm{var}(K - Z_3) = r,$$

where K is the value in node k.

We should not consider W as an estimate for K because $\mathrm{var}(W - Z_3)$

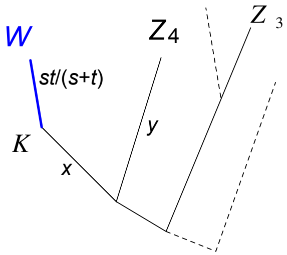
$$\begin{aligned} &= \mathrm{var}\left(\frac{s}{t + s} \cdot Z_1 + \frac{t}{t + s} \cdot Z_2 - Z_3 \right) \;=\; \mathrm{var}\left(\frac{s}{t + s} \cdot (Z_1 - K) + \frac{t}{t + s} \cdot (Z_2 - K) + K - Z_3 \right) \\ &= \left(\frac{s}{s + t} \right)^2 \cdot \mathrm{var}(Z_1 - K) + \left(\frac{t}{s + t} \right)^2 \cdot \mathrm{var}(Z_2 - K) + \mathrm{var}(K - Z_3) \\ &= \frac{s^2}{(s + t)^2} \cdot t + \frac{t^2}{(s + t)^2} \cdot s + r \;=\; \frac{st}{s + t} + r \;>\; \mathrm{var}(K - Z_3). \end{aligned}$$

Thus, we can imagine that we prune the subtree of b_1 and b_2 from the tree and extending the branch to k by length $\frac{st}{s+t}$. To the new tip at the end of this extended branch we assign a value of W. The contrast $Z_1 - Z_2$ is uncorrelated to all values at tips of this new tree and thus also to any contrasts that we can compute from them.

This means, we continue with this tree:

For the next contrast we can use $W - Z_4$.

We repeat this pruning step until we have $m - 1$ independent contrasts.



Dividing all contrasts by their standard deviations leads to a standard-normally distributed vector of contrasts:

$$\frac{Z_2 - Z_1}{\sqrt{s+t}}, \quad \frac{W - Z_4}{\sqrt{x + y + \frac{st}{s+t}}}, \quad \ldots$$

All this is only true under the null hypothesis of neutral evolution. We can reject this null hypothesis if the vector of standard-normalized contrast deviates signifcantly from the normal distribution. Since the contrasts are associated with branchs of the tree, we can then identify which branch of the tree shows evidence for process of adaptation. (Here we assume that the phylogeny is known.)

In principle, we can also use quantitative characters to estimate the tree, but usually the amount of available data is insufficient to infer the tree, adaptation processes and correlation between different quantitative traits. It ususally makes more sense to estimate the tree from molecular data and then use the independent contrasts method to analyse the evolution of the quantitative traits along the tree.

10.5 Software

Phylip: contrast

http://evolution.genetics.washington.edu/phylip/doc/contrast.html

Can deal with variation of traits within species (Above we have always assumed only one value for per trait for each species. This should be the average value, which, however, can ususally not be estimated with high precision.)

Note that correlation of different traits within species is usually different from correlation between species.

References

[F08] J. Felsenstein (2008) Comparative Methods with Sampling Error and Within-Species Variation: Contrasts Revisited and Revised. *American Naturalist* **171(6)**: 713–725

Note: when calculating correlations of contrasts x and y of two traits (that is, phylogeny-corrected correlations of the traits), `phylip contrast` assumes that the expected values of the changes are 0 and therefore estimates the correlation by $\frac{\sum_i x_i \cdot y_i}{\sqrt{\sum_i x_i^2} \cdot \sqrt{\sum_j y_j^2}}$ instead of $\frac{\sum_i (x_i - \overline{x}) \cdot (y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2} \cdot \sqrt{\sum_j (y_j - \overline{y})^2}}$.

BayesTraits

The software package BayesTraits from Mark Pagel's group provides several Bayesian and Likelihood-based methods for inferring the evolution of continuous and discrete traits along phylogenetic trees.

http://www.evolution.reading.ac.uk/BayesTraits.html

Coevol

References

[LP11] N. Lartillot, R. Poujol (2011) A phylogenetic model for investigating correlated evolution of substitution rates and continuous phenotypic characters *Mol. Biol. Evol.* **28**:729–744.

`https://megasun.bch.umontreal.ca/People/lartillot/www/downloadcoevol.html`

Some of the things you should be able to explain:

- why and how to correct for phylogenetic correlation when comparing quantitative traits among species
- what is the Brownian motion model for the evolution of quantitative traits
- how are quantitative traits correlated when they evolved neutrally along a given phylogeny
- properties of multidimensional normal distribution
- what is the REML approach and in which cases do we need to use it instead of ML?
- how to calculate the (reduced) likelihood of a tree for such data with a pruning algorithm
- What are potential problems if we want to estimate a tree from quantitative traits without any molecular data?

10.6 Extra topic, so far not covered in the lecture: Pruning algorithm for the Ornstein–Uhlenbeck model

In this section we write

$$f(x \mid \mu, v) := \frac{e^{-\frac{(x-\mu)^2}{2v}}}{\sqrt{2\pi v}}$$

for the Gaussian density function of a normal distribution with mean μ and variance v.

The Ornstein-Uhlenbeck process is a generalization of the Brownian motion. It combines the random fluctuations of the Brownian motion with the tendency to move toward some value θ. Thus, it can be applied in biology to model the evolution of a quantitative trait with a fitness optimum of θ. If the process is in a value x at some time point, its state at an infinitesimally time dt later is normally distribted with expected value $\alpha \cdot (\theta - x) \cdot dt$ and variance $\sigma^2 \cdot dt$. This has the consequence that after a longer time span t (or, in the context of phylogeny, after a branch of length t) the state of the process is normally distributed with mean $\theta + (x - \theta) \cdot e^{-\alpha t}$ and variance $\frac{\sigma^2}{2\alpha}\left(1 - e^{-2\alpha t}\right)$, in other words, has a Gaussian distribution density

$$y \mapsto f\left(y \,\middle|\, \theta + (x-\theta) \cdot e^{-\alpha t}, \frac{\sigma^2}{2\alpha}\left(1 - e^{-2\alpha t}\right)\right),$$

see e.g. Karlin, Taylor "An Introduction To Stochastic Modeling" (1998, 3rd Ed.).

The pruning algorithm of e.g. FitzJohn (2012) and Freckleton (2012) to compute the likelihood in this model uses the fact that partial likelihoods in this model are again Gaussian functions of the trait value of the focal node. (A Gaussian function is a product of a Gaussian probability density an a scaling factor.) Extenstions of the Ornstein–Uhlenbeck model allow for example that the optimal trait values vary in the tree. The RevBayes packages provides functions to analyse data based on this model. Here, we cover only the basic phylogenetic Ornstein–Uhlenbeck model.

To derive the pruning algorithm, we need the following three properties of Gaussian functions:

$$f(y \mid a + bx, w) = \frac{1}{b} \cdot f\left(x \,\middle|\, \frac{y-a}{b}, \frac{w}{b^2}\right) \tag{1}$$

$$f(x \mid \lambda, w) \cdot f(x \mid \mu, v) = f(\lambda \mid \mu, v + w) \cdot f\left(x \,\middle|\, \frac{\lambda v + \mu w}{v + w}, \frac{vw}{v + w}\right) \tag{2}$$

$$\int_{-\infty}^{\infty} f(x \mid \lambda, w) \cdot f(x \mid \mu, v) dx = f(\lambda \mid \mu, v + w) = f(\mu \mid \lambda, v + w) \tag{3}$$

Note that equation 1 implies for a tree consiting of a single branch of length t and a value of y at the tip that the likelihood is Gaussian function of the initial value x:

$$\begin{aligned}
& f\left(y \;\middle|\; \theta + (x-\theta)\cdot e^{-\alpha t}, \frac{\sigma^2}{2\alpha}\left(1-e^{-2\alpha t}\right)\right) \\
= \;& f\left(y \;\middle|\; \theta\cdot(1-e^{-\alpha t}) + x\cdot e^{-\alpha t}, \frac{\sigma^2}{2\alpha}\left(1-e^{-2\alpha t}\right)\right) \\
= \;& e^{\alpha t}\cdot f\left(x \;\middle|\; \left(y-\theta\cdot\left(1-e^{-\alpha t}\right)\right)\cdot e^{\alpha t}, \frac{\sigma^2}{2\alpha}\cdot\left(e^{2\alpha t}-1\right)\right) \qquad (4)
\end{aligned}$$

We can derive equation 1 as follows:

$$\begin{aligned}
f(y \mid a+bx, w) &= \frac{1}{\sqrt{2\pi w}}\exp\left(-\frac{(y-a-bx)^2}{2w}\right) \\
&= \frac{1}{\sqrt{2\pi w}}\exp\left(-\frac{\left(\frac{y-a}{b}-x\right)^2}{2w/b^2}\right) \\
&= \frac{1/b}{\sqrt{2\pi w/b^2}}\exp\left(-\frac{\left(x-\frac{y-a}{b}\right)^2}{2w/b^2}\right) \\
&= \frac{1}{b}\cdot f\left(x \;\middle|\; \frac{y-a}{b}, \frac{w}{b^2}\right)
\end{aligned}$$

To proof equation 2, we first expand

$$f(x \mid \lambda, w)\cdot f(x \mid \mu, v) = \frac{1}{\sqrt{2\pi w}}\frac{1}{\sqrt{2\pi v}}\exp\left(-\frac{(x-\lambda)^2}{2w}-\frac{(x-\mu)^2}{2v}\right) \qquad (5)$$

and then rewrite the exponent with the aim to obtain a single x in a term as it appears in the exponent in a Gaussian:

$$\begin{aligned}
& -\frac{(x-\lambda)^2}{2w}-\frac{(x-\mu)^2}{2v} \\
= \;& -\frac{v\cdot(x-\lambda)^2 + w\cdot(x-\mu)^2}{2vw} \\
= \;& -\frac{v\cdot(x^2-2x\lambda+\lambda^2) + w\cdot(x^2-2x\mu+\mu^2)}{2vw} \\
= \;& -\frac{x^2\cdot(v+w) - 2x\cdot(\lambda v+\mu w) + v\lambda^2 + w\mu^2}{2vw} \\
= \;& -\frac{x^2 - 2x\cdot\frac{\lambda v+\mu w}{v+w} + \frac{v\lambda^2+w\mu^2}{v+w}}{2\frac{vw}{v+w}} \\
= \;& -\frac{x^2 - 2x\cdot\frac{\lambda v+\mu w}{v+w} + \left(\frac{\lambda v+\mu w}{v+w}\right)^2 - \left(\frac{\lambda v+\mu w}{v+w}\right)^2 + \frac{v\lambda^2+w\mu^2}{v+w}}{2\frac{vw}{v+w}} \\
= \;& -\frac{\left(x-\frac{\lambda v+\mu w}{v+w}\right)^2}{2\frac{vw}{v+w}} + \frac{\left(\frac{\lambda v+\mu w}{v+w}\right)^2 - \frac{v\lambda^2+w\mu^2}{v+w}}{2\frac{vw}{v+w}} \\
= \;& -\frac{\left(x-\frac{\lambda v+\mu w}{v+w}\right)^2}{2\frac{vw}{v+w}} + \frac{(\lambda v+\mu w)^2 - (v\lambda^2+w\mu^2)\cdot(v+w)}{2vw\cdot(v+w)} \\
= \;& -\frac{\left(x-\frac{\lambda v+\mu w}{v+w}\right)^2}{2\frac{vw}{v+w}} + \frac{\lambda^2v^2 + 2\lambda v\mu w + \mu^2w^2 - v^2\lambda^2 - w^2\mu^2 - vw\lambda^2 - vw\mu^2}{2vw\cdot(v+w)}
\end{aligned}$$

$$\begin{aligned}
&= -\frac{\left(x-\frac{\lambda v+\mu w}{v+w}\right)^2}{2\frac{vw}{v+w}} + \frac{2\lambda v\mu w - vw\lambda^2 - vw\mu^2}{2vw\cdot(v+w)} \\
&= -\frac{\left(x-\frac{\lambda v+\mu w}{v+w}\right)^2}{2\frac{vw}{v+w}} - \frac{-2\lambda\mu + \lambda^2 + \mu^2}{2(v+w)} \\
&= -\frac{\left(x-\frac{\lambda v+\mu w}{v+w}\right)^2}{2\frac{vw}{v+w}} - \frac{(\lambda-\mu)^2}{2(v+w)} \qquad (6)
\end{aligned}$$

Inserting this into equation 5 we obtain

$$\begin{aligned}
&f(x \mid \lambda, w)\cdot f(x \mid \mu, v) \\
&= \frac{1}{\sqrt{2\pi w}}\frac{1}{\sqrt{2\pi v}}\exp\left(-\frac{\left(x-\frac{\lambda v+\mu w}{v+w}\right)^2}{2\frac{vw}{v+w}} - \frac{(\lambda-\mu)^2}{2(v+w)}\right) \\
&= \frac{1}{\sqrt{2\pi(v+w)}}\frac{1}{\sqrt{2\pi\frac{vw}{v+w}}}\cdot\exp\left(-\frac{(\lambda-\mu)^2}{2(v+w)}\right)\cdot\exp\left(-\frac{\left(x-\frac{\lambda v+\mu w}{v+w}\right)^2}{2\frac{vw}{v+w}}\right) \\
&= f(\lambda \mid \mu, v+w)\cdot f\left(x \,\middle|\, \frac{\lambda v+\mu w}{v+w}, \frac{vw}{v+w}\right)
\end{aligned}$$

Equation 3 has to do with the fact that sum of two normally distributed random variables is normally distributed, but also follows easisly from the above and the fact that the area under a probability density is 1:

$$\begin{aligned}
&\int_{-\infty}^{\infty} f(x \mid \lambda, w)\cdot f(x \mid \mu, v)dx \\
&= \int_{-\infty}^{\infty} f(\lambda \mid \mu, v+w)\cdot f\left(x \,\middle|\, \frac{\lambda v+\mu w}{v+w}, \frac{vw}{v+w}\right)dx \\
&= f(\lambda \mid \mu, v+w)\cdot \int_{-\infty}^{\infty} f\left(x \,\middle|\, \frac{\lambda v+\mu w}{v+w}, \frac{vw}{v+w}\right)dx \\
&= f(\lambda \mid \mu, v+w)
\end{aligned}$$

To move on to the pruning recursion, let R be a node with two daughter nodes N and M with branches of legths t_N and t_M, and let d_N, d_M and d_R be the vectors of trait values at all leaves that stem from the nodes N, M, or R, respectively. Let $k_N \cdot f(x \mid \mu_N, v_N)$ and $k_M \cdot f(x \mid \mu_M, v_M)$ be the partial likelihoods at nodes N and M. That is, e.g. $k_N \cdot f(x \mid \mu_N, v_N)$ is the multi-dimensional probability density of d_N, assuming that the trait value in N is x and assuming given values for the parameters θ, α, σ of the Ornstein–Uhlenbeck process. We already assume here that the partial likelihoods are Gaussian – including the possibility of a fixed value at μ_N with $v_N = 0$ if N is a tip – and show that the partial likelihood at R is then also Gaussian, which then justifies the assumption as it shows that we stay in the family of Gaussian functions.

The Likelihood in R starting in x is the product of the probability densities of d_N and d_M, both starting in x in R. If N is a leaf with value y, the density is given by equation 4, which is a Gaussian function of x. Otherwise, the density of d_N is calculated taking all possible values y in N into account and apply first equation 3 and then equation 1 as follows:

$$\begin{aligned}
&\int_{-\infty}^{\infty} f\left(y \;\middle|\; \theta + (x-\theta)\cdot e^{-\alpha t}, \frac{\sigma^2}{2\alpha}\left(1-e^{-2\alpha t}\right)\right)\cdot k_N \cdot f(y \mid \mu_N, v_N)dy \\
&= \quad k_N \cdot f\left(\mu_N \;\middle|\; \theta\cdot\left(1-e^{-\alpha t}\right) + x\cdot e^{-\alpha t}, v_N + \frac{\sigma^2}{2\alpha}\left(1-e^{-2\alpha t}\right)\right) \\
&= \quad k_N e^{\alpha t} f\left(x \;\middle|\; e^{\alpha t}\cdot\left(\mu_N - \theta\cdot\left(1-e^{-\alpha t}\right)\right), v_N\cdot e^{2\alpha t} + \frac{\sigma^2}{2\alpha}\left(e^{2\alpha t}-1\right)\right) \\
&= \quad k_N e^{\alpha t} f\left(x \;\middle|\; e^{\alpha t}\cdot(\mu_N-\theta)+\theta, v_N\cdot e^{2\alpha t} + \frac{\sigma^2}{2\alpha}\left(e^{2\alpha t}-1\right)\right)
\end{aligned}$$

Note that this is a Gaussian function of x, and analogously we get the density of the data d_M, which is again a Gaussian function that has the same structure as above, with each N replaced by M. As Gaussian functions are determined by tree parameters (mean and variance of the density part and scaling factor), we only need to calculate these three parameters when we implement the algorithm. To obtain the partial likelihood function in R, which assigns to each x the denstity of d_R, assuming the value x in R, we multiply the two Gaussians and obtain again a Gaussian function of x according to equation 2. This leads to

$$\begin{aligned}
k_R &= k_N k_M e^{2\alpha t}\cdot f\left(e^{\alpha t}\cdot(\mu_N-\theta)+\theta \;\middle|\; e^{\alpha t}\cdot(\mu_M-\theta)+\theta,\ (v_N+v_M)\cdot e^{2\alpha t} + \frac{\sigma^2}{\alpha}\left(e^{2\alpha t}-1\right)\right) \\
&= k_N k_M e^{2\alpha t}\cdot f\left(e^{\alpha t}\cdot\mu_N \;\middle|\; e^{\alpha t}\cdot\mu_M,\ (v_N+v_M)\cdot e^{2\alpha t} + \frac{\sigma^2}{\alpha}\left(e^{2\alpha t}-1\right)\right) \\
&= k_N k_M e^{\alpha t}\cdot f\left(\mu_N \;\middle|\; \mu_M,\ v_N+v_M + \frac{\sigma^2}{\alpha}\left(1-e^{-2\alpha t}\right)\right) \\
\mu_R &= \frac{\left(e^{\alpha t}(\mu_N-\theta)+\theta\right)\cdot\left(v_M+\frac{\sigma^2}{2\alpha}\cdot\left(1-e^{-2\alpha t}\right)\right) + \left(e^{\alpha t}(\mu_M-\theta)+\theta\right)\cdot\left(v_N+\frac{\sigma^2}{2\alpha}\cdot\left(1-e^{-2\alpha t}\right)\right)}{v_N+v_M+\frac{\sigma^2}{\alpha}\left(1-e^{-2\alpha t}\right)} \\
v_R &= \frac{e^{2\alpha t}\cdot\left(v_N+\frac{\sigma^2}{2\alpha}\left(1-e^{-2\alpha t}\right)\right)\cdot\left(v_M+\frac{\sigma^2}{2\alpha}\left(1-e^{-2\alpha t}\right)\right)}{v_N+v_M+\frac{\sigma^2}{\alpha}\left(1-e^{-2\alpha t}\right)}.
\end{aligned}$$

11 Model selection

11.1 Concepts: AIC, hLRT, BIC, DT, Model averaging, and bootstrap again

AIC

The likelihood of a model M,

$$L_D(M) = \max_{\text{``}\theta\in M''} L_D(\theta) = \max_\theta \Pr_{M,\theta}(D)$$

tells us how well M fits the data D. The more parameter dimensions d (i.e. $\theta = (\theta_1,\theta_2,\ldots,\theta_d)$) the higher the likelihood and the higher the risk of *overfitting*!

Under certain assuptions (with normal distributions, not phylogenies), the error of future predictions in terms of Kullback-Leibler-Information can be estimated by Akaike's Information Criterion:

$$\mathrm{AIC} = -2\cdot\log L_D(M) + 2\cdot d.$$

One approach: use the model of lowest AIC.

Model selection via LRT

If we have a model M_1 with $n-d$ parameters nested in a model M_2 with n parameters, then under the null-hypothesis that the data come from the more simple model M_1, the double log likelihood ratio is under certain conditions approximately chisquare-distributed with d degrees of freedom,

$$\mathcal{L}_{M_1}\left(2\cdot\log\frac{L_D(M_2)}{L_D(M_1)}\right) = \mathcal{L}_{M_1}\left(2\cdot\left(\log L_D(M_2) - \log L_D(M_1)\right)\right) \approx \chi^2_d,$$

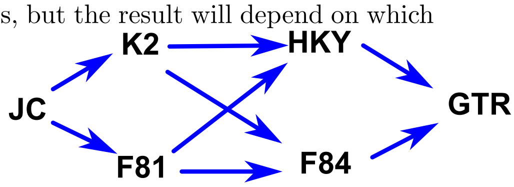
where the likelihood of a model $L_D(M_i)$ is maximum likelihood obtained by optimization over all parameters of the model.

In cased where the χ^2_d approximation is dubious (e.g. when the models are not nested) one can simulate the likelihood ratio distribution under the null hypothesis.

One approach of model selection is to accept the more complex model only if the simpler model is significantly violated.

Problems of this LRT approach

- Model selection is different from the original idea of testig. If a test does not show significance, one cannot conclude anyzthing, and especially not that the null hypothesis (the simpler model) ist favorable.
- One can inprinciple apply this to a hierarchy of nested models, but the result will depend on which intermediate steps are allowed.



Bayesian model selection

Each model M_i has a prior probability $\Pr(M_i)$. Its posterior probability is then

$$\Pr(M_i|D) = \frac{\Pr(D|M_i) \cdot \Pr(M_i)}{\sum_j \Pr(D|M_j) \cdot \Pr(M_j)}$$

with

$$\Pr(D|M_i) = \int_\theta \Pr(D|M_i, \theta) \cdot \Pr(\theta|M_i) d\theta.$$

Note the difference between $\Pr(D|M_i)$, where we integrate over θ, and $L_D(M_i)$ where we maximize over θ! The sum over all models in the denominator above cancels if we compare two models by taking the ratios of their posteriors:

$$\frac{\Pr(M_1|D)}{\Pr(M_2|D)} = \frac{\Pr(D|M_1)}{\Pr(D|M_2)} \cdot \frac{\Pr(M_1)}{\Pr(M_2)}$$

The fraction $\Pr(D|M_1)/\Pr(D|M_2)$ is called the *Bayes factor* of the models M_1 and M_2.

To avoid the priors of the models we use the Bayes factors rather than the posterior distributions to decide between models. If the Bayes factor $\Pr(D|M_1)/\Pr(D|M_2)$ is larger than 1 we may favor M_1 over M_2. The rule of thumb says that a Bayesfactor between 1 and 3 is not worth mentioning, between 3 and 20 it indicates some evidence, between 20 and 150 strong evidence, and over 150 very strong evidence.

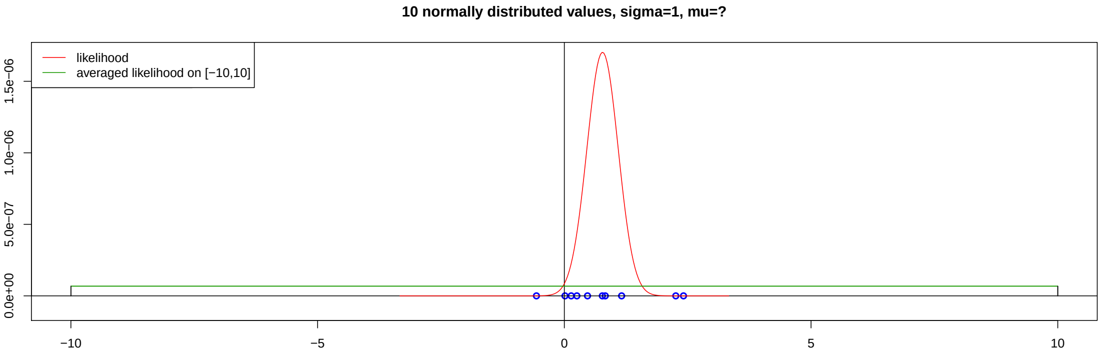
It is important to note that even if the priors $\Pr(M_i)$ of the models do not matter, the priors $\Pr_{M_i}(\theta)$ within the model may have a strong influence. An important difference between Bayesian parameter estimation and Bayesian model selection is that priors become less important for paramterestimation as more data is added. This is not the case in model selection, where priors for the model parameters will always have an important impact!

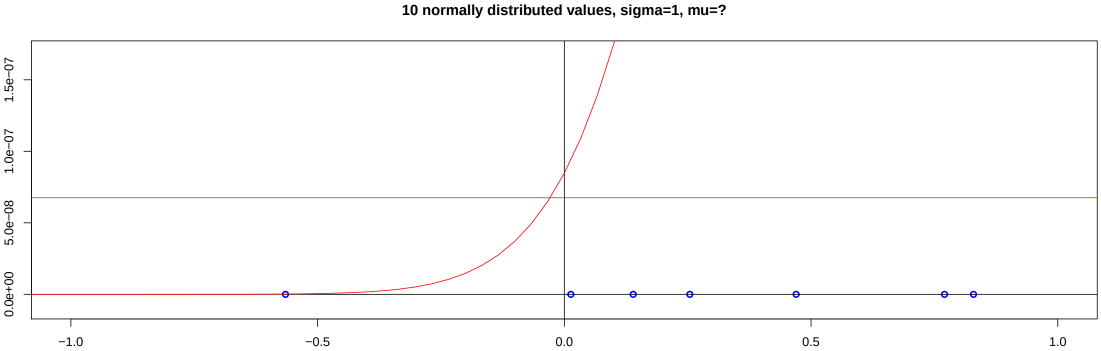
Some people find the following properties of posterior probabilities counter-intuitive:

Lindley's paradox In the limit of uninformative priors, the simplest model is always preferred.

Star-tree paradox If all internal node have length (almost) 0, there will often be a fully resolved tree with high posterior probability (deciding between topology can be considered as model selection).

Fair-coin paradox If a (almost) fair coin is tossed many times, but the models compared allow only for one or the other side to have probability larger than 0.5, it will often be the case that one of the two models have a high posterior probability.





Computation of Bayes factors from MCMC runs

If $\theta^{(1)}, \dots, \theta^{(m)}$ are (approximately) independent samples $\Pr(\theta|D,M)$ we can compute $\Pr(D|M)$ by importance sampling approximation:

$$\begin{aligned}
\Pr(D|M) &= \frac{\Pr(D|M)}{\int_\theta \Pr(\theta|M)d\theta} = \frac{1}{\int_\theta \frac{\Pr(\theta|M)}{\Pr(D|M)}d\theta} \\
&\approx \frac{1}{\frac{1}{m}\sum_{\theta^{(i)}} \frac{\Pr(\theta^{(i)}|M)}{\Pr(D|M)\cdot\Pr(\theta^{(i)}|D,M)}} \\
&= \frac{m}{\sum_{\theta^{(i)}} \frac{\Pr(\theta^{(i)}|M)}{\Pr(D,\theta^{(i)}|M)}} = \frac{m}{\sum \frac{1}{\Pr(D|M,\theta^{(i)})}}
\end{aligned}$$

(note that this harmonic mean estimator may be numerically unstable.)

BIC

For a model M with a d-dimensional parameter θ and data D consisting of N independent samples, we can under certain conditions approximate

$$\log \Pr(D|M) \approx \log \Pr(D|M,\widehat{\theta}) - \frac{d}{2}\cdot \log N$$

We call $BIC(M) = -2 \cdot \log \Pr(D|M,\widehat{\theta}) + d \log N$ the *Bayesian Information Criterion* or *Schwartz Criterion*, and favor models of low BIC. Moreover,

$$\frac{\Pr(D|M_1)}{\Pr(D|M_2)} \approx e^{(BIC(M_2)-BIC(M_1))/2}.$$

Minin, Abdo, Joyce, Sullivan (2003): “[..] rather than worry about the somewhat artificial criterion whether or not a model is correct, we will focus on the accuracy of the branch lengths estimated under various models”

- Assume that the unrooted phylogeny with k tips is known or use some inital guess.
- Candidate modles: $M_1, \ldots, M_m$ with uniform prior $\Pr(M_i) = 1/m$.
- Branch lengths estimated with model M_i:

$$B_i = (\widehat{B_{i,1}}, \ldots, \widehat{B_{i,2k-3}})$$

$$||B_i - B_j|| := \sqrt{\sum_{\ell=1}^{2k-3} \left(\widehat{B_{i,\ell}} - \widehat{B_{j,\ell}}\right)^2}$$

- Risk when choosing Model M_i:

$$R_i = \sum_{j=1}^{m} ||B_i - B_j|| \cdot \Pr(M_j|D) \approx \sum_{j=1}^{m} ||B_i - B_j|| \cdot \frac{e^{-\mathrm{BIC}(M_j)/2}}{\sum_{h=1}^{m} e^{-\mathrm{BIC}(M_h)/2}}$$

DT

The Decision-Theoretic (DT) criterion of Minin, Abdo, Joyce, Sullivan (2003) is to choose the model with the minimal risk

$$R_i \approx \sum_{j=1}^{m} ||B_i - B_j|| \cdot \frac{e^{-\mathrm{BIC}(M_j)/2}}{\sum_{h=1}^{m} e^{-\mathrm{BIC}(M_h)/2}}$$

based on the initial tree.

In a follow-up paper they study the robustness of this approach against uncertainty about the initial tree.

Model averaging

Let θ be the vector of parameters and $s(\theta)$ some interesting aspect of the parameters. s must have the same meaning in all considered models $M_1, \ldots, M_m$. We can then estimate:

$$\Pr(s(\theta)|D) \approx \sum_{i=1}^{m} \Pr(s(\theta)|D, M_i) \cdot \Pr(M_i|D)$$

One possible implementation of Model averaging is reversible-jump MCMC, see Huelsenbeck, Larget, Alfaro (2004)

Reversible-Jump MCMC

If an MCMC procedure shall sample from a state space that has several continuous components of different dimensions (e.g. for averaging over several models with different numbers of parameters), the problem arises that a density of n dimensions cannot be directly compared to a density in e.g. $n+1$ dimensions in a Metropolis-Hastings ratio.[1.5ex] Simple approach is to add an artificial parameter to the state of n dimensions, which has a uniform distribution on $[0,1]$ and no influence on the probability of the data. [1.5ex] Then you can apply Metropolis-Hasting to perform *reversible jumps* between the components of dimension n and dimension $n+1$.

Parametric bootstrap approach

If different models lead to different results, and it is not clear which model fits best, one should ask for all i and j:

If model M_i was right, how accurate would an analysis based on model M_j be?

do for each i:

1. $\widehat{\theta}_i$:= estimate θ based on M_i
2. repeat for $k = 1, \ldots, 1000$:
   (a) $D_{i,k}$: simulated data based on M_i and $\widehat{\theta}_i$
   (b) For all j: let $\widetilde{\theta}_{i,k,j}$ the M_j-based estimation for dataset $D_{i,k}$
3. Analyse for all j how close the average $\overline{\widetilde{\theta}_{i,\cdot,j}}$ is to $\widehat{\theta}_i$.

11.2 Does model selection matter?

Substitution models for phylogeny reconstruction

Study with wide range of data sets for : Ripplinger and Sullivan (2008)

- Different model selection methods led to different models in 80% of the cases
- use of different best-fit models changes the optimal tree topology in 50% of the cases, but only for poorly supported branches.
- BIC and DT selected simpler models than hLRT and AIC. The simpler models performed at least as well as the more complicated.
- Use of models supported by model selection in ML gave better trees than MP or ML with K2P.
- Trees based on models favored by different model selection strategies gave similar results in hypothesis tests.
- Recommend to use the simpler BIC- and DT-selected models.

From Lin Himmelmann's PhD thesis

Simulation study to compare (relaxed) molecular-clock models

MC strict molecular clock model

CPP compound Poisson process

DM Dirichlet model (rate factors on branches add up to 1, no correlation of neighboring branches)

ULN uncorrelated log-normal

UEX uncorrelated exponential

Results of Lin's model comparison

Data origin	Performance of models in analysis
MC	MC best CPP, DM, ULN almost as good UEX much worse
CPP, DM, ULN	MC, CPP, DM, ULN give good results UEX slightly worse
UEX	DM, ULN best UEX slightly worse CPP worse MC worst

Lin recommends: DM, ULN okay for most situations
More severe than substitution model selection may be:

- Alignment
- Confusion of paralogs
- Gene trees can differ due to recombination combined with
  - Incomplete Lineage Sorting (ILS, details on white board)
  - Horizontal gene transfer (HGT)
  - Coalescence of lineages further back than speciation
  - Introgression
- Phylogenetic methods can be confused by incompatible trees

Some of the things you should be able to explain:

- criteria AIC, BIC, hLRT, Bayes factors, DT and model averaging to decide which model to use for your data (e.g. substitution model for sequence data)
- Lindley's paradox and other differences between Bayesian and frequentist approach
- how relevant is model selection, also compared to other potential problems in your data

12 Insertion-Deletion Models for Statistical Alignment

12.1 Alignment sampling with pairHMMs

To Do: estimate mutation rates from sequences

```
ACTCGCGCTT
 ACGTCGATT
```

Classical Approach:

1. Take best Alignment:

```
AC_TCGCGCTT
ACGTCGA__TT
```

2. Count Mutations in best Alignment:

   1 Mismatch : 7 Matches

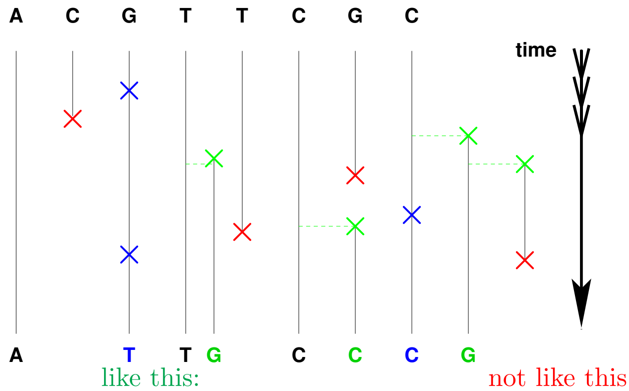
   2 Indels (3 Sites) : 8 homologous Sites

Problem: underestimation of mutation rates, since alignment fits too well!
What are typical Alignments and Mutation Rates for given sequences?
Idea: Generate many random alignments A with corresponding mutation rates M according to

$$\Pr\big(\,(A, M) \mid \text{sequences}\,\big)$$

Needed: A model of sequence evolution with insertions, deletions and substitutions. Otherwise $\Pr(\dots)$ has no meaning!

Model of Sequence Evolution
Thorne, Kishino, Felsenstein (1991):
Deletions with rate μ at each site.
Insertions with rate λ right of each site & at the very left.
Substitutions with Rate s at each site.



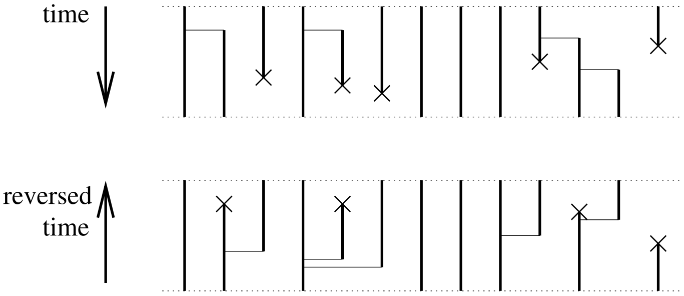
like this: not like this:

TKF alignment convention:

```
ACGT_TC_GC_
A_TTG_CC_CG
```

```
ACGT_TCG_C_
A_TTG_C_CCG
```

Reversibility?

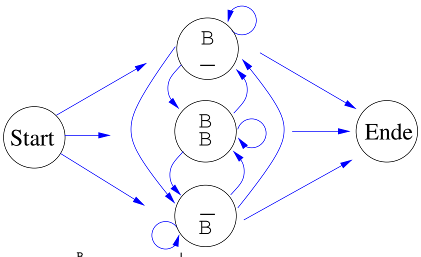


Consequence of TKF convention

The bare alignment

```
BBBB_BB_BB_
B_BBB_BB_BB
```

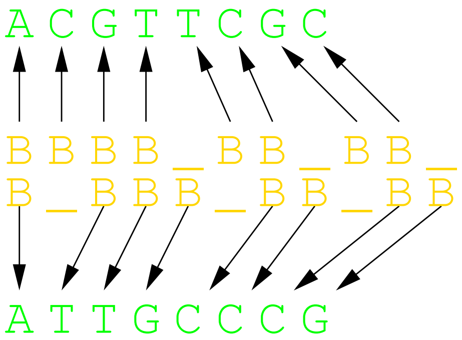
is generated by a Markov chain:



from \ to	B B	B _	_ B
B B	$(1-\lambda\beta)\frac{\lambda}{\mu}e^{-\mu}$	$(1-\lambda\beta)\frac{\lambda}{\mu}(1-e^{-\mu})$	$\lambda\beta$
B _	$\lambda\beta\,\frac{e^{-\mu}}{1-e^{-\mu}}$	$\lambda\beta$	$\frac{1-e^{-\mu}-\mu\beta}{1-e^{-\mu}}$
_ B	$(1-\lambda\beta)\frac{\lambda}{\mu}e^{-\mu}$	$(1-\lambda\beta)\frac{\lambda}{\mu}(1-e^{-\mu})$	$\lambda\beta$

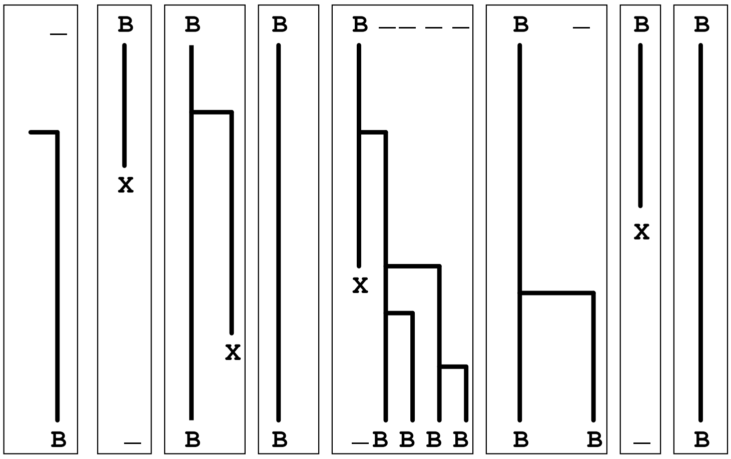
transition probabilies im (model: TKF'91), $\beta = \frac{1-e^{\lambda-\mu}}{\mu-\lambda e^{\lambda-\mu}}$

The Markov chain (the alignment) is hidden, observable is the pair of sequences emitted by the alignment.



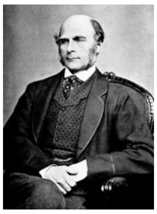
pair Hidden Markov Model (pair HMM)

Why Markov?



Galton and Watson

Sir Francis Galton
1822–1911



Henry William Watson
1827–1903

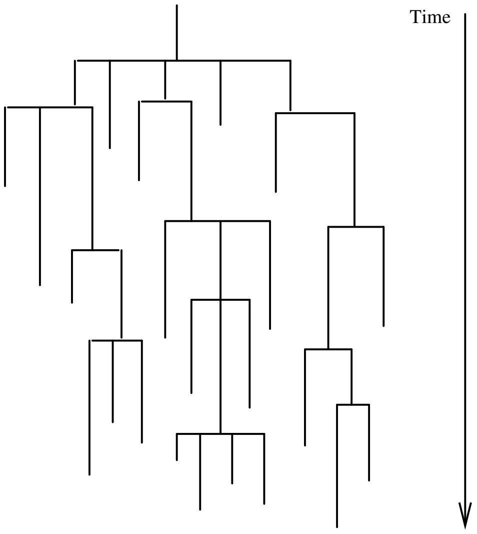
Galton Watson Tree

$X_k :=$ number of offsprings at node k

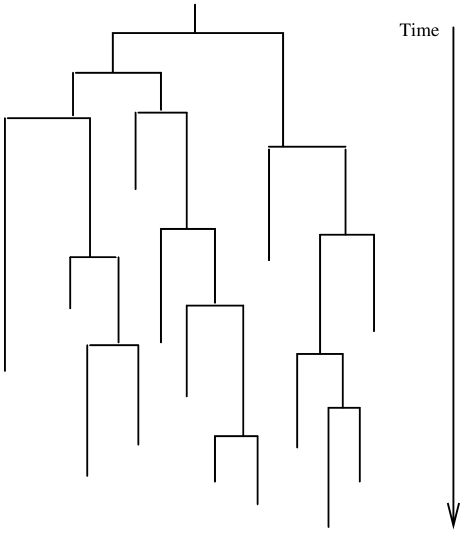
$X_1, X_2, X_3, \ldots$ i.i.d. random variables

$\mathbb{E}X_k < 1$: "subcritical"
$\mathbb{E}X_k = 1$: "critical"
$\mathbb{E}X_k > 1$: "supercritical"

Galton-Watson Process in continuous time



Binary Branching GW Process in cont. time

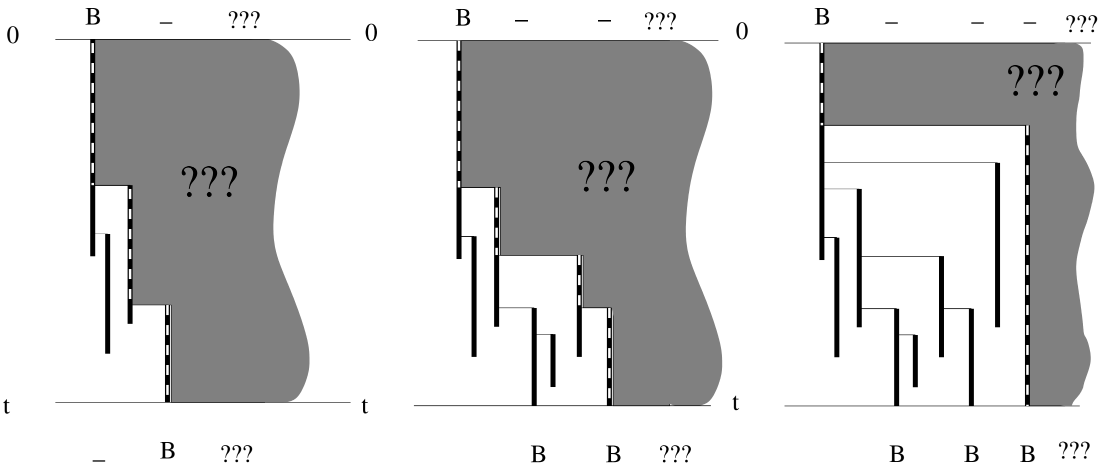


Theorem 6 *If a Galton-Watson process with binary branching or geometric offspring distribution (on* $\{0, 1, 2, \dots\}$*) is still alive at time* t*, then the number of survivors at time* t *is geometrically distributed (on* $\{1, 2, 3, \dots\}$*).*

on $\{0, 1, 2, \dots\}$: $\Pr(X = k) = (1-p)^k \cdot p$, $\mathbb{E}(X) = (1-p)/p$
on $\{1, 2, \dots\}$: $\Pr(X = k) = (1-p)^{k-1} \cdot p$, $\mathbb{E}(X) = 1/p$

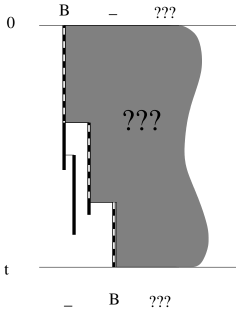
The geometric distribution is the only one on $\{(0,)1, 2, 3, \dots\}$ without memory: $\Pr(X = n+k \mid X > n) = \Pr(X = k)$

Why Geometric Distribution?



Computing transition probabilies

Simplification: $\lambda = \mu$



$X :=$ number of survivors at time t $\quad \mathbb{E}(X) = 1$

$\Pr(X = k \mid X > 0) = (1-p)^{k-1} \cdot p$

$\frac{1}{p} = \mathbb{E}(X \mid X > 0) = 1 + t \cdot \lambda \qquad \Rightarrow p = 1/(1 + t \cdot \lambda)$

$$\begin{aligned}
\Pr\left(\begin{matrix}\texttt{-}\\ \texttt{B}\end{matrix} \to \begin{matrix}\texttt{-}\\ \texttt{B}\end{matrix}\right) &= 1 - \frac{1}{1+t\lambda} = \frac{t\lambda}{1+t\lambda} = \Pr\left(\begin{matrix}\texttt{B}\\ \texttt{B}\end{matrix} \to \begin{matrix}\texttt{-}\\ \texttt{B}\end{matrix}\right)\\
\Pr\left(\begin{matrix}\texttt{-}\\ \texttt{B}\end{matrix} \to \begin{matrix}\texttt{B}\\ \texttt{B}\end{matrix}\right) &= \frac{1}{1+t\lambda} \cdot e^{-t\lambda} = \Pr\left(\begin{matrix}\texttt{B}\\ \texttt{B}\end{matrix} \to \begin{matrix}\texttt{B}\\ \texttt{B}\end{matrix}\right)
\end{aligned}$$

$$\begin{aligned}
1 &= \mathbb{E}(X)\\
&= \Pr(X = 0) \cdot \mathbb{E}(X \mid X = 0)\\
&\quad + \Pr(X > 0) \cdot \mathbb{E}(X \mid X > 0)\\
&= \Pr(X > 0) \cdot (1 + t \cdot \lambda)
\end{aligned}$$

$$\Rightarrow \Pr\begin{pmatrix}\texttt{B}\\ \texttt{B}\end{pmatrix} + \Pr\begin{pmatrix}\texttt{B} & \texttt{-}\\ \texttt{-} & \texttt{B}\end{pmatrix} = \Pr(X > 0) = \frac{1}{1 + t \cdot \lambda}$$

$$\begin{aligned}
\Pr\begin{pmatrix}\texttt{B} & \texttt{-}\\ \texttt{-} & \texttt{B}\end{pmatrix} &= \frac{1}{1 + t \cdot \lambda} - e^{-t\lambda}\\
\Pr\left(\begin{matrix}\texttt{B}\\ \texttt{-}\end{matrix} \to \begin{matrix}\texttt{-}\\ \texttt{B}\end{matrix}\right) &= \frac{\Pr\left(\begin{smallmatrix}\texttt{B} & \texttt{-}\\ \texttt{-} & \texttt{B}\end{smallmatrix}\right)}{\Pr\left(\begin{smallmatrix}\texttt{B}\\ \texttt{-}\end{smallmatrix}\right)} = \frac{\frac{1}{1+t\cdot\lambda} - e^{-t\lambda}}{1 - e^{-t\lambda}}\\
&= \frac{1 - e^{-t\lambda} \cdot (1 + t\lambda)}{(1 + t\lambda) \cdot (1 - e^{-t\lambda})}
\end{aligned}$$

From the previous calculation we obtain that after $\begin{smallmatrix}\texttt{B}\\ \texttt{-}\end{smallmatrix}$ the probability that this site has no surviving offspring is

$$1-\frac{1-e^{-t\lambda}\cdot(1+t\lambda)}{(1+t\lambda)\cdot(1-e^{-t\lambda})}=\frac{t\lambda}{(1+t\lambda)\cdot(1-e^{-t\lambda})}.$$

As the probability that the next B survives is $e^{-t\lambda}$, we obtain

$$\begin{aligned}\Pr\begin{pmatrix}\texttt{B} & \rightarrow & \texttt{B}\\ \texttt{-} & & \texttt{B}\end{pmatrix} &= \frac{t\lambda e^{-t\lambda}}{(1+t\lambda)\cdot(1-e^{-t\lambda})}\\ \Pr\begin{pmatrix}\texttt{B} & \rightarrow & \texttt{B}\\ \texttt{-} & & \texttt{-}\end{pmatrix} &= \frac{t\lambda\cdot\left(1-e^{-t\lambda}\right)}{(1+t\lambda)\cdot(1-e^{-t\lambda})} = \frac{t\lambda}{(1+t\lambda)}\end{aligned}$$

Aim: Sequences are given. Generate alignments A and mutation rates $M=(\lambda,\mu,s)$ according to

$$\Pr\left(\ (A,M)\mid \text{sequences}\ \right)$$

partial steps:

1. Assume that the mutation rates M are known. Generate alignments A according to

$$\Pr\left(\ A\mid \text{sequences}, M\ \right)$$

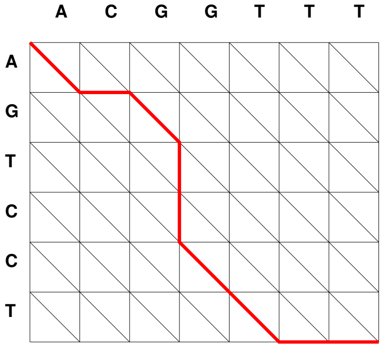
2. Assume that the alignment A is known. Generate values for the mutation rates M according to

$$\Pr\left(\ M\mid \text{sequences}, A\ \right)$$

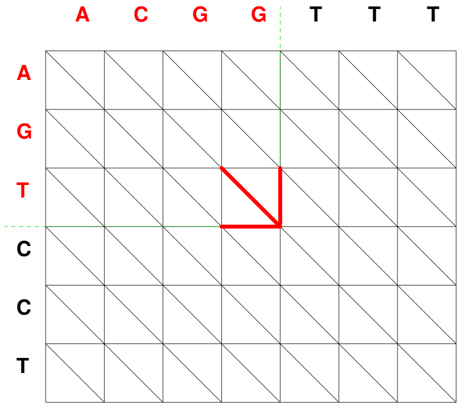
3. combine 1. and 2.

Path representation of an alignment

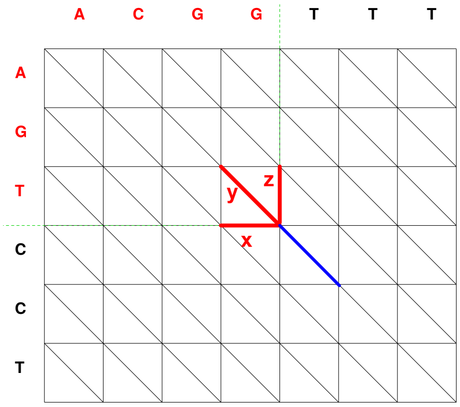
```
A_GTCCT__
ACG__GTTT
```



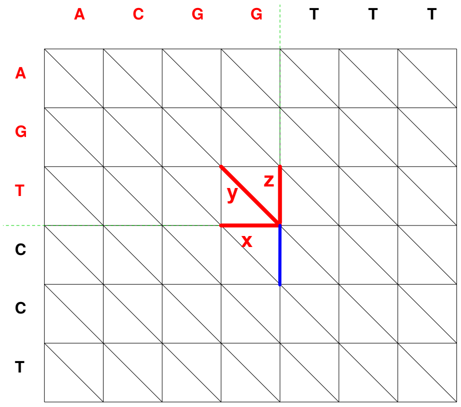
$$\Pr(\text{sequences} \mid M) = \sum_{\text{alignment } A} \Pr(A,\text{sequ.} \mid M)$$



Summing efficiently: label each edge with
Pr(Alignment contains this edge and generates the sequences so far | M)

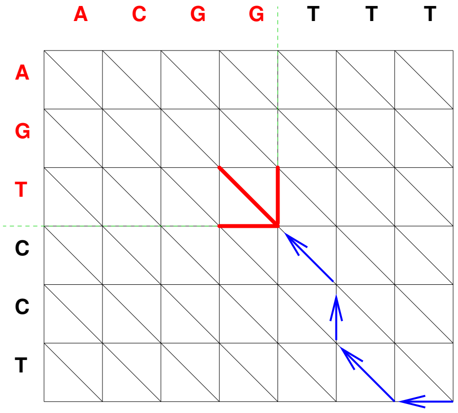


$$\left(x \cdot \Pr\left(\begin{matrix}\texttt{-}\\ \texttt{B}\end{matrix} \to \begin{matrix}\texttt{B}\\ \texttt{B}\end{matrix}\right) + y \cdot \Pr\left(\begin{matrix}\texttt{B}\\ \texttt{B}\end{matrix} \to \begin{matrix}\texttt{B}\\ \texttt{B}\end{matrix}\right) + z \cdot \Pr\left(\begin{matrix}\texttt{B}\\ \texttt{-}\end{matrix} \to \begin{matrix}\texttt{B}\\ \texttt{B}\end{matrix}\right)\right) \cdot \pi_C \cdot P_{C\to T}$$

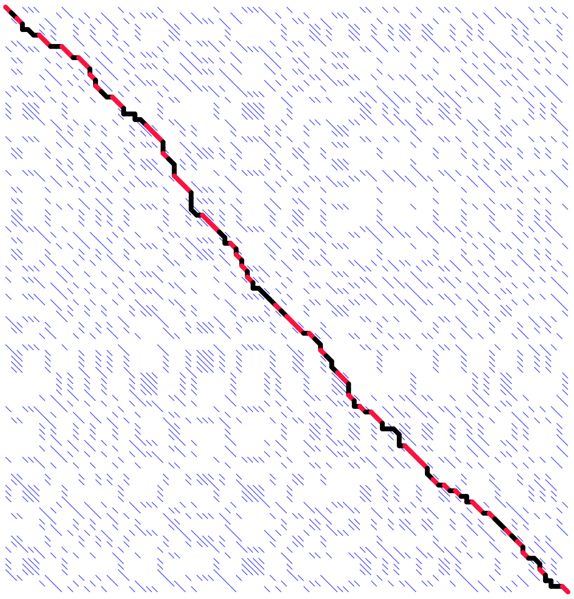


$$\left(\color{red}{x} \cdot \Pr\left(\begin{matrix}\texttt{-}\\ \texttt{B}\end{matrix} \to \begin{matrix}\texttt{B}\\ \texttt{-}\end{matrix}\right) + \color{red}{y} \cdot \Pr\left(\begin{matrix}\texttt{B}\\ \texttt{B}\end{matrix} \to \begin{matrix}\texttt{B}\\ \texttt{-}\end{matrix}\right) + \color{red}{z} \cdot \Pr\left(\begin{matrix}\texttt{B}\\ \texttt{-}\end{matrix} \to \begin{matrix}\texttt{B}\\ \texttt{-}\end{matrix}\right) \right) \cdot \pi_C$$

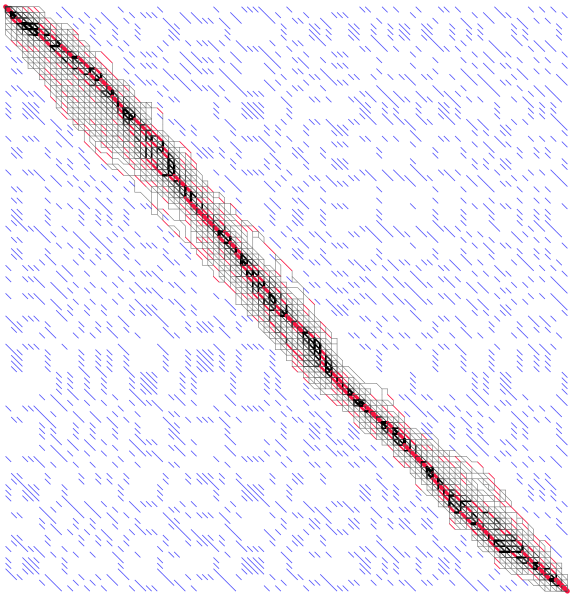
After labeling all edges, generate alignment backwards.



Random decisions in each step depend on edge labels and Markov transition probabilities.



True alignment for simulated sequence pair of lenght 100 with indel rate 0.3 and substitution rate 0.4.



5000 sampled alignments for simulated sequence pair of lenght 100 with indel rate 0.3 and substitution rate 0.4

partial steps:

1. Assume that the mutation rates M are known. Generate alignments A according to

$$\Pr(\ A \mid \text{sequences}, M\ )$$

(as explained before)

2. Assume that the alignment A is known. Generate values for the mutation rates M according to

$$\Pr(\ M \mid \text{sequences}, A\ )$$

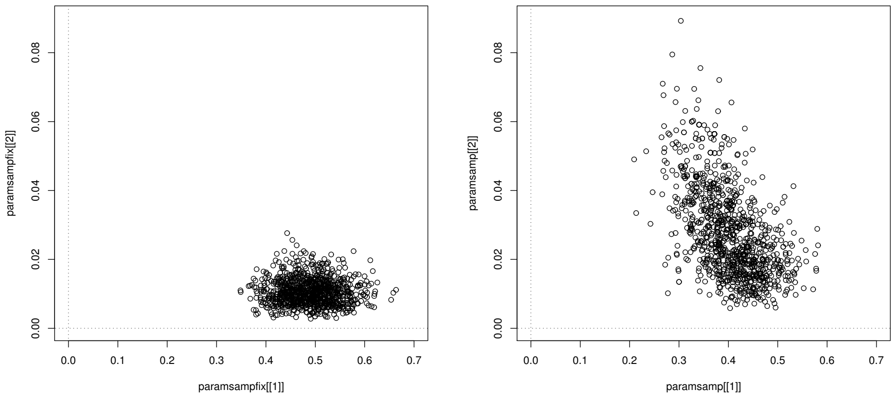
by Metropolis Hastings Algorithm

3. Combine 1. and 2. (Gibbs-Sampling)

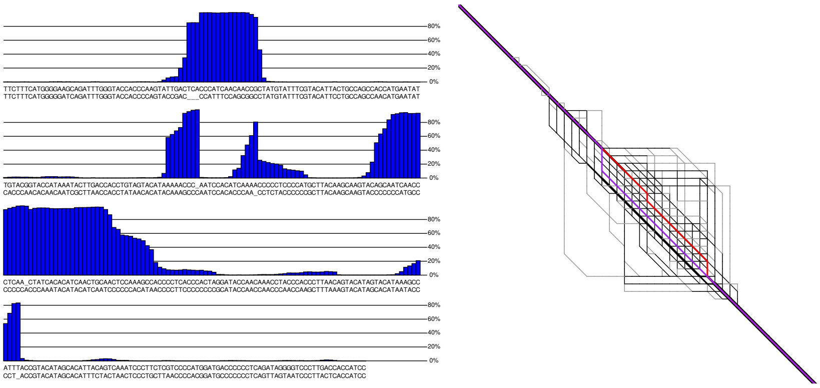
=> Markov chain Monte Carlo Method for sampling (A,M) according to

$$\Pr(\ A, M \mid \text{sequences}\ )$$

posterior probability samplings of mutation parameters for HVR-1 of human and orangutan with alignment given in data base (left) and alignments sampled simultaneously with parameters (right)



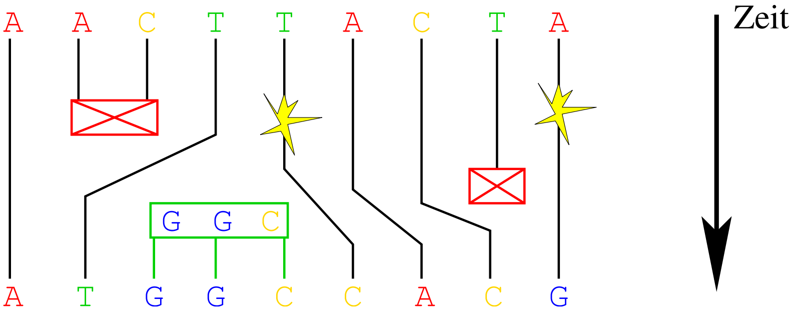
Alignment Accuracy: HVR1 of Human and Orang



D. Metzler, R. Fleißner, A. Wakolbinger, A. von Haeseler (2001) *J. Mol Evol.* 53:660-669.

12.2 Insertions and deletions of more than one site

InDels are usually longer than 1 position



References

[TKF92] J.L. Thorne, H. Kishino, J. Felsenstein (1992) Inching towards reality: an improved likelihood model for sequence evolution. *J. Mol. Evol.* **34**, 3-16.

[M03] D. Metzler (2003) Statistical alignment based on fragment insertion and deletion models. *Bioinformatics* **19**:490-499.

FID Model (also a pairHMM):

- instead of single nucleotides, fragments are inserted an deleted with rate λ.
- Length of the fragments: geometrically distributed, mean length: γ.

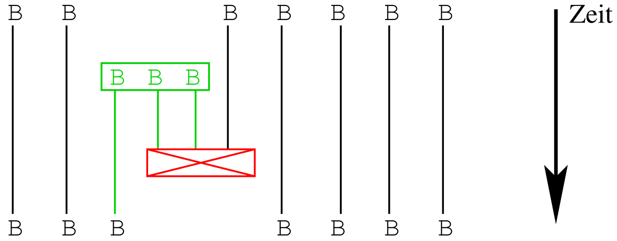
$$\Pr(L = k) = \frac{1}{\gamma}\left(1 - \frac{1}{\gamma}\right)^{k-1}$$

FID transition probabilities

The transition probability of the FID model can be derived from the transition probabilities of the simplified ($\mu = \lambda$) TKF91 model, taking into account that with a probability of $1 - \frac{1}{\gamma}$ the position is in the same fragment as its left neighbor and thus is in the same state. With the probability $1/\gamma$ the fragment ends and the state of the next fragment is chosen according to the transition probabilities of the simplified TKF91 model. This leads to the following transition probabilities between sites:

$P(x \to y)$	$y = \genfrac{}{}{0pt}{}{\mathtt{B}}{\mathtt{B}}$	$y = \genfrac{}{}{0pt}{}{\mathtt{-}}{\mathtt{B}}$	$y = \genfrac{}{}{0pt}{}{\mathtt{B}}{\mathtt{-}}$
$x = \genfrac{}{}{0pt}{}{\mathtt{B}}{\mathtt{B}}$	$1 - \frac{1+t\lambda-e^{-t\lambda}}{\gamma(1+t\lambda)}$	$\frac{t\lambda}{\gamma(1+t\lambda)}$	$\frac{1-e^{-t\lambda}}{\gamma(1+t\lambda)}$
$x = \genfrac{}{}{0pt}{}{\mathtt{-}}{\mathtt{B}}$	$\frac{e^{-t\lambda}}{\gamma(1+t\lambda)}$	$\frac{\gamma(1+t\lambda)-1}{\gamma(1+t\lambda)}$	$\frac{1-e^{-t\lambda}}{\gamma(1+t\lambda)}$
$x = \genfrac{}{}{0pt}{}{\mathtt{B}}{\mathtt{-}}$	$\frac{t\lambda e^{-t\lambda}}{\gamma(1-e^{-t\lambda})(1+t\lambda)}$	$\frac{1-e^{-t\lambda}(1+t\lambda)}{\gamma(1-e^{-t\lambda})(1+t\lambda)}$	$\frac{\gamma(1+t\lambda)-1}{\gamma(1+t\lambda)}$

forbidden in TKF92 and FID:

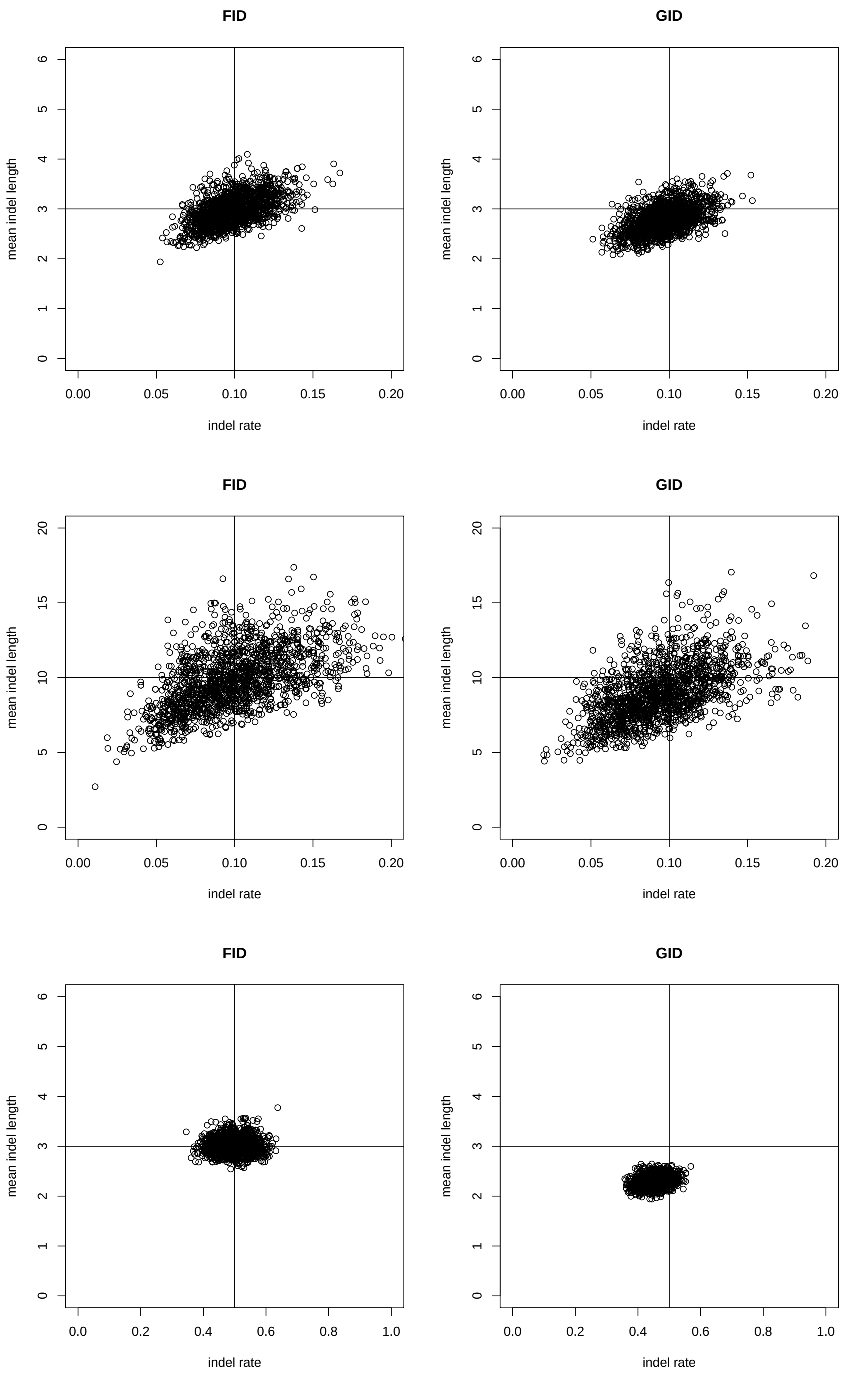


GID Model:

- ↑ this is allowed
- **no** *hidden Markov* structure

Use GID to simulate data and test robustness of FID
Test robustness of ML estimates for mutation rates

- Generate sequence pairs according to FID and GID
- Tell FID-based estimator which positions are homologous
- Are estimates for GID data worse than FID data? (This will be the case only when true parameter values are extreme.)
- Differences should be lower when estimates are based on sequences instead of homology structures.



How good are FID-based methods when GID/"Long Indel Model" is true?

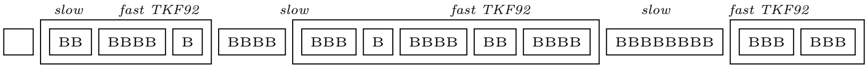
- no problem for parameter estimations (Metzler, 2003)
- alignment accuracy can be decreased (Miklos, Lunter, Holmes, 2004)

Maybe generate mixed-geometric gap-length with different types of fragments.
Along a tree fragmentation may change from edge to edge.

InDel Model for detecting conserved regions

References

[AMP07] A. Arribas-Gil, D. Metzler, J.-L. Plouhinec (2007) Statistical alignment with a sequence evolution model allowing rate heterogeneity along the sequence *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **6**(2): 281-295

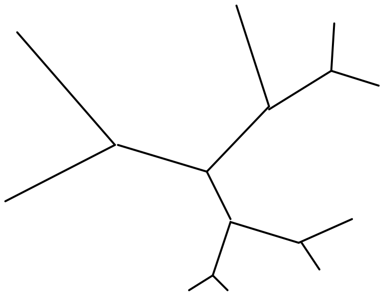


12.3 Multiple Alignments

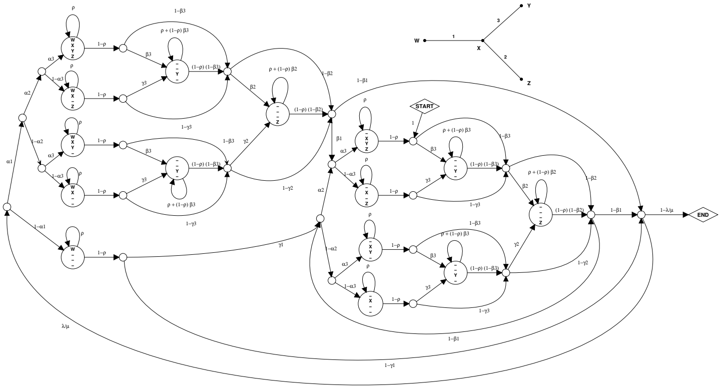
References

[HB01] I. Holmes, W. J. Bruno (2001) Evolutionary HMMs: a Bayesian approach to multiple alignment *Bioinformatics* **17**:803-820.

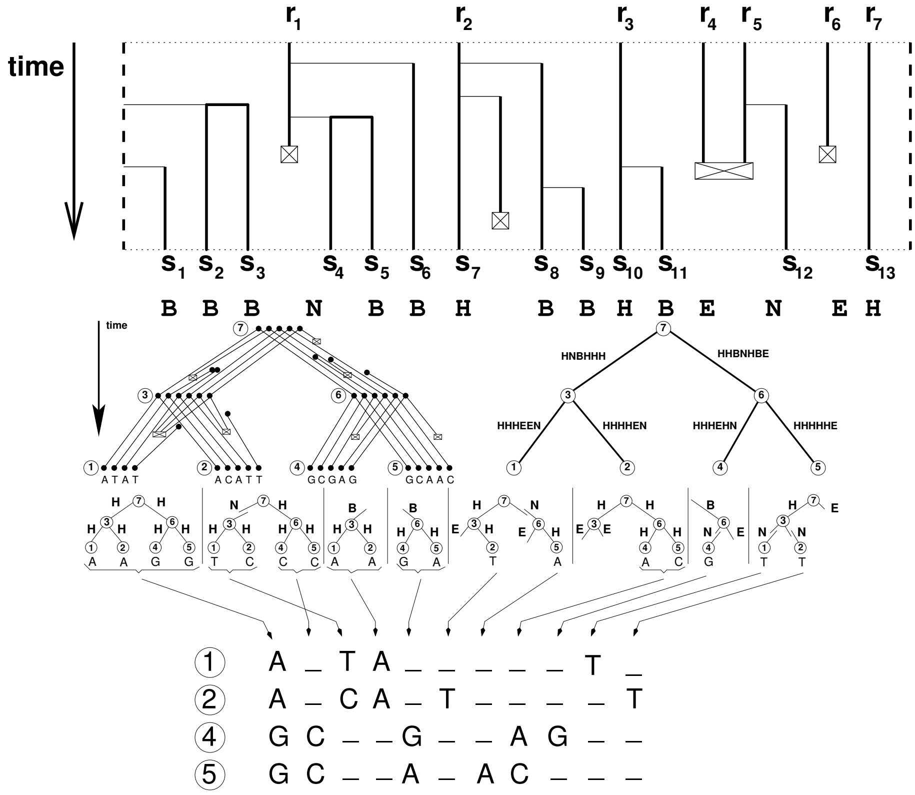
[MFvH05] R. Fleißner, D. Metzler, A. von Haeseler (2005) Simultaneous statistical multiple alignment and phylogeny reconstruction. *Systematic Biology* **54**(4):548-61.



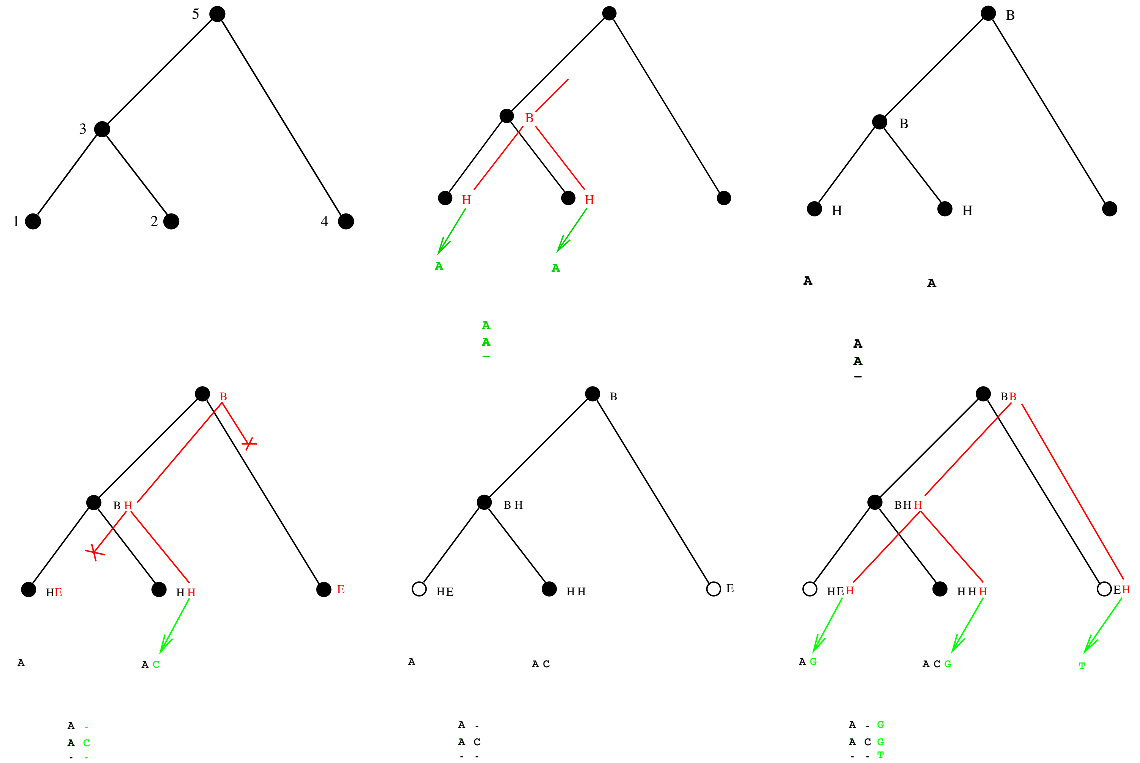
multiple HMM for sampling a sequence given its neighbours

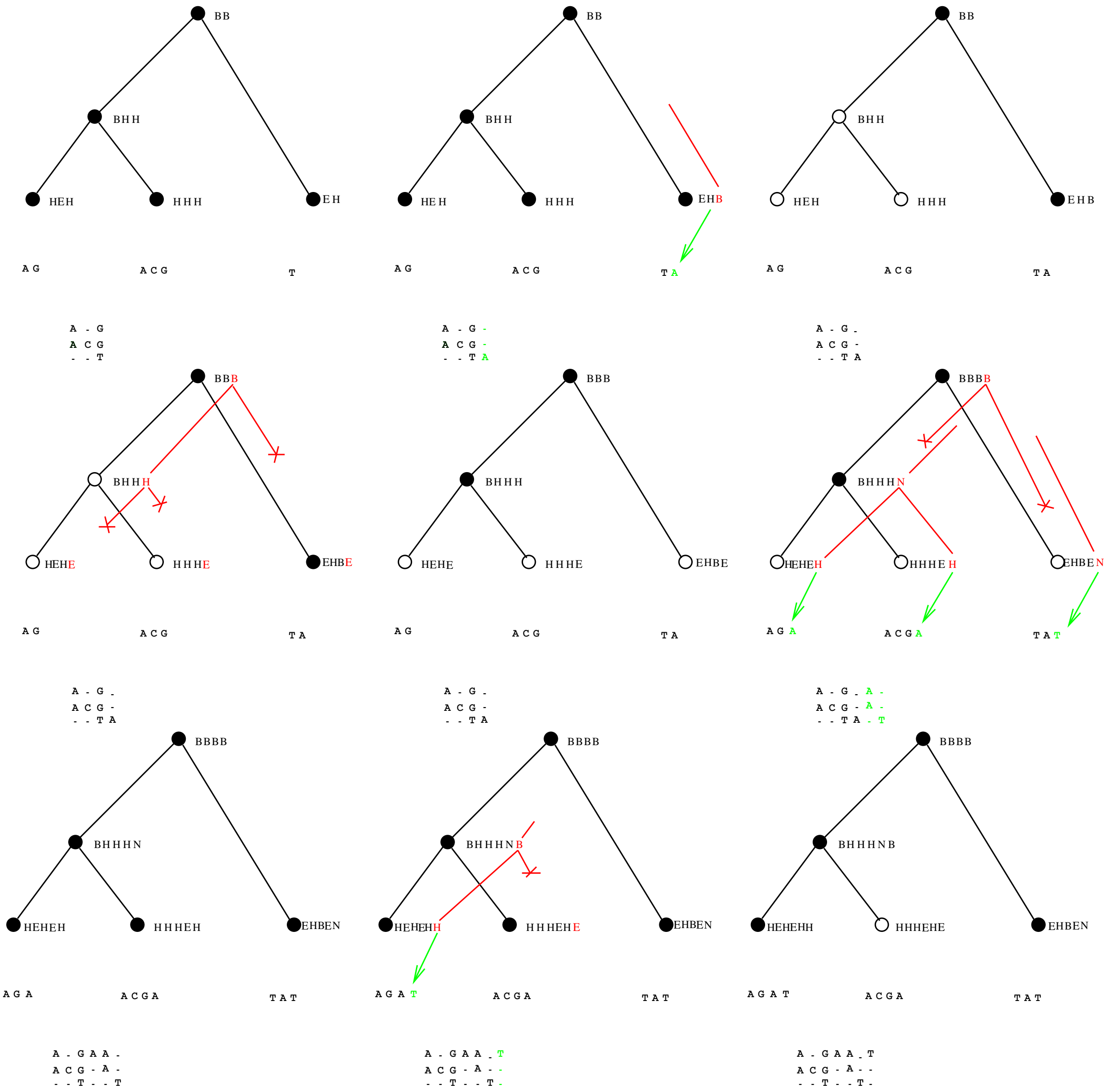


G.A. Lunter, I. Miklós, Y.S. Song, J. Hein (2003) An efficient algorithm for statistical multiple alignment on arbitrary phylogenetic trees. *J. Comp. Biol.* 10(6):869-889.



Tree-Indexed Heirs Line =: TIHL





TKF91: states of hidden Markov chain are the Sets Of Active Nodes (soans).

$$P_{\mathcal{S}}(k) \quad = \sum_{(\mathcal{R},e)\ :\ \mathcal{S}=[\mathcal{R},e]} p(e)q(e)P_{\mathcal{R}}(k-v_e)\vartheta(e,k)$$

where

k : Multi-index of Positions in sequences at leaves

$\mathcal{S}=[\mathcal{R},e]$: tihl e turns soan $\mathcal{S}$ into soan$\mathcal{R}$

$P_{\mathcal{S}}(k)$: Pr(sequences up to k are generated and end there)

$p(e)$ = Pr(indel history of e)

$q(e)$ = Pr(no inserts at nodes in e)

$\vartheta(e,k)$ = Pr(e emits base given in data types at k)

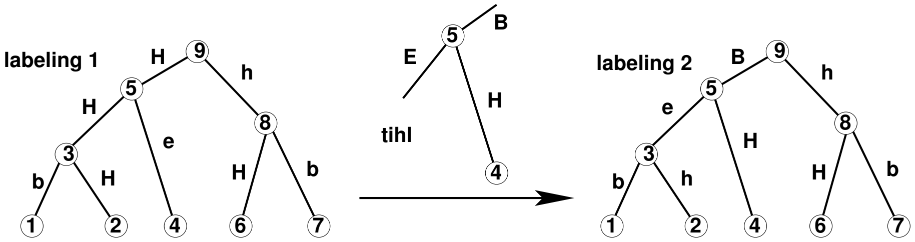
$v_e \in \{0,1\}^n$: indicates postions in leaf-sequences to which e emits

TKF91: states of hidden Markov chain are the Sets Of Active Nodes (soans).

Transfer this to FID or TKF92 (fragmentation may change from edge to edge)

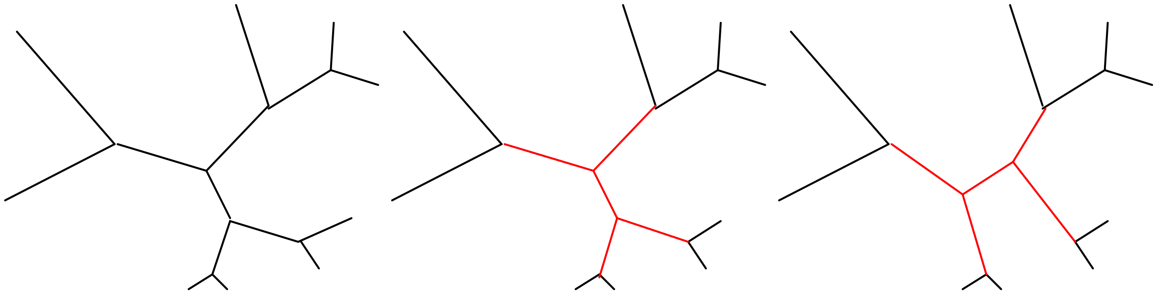
- D. Metzler, R. Fleißner, A. Wakolbinger, A. von Haeseler (2005) Stochastic insertion-deletion processes and statistical sequence alignment.
- D. Metzler, R. Fleißner (2007) Sequence Evolution Models for Simultaneous Alignment and Phylogeny Reconstruction.

state space: edge-labellings with $\{B, H, e, b, h\}$.



tihl = tree indexed heirs line
Example: 3-leaved tree
TKF91: $2^3 = 8$ possible sets of active nodes
TKF92/FID: $5^3 = 125$ possible labellings, 41 of them are relevant

When changing the tree topology...



...keep alignments of exterior sequences fixed. (TKF91: 32 SOANS; FID: 437 relevant labellings)

Why Statistical Alignment is Important

- Over-optimization of alignments can bias your analysis.
- Without statistical alignment methods, like Bayesian tree sampling and bootstrapping will be by far to optimistic about the uncertainty in phylogeny inference.
- Statistical alignment allows you to use the information contributed by insertions and deletions.

12.4 Software for joint estimation of phylogenies and alignments

BAli-Phy

http://www.bali-phy.org/

References

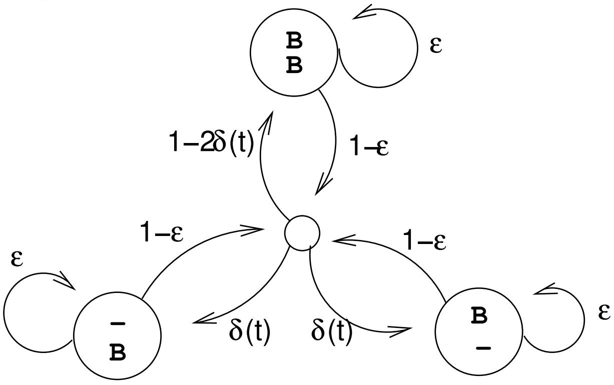
[RS05] B.D. Redelings, M.A. Suchard (2005) Joint Bayesian Estimation of Alignment and Phylogeny *Systematic Biology* **54(3)**:401-418

[SR06] M.A. Suchard, B.D. Redelings (2006) BAli-Phy: simultaneous Bayesian inference of alignment and phylogeny *Bioinformatics* **22**:2047-2048

[RS07] B.D. Redelings, M.A. Suchard (2007) Incorporating indel information into phylogeny estimation for rapidly emerging pathogens. *BMC Evolutionary Biology* **7**:40

pairHMM in BAli-Phy

The alignment consists of a geometically distributed number of fragments. It is generated according to the pairHMM



with
$\delta(t) = 1 - e^{-\lambda t/(1-\varepsilon)}$.

MCMC steps in BAli-Phy

- Parts of the pairwise alignments along branches of the current tree are re-sampled. Felsenstein wildcards are used for the nucleotide or amino acid types, i.e. probability distributions conditioned on the sequences at the tips of the tree.
- SPR steps for updating the tree.
- After an SPR step a pairwise alignment along the new branch is sampled. For efficiency, it keeps the alignments within each of the two partial trees fixed.

Statistical alignment software StatAlign

StatAlign `https://statalign.github.io/`

1. simultaneous statistical alignment and phylogeny reconstruction
2. optionally also simultaneous RNA secondary structure prediction
3. extension StructAlign can account for protein or RNA secondary structure prediction
4. provides a graphical user interface where you can watch the changes in the alignment and the phylogeny

References

[NMLH08] A. Novák, I. Miklós, R. Lyngsø, J. Hein (2008) StatAlign: an extendable software package for joint Bayesian estimation of alignments and evolutionary trees. *Bioinformatics* **24(20)**: 2403–2404

[AEG+13] Arunapuram P, Edvardsson I, Golden M, Anderson JWJ, Novák Á, Sökösd Z and Hein J (2013) StatAlign 2.0: Combining statistical alignment with RNA secondary structure prediction. *Bioinformatics* **29**(5): 654–655

[HCN+19] Herman JL, Challis CJ, Novák Á, Hein J and Schmidler, SC (2014) Simultaneous Bayesian estimation of alignment and phylogeny under a joint model of protein sequence and structure. *Molecular Biology and Evolution* **31**(9): 2251–2266

Some of the things you should be able to explain:

- Why and how can optimized alignments bias a phylogeny analysis?
- Advantages of statistical alignment.
- What is a pairHMM?
- How is dynamic programming used in alignment and why are hidden Markov structures a prerequisite for this?
- What model assumptions equip insertion–deletion models with a hidden Markov structure, also in the case of longer indels?
- Approaches for multiple statistical alignment and their complexity.

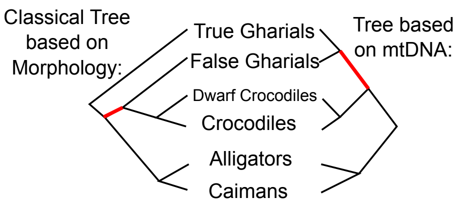
13 Tests for trees and branches (extra material; not part of WS21/22 course)

13.1 The Kishino–Hasegawa (KH) test

Application Example

References

[HHB+03] J. Harshman, C.J. Huddleston, J.P. Bollback, T.J. Parsons, M.J. Braun (2003) True and False Gharials: A Nuclear Gene Phylogeny of Crocodylia *Systematic Biology* **52**:386–402 https://doi.org/10.1080/10635150390197028



- KH test with new data: has one tree significantly higher likelihood than the other?
- Nuclear DNA from 8 crocodylian and 6 avian species
- Result: the tree with the true and false gharials grouped together has significantly higher likelihood for the nuclear data.

The KH test compares two given trees. The null hypothesis is that differences in their likelihoods are only due to "sampling error", i.e. the mutations that randomly occurred at the sites in our dataset.Several versions of the KH test exist, one of them is as follows:

- Given an alignment of length S let for each $k \leq S$ be $\ell_1^{(k)}$ and $\ell_2^{(k)}$ the log-likelihoods of the two trees for the k-th column of the alignment.
- define $\delta_k = \ell_1^{(k)} - \ell_2^{(k)}$
- estimate the variance of all δ^k by $\widehat{\sigma}^2 = \frac{\sum_k (\delta_k - \overline{\delta}_.)^2}{S-1}$, where $\overline{\delta}_.$ is the mean over all δ_k.
- Under the null hypothesis (and model assumptions like independence of sites etc.), the log likelihood-ratio $\ell_1 - \ell_2$ is normally distributed with mean 0 and variance $S \cdot \sigma^2$.

  Hence, reject the null hypothesis on the 5% level if $|\ell_1 - \ell_2| > 1.96 \cdot \sqrt{S}\widehat{\sigma}$

(other variants of the test use log likelihood-ratios of bootstrapped trees instead od site-wise log likelihood-ratios)

Note that the selection of trees to be tested must be independent from the data that is used in the KH test!

If one of the trees has been selected because of its high likelihood for this dataset, the other tree will be rejected too often!

To apply the KH test to more than two trees, some multiple-testing correction is needed.

Basic version does not account for variation between genomic regions (ILS etc..). But could be done for regions like for sites.

13.2 The Shimodaira–Hasegawa (SH) test

Application example

References

[MTW20] Murahwa, A.T., Tshabalala, M., Williamson, A.-L. (2020) Recombination Between High-Risk Human Papillomaviruses and Non-Human Primate Papillomaviruses: Evidence of Ancient Host Switching Among Alphapapillomaviruses *J Mol Evol* **88**: 453–462 `https://doi.org/10.1007/s00239-020-09946-0`

- Genomic regions with evidence of recombination
- Are trees for these regions significantly different from tree for rest of the genome?
- (Be cautious: is evidence of recombination independent of phylogenetic signal?)

References

[1] Shimodaira H, Hasegawa M (1999) Multiple Comparisons of Log-Likelihoods with Applications to Phylogenetic Inference *Molecular Biology and Evolution* **16** (8): 1114–1116 `https://doi.org/10.1093/oxfordjournals.molbev.a026201`

- Assume that a set of trees is given that includes the true tree.
- Again, the choice of the set of trees must be independent of the data. The null hypothesis is that differences in the likelihoods of the trees are only due to “sampling error”.

SH-Test

1. Make R bootstrap samples from the S sites and compute the log likelihood $\ell_{t,r}$ for each tree t in the set and each bootstrapped data set r.
2. $\widetilde{R}_{t,r} := \ell_{t,r} - \frac{1}{R}\sum_{k=1}^{R} \ell_{t,k}$
3. $D_{t,r} := \max_s \widetilde{R}_{s,r} - \widetilde{R}_{t,r}$
4. A 95 % confidence range of trees consists of that trees t for which more than 5 % of the $D_{t,r}$ are larger than $\max_s \ell_s - \ell_t$.

t	ℓ_t
1	ℓ_1
2	ℓ_2
3	ℓ_3
$\vdots$	$\vdots$
T	ℓ_T

r				
1	2	3	$\dots$	R
$\ell_{1,1}$	$\ell_{1,2}$	$\ell_{1,3}$	$\dots$	$\ell_{1,R}$
$\ell_{2,1}$	$\ell_{2,2}$	$\ell_{2,3}$	$\dots$	$\ell_{2,R}$
$\ell_{3,1}$	$\ell_{3,2}$	$\ell_{3,3}$	$\dots$	$\ell_{3,R}$
			$\vdots$	
$\ell_{T,1}$	$\ell_{T,2}$	$\ell_{T,3}$	$\dots$	$\ell_{T,R}$

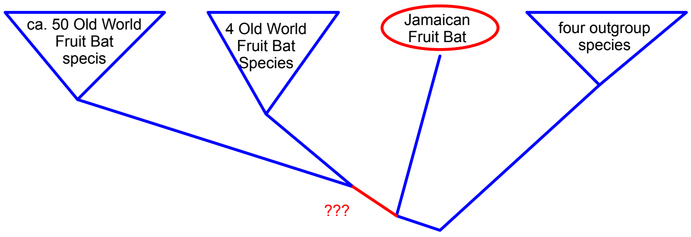
Note that this includes some multiple testing correction, but not completely.

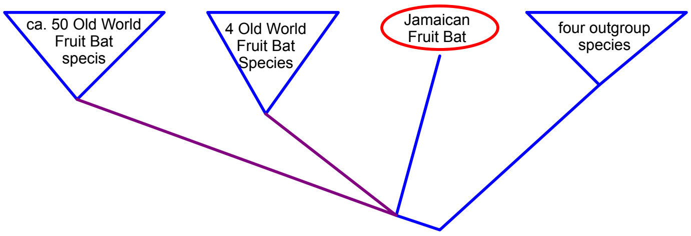
13.3 The SOWH test

Application Example

References

[AGD11] Almeida, F.C., Giannini, N.P., DeSalle, R. et al. (2011) Evolutionary relationships of the old world fruit bats (*Chiroptera*, *Pteropodidae*): Another star phylogeny? *BMC Evol Biol* **11**:281 `https://doi.org/10.1186/1471-2148-11-281`


ca. 50 Old World Fruit Bat specis
4 Old World Fruit Bat Species
Jamaican Fruit Bat
four outgroup species
???


ca. 50 Old World Fruit Bat specis
4 Old World Fruit Bat Species
Jamaican Fruit Bat
four outgroup species

- data: sequenced 8 genes (4 mt, 4 nuclear), complemented by data from genbank
- "The SOWH test confirmed that basal branches' lengths were not different from zero, which points to closely-spaced cladogenesis as the most likely explanation for the poor resolution of the deep pteropodid relationships."
- caution: In general don't draw conclusions from non-significance!
- But they also state: "Simulations suggest that an increase in the amount of sequence data is likely to solve this problem."

References

[GAR00] N. Goldman, J.P. Anderson, A.G. Rodrigo (2000) Likelihood-Based Tests of Topologies in Phylogenetics *Syst. Biol.* **49(4)**: 652–670

[SOWH] D.L. Swofford, G.J. Olsen, P.J. Waddell, D.M. Hillis (1996) Phylogenetic inference in: D.M. Hillis, C. Moritz, B.K. Mabe (eds.) *Molecular Systematics*, Sinauer.

To test whether a tree T_0 can be rejected (H_0: "T_0 is the true tree"), use as a test statistic the difference $\delta = \ell_{ML} - \ell_0$ between the maximum log likelihood ℓ_{ML} and the log likelihood ℓ_0 of T_0.

Simulate many datasets d by parameteric bootstrapping using T_0 and the corresponding estimates of all parameters (mutation rates, branch lengths etc.).

Let $\ell_{0,d}$ be the log likelihood of T_0 based on bootstrap data set d with new estimations for all parameters, and let $\ell_{ML,d}$ be the same maximized over all tree topologies.

Use all $\delta_d = \ell_{ML,d} - \ell_{0,d}$ (for all d) to estimate the distribution of the test statistic δ under the null hypothesis that T_0 is correct.

Reject T_0 on the 5% level if less than 5% of the δ_d are larger than δ.

References

[B02] T.R. Buckley (2002) Model Misspecification and Probabilistic Tests of Topology: Evidence from Empirical Data Sets *Syst. Biol.* **51(3)**: 509–523

Shows examples where SOWH test and posterior probabilities falsely reject too many trees because of using the wrong substitution models. The SH test does not have this problem and rather tends to be too conservative.

Uses real data with phylogeny more or less well known.

Advantage: Realistic because all substitution model used in simulation study are somehow idealized.

Drawbacks: Only a few such datasets are available and results may not be representative. In principle, the assumed phylogenies could still be erroneous.

13.4 Anisimova and Gascuel's approximate Likelihood-Ratio Test (aLRT)

References

[AG06] M. Anisimova, O. Gascuel (2006) Approximate Likelihood-Ratio Test for Branches: A Fast, Accurate, and Powerful Alternative *Syst. Biol.* **55(4)**: 539–552

- We want to show significance of a particular branch in the tree, i.e. the null hypothesis is that this branch has length 0.
- We assume, however, that with any other respect, the topology of the tree is true.
- A likelihood-ratio test:
  - H_0: Length of this branch is 0.

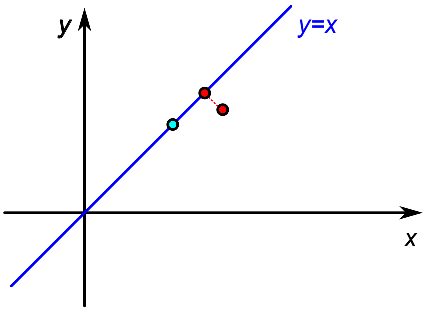
  - H_1: Length of this branch > 0.

- Thus, we have to approximate the distribution of the log likelihood-ratio under the null hypothesis.

Ususally, if we have a model M_1 with $n-d$ parameters nested in a model M_2 with n parameters, then under the null-hypothesis that the data come from the more simple model M_1, the double log likelihood ratio is approximately chisquare-distributed with d degrees of freedom,

$$\mathcal{L}_{M_1}\left(2 \cdot \log \frac{L_D(M_2)}{L_D(M_1)}\right) = \mathcal{L}_{M_1}\left(2 \cdot (\log L_D(M_2) - \log L_D(M_1))\right) \approx \chi^2_d,$$

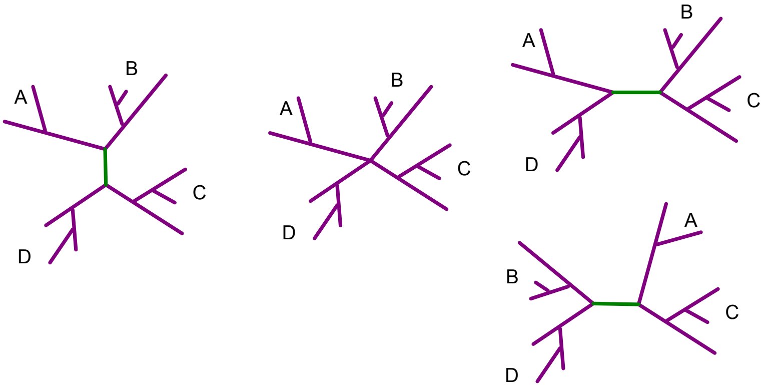
where the likelihood of a model $L_D(M_i)$ is maximum likelihood obtained by optimization over all parameters of the model.



This only works if the model M_1 is in the inner of the model M_2. In our case, the null hypothesis is at the boundary of the more general model, because the branch length 0 is on the boundary of the set of allowed branch lengths.

Therefore, the following correction proposed: The distribution of $2 \cdot (\log L_D(M_2) - \log L_D(M_1))$ is approximated by a distribution that puts weight 0.5 on 0 and half of the density of χ^2_1 on all positive values.

Anisimova, Gascuel (2006): Let ℓ_1 be the log likelihood of the ML tree, ℓ_0 that of the topology with the length of the focal branch removed, and $\ell_2 > \ell_3$ the log likelihoods of the two topologies where the focal branch is removed in an NNI step and (see Figure 1 in Anisimova, Gascuel (2006))[1.5ex]



For more robustness, $2(\ell_1 - \ell_2) \leq 2(\ell_1 - \ell_0)$ is used as a test statistic. (Maybe the idea is that the null hypothesis should be that one of the other fully resolved trees is right.)[1.5ex] The likelihood of a topology is the maximum likelihood of a tree with this topology. Thus, each value ℓ_0, ℓ_2, ℓ_3 needs own optimization of all branch lengths. Here, Anisimova and Gascuel use an approximation by optimizing only the four neighboring branches of the focal branch and its alternative branch in the case of ℓ_2 and ℓ_3.[1.5ex] If the null hypothesis is true, any of the three possible fully resolved topologies can get the highest likelihood. Therefore, a multiple-testing correction is needed. The Bonferroni correction is applied, which means that the α-level is replaced by its third.

Anisimova and Gascuel conclude from simulations that

- Approximate likelihood-ratio test (aLRT, i.e. with optimization over only five branches) has accuracy and power similar to standard LRT.
- aLRT is robust against mild model misspecifications.
- aLRT was slightly more accurate w.r.t. 5% type I error than ML bootstrap.
- In contrast to wide-spread belief, bootstrap was a bit too liberal, i.e. its type I error rate was higher than the significance level.
- Bayesian methods were a bit too conservative in this simulation study.

A Basic concepts from probability theory

This is a very quick overview of some basic concepts from probability theory.

I give a more thorough introduction in my videos and handouts of the statistics course, see
`http://evol.bio.lmu.de/_statgen/StatEES/20SS/Videos/StatEES_3a.mp4`
`http://evol.bio.lmu.de/_statgen/StatEES/20SS/Videos/StatEES_3b.mp4`
`http://evol.bio.lmu.de/_statgen/StatEES/stochbasics_handout.pdf`

And of course there are many excellent textbooks and other internet resources.

A.1 Events and their probabilities

Event in probability is something that takes place with a certain probability, but is not necessarily associated to a certain time or place.

Examples of events:

- $A =$ {The next time I role a dice, it is a six.}
- $B =$ {The next time I role a dice, the result is an even number.}
- There are five segregating sites in this alignment.
- No mutation happend in this genomic region.

Conditional Probability of A, given B:

$$\Pr(A \mid B) = \frac{\Pr(A,B)}{\Pr(B)} = \frac{1/6}{1/2} = \frac{1}{3}$$

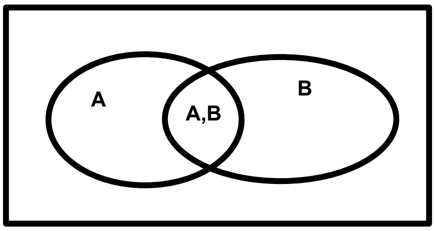
$$\Pr(A,B) = \Pr(B) \cdot \Pr(A \mid B)$$

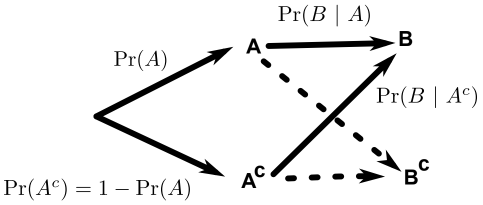
Two events A and B are *stochastically independent* if and only if

$$\Pr(A,B) = \Pr(A) \cdot \Pr(B).$$

If A and B are independent, then $\Pr(B \mid A) = \Pr(B)$ and $\Pr(A \mid B) = \Pr(A)$.

$$\begin{aligned} \Pr(B) &= \Pr(A,B) + \Pr(A^c, B) \\ &= \Pr(A) \cdot \Pr(B \mid A) + \Pr(A^c) \cdot \Pr(B \mid A^c) \end{aligned}$$



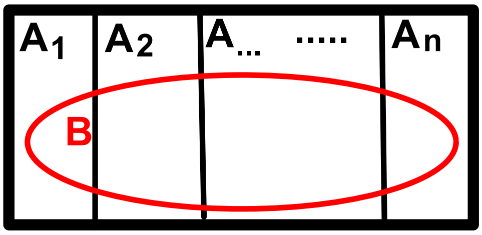


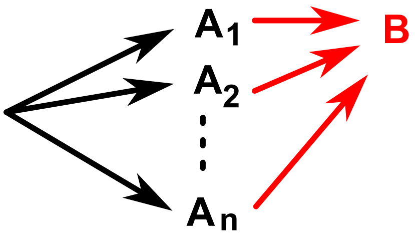
A.2 Law of total probability

Law of total probability

If one and only one of the events $A_1, A_2, \ldots, A_n$ will take place, then

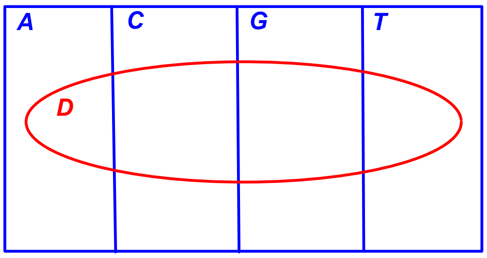
$$\Pr(B) = \sum_{i=1}^{n} \Pr(A_i) \cdot \Pr(B \mid A_i).$$

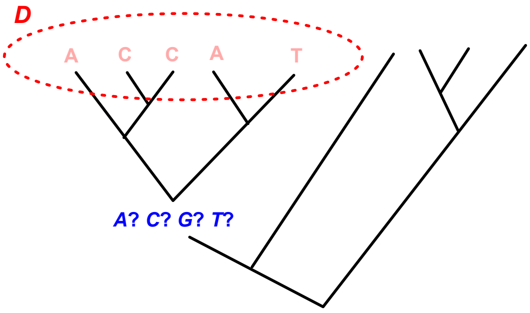




Law of total probability in Felsenstein's prunig algorithm

$$\Pr(D) = \Pr(A)\cdot\Pr(D|A) + \Pr(C)\cdot\Pr(D|C) + \Pr(G)\cdot\Pr(D|G) + \Pr(T)\cdot\Pr(D|T)$$





A.3 Random Variables and their Distributions

Examples for random variables:

Roll a dice two times

X**:** result of the first throw

Y**:** result of the second throw

$S =\ X + Y$

$M =$ number of mutations in some genomic region

$N =$ number of mutations on a branch of a tree

I_U indicator variable of some event U:

$I_U = 1$ if event takes place

$I_U = 0$ if event does not take place

$B =$ the nucleotide type A, C, G or T of the next mutation

The set of possible values of a random variable is calles state space.

Distribution of a random variable

If X is a random variable with discrete state space S (e.g. a finite set like $\{A, C, G, T\}$ or $\mathbb{N}$ or $\mathbb{Z}$), the *distribution* of X is a function that assigns to each subset $U \subset S$ the probability

$$\Pr(X \in U) = \sum_{k \in U} \Pr(X = k).$$

If Z is a random variable with a density f on a continuous state space R (e.g. $\mathbb{R}$ or $\mathbb{R}_+$), the *distribution* of Z is a function that assigns to each measurable subset $U \subset R$ the probability

$$\Pr(Z \in U) = \int_U f(x)dx,$$

where measurable means that the integral is defined.

A.4 Expected Values

If the random variable Y has a discrete state space $S \subset \mathbb{R}$:

$$\mathbb{E}Y = \sum_{k \in S} k \cdot \Pr(Y = k)$$

If Z is a continuous random variable with state space $R \subseteq \mathbb{R}$ and probability density f:

$$\mathbb{E}Z = \int_R x \cdot f(x)\ dx$$

(More generally also works if $S \subseteq \mathbb{R}^n$.) If A and B are random variables and c is a (non-random) number, the expectation value is linear. This means:

$$\begin{aligned} \mathbb{E}(A+B) &= \mathbb{E}(A)+\mathbb{E}(B) \\ \mathbb{E}(c \cdot A) &= c \cdot \mathbb{E}(A) \end{aligned}$$

If A and B are stochastically independent, then

$$\mathbb{E}(A \cdot B) = \mathbb{E}(A) \cdot \mathbb{E}(B),$$

but note that this is in general **not true if A and B are stochastically dependent**.

Law of total expectation

Conditional expectation of a discrete random variable X given an event A:

$$\mathbb{E}(X \mid A) = \sum_k k \cdot \Pr(X = k \mid A).$$

If one and only one of the events $A_1, A_2, \ldots, A_n$ will take place, then

$$\mathbb{E}X = \sum_{i=1}^{n} \Pr(A_i) \cdot \mathbb{E}(X \mid A_i).$$