Probabilistic Models II

Substitution Matrices and Phylogenetic Trees

Scott C.-H. Pegg

BMI203   May 25, 2004

Where we're going today

- Substitution matrices
- Significance of alignment scores
- Phylogenetic trees
- Homework

Substitution Matrices

Formally:

Given an alphabet **A** of symbols, a substitution matrix is an |**A**| x |**A**| matrix where element a_{ij} represents a "score" for the substitution of symbol a_i with symbol a_j.

But what does a "score" represent?

In an evolutionary model, it's a function of the likelihood for one symbol to be replaced by the other.

$$\boxed{S(a, b) = f\,(\,P(a,b)\,)}$$

Typically, f is a function that takes the log of the probability.

Substitution Scores

So how do we get the scores?

We could try to estimate P(a,b) by first principles (size, electrostatics, etc.).

What are the potential problems with this?

- We may not understand these principles well enough
- This may not include other "hidden" selective pressures

Substitution Scores

We can also look at "good" (i.e. trusted) alignments and use them to estimate P(a,b).

There are two fundamental difficulties with this approach:

1. Getting a valid (i.e. random) sample of alignments.

   Sequences (especially proteins) tend to come in families, which may have particular substitution restrictions.

Substitution Scores

2. Substitution is a function of time.

   Not all alignments can be considered equally. The probability of substitution in a pair or sequences that are evolutionarily far apart is higher than in a pair that are close.

distantly related	closely related
TYGGLPA	TYGGLPA
TFAGIPV	TYAGLPA

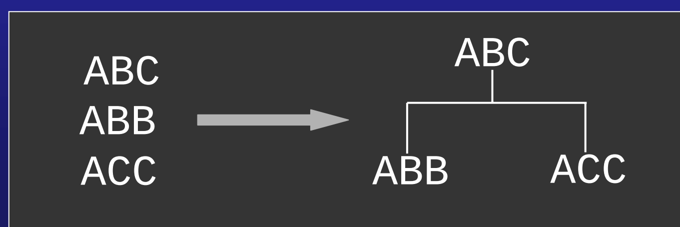
Is P(G,A) the same in both cases?

Dayhoff, et. al. (PAM) Method

To account for the variability between families, they used sequences from 71 different protein families.

Within each family, each pair of sequences was at least 85% sequence identical.

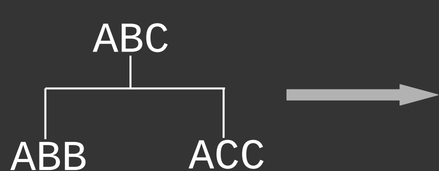
For each family, they built a phylogenetic tree (using a parsimony method).



Dayhoff, et. al. (PAM) Method

From the trees, they built a matrix X, where

x_{ab} = frequency at which symbol a was paired with symbol b between a sequence and its immediate ancestor



$$\begin{array}{c c} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 1 \end{bmatrix} \end{array}$$

Dayhoff, et. al. (PAM) Method

Next, they calculated

$P(b|a)$ = the probability that a is substituted for b

$$= \frac{x_{ab}}{\sum_c x_{ac}}$$

$$= y_{ab}$$

So now there's a matrix Y with probabilities as its elements.

But they still haven't accounted for the differences in evolutionary time.

Dayhoff, et. al. (PAM) Method

They define a substitution matrix to be 1 PAM (point accepted mutation) if the expected number of substitutions in a given sequence is 1%.

The expected number of substitutions is

$$\sum_{a,b} q_a \, q_b \, y_{ab}$$

Where q_x is the frequency of occurrence of x in the sequence

To make this sum equal 0.01, they scaled the values in matrix Y.

Dayhoff, et. al. (PAM) Method

They made a new matrix Z, where

$$z_{ab} = \sigma\, y_{ab}$$

$$z_{aa} = \sigma\, y_{aa} + (1 - \sigma)$$

z_{ab} is now considered P(b|a, t = 1), and matrix Z is denoted as S(1), or a 1PAM matrix.

To extrapolate to longer times, we simply raise S(1) to a power.

So PAM250 = $S(1)^{250}$

Dayhoff, et. al. (PAM) Method

To get the final scoring matrix, we convert from the probabilities in the $S(1)^n$ matrix to scores,

$$S(a,b) = \text{Log}\left[\frac{P(b|a, t = n)}{q_b}\right]$$

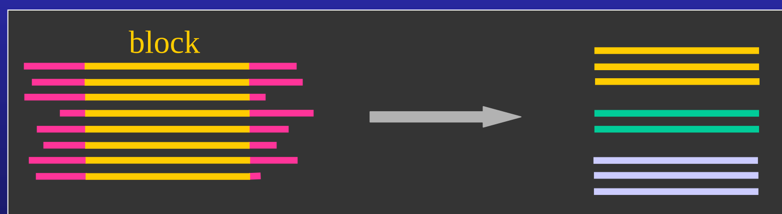
These values are then scaled and rounded to the nearest integer.

PAM250 is scaled by $\frac{3}{\text{Log } 2}$

Henikoff (BLOSUM) Method

Henikoff & Henikoff used sets of multiple alignments from their BLOCKS database.

To account for the variance between familes, they used "blocks" from many different families. Within each block, sequences were clustered by % identity.



A sequence was allowed in a cluster if it was at least *L*% identical to at least one member of the cluster.

Henikoff (BLOSUM) Method

In each BLOCK, they count the number of times symbol a_i from one cluster was matched with symbol a_j from another cluster.

GYAGFPA GFAGFPG GYAAFPA	$f_{GA} = f_{AG} = 3$ $f_{GG} = 2$
AYAGFPA AYAAYPA	$f_{AA} = 1$

Each count is weighted by $\dfrac{1}{n_1\, n_2}$

where n_x = number of sequences in cluster x

Henikoff (BLOSUM) Method

The observed probability of an a_i, a_j pairing is

$$q_{ij} = \frac{f_{ij}}{\sum_{i=1}^{20} \sum_{j=1}^{i} f_{ij}}$$

The background probability of an a_i, a_j pairing is

$$b_{ij} = \begin{cases} p_i \, p_j & \text{for } i = j \\ 2 \, p_i \, p_j & \text{for } i \neq j \end{cases}$$

where

$$p_i = \sum_{j,\, i \neq j} q_{ij} + \frac{q_{ii}}{2}$$

Henikoff (BLOSUM) Method

The matrix score is calculated as

$$S(a_i, a_j) = s_{ij} = 2 \, \mathrm{Log}_2 \frac{q_{ij}}{b_{ij}}$$

and rounded to the nearest integer.

Alignment Scores

How do I know if an alignment score is significant?

We can look at this probabilistically by assuming that symbols occur randomly at all positions according to their background frequencies.

When two random sequences of lengths m and n are compared, the probability of scoring at least S is

$$1 - e^{-Kmne^{-\lambda S}}$$

extreme value distribution

where λ is the unique solution to $\sum_{ij} p_i \, p_j \, e^{\lambda s_{ij}} = 1$

and K is a constant

Alignment Scores

Note that the expected frequency of a substitution is

$$q_{ij} = p_i \, p_j \, e^{\lambda s_{ij}}$$

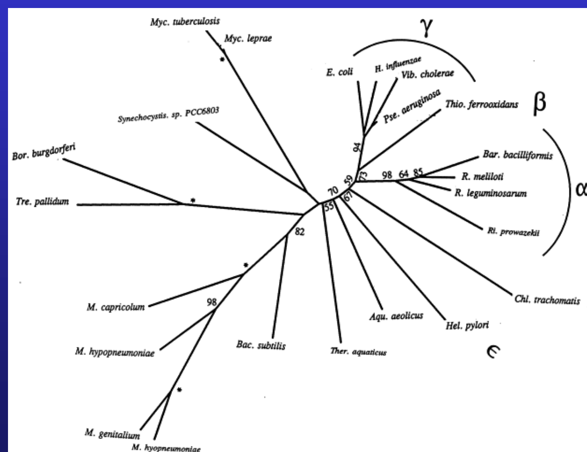
If we rearrange this, we get

$$s_{ij} = \frac{\ln \dfrac{q_{ij}}{p_i \, p_j}}{\lambda}$$

which is the basic log-odds formula used to build substitution matrices.

Phylogenetic Trees

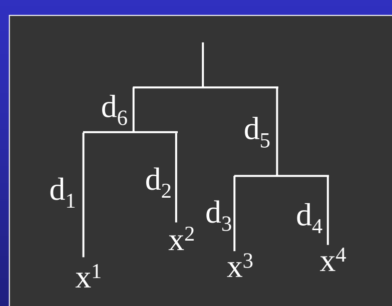
Phylogenetic trees are a way of looking at the evolution and divergence of sequences.



Gupta, R.S. (2000) FEMS Microbiology Reviews, v.24, p. 367-402.

Phylogenetic Trees

We already know of one method to make trees.



Agglomerative hierarchical clustering algorithms.

© 2004 by Scott C.-H. Pegg

Parsimony

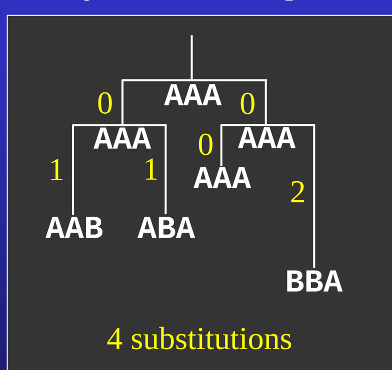
1. Unusual or excessive frugality; extreme economy or stinginess
2. Adoption of the simplest assumption in the formulation of a theory or in the interpretation of data

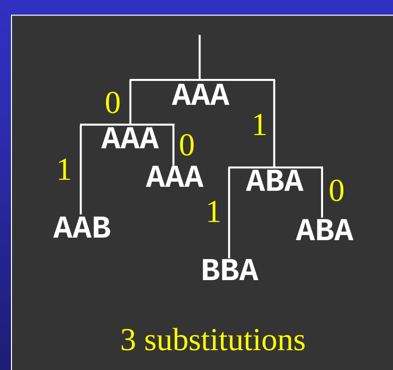
American Heritage Dictionary, 3rd edition

Choose the tree that requires the minimum number of substitutions

Parsimony

Say we have sequences **AAA**, **AAB**, **ABA**, and **BBA**





Cost of a tree is the sum of the substitutions

Each site in the sequence is treated independently

Parsimony Method

We can break the parsimony method into two parts

1. Computing the cost of a given tree
2. Searching the set of all possible trees to find the one with the minimum cost

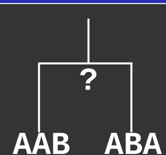
When considering the cost of a tree, we can use the cost of a substitution, S(a,b), instead of just counting the number of substitutions.

weighted parsimony

Cost of a Parsimony Tree

We start with n sequences, each of length L.

We're given a tree with a topology and an assignment of sequences to the n leaves.



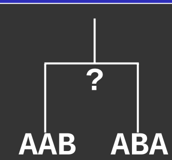
but we don't know the Intermediate sequences

We calculate the minimum cost of a tree by summing the minimum cost at each site of the sequences,

$$\text{Cost of tree} = \sum_{u=1}^{L} \text{cost of tree at position } u$$

Cost of a Parsimony Tree

We can compute the minimum cost at site *u* via a recursive algorithm



but we don't know the Intermediate sequences

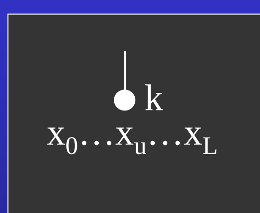
Let $S_k(a)$ = the minimum cost of assigning symbol *a* to node k at site *u*

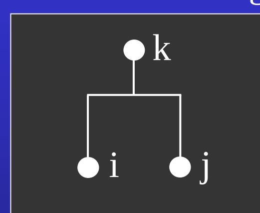
Step 1: Set k = 2n – 1 (the root node)

Step 2: Compute the cost $S_k(a)$ for all *a*

Cost of a Parsimony Tree

A node k is either a leaf node or a branching node





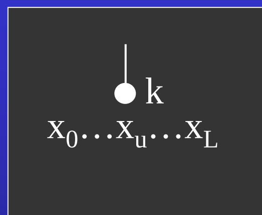
If k is a leaf node, $S_k(a) = 0$ if $a = x_u^k$

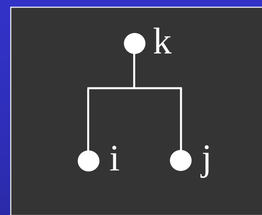
$S_k(a) = \infty$ otherwise

If k is a branching node,

$$S_k(a) = \min_b (S_i(b) + S(a,b)) + \min_b (S_j(b) + S(a,b))$$

Cost of a Parsimony Tree





While k starts at the root, we must calculate the minimum costs for all children i and j first

$$S_k(a) = \min_b \left(S_i(b) + S(a,b)\right) + \min_b \left(S_j(b) + S(a,b)\right)$$

so we're really working from the bottom up.

post-order traversal

Cost of a Parsimony Tree

Step 3: The minimum cost for site u is

$$\min_a S_{2n-1}(a)$$

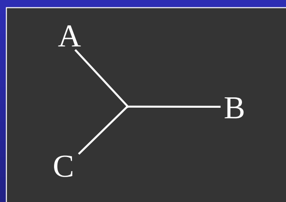
The minimum cost of the entire tree is the sum of the minimum costs for each site (position in the sequence)

$$\sum_u \min_a S_{2n-1}(a)$$

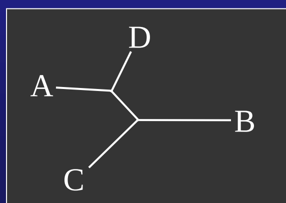
Parsimony Method

Now that we know how to find the cost of a given tree, we want to find the tree with the minimum cost.

For n sequences, there are a lot of trees



how many choices do I have to add a new edge to a binary tree?



now how many?

$3 \cdot 5 \cdot \ldots \cdot (2n\text{-}5)$

$3 \cdot 5 \cdot \ldots \cdot (2n\text{-}3)$ for rooted

$O(n!)$

Parsimony Method

Finding the optimal tree is known to be NP-complete.

One strategy is to use a branch-and-bound algorithm.

Basic idea:

Build trees systematically, but abandon the construction of a tree when adding one more node would exceed the cost of the cheapest tree already constructed.

Clever starting trees and enumeration can help.

Guarantees the optimal tree, but often runs too slowly for use large numbers of sequences.

The Probabilistic Approach

What we ultimate want to compute is $P(X \mid T,D)$

where X = the set of sequences
T = the tree topology
D = the lengths (distances) of the edges

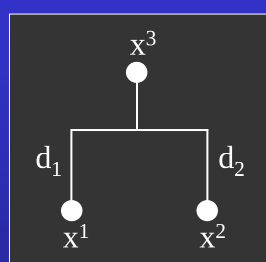
Or, in the Bayesian view, $P(T,D \mid X)$

We start by defining

$P(x \mid y, d)$ = the probability that sequence y changes to sequence x along an edge length d

Assuming independence of the nodes, $P(X \mid T,D)$ is the product of $P(x \mid y, d)$ for each node.

The Probabilistic Approach



$$P(x^1,x^2,x^3 \mid T,D) = P(x^1 \mid x^3, d_1)\, P(x^2 \mid x^3, d_2)\, P(x^3)$$

We don't know x^3 exactly, so we have sum over all possible x^3's,

$$P(x^1,x^2 \mid T,D) = \sum_{x^3} P(x^1 \mid x^3, d_1)\, P(x^2 \mid x^3, d_2)\, P(x^3)$$

The Probabilistic Approach

Given that we can calculate P(x | y, d), we can calculate the likelihood of a given tree.

We now want to choose the tree with the highest value of this likelihood.

This requires searching over two spaces simultaneously

1. All possible topologies T
2. For each topology, all possible edge lengths D

We can search topologies using brand-and-bound.

We can search edge lengths using a variety of optimization methods.

The Probabilistic Approach

We can search both at once by using sampling methods.

Basic idea:

Sample randomly from the space of all possible trees according to the posterior distribution

$$P(T,D \mid X) = \frac{P(X \mid T,D)\, P(T,D)}{P(X)}$$

The frequency of properties in the sample will converge to the posterior probability as the number of samples increases.

Metropolis Method

Here's an adaptation of the Metropolis method by Mau et. al.

Given: A procedure *f* that will generate a tree $(\tilde{T}, \tilde{D})$ randomly when given tree (T, D) as input by sampling from a proposed distribution.

Let $P_1 = P(T,D \mid X)$ and $P_2 = P(\tilde{T},\tilde{D} \mid X)$

Step 1: Build a random tree (T, D) and calculate P_1

Step 2: Build a new tree $f(T, D) = (\tilde{T}, \tilde{D})$ and calculate P_2

Metropolis Method

Step 3: Accept the new tree if $P_2 > P_1$

If $P_2 < P_1$, accept P_2 with probability $\frac{P_2}{P_1}$

If P_2 is accepted, it represents a sampled tree.
Otherwise, P_1 represents a sampled tree.

Step 4: If an appropriate number of samples have been taken, stop.
Else, go to Step 2.

The Probabilistic Approach

Any probabilistic approach requires that we can calculate $P(X \mid T, D)$.

$$P(X \mid T, D) = \prod_{\text{nodes}} P(x \mid y, d)$$

This requires an ability to calculate $P(x \mid y, d)$.

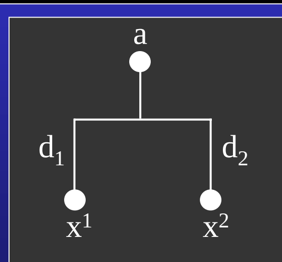
We start by making some assumptions:

1. Evolution works only via substitutions.
2. Substitutions at each site in a sequence are independent.
3. Substitutions follow a first-order Markov process.
4. The Markov process is identical at each site.

The Probabilistic Approach

Our assumptions allow us to compute

$$\boxed{P(x \mid y, d) = \prod_{u=1}^{L} P(x_u \mid y_u, d)}$$



$$\boxed{P(x_u^1, x_u^2, a_u \mid T, d_1, d_2) = P(x_u^1 \mid a_u, d_1)\, P(x_u^2 \mid a_u, d_2)\, P(a_u)}$$

ⓒ 2004 by Scott C.-H. Pegg

The Probabilistic Approach

We don't know the sequence of a exactly, so we sum over all possibilities

$$P(x_u^1, x_u^2 \mid T, d_1, d_2) = \sum_a q_a P(x_u^1 \mid a_u, d_1) P(x_u^2 \mid a_u, d_2)$$

and calculate the likelihood of the tree as

$$P(x^1, x^2 \mid T, d_1, d_2) = \prod_{u=1}^{L} P(x_u^1, x_u^2 \mid T, d_1, d_2)$$

This is usually done using a recursive algorithm very similar to the one used in parsimony cost evaluation.

The Probabilistic Approach

Note that we still have distances in our probability calculations.

$$P(x_u^1, x_u^2 \mid T, d_1, d_2) = \sum_a q_a P(x_u^1 \mid a_u, d_1) P(x_u^2 \mid a_u, d_2)$$

This requires probabilities of substitutions that depend on time. In general, we want

$$P(a \mid c, t+s) = \sum_b P(a \mid b, t) P(b \mid c, s)$$

It also makes things a bit easier if

$$P(a \mid b, t) = P(b \mid a, t)$$

The Probabilistic Approach

So in general, we'd like to have an |**A**| x |**A**| matrix of probabilities that's symmetric, and for which

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

Where do we get one of these?

PAM (and other) matricies

The Probabilistic Approach

We started by making some assumptions:

1. Evolution works only via substitutions.
2. Substitutions at each site in a sequence are independent.
3. Substitutions follow a first-order Markov process.
4. The Markov process is identical at each site.

Given what we know about the process of evolution, these assumptions seem pretty lousy.

How can we relax them?

The Probabilistic Approach

1. Evolution works only via substitutions.

   We can add a gap symbol to the alphabet to allow deletions and insertions.

2. Substitutions at each site in a sequence are independent.

   ?

3. Substitutions follow a first-order Markov process.

   ?

4. The Markov process is identical at each site.

   We could use a different scoring matrix at each site.

Phylogenetic Trees

So now we've seen 3 different methods of creating a phylogenetic tree.

1. Distance methods (agg. hierarchical clustering)

   Fastest of the three, so it's good for lots of sequences, but can build incorrect topologies.

2. Parsimony

   Includes assumptions about the evolutionary process to make better trees, but can be very slow.

3. Probabilistic methods (maximum likelihood, sampling)

   ML is slow, but sampling methods can provide the likelihood of particular sub-topologies and distances in trees.

Takeaway

- Substitution matrices
  - PAM, Henikoff
- Significance of alignment scores
- Phylogenetic trees
  - parsimony, probabilistic methods