Introduction to Computational Biology
Lecture # 21: Improvements on the evolutionary model
Barak Segal
12/5/2008

1 Previous lectures review

In previous lectures we discussed evolutionary models on series. The process was described as a tree whose height
describes the time frame as can be seen in Figure 1.

![Figure 1: An evolutionary tree. The height of the tree corresponds to the time passed. Each internal node is a branching
event. The leaves are the present day species i.e our observations.](image)

We also made several assumptions in the model, these assumptions will be described later on.

In order to describe the model, we require several components:

- **T** - Tree topology - the structure of the tree, which nodes branched from which and so on.
- **τ** - Branch lengths - evolutionary time on each branch.
- **R** - Rate matrix - $R_{ab}$ is the rate $a$ changes to $b$.

As mentioned before, the leaves describe the present day observations. We’ll denote these observations as: $X_1, \ldots, X_n$, and the internal tree nodes as $X_{n+1}, \ldots, X_N$ (where $N$ equals $2n+1$). We wanted to describe the proba-
bility to see a certain sequence of leaves. Thus what we were seeking was:

$$P(X_1, \ldots, X_n) = \sum_{X_{n+1}} \ldots \sum_{X_N} P(X_1, \ldots, X_n, X_{n+1}, \ldots, X_N)$$

Let $b$ be the root (pervious annotation for the root was $r$, but we’ll change it to $b$ so we won’t get mixed up with later annotations), and $p(i)$ the parent of $i$. Under our assumptions, we gathered that:

$$P(X_1, \ldots, X_N) = P(X_b) \prod_{i \neq b} P(X_i \mid X_{p(i)})$$

(1)

We had used the Upwards and Downwards algorithms to answer several questions regarding the tree model, and
used the Expectation Maximization method to estimate the tree branches lengths ($\tau$).
2 Further questions

1. How to reconstruct the tree topology - T.
2. How to learn the rate matrix - R.
3. How to best reconsider and fix some of the effects of our assumptions.

We will now deal with the third problem. These are the assumptions we made in our model:

- Independence between positions.
- The process is a Markovian process - each node depends only on its parent node.
- The process is a homogenous process - the probability for a certain change to occur in \( \Delta t \) time span is independent of when the change has started.
- Same rate matrix for all positions. This is not always correct, for example: there is a different rate of change in amino acids for an alpha helix amino acid and a beta sheet amino acid.
- Rate is homogenous between positions. This is also not always correct since some areas in the protein/DNA are more conserved, and therefore have more selection and less opportunity for change.

The last two assumptions are actually sub assumptions that are derived from the first three. Let us explore them further.

3 Evolutionary Rate considerations

3.1 Affect of conservation on the evolutionary rate

A protein/DNA consists of areas which are highly variable and areas that are somewhat conserved and areas that are highly conserved such as the catalytic site in proteins, as can be seen in Figure 2.

Figure 2: A protein can be hypothetically divided into super conserved sections, conserved sections and variable sections.

What can we learn from the conserved or variable sites? In very close species, the conserved and highly conserved sites do not change, but the variable sites are actually more informative since they are part of what differentiate between the species, and may help determine the structure of sub trees. However, under distant areas in the tree, the variable sites only produce noise due to their fast mutation rate. This is where the conserved sites come in to provide information. To summarize, it can be said that areas with different selection rates tell a story on a different time scale.

3.2 ‘Fixing’ our model

We would like to complicate our model a little more in order to incorporate more data on the conservation. In order to fix the bad outcomes of the ‘Homogenous rate between positions’ assumption, we can add a new parameter to the previous three parameters that represented the tree (the three being: T, R, \( \tau \)):

Definition 3.1 Position rate vector: \( r \) s.t: \( \forall l \ r_l = \text{rate of position } l \)

Now, if we look at the equation (1), we can add the new information, and receive:

\[
P(X_1, \ldots, X_N | r) = P(X_b) \prod_{i \neq b} P(X_i | X_{p(i)}) e^{r_{i,p(i)} r_{i,p(i)} - r_i r_{p(i)}}
\]  \hspace{1cm} (2)
Note that now \( P(X_i \mid X_{p(i)}) \) is equal to \( e^{\tau_{i,p(i)} r_l R_{i,p(i)}} \).

So what does different values of \( r_l \) mean?
When \( r_l = 1 \) then the rate is equal to the global rate.
When \( r_l = 2 \) then the mutation rate is larger, the branches are longer (since they are given more opportunity to change).
When \( r_l = 0.5 \) then the mutation rate is smaller, the branches are shorter (since they are given less opportunity to change).

There are still some problems in this model:

- The \( \tau_{i,p(i)} \) and \( r \) parameters are always multiplied together. Therefore, if \( \tau \), for example, is multiplied by 2 than by multiplying \( r \) by 2 will get us the same results. Therefore, we have infinite solutions for our problem. This can be solved by making sure the rate’s expectation is 1, that is, setting the most common event as the event of no change in the site. Thus, when later handling its distribution, we will fix the mean to 1, and have only one solution.

- Up until now, the number of free parameters was dependent on the species amount (that is, our leaves), and the positions were independent of each other. Now we have \( l \) new parameters (composing \( r \)), one for each position. When learning these parameters with a too small amount of species, there might be overfitting.

How can we use \( r \) correctly to solve the latter problem? Let’s look at possible solutions:

1. Regulation - set a preliminary guess for \( r \).
2. Treat \( r \) as a hidden variable, and not as a parameter, and then we can change the likelihood probability equation to:

\[
P(X_1, \ldots, X_n, r) = \sum_r \sum_{X_{n+1}} \cdots \sum_{X_N} P(X_1, \ldots, X_n, X_{n+1}, \ldots, X_N, r)
\]

\[
P(X_1, \ldots, X_n, r) = P(r) \prod_{i \neq b} P(X_i \mid X_{p(i)}, r)
\]

3.3 Adding further complications to the model

Say we are not content with only \( r \), and wish to add more information that will further complicate our model, but will probably make it more accurate biologically. We’ll add structural information for each position, that represents which secondary structure the position belong to (alpha helix, beta-sheet, loop, etc). This will be denoted using \( s \), and each value of \( s \) fits a certain rate matrix which will be denoted as \( R_s \). When adding \( s \), we receive:

\[
P(X_1, \ldots, X_n, r, s) = P(r) P(s) P(X_b) \prod_{i \neq b} P(X_i \mid X_{p(i)}, r, s)
\]

3.4 Defining the priors

How can we define the priors, \( P(r) \) and \( P(s) \)?

We can define \( P(s) \) as a multinomial distribution, with the parameter amount defined as the number of possible values \( s \) can receive minus one (for example, if \( s \) can be: alpha helix, beta-sheet, loop, then it has 3 values it can receive and the parameter amount is 3-1=2).

\( P(r) \), on the other hand, is more problematic. We know that \( r \) is continuous and non-negative. Therefore, we would like to define it as a function in order to perform an integral on \( r \) to get all of its possible values. Usually, if our parameter is continues and with no constraints, we would use the Gaussian distribution, but this is not the case. Therefore, we would use the Gamma distribution, which is quite similar to Gaussian distribution, but realizes our non negativity constraint.

Generally, the Gamma distribution is defined as:
**Definition 3.2** Gamma Distribution

\[ X \sim \Gamma(\alpha; \beta) \]

\[ P(X = r) = r^{\alpha-1} \beta^\alpha e^{-\beta r} / \Gamma(\alpha) \]

*Where \( \Gamma \) is the Gamma function.*

The distributions of this family are also (like the Gaussian distribution) determined by two parameters - mean and variance. We require the mean to be 1. This requirement causes \( \beta \) to be equal to \( \alpha \). There remains only one parameter \( \alpha \), that represents the variance and determines the distribution. Thus, we have to evaluate only one parameter for all of the rate coefficients \( r \) of all the sites in the sequence. We get the following:

**Definition 3.3** Gamma Distribution (mean == 1)

\[ X \sim \Gamma(\alpha) \]

\[ P(X = r) = r^{\alpha-1} \alpha^\alpha e^{-\alpha r} / \Gamma(\alpha) \]

Figure 3: Gamma Distribution - several graphs with different parameters (taken from Wikipedia). In relation to the definition 3.2, \( \alpha = k \) and \( \beta = \theta \). The mean is \( k \theta \) and the variance is \( k \theta^2 \).

In order to calculate the probability we need to use integration. For that we’ll use numerical integration.

### 3.5 Numerical Integration

The integral of equation 2 can be approximated using numeric methods. The definite integral is defined in the analytical sense as:

\[ \int_{a}^{b} f(x) dx := F(b) - F(a) \]

Where \( F(x) \) is defined as \( F'(x) = f(x) \). Finding \( F(x) \) is not always possible or easy, so we can use a numerical approximation. We can approximate the value of the integral by dividing the relevant range to many short segments,
sampling a point from each one and summing up the area of the rectangles (see Figure 4A). If we divide the range into n segments, then the integral is calculated as:

\[
\int_a^b f(x) \, dx = \frac{b-a}{n} \sum_{i=1}^n f(a + \left(\frac{b-a}{n}\right)(i - \frac{1}{2}))
\]

Note: we use the middle of the segment as the step’s height.

Of course, there is a trade-off between the number of the sampled points and the accuracy of the approximation. A bit more accurate approximation is achieved by using sum trapezes instead of rectangles (see Figure 4B). This is called the Simpson approximation and is calculated as:

\[
\int_a^b f(x) \, dx = \sum_{i=1}^n (x_i - x_{i-1}) \left( \frac{f(x_i) + f(x_{i-1})}{2} \right)
\]

Basically, the numeric integration works in the following manner:

\[
\int_a^b f(x) \, dx = \sum_{i=1}^n W_i f(x_i)
\]

Where \( W_i \) is a weight given to \( f(x_i) \).

Figure 4: An illustration of numerical integration: (A) sum of rectangles, (B) sum of trapezes.

### 3.6 Using the integration

Now we can use numeric integration to evaluate the probability of the observations (we’ll ignore \( s \) for now):

\[
P(X_1, \ldots, X_n) = \int_0^\infty \sum_{X_{n+1}} \ldots \sum_{X_N} P(X_1, \ldots, X_n, X_{n+1}, \ldots, X_N, r) \, dr \\ 
\approx \sum_j \sum_{X_{n+1}} \ldots \sum_{X_N} P(X_1, \ldots, X_n, X_{n+1}, \ldots, X_N, r_j) W_j
\]

Note: performing numeric integration on Gamma allows us to choose a good segmentation on \( r \).

For every value of \( r \) we run the Upwards and Downwards (note that \( r_j \) represents a value for the \( r \) vector). We perform dynamic programming for all values of \( r \):

\[
P(X_1, \ldots, X_n) = \sum_j \frac{P(X_1, \ldots, X_n, r_j)}{P_{r_j}} W_j
\]
The complexity is multiplied by a constant. If we also add $s$ than the complexity again is multiplied by a constant.

We can now also try to evaluate the rate of specific positions using our observations and thus learn about the conservation of different positions. Using this information we can decide, for example, if a certain SNP is evolutionary conserved, and thus conclude if a certain change in it might cause damage or disease.

4 Dealing with the position independence assumption

We assumed that the positions are independent of each other. This is of course a false assumption (for example, if a position is a part of an alpha helix, than it's quite probable that the positions next to it are also part of an alpha helix). Therefore, it is safe to claim that both $s$ is not position wise independent, and the same goes for $r$.

When we discussed HMMs we had a transition probability between positions. We can define an HMM in here as well. Now $s$ is our hidden parameter (analogues to $H$ when we learned HMM) and our observations are the MSA's (multiple sequence alignment) of the sequences (what we treated as observations before). We need to use the following two elements in the model: $P(s_{j+1}|s_j), P(O_j|s_j)$ (Where $O_j$ is $x_1[j] \ldots x_n[j]$). This HMM is not solved by forward and backwards, but by Upwards and Downwards.