Likelihood computation under TKF91 model for a 3 taxa tree

Problem

Given a 3 taxa phylogenetic tree (Figure 1), where **Ω** denotes the root node the model parameters **θ**(substitution rates, insertion rate per link **λ** and deletion rate per link **µ**) and an MSA **α**, the proposed algorithm calculates the likelihood under the TKF91 model. We assume that the root node is the central node and 3 leaves were generated from it.

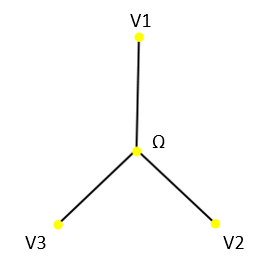


Figure 1: 3 taxa tree

Solution

In this section, I will explain the Likelihoood algorithm in detail. The detailed description is given for a star tree with three sequences, the generalization is straightforward.

Let us consider a star tree $\tau$ rooted at \omega with leaves $v\_1, v\_2, v\_3$. The associated divergence time with each edge be $t\_1, t\_2, t\_3$. For sequences $A\_1, A\_2, A\_3$ at leaves $\alpha$ be an alignment and $\theta$ represent the model parameters (substitution rates, insertion rate per link λ and deletion rate per link µ). The alignment $\alpha(A\_1, A\_2, A\_3)$ represents the fate of the ancestral sequence. The joint probability calculation for observing $ A\_1, A\_2, A\_3$ at leaves $v\_1, v\_2, v\_3$ takes into consideration every possible character on each ancestral site. Since ancestral characters are not represented in $\alpha(A\_1, A\_2, A\_3)$, an alignment involves a possible fate of several ancestral sequences. Therefore, it is important that the approach marginalizes over all the possible ancestral sequences. The probability of specific transition path represented by alignment \alpha, $P(\alpha|\theta) can be decomposed into two components, $P(\alpha’|\theta)$(transition probability of insertion-deletion and ) P(\alpha|\alpha’, \theta) the substitution probability of the given alignment \alpha.[TKF paper]. In other words,

$P(\alpha|\theta)= P(\alpha,\alpha’|\theta)= P( \alpha’|\theta) P(\alpha|\alpha’, \theta)$

$P(\alpha’|\theta)$ calculates the geometric probability, $\gamma\_N(\lambda,\mu)$ that the ancestral sequence has N links and considers only the structure of MSA $\alpha’$, insertion and deletions, converting the MSA into Character\gap state (#/-) while $P(\alpha|\alpha’,\theta)$ accounts for the substitution process given the MSA structure and model parameters. If the fate of one single ancestral character is represented as a single block, for an alignment \alpha with ancestral sequence having ‘i’ ancestral characters will constitute of ‘i’ blocks and $P(\apha’|\theta)$ can be calculated as

$P(\alpha’|\theta)= \gamma\_N(\lambda,\mu)$ \prod\_{i}LK\_i $

Where LK\_i is the is the likelihood of block i. The probabilities for the different possible fates of a single link (character) evolving along a tree branch of length t are given by eqs in section

Survived ancestral links are always aligned to their respective homologous site in the ancestral sequence and their descendants are placed to their right side. A gap symbol is introduced if

• an ancestral link has died;

• an ancestral link has a different number of descendants on different branches.

In each sequence the newborn links follow their parents immediately. Thus both homologous and non-homologous scenarios are aligned in this type of alignment. However, homologous and non-homologous are distinguished both in the alignment and in the probability calculations. Homologous characters are associated to an ancestral mortal link (\*) and the probability calculations are different in the homologous and non-homologous case. For example, Fig. 1 shows that the immortal link (o) has no descendant mortal link (\*) in *A*1 and has one mortal link in *A*2 and *A*3. The first ancestral mortal link survives in each sequence and has a newborn link in *A*2. From the second link to the next-to-the-last one all links died out. The last ancestral link survived in sequence *A*2 and *A*3, and left one newborn link both in *A*1 and *A*2. The probability of this special transition is

The developed algorithm take advantage of the homologous characteristics to find the obvious mortal links present in all possible ancestral sequences. This information could be utilised to block the alignment. Later the likelihood for each block is solved locally considering the possible ancestral sequences for each block. Finally their product will give us the final likelihood.

Accomodating all scenarios:

In this section we will now verify whether blocking the MSA and calculating the LK considering block Lks considering all ancestral scenarios in each block will give the same LK as considering all ancestral sequences possible for the whole MSA. Let us consider an MSA with 3 blocks and (a,b), (1,2,3), (x,y) be the respective set of all possible ancestral scenarios for block1, block2 and block3. If L\_i(m) gives the likelihood underTKF91 block i where m is a possible ancestral scenario. The total Likelihood is given by

Possible combinations:

L(MSA) = L (a1x + a1y + a2x + a2y + a3x + a3y + b1x + b1y + b2x + b2y + b3x + b3y)

= L (a1(x + y) + a2(x + y) + a3(x + y) + b1(x + y) + b2(x + y) + b3(x + y))

= L ((a + b) (1 + 2 + 3) (x + y))

= L (a + b) \* L (1 + 2 + 3) \* L (x + y)

Where L(a1x) is the MSA likelihood for ancestral sequence a1x. Hence it is evident from the final eq that blocking the MSA and solving for each block will give the LK of the MSA for all possible ancestral sequences.

The current algorithm calculates the likelihood in the following way:

1. Block the MSA into immortal and mortal blocks
2. For each block find all possible ancestral scenarios
3. Compute the block likelihood
4. Obtain P(\alpha|\theta) by multiplying Block likelihood

In the following sections we will look into each steps of the algorithm in detail.

# Blocking the MSA

The initial task is to seperate the MSA into blocks. In TKF91 it is convinient to handle the MSA as blocks as the blocks represents the fate of each ancestral link . Likelihood is computed for each blocks and finally multiplied together to obtain the total likelihood. In order to block an MSA we first apply the basic rule when a column in MSA contains atleast 2 characters which means there were atleast 2 survivals at the leaves it is obvious that they are homologus and must be from a common ancestor. Thus there should be an ancestor at the root. Once we find all such homologous columns we perform the blocking as below

1. Let START = end column index of the MSA and POSITION= end column index of the MSA.
2. Check START= first column index of the MSA. If true go to 3 else to 4
3. Extract the columns from START to POSITION of the MSA and this is stored as a block. Go to 8
4. Check Homologous condition. If True go to 5 else to 6
5. we extract the columns from START to POSITION of the MSA and this is stored as a block. Set POSITION= START-1
6. START=START-1
7. Go to 2
8. Stop

Block 1

Block 2

Block 3

Block 4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ω | o |  | \* |  |  |  | \* | \* |
|  | o − | C \* | T \* | G \* | − | − | A \* | − |
|  | o − | − | C \* | − | A \* | − | A \* | C \* |
|  | o A \* | − | C \* | − | − | C \* | − | G \* |

For example, let us consider the MSA in figure. This can be blocked into

Block1

|  |  |
| --- | --- |
| o − | C \* |
| o − | − |
| o A \* | − |

Block2

|  |  |  |  |
| --- | --- | --- | --- |
| T \* | G \* | − | − |
| C \* | − | A \* | − |
| C \* | − | − | C \* |

Block3

|  |
| --- |
| A \* |
| A \* |
| − |

Block4

|  |
| --- |
| − |
| C \* |
| G \* |

# Likelihood Computation of a block

In general, for any MSA there are 2 types of block, immortal and mortal. Immortal block represents the fate of the immortal link present in the ancestral sequence and the mortal blocks represent the fate of the mortal links present the ancestral sequence. In our approach we handle them separately. First, let us discuss the mortal block.

## Computing LK for a mortal block

For a mortal block the first column is always homologous (always contains at least two survivals on leaves) and contains an ancestor at the root. For example, consider block2. However, the other columns in the block where a character is present at only one leave there can be 2 possibilities:

1. Are survivals and has its own ancestral link.
2. Or are descendants to any ancestral links in the block.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ω | # \* | # \*/ − | # \*/ − | # \*/ − |
|  | T \* | G \* | − | − |
|  | C \* | − | A \* | − |
|  | C \* | − | − | C \* |

Since the ancestral sequence is unknown for calculating the likelihood we must marginalize over all possible ancestral sequences. This can be done by finding all possible combinations at the root sequence for all character/ gap occurrences at each columns. For Block1 in the current example we have for first column only one possibility (# \*) and for each of the other 3 columns have 2 possibilities of either having ‘# \*‘ or ‘− ‘ . Thus in total we have 1\*2\*2\*2 i.e. 8 possibilities in figure.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | # \* | − | − | − |
|  | # \* | − | − | # \* |
|  | # \* | − | # \* | − |
|  | # \* | − | # \* | # \* |
|  | # \* | # \* | − | − |
|  | # \* | # \* | − | # \* |
|  | # \* | # \* | # \* | − |
|  | # \* | # \* | # \* | # \* |

In the table we could observe a pattern in the possible ancestral sequences. If we replace ‘# ‘ with 1 and ‘– ‘with 0, we could see that the possible ancestral combinations are the binary form of integer numbers from decimal of ‘1 0 0 0’ to ‘1 1 1 1’ Refer figure/table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 0 | 0 | 0 | 8 |
|  | 1 | 0 | 0 | 1 | 9 |
|  | 1 | 0 | 1 | 0 | 10 |
|  | 1 | 0 | 1 | 1 | 11 |
|  | 1 | 1 | 0 | 0 | 12 |
|  | 1 | 1 | 0 | 1 | 13 |
|  | 1 | 1 | 1 | 0 | 14 |
|  | 1 | 1 | 1 | 1 | 15 |

We shall use this information to generate all possible ancestral scenarios for a given block. In general for a mortal block there will be 2^(n-1) possible ancestral scenarios, where n is the size of the block. For mortal block the integer set can be generalised as an integer set from 2^(n-1) to (2^n)-1. For each of this possible ancestors we will compute the $P(\alpha‘|\theta)$ and $P(\alpha |\alpha‘,\theta)$. The geometric probability $\gamma\_n$ for the mortal block is considered as $(frac{\lambda}{\mu})^n$. This is done inorder to avoid overestimating the probability. The product of these for each ancestral scenario is summed to obtain the complete likelihood of the block.

### Computing for each ancestral scenario

For each ancestral scenario the algorithm will perform the following steps to compute block likelihood

1. Block the orginal block into sub blocks

Blocking is performed using the information about the presence and absence of an ancestral character. Each subblock will consist of only one ancestral link and represents the fate of the same.

1. the $P(\alpha‘|\theta)$ and $P(\alpha |\alpha‘,\theta)$ is computed for each subblocks
2. The product of each subblock are multiplied together to obtain the final block likelihood

We will now calculate the likelihood for 2 of the ancestral scenarios from the table in detail for an understanding.

For the ancestral sequence ‘1 0 0 0’ at root we can represent the block as below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ω | # \* | − | − | − |
|  | T \* | G \* | − | − |
|  | C \* | − | A \* | − |
|  | C \* | − | − | C \* |

In this case we have only 1 ancestral link at column 1 which survived in all the 3 leaves and also gave birth to a character in each leaf. The Indel LK, $P(\alpha‘|\theta)$ for this block under TKF91 can be calculated as:

Now consider the ancestral sequence ‘1 1 0 0’ at root we can represent the block as below

Sub block 1

Sub block 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ω | # \* | # \* | − | − |
|  | T \* | G \* | − | − |
|  | C \* | − | A \* | − |
|  | C \* | − | − | C \* |

In this case we have 2 ancestral links. One in column 1 which survived in all the 3 leaves and Other in column 2 which survived in leaf 1 and died in leaf 2 and 3 but gave birth to one character each at leaf 2 and 3. In this case the algorithm blocks the block into 2 sub block in fig. The Indel LK, $P(\alpha‘|\theta)$ for this block under TKF91 can be calculated as the product of $P(\alpha‘|\theta)$ for each of that subblock:

Subblock1

|  |
| --- |
| # \* |
| T \* |
| C \* |
| C \* |

Subblock2

|  |  |  |
| --- | --- | --- |
| # \* | − | − |
| G \* | − | − |
| − | A \* | − |
| − | − | C \* |

The Substitutional likelihood, $P(\alpha |\alpha‘,\theta)$ is calculated in a similar way. Here in this implementation I have used the simple substitution model from Felsenstine(1981).

## Computing LK for an immortal block

The immortal block represents the fate of the ancestral immortal link at the root. While blocking the MSA if there are columns at the beginning with only one character meaning we have the possibilities of homologous or non-homologous columns. The non-homologous possibility is addressed using the immortal link. In the MSA considered as the example Block1 is the immortal block

Block1

|  |  |  |
| --- | --- | --- |
|  | O − | C \* |
|  | O − | − |
|  | O A \* | − |

Since the immortal link ‘O’ cannot die and give birth it is possible for each columns in the block to take either ‘#\*’ or ’−’ in figure . i.e in this case they have 2^2=4 cases (shown in fig).

Sub block 2

Sub block 1

|  |  |  |  |
| --- | --- | --- | --- |
| Ω | O | − | # \* |
|  | O | − | C \* |
|  | O | − | − |
|  | O | A \* | − |

|  |  |  |  |
| --- | --- | --- | --- |
|  | O | − | − |
|  | O | − | # \* |
|  | O | # \* | − |
|  | O | # \* | # \* |

Here also we could observe a similar pattern exactly like in the mortal block. In General for an immortal block we have 2^n scenarios, where n is the size of the block. For immortal block the integer set can be generalised as a set from 0 to (2^n)-1. If we split the possiblities into two sets, Set 1 from 0 to (2^n-1)-1 and Set 2 from 2^(n-1) to (2^n)-1. The binary form for Set 1 and Set 2 for the Block1 is given in fig. The set 2 represents cases where the immortal link only survives and does not give birth to any character at leaves. This is exactly the case of a mortal block and could be solved.

Set 1

|  |  |  |
| --- | --- | --- |
| Ω | O − | − |
| Ω | O − | # |

Set 2

|  |  |  |
| --- | --- | --- |
| Ω | O # | − |
| Ω | O # | # |

In order to solve set 1 we split the immortal block again into 2 sub blocks sub\_block1 and sub\_block2 depending on the presence of 1st ancetral link at root. The sub\_block1 will give the fate of immortal link in the root and hence the likelihood for the sub\_block1 can be computed using , where i is the leaf and k is the number of children at leaf i. The geometric probability $\gamma\_n$ for the immortal block is considered as $(1-frac{\lambda}{\mu})$. This is done inorder to avoid overestimating the probability. The sub\_block2

is similar to a normal block and thus treated as same like before. The output from sub\_block1 and sub\_block2 will be multiplied for each case. This product is later summed for each ancestral scenarios defined in Set 1.

Substitution LK will also be calculated in a similar fashion at each step and incoporated through multiplication at each step. Fig shows the result of blocking of ancestral scenario ‘O − # \* ‘ from the example.

Immortal Block

|  |  |  |
| --- | --- | --- |
| Ω | O − | # |
|  | O − | C \* |
|  | O − | − |
|  | O A \* | − |

Sub\_block 1

|  |
| --- |
| O − |
| O − |
| O − |
| O A \* |

Sub\_block 2

|  |
| --- |
| # |
| C \* |
| − |
| − |

## Accounting for the hidden scenarios

Consider the case of an ancestral link at the root which died in all 3 leaves. This is a possible scenario which is not visible in the MSA. There could have been 1 to \infty such columns and it is important to account for these. A

## Likelihood of the MSA

The final likelihood, $P(A\_1, A\_2, A\_3)=P(immortal\_block)\*P(mortal\_blocks)

Previous work:

An Ef. cient Algorithm for Statistical Multiple

Alignment on Arbitrary Phylogenetic Trees

@article{lunter2003efficient,

title={An efficient algorithm for statistical multiple alignment on arbitrary phylogenetic trees},

author={Lunter, GA and Mikl{\'o}s, Istv{\'a}n and Song, Yun S and Hein, Jotun},

journal={Journal of Computational Biology},

volume={10},

number={6},

pages={869--889},

year={2003},

publisher={Mary Ann Liebert, Inc.}

}

**Bayesian coestimation of phylogeny and sequence alignment**

@article{lunter2005bayesian,

title={Bayesian coestimation of phylogeny and sequence alignment},

author={Lunter, Gerton and Mikl{\'o}s, Istv{\'a}n and Drummond, Alexei and Jensen, Jens Ledet and Hein, Jotun},

journal={Bmc Bioinformatics},

volume={6},

number={1},

pages={83},

year={2005},

publisher={Springer}

}

**Applying TKF91 model to sequence evolution on a star shaped tree**

The alignment is obtained conventionally with dynamic

programming algorithms (Needleman and Wunsch, 1970; Sankoff and Kruskall,

1983). The weakness of the basic dynamic programming technique based on distance

or similarity functions is the lack of a strong statistical basis. In the pioneering

work (Bishop and Thompson, 1986) an approximate likelihood calculation was

introduced. Thorne *et al*. wrote a landmark paper (Thorne *et al*., 1991), in which

they introduced an exact maximum likelihood algorithm (TKF91). They estimated

the evolutionary distance between two sequences involving all possible alignments

into the likelihood calculation.

Alternative approaches appear in some papers (Allison and Wallace, 1994; Zhu

*et al*., 1998; Mitchinson, 1999), but these are based directly on a probability measure

on alignments and not on an evolutionary process.

The algorithm of Steel and Hein has O(*l*2*k*) running time, where *l* is the geometric

average of sequence lengths, *k* is the number of sequences. This amount of computation

time is not suitable in practice even for three sequences with average length

of 100. A considerable improvement to O(22*klk*) running time is presented in this

paper.

@article{miklos2002improved,

title={An improved algorithm for statistical alignment of sequences related by a star tree},

author={Mikl{\'o}s, Istv{\'a}n},

journal={Bulletin of mathematical biology},

volume={64},

number={4},

pages={771--779},

year={2002},

publisher={Elsevier}

}

Previous work:

An Ef. cient Algorithm for Statistical Multiple

Alignment on Arbitrary Phylogenetic Trees

**Bayesian coestimation of phylogeny and sequence alignment**

**Applying TKF91 model to sequence evolution on a star shaped tree**

The alignment is obtained conventionally with dynamic

programming algorithms (Needleman and Wunsch, 1970; Sankoff and Kruskall,

1983). The weakness of the basic dynamic programming technique based on distance

or similarity functions is the lack of a strong statistical basis. In the pioneering

work (Bishop and Thompson, 1986) an approximate likelihood calculation was

introduced. Thorne *et al*. wrote a landmark paper (Thorne *et al*., 1991), in which

they introduced an exact maximum likelihood algorithm (TKF91). They estimated

the evolutionary distance between two sequences involving all possible alignments

into the likelihood calculation.

Alternative approaches appear in some papers (Allison and Wallace, 1994; Zhu

*et al*., 1998; Mitchinson, 1999), but these are based directly on a probability measure

on alignments and not on an evolutionary process.

The algorithm of Steel and Hein has O(*l*2*k*) running time, where *l* is the geometric

average of sequence lengths, *k* is the number of sequences. This amount of computation

time is not suitable in practice even for three sequences with average length

of 100. A considerable improvement to O(22*klk*) running time is presented in this

paper.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | − | C | T | G | − | − | A | − |
|  | − | − | C | − | A | − | A | C |
|  | A | − | C | − | − | C | − | G |