PASTML v0.1

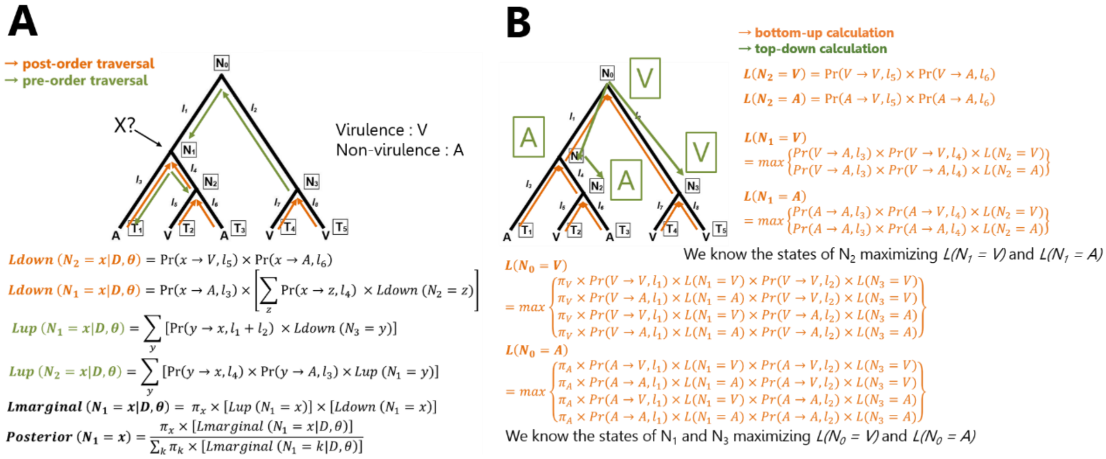
July 21 2017
Sota Ishikawa[1,2], Tomochika Fujisawa[1], Olivier Gascuel[1]
1 Unité Bioinformatique Evolutive, C3BI, Institut Pasteur, Paris, France
2 Graduate School of Science, the University of Tokyo, Tokyo, Japan

Introduction

NGS pipelines are rapidly becoming a routine repertoire in evolutionary, ecological, and epidemiological studies. Yet, only a small part of the several millions of short-length sequence fragments generated by NGS experiments, many of which are expected to be of viral origin, can be analyzed with current methods in bioinformatics. Even for well-known pathogenic viruses, proper epidemiological analyses are becoming more and more difficult due to the lack of bioinformatics tools that can handle the large and growing size of datasets. The VIROGENESIS consortium aims at overcoming these bioinformatics obstacles by developing a software platform for end-users with tools underpinned by novel algorithms and models. We, as a part of VIROGENESIS consortium, focus on the phylogenetic approach to trace the origin and evolution of virus epidemics, by combining large virus trees with extrinsic characters (e.g. geographic location, risk group, presence of a given resistance mutation).

In this context, we developed PASTML (Prediction of Ancestral STates using Maximum-Likelihood), to estimate ancestral characters given an annotated phylogenetic tree. It is built upon a probabilistic, model-based approach and implements the two major ML methods of ACR by considering either joint [1] or marginal [2] posterior probabilities of character state at each tree node. While both methods give accurate results, the joint method does not take into account cases where several characters have similar probabilities at a given node, and predicts a unique ancestral character state per internal node. On the other hand, the marginal method does not provide promising evolutionary scenario(s), as it proposes all possibilities of the character evolution in the data under considering, even when some states clearly emerge while others have very low probabilities, That is why we developed a novel algorithm that provides an intermediate prediction in between these two extremes. It uses a likelihood-based criterion and a statistical score to evaluate predictive accuracy, and provides the user with better and more interpretable predictions. In this document, we present the performances of PASTML by analyzing simulated & real-world datasets.

Marginal reconstruction


A
→ post-order traversal
→ pre-order traversal
N₀
N₁
N₂
N₃
X?
l_1
l_2
l_3
l_4
l_5
l_6
l_7
l_8
A T₁
V T₂
A T₃
V T₄
V T₅
Virulence : V
Non-virulence : A
$Ldown\ (N_2 = x|D,\theta) = \Pr(x \to V, l_5) \times \Pr(x \to A, l_6)$
$Ldown\ (N_1 = x|D,\theta) = \Pr(x \to A, l_3) \times \left[\sum_z \Pr(x \to z, l_4) \times Ldown\ (N_2 = z)\right]$
$Lup\ (N_1 = x|D,\theta) = \sum_y [\Pr(y \to x, l_1 + l_2) \times Ldown\ (N_3 = y)]$
$Lup\ (N_2 = x|D,\theta) = \sum_y [\Pr(y \to x, l_4) \times \Pr(y \to A, l_3) \times Lup\ (N_1 = y)]$
$Lmarginal\ (N_1 = x|D,\theta) = \pi_x \times [Lup\ (N_1 = x)] \times [Ldown\ (N_1 = x)]$
$Posterior\ (N_1 = x) = \frac{\pi_x \times [Lmarginal\ (N_1 = x|D,\theta)]}{\sum_k \pi_k \times [Lmarginal\ (N_1 = k|D,\theta)]}$
B
→ bottom-up calculation
→ top-down calculation
V
A
A
V
$L(N_2 = V) = \Pr(V \to V, l_5) \times \Pr(V \to A, l_6)$
$L(N_2 = A) = \Pr(A \to V, l_5) \times \Pr(A \to A, l_6)$
$L(N_1 = V) = max \begin{cases} Pr(V \to A, l_3) \times Pr(V \to V, l_4) \times L(N_2 = V) \\ Pr(V \to A, l_3) \times Pr(V \to A, l_4) \times L(N_2 = A) \end{cases}$
$L(N_1 = A) = max \begin{cases} Pr(A \to A, l_3) \times Pr(A \to V, l_4) \times L(N_2 = V) \\ Pr(A \to A, l_3) \times Pr(A \to A, l_4) \times L(N_2 = A) \end{cases}$
$L(N_0 = V) = max \begin{cases} \pi_V \times Pr(V \to V, l_1) \times L(N_1 = V) \times Pr(V \to V, l_2) \times L(N_3 = V) \\ \pi_V \times Pr(V \to A, l_1) \times L(N_1 = A) \times Pr(V \to V, l_2) \times L(N_3 = V) \\ \pi_V \times Pr(V \to V, l_1) \times L(N_1 = V) \times Pr(V \to A, l_2) \times L(N_3 = A) \\ \pi_V \times Pr(V \to A, l_1) \times L(N_1 = A) \times Pr(V \to A, l_2) \times L(N_3 = A) \end{cases}$
$L(N_0 = A) = max \begin{cases} \pi_A \times Pr(A \to V, l_1) \times L(N_1 = V) \times Pr(A \to V, l_2) \times L(N_3 = V) \\ \pi_A \times Pr(A \to A, l_1) \times L(N_1 = A) \times Pr(A \to V, l_2) \times L(N_3 = V) \\ \pi_A \times Pr(A \to V, l_1) \times L(N_1 = V) \times Pr(A \to A, l_2) \times L(N_3 = A) \\ \pi_A \times Pr(A \to A, l_1) \times L(N_1 = A) \times Pr(A \to A, l_2) \times L(N_3 = A) \end{cases}$

We know the states of N_2 maximizing $L(N_1 = V)$ and $L(N_1 = A)$

We know the states of N_1 and N_3 maximizing $L(N_0 = V)$ and $L(N_0 = A)$

Figure 1. Marginal (A) and Joint (B) methods

With the marginal reconstruction method, down-likelihood (L_{down} in Fig. 1A) and up-likelihood (L_{up} in Fig. 1A) of a certain ancestral state at a given internal node is computed using two successive procedures: one pre-order and one post-order tree traversal. The down-likelihood at a node of interest can be calculated based on information from its descendant subtree, which computation is similar to Felsenstein's pruning algorithm [3]. For example, the down-likelihood of the node N_2 having the state x (either V or A in the case in Fig. 1) can be computed by multiplying two probabilities of substitution from N_2 to its two descendant states (i.e., $T_2 = V$ and $T_3 = A$), i.e. substitutions from x to V and A during time l_5

and l_6. Down-likelihood of N_1 having a certain state x can be computed in the same way, considering substitutions to T_1 and N_2. We consider all possible states at N_2, shown as z in Fig.1A, and calculate the sum of substitution probabilities from x to z multiplied by the down-likelihood of $N_2 = z$. We calculate the down-likelihood of N_3 in the same manner. The up-likelihood of a given node can be computed based on the information of its external side of the tree. For example, to compute up-likelihood of $N_1 = x$, we consider substitutions from its parent node N_3 (note that we ignore the root and consider the tree as unrooted). We then multiply each down likelihoods of $N_3 = y$ by the probability of substitution from y to x during the time $l_1 + l_2$, and sum them. The up-likelihood of N_2 having a certain state x can then be calculated by summing up-likelihoods of N1 having state y, multiplying each of them by the probability of substitutions among N_1 and T_1 ($y \rightarrow A$, in l_3), and N_1 and N_2 ($y \rightarrow x$, in l_4). We calculate up-likelihoods for N_3 in same manner. Finally, the marginal likelihood of a certain state can be computed by multiplying up- and down-likelihoods of the corresponding state, taking the prior probability of that state (π) into account; note that π stands for the frequency of the corresponding state in the input annotation data. Posterior probability of each state can be computed based on its marginal likelihood as we shown in Fig. 1A.

Joint reconstruction

The joint reconstruction algorithm first traverses the tree from the tips toward the root, as we denote as 'bottom-up calculation' in Fig. 1B. Upon visiting an internal node N, we compute for each character state x a likelihood $L(N = x)$, based on the information from its descendant subtree. This procedure is similar to the post-order traversal in the marginal method, however, in that case we compute the 'best' reconstruction for the subtree given that N is assigned a certain state x. For example, upon visiting N_1 we compare $L(N_1 = V)$ on the two different conditions of state reconstruction at N_2 considering their likelihood values $L(N_2 = V)$ and $L(N_2 = A)$, which are same with the down-likelihood of $N_2 = V$ *or* A. We then select which state at N_2 can maximize $L(N_1 = V)$. We compute $L(N_1 = A)$ in the same way and maximized joint probabilities of N_1 can be used for the calculation at the root (N_0), along with $L(N_3 = V)$ and $L(N_3 = A)$ computed as $L(N_2=V)$ and $L(N_2=A)$.At the final step of the bottom-up calculation, we know which states at N_1 and N_3 can maximize the joint probability at the root given a certain state. Then, we compare the joint probability of V and A, multiplying them by the prior probability π_A and π_V, and select the most likely state at the root. We traverse the tree from the root in the direction of tips (denoted as the 'top-down calculation' in Fig. 1B), assigning to each node its most likely state given its parent state as we already computed in the bottom-up calculation. Finally we obtain the 'joint' reconstruction on the tree.

New algorithm to find the best intermediate prediction

Both marginal and joint likelihood computation methods are suggested to be accurate on a variety of data [9]. However, as we show in Fig.2 joint method gives only one character state (= prediction) for each node even when several states have similar probabilities. Thus, it potentially ignores some 'nearly best' predictions to be informative to investigate the character evolution in the data of interest. In contrast, marginal method does not provide a promising character evolution process on the tree even if some of the states clearly emerge with high posterior probabilities while the others have low probabilities. Thus, deciding on the most likely evolutionary scenario matching the data is still challenging with the above two methods, especially when dealing with the evolution of virus epidemics (i.e. geographic location, risk group, presence of a given resistance mutation) on large phylogenetic trees. Therefore, we developed a new approach to search the best intermediate prediction between the above two extremes and implemented it in PASTML.

This algorithm starts with the marginal likelihood prediction computed as explained above. Then, the least likely state not being predicted by the joint method is removed iteratively, until the joint prediction is reached (Fig. 2). At each step of this 'marginal-to-joint' algorithm, marginal likelihoods at all ancestral nodes are recomputed using pre- and post-order traversal algorithm (Fig. 1A), ignoring previously removed states by assigning them a null marginal likelihood. Consequently, we obtain updated posterior probabilities of remaining states that will be used at the next step (Fig. 2). The likelihood of the tree is also recomputed at each step by summing marginal likelihoods (incorporating priors) recomputed at the root. Then we calculate 'likelihood fraction' of the present prediction (see below) that is used to stop the iterative process when a given threshold is met.

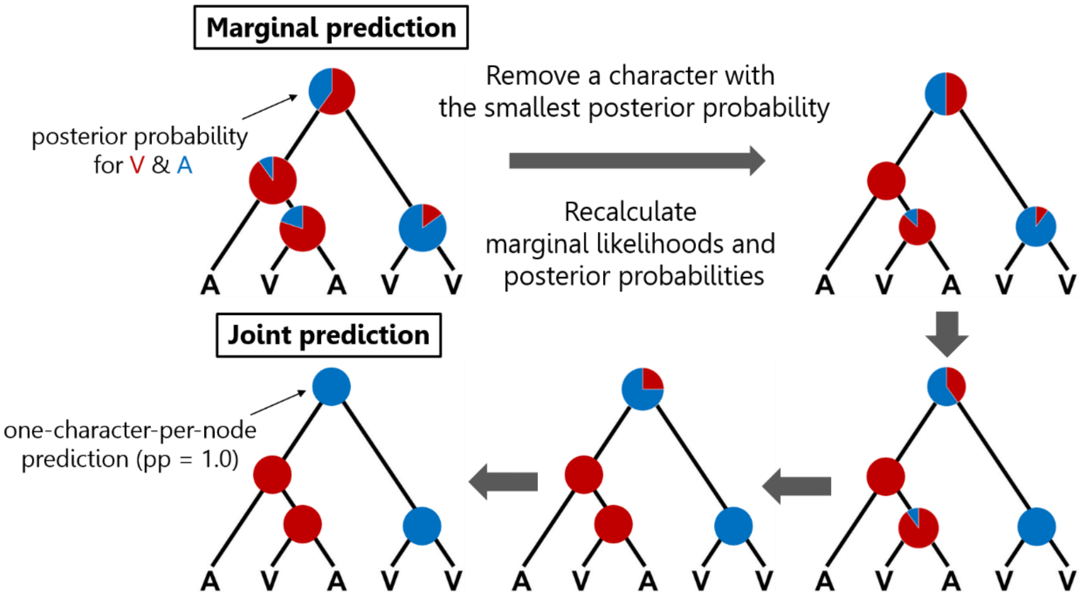


Figure 2. Example of the PASTML workflow

The present algorithm stops when it gets a certain prediction having the likelihood fraction closest to a user-defined value 1/X, otherwise it stops when it reaches the joint prediction.

Likelihood fraction

We define F_i, the likelihood fraction for the i_{th} intermediate prediction, as:

$$F_i = \frac{L_i}{L_{marginal}}$$

where $L_{marginal}$ is the likelihood of the marginal prediction that can be calculated by Felsenstein pruning algorithm [3]. It is computed as the sum of marginal likelihoods of all possible states at the root incorporating prior probability of each state. L_i is the likelihood of the i_{th} intermediate prediction, computed the same way. F_i is computed at each step i and is compared with the user-defined value (X, see Usage Example & Command-line Options). If an intermediate prediction shows the closest value to 1/X (the minimum value of $|F_i - X|$), marginal-to-joint algorithm stops and returns the current prediction.

Rescaling tree

In order to combine phylogenetic trees, inferred from molecular sequences, with extrinsic character data, it is necessary to re-scale branch lengths of the input tree before all calculations. To do so, we optimize a tree-scaling factor (ρ) that we subsequently multiply to each branch length of the tree. We chose the Golden Section Search (GSS) method [4] to optimize ρ. At each step of the optimization, we re-compute the likelihood of the rescaled tree by the proposed values of ρ, and iterate it until we find the maximum of the likelihood function over ρ.

How it performs

In the following section, we present the results of our simulation and real-data analysis, and we show how PASTML compares to previous methods. We also detail the theoretical and experimental complexity of the present algorithm, as well as the total execution time on datasets of different size.

Simulation experiment

Our simulation procedure is detailed below:

1. 1000-tips pure-birth trees were generated based on different birth rate (0.1, 0.2, … 1.0, 2.0, …, 10.0, i.e. 5, 2.5, …, 0.5, 0.25, …, 0.05 mutations/branch). Fifty trees were generated for each birth rate, and one 4-characters dataset were generated from each

tree, using the HKY85 (HKY) model [5] , ts/tv ratio = 8.0 and π_T = 40%, π_C = 10%, π_A = 20%, and π_G = 30%. To do so, we applied a Monte-Carlo simulation procedure [6] by traversing the tree from the root to the tips and evolving states using HKY model.

At this stage, we obtain 950 trees, states of the tips, and known ancestral states of internal nodes (including the root). We denote this as the 'true scenario'.

2. Simulated trees and character datasets were then analyzed by three approaches: 1) A standard parsimony approach [7], 2) Joint and marginal methods using approximate JC model [8], and 3) the present method. For the above setting of models used in the simulation, HKY and JC, we referred to the previous procedure presented by [9]. Marginal-to-joint algorithm (Fig. 2) were stopped at different likelihood fraction thresholds: 1/3 and 1/20. Likelihood fractions are in some way analogous to Bayesian factors, as they can be seen as the likelihood-ratio between two different models of prediction, i.e. marginal (*M_1*) and any intermediate prediction (*M_2*). As we follow Kass and Raftery (1995) [10], 1/3 fraction means 'positive' strength of evidence of *M_2* being more strongly supported by the data than *M_1*, while 1/20 fraction means 'strong' strength of evidence of that. To assess that the above assumption is appropriate, we compared the prediction of 1/3 and 1/20 fractions with parsimony, marginal, joint, and the best prediction.
3. To evaluate and compare the predictions of the above methods, we used their Brier score (BS) [11] against the true scenario. BS can be computed as below;

$$(1) \qquad BS_{N_i} = \sum_{k}^{C_{N_i}} (P_k - f_k)^2$$

$$(2) \qquad P_k = \begin{cases} \dfrac{1}{C_{N_i}}, & \text{if } k \text{ is predicted at } N_i \\ 0, & \text{if } k \text{ is not predicted} \end{cases}$$

$$(3) \qquad f_k = \begin{cases} 1, & \text{if } k \text{ is TRUE} \\ 0, & \text{if } k \text{ is FALSE} \end{cases}$$

$$(4) \qquad \mathrm{BS} = \frac{1}{N} \sum_{N_i}^{N} B_{N_i}$$

where *N* is the number of ancestral nodes (including the root) and *C_{Ni}* is the set of possible character states at the node *N_i*. *BS* = 0 means that the prediction corresponds perfectly to the true scenario. As we assume that posterior probabilities do not provide promising evolutionary history, we provide simplified probability, $^{1}/_{C_{N_i}}$, for each predicted character instead of using its (re-calculated) posterior probability. The worst case is when we have equally probable states, e.g. 50% of a certain state and 50% of another state. In such case, there is no way to make a decision based on posterior probabilities. Therefore, to evaluate each prediction, we focus on 'how many FALSE characters we could remove keeping a TRUE character,' using simplified probability (*P*) and 0 or 1 function (*f*) for each predicted state.

4. As we know the true scenario, the 'truly best' prediction can be selected by comparing BSs of joint, marginal, and each intermediate prediction. This simply corresponds to the minimum point of the empirical curve of BS between joint and marginal predictions. Our goal is to find a good estimation of F_i providing the optimal stopping point as close as possible to the true best prediction. Therefore, we compare BS of predictions obtained from different methods including parsimony, joint, marginal reconstructions, and the present algorithm stopped at 1/3 & 1/20 fraction to assess whether our approach can provide better predictions than any other existing methods.

Figure 3A represents the BS of five predictions, parsimony (represented by "open circle"), joint ("cross"), marginal ("plus"), and 1/3 ("triangle") & 1/20 ("diamond") fractions, as well as the best prediction ("orange closed circle"), on 0.1 – 10.0 birth rates.

We can already note that maximum-likelihood based methods (i.e. joint, marginal, and fractions) are better than MP-based (maximum parsimony) method. At birth rate = 2.0, the predictive accuracy of MP was not better than random prediction in which we randomly reconstructed one ancestral character per branch and computed its BS as 1/4*(1/4 - 0)^2 + 3/4*(1/4 - 0)^2 = 0.1875 under the present simulation setting. Thus, the MP-based method can be even worse than random prediction if we have more than 0.25 mutations/branch, i.e. it is quite sensitive to the saturation of substitutions.

Since we used simplified probabilities to calculate BS of marginal prediction, its value was not changed regardless of birth rates and equivalent with that of the random prediction. In contrast, BS of the joint prediction can be higher or lower than that of the marginal prediction, depending on birth rates. At birth rate = 0.1 ~ 1.0, where the average of branch lengths of the corresponding tree is 0.5– 5, joint prediction was worse than marginal prediction because of the difficulty of one-character-per-branch prediction considering multiple substitutions happened on such long trees. In other hand, joint prediction became better than marginal prediction at birth rate = 2.0 ~ 10.0 (average branch length = 0.25- 0.05), where the true scenario on such short trees experienced just a few substitution so that the joint reconstruction can be easy.

Intriguingly, at birth rate = 0.1, which is the most difficult case of ACR in this simulation, both 1/3 and 1/20 fractions provide the closest predictions to the best (Fig. 3A). At birth rate = 0.2 ~ 1.0, 1/3 and 1/20 fractions also provided better predictions than other approaches (parsimony, marginal and joint) albeit they need to remove further 1 - 2 characters in average at each node to be closed to the best (Fig 3B). At birth rate = 2.0 ~ 8.0, 1/20 fraction-based prediction and joint prediction showed nearly equivalent BS value, while former was still useful than later because it can consider of several likely states in addition to a unique prediction provided by the joint method (Fig. 3B). At birth rate = 9.0 & 10.0, all ML-based methods except for the marginal method provide equally accurate prediction compared to the best one because of the simplicity of ACR as we mentioned above.

In conclusion, our new approach using likelihood fraction provides better prediction than other existing methods regardless of the tree length. Furthermore, PASTML is robust to model misspecifications that we added between true (HKY) and approximate (JC) models. It suggests that it can be useful in real-world data analyses for which we do not know the true model that generated the data. The 1/20 fraction always provided better prediction than that of 1/3 fraction. This is concordant with Bayes factor as we mentioned above, as Kass and Raftery (1995) stated that a Bayes factor greater than 20 between M_1 and M_2 means a ‘strong’ evidence of M_2 as being more supported by the data than M_1. Thus, we recommend our end-users to apply 1/3 – 1/20 fractions (i.e. 3 – 20 for the ‘Fraction’ option of the program, see Usage Example & Command Line Options), and crosscheck predictions from a variety of fractions, checking whether there is significant difference on the number of characters predicted at each node between them. Given the results of our simulations, lower number means better prediction (Fig. 3B).

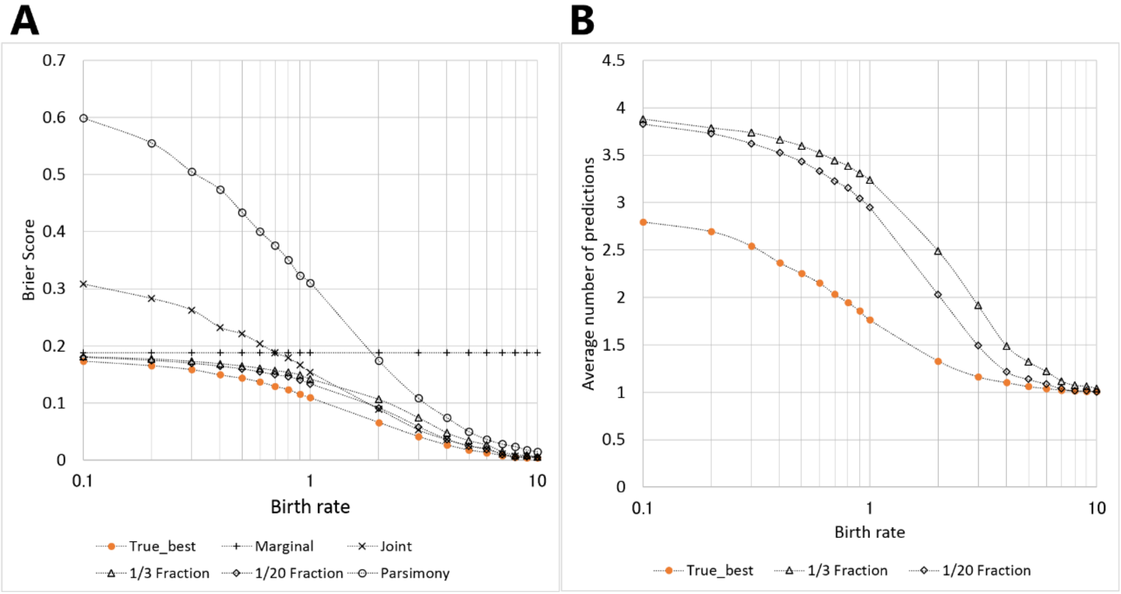


Figure 3. Results of simulation

(A) Brier score vs. Birth rate for five different methods: parsimony, joint, marginal, and 1/3 & 1/20 fractions, as well as the best prediction. Horizontal axis represents the birth rate of the simulated tree (0.1 – 10.0). (B) Similar to (A) but with the average number of characters predicted at each node is shown for 1/3 & 1/20 fractions, and compared with that of the best prediction.

Real-data analysis with Albanian virus dataset

We applied PASTML to the epidemiological history of HIV-1 subtype A in Albania (Albanian virus). The dataset was taken from the MP-based 'phylotype' analysis of [12]. A phylogenetic tree of 152 strains was inferred from *pol* sequences that were derived from the study by Salemi et al. (2008) [13]. Primary annotation of the strains consists of five geographic zones (Africa, Western Europe, Eastern Europe, Greece and Albania) where the sequences were collected. We attached all input/output files in the 'Example' directory of source code. We applied JC model to analyze the data, and a fraction threshold of 1/20 to stop the marginal-to-joint algorithm. Result is provided in Figure 4 in the form of a re-scaled tree with colored branches and pie charts. External branches are colored by their tip-annotations, while internal branches are colored according to their prediction provided by the joint method. On this illustration of the joint prediction, we added some possibilities of ancestral states predicted with 1/20 likelihood fraction. Pie charts at seven internal nodes represent one additional possibility with its recomputed posterior probability, as well as that of the joint prediction (note that a joint prediction of node with pie chart is shown in same color as its parent branch).

The results are consistent with MP-based method [12] with regards to major transmission of HIV-1 subtype A strains, i) from Africa to East Europe, ii) from Africa to Greece, iii) from Greece to Albania, as well as several multiple introductions of African strains into Europe including Greece. Furthermore, the present method is more informative because it allows us to consider alternative scenarios for the epidemiological history in addition to the unique prediction of the joint method. For example, according to the joint prediction, strains named as 97YUAF9960, 02GRAY0270, 97FRAJ0558, 98FRAJ0552, 00GRAF5753, and 03ESAY2111 are sampled in different zones, West/East Europe and Greece, while they are originated from a single African strain (surrounded by broken line in Fig.4). However, our results suggests additional intriguing questions for virus transmission in the corresponding clade, if we look at pie charts represented at three internal nodes of that particular clade. That is;

I. Which zone was firstly introduced by the common ancestral African strain, Greece or West Europe?

II. How many times virus transmission occurred between West Europe and Greece? Are there frequent virus transmission between populations in these zones?

Consequently, by considering some possibly important alternative scenarios on virus epidemics, our method gives a new insight of the data, and constitutes a good tradeoff between the complexity of marginal predictions (i.e. having pie charts of all possibilities at all nodes) and the over simplicity of joint predictions..

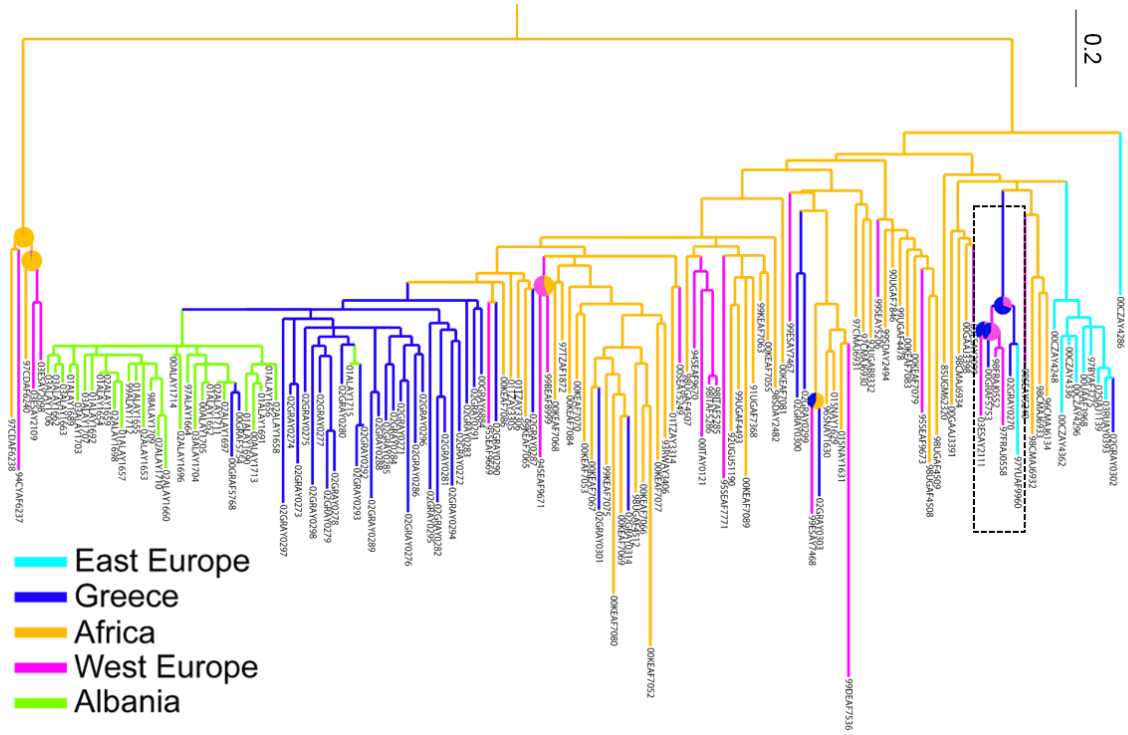


Figure 4. A prediction of 1/20 fraction on Albanian virus dataset

Tree is re-scaled by a scaling factor optimized by GSS. External branches are colored by their tip-annotations, while internal branches are colored by the joint prediction. Pie charts represent re-computed posterior probability of states predicted with 1/20 fraction.

Computational performance

We describe the computational complexity of PASTML below. Overall workflow is illustrated in Fig. 5. First, we need to compute marginal and joint predictions. Complexity of marginal and joint methods, as we explained above (Fig. 1), is theoretically $O(n \times m^2)$ if we analyze *n* tips tree and *m* possible character states as input data. Second, we run the marginal-to-joint algorithm (Fig. 2) removing most unlikely states step-by-step. At each step of this algorithm, we need to re-compute marginal likelihoods (and posterior probabilities) of remaining states. Since this re-computation can be done by using the same algorithm as shown in Fig. 1A, its order can be assumed as $O(n \times m^2)$. We also need to compute the likelihood fraction of the prediction at each step. It can be performed in $O(n \times m^2)$ as we applied Felsenstein pruning algorithm [3] to compute the likelihood of each prediction. The number of iteration in the above algorithm can be changed according to the user-defined value of likelihood fraction (*X*). However, the maximum number of iteration (i.e. *X = 0* and we reach the joint prediction without stopping) is $(n-1)(m-1)$. Thus, the most complicated part of PASTML is the above iteration procedure and complexity of the overall computation can be theoretically assumed as $O(n^2 \times m^3)$ if we set *X* = 0 option.

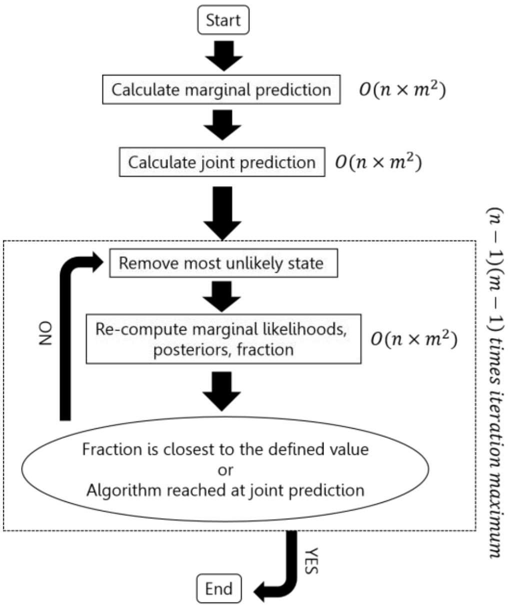

Start
Calculate marginal prediction
$O(n \times m^2)$
Calculate joint prediction
$O(n \times m^2)$
Remove most unlikely state
Re-compute marginal likelihoods, posteriors, fraction
$O(n \times m^2)$
NO
Fraction is closest to the defined value
or
Algorithm reached at joint prediction
$(n-1)(m-1)$ times iteration maximum
YES
End

Figure 5. Flowchart of PASTML and computational order of main algorithms

Computational complexity of marginal and joint methods, as well as the present iteration algorithm to compute intermediate predictions, is shown.

To assess the practical complexity of our program, we simulated datasets of different number of tips or character states, and analyzed them with PASTML applying JC model and *X* = 20 option. Results are shown in Fig. 6. PASTML showed nearly quadratic order of computation against increasing number of tips up to 128 (Fig. 6A). Although its practical complexity grow up bigger than quadratic order in analyses on larger trees of thousands of tips, it is still acceptable in the point of view of absolute computational time, i.e. it can run within one hour with ~8000 tips tree (Fig. 6A). On the other hand, Fig. 6B showed significantly better performance than cubic order of computation against increasing number of possible character states up to 64. This is because we stopped the iteration with 1/20 fraction where we removed one most unlikely state per node in average (see diamond plot at birth rate = 1.0 in Fig. 3B). Thus, re-computation of likelihoods was iterated just *n* times so that theoretical complexity against number of character states, *m*, can be assumed as $O(m^2)$.

In conclusion, PASTML code is performing well according to its expected complexity. It is also valid to note that the present code can provide our users with useful prediction in practical computation time, even if they analyze very large virus trees with multiple character states.

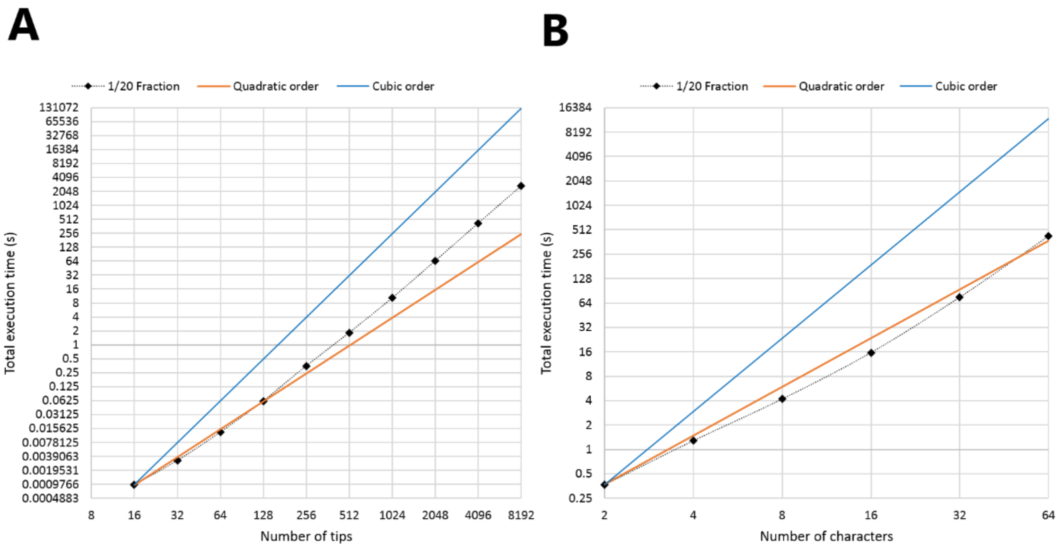


Figure 6. Computational performance on a variety of datasize

Total execution time is compared between different size of input data. A) Number of tips of the tree was increased up to 8192, and for each tree-size we generated fifty 4-character data and analyzed them to calculate the average of total execution time until we obtained the prediction with 1/20 fraction. B) Number of characters was increased up to 64. We generated fifty dataset of each character-size on the 512 tips tree and calculated the average of total execution time. Orange line means execution time inferred from theoretical quadratic order based on the time at 16 tips or 2 characters. Blue line represents that coming from theoretical cubic order. All trees were generated with birth rate = 1.0

Building PASTML

After downloading the source code and example files from https://github.com/saishikawa/PASTML, you can compile PASTML by just typing;

```
$ make
```

This should produce an executable called PASTML.

Usage Example & Command Line Options

Example of usage of PASTML command line:

```
$ ./PASTML –a <annotation file> –t <input tree file> –c <#characters> -# <#tips>-x <denominator of likelihood fraction cutoff> –m <model> –f <frequencies>
```

-a: name of input annotation file with the following format;

ID of tips,	Annotation
Tip 1,	Annotation 1
Tip 2,	Annotation 2
…,	…
Tip N,	Annotation M

You can look at the example in Example/Annotation.Albanian.5chars.txt

-t: input tree file (NEWICK format, **must be rooted**)

-x: specify *X* (integer), where you want to stop the algorithm and obtain the intermediate prediction with corresponding 1/*X* of likelihood fraction

-m: specify probabilistic model to be used in likelihood calculations

JC: Jukes and Cantor 1969 [8] like model assuming same probabilities for all subsitutions among possible states.

F81_E: Felsenstein 1981 [14] like model, in which probability of substitution from acertain state *i* and *j* can be calculated based on character frequencies (π) estimated from the annotation data as below.

$$\mathbf{Pr}(\boldsymbol{i} \rightarrow \boldsymbol{j},\ \boldsymbol{l}) = \begin{Bmatrix} (1 - e^{-\mu l})\pi_j & if\ i \neq j \\ e^{-\mu l} + (1 - e^{-\mu l})\pi_i & if\ i = j \end{Bmatrix}$$

$$\mu = 1 \Big/ \left(1 - \sum_i \pi_i^2\right)$$

F81_U: same as F81_E but it applies user-defined character frequencies

-f: if you specify F81_U type character names and frequencies, in order, as below;

character1 character2 character3 … 0.1 0.2 0.3 …

note that the sum of character frequencies should be 1.0

You can run the test-data analysis by typing;

```
$ cd Example
$ ../PASTML –a Annotation.Albanian.5chars.txt –t Albanian.tree.152tax.tre –c 5
-# 152 –x 20 –m F81_E
$ ../PASTML –a Annotation.Albanian.5chars.txt –t Albanian.tree.152tax.tre –c 5
-# 152 –x 20 –m F81_U –f Greece Albania EastEurope Africa WestEurope
0.256579 0.203947 0.065789 0.328947 0.144738
```

Outputs

PASTML generates two output files :

1. Result_treeIDs.N.taxa.M.states.tre

   This is the same tree as the input file (Newick format), but with rescaled branch lengths by multiplying them by the scaling factor ρ as we discussed above. In addition, internal node IDs are written as branch labels (like branch supports).

2. Result_states_probs.N.taxa.M.states.txt

   This is a tab separated file containing the resulting prediction incorporating probability of each state, as explained below;

Node Id	Character 1	Character 2	…	Character M
Internal 1	P_1	P_2	…	P_M
…	…	…	…	…
ROOT	P_1	P_2	…	P_M
Tip 1	1.0	0.0	…	0.0
…	…	…	…	…
Tip N	0.0	0.0	…	1.0

P is the probability of corresponding character given a node or a tip. P should be 1.0 or 0.0 at every tips since their annotation is known. For each internal node (including the root), we print the posterior probability (PP) of each corresponding character state. As we mentioned above we re-calculate all PPs after we remove the most unlikely character at each step, ignoring removed states. Thus, P should be 0.0 if the corresponding character is not predicted (i.e. removed) at the node otherwise it is the re-calculated PP for the corresponding prediction.

Acknowledgement

This work is supported by the VIROGENESIS H2020 European Project (N° 634650).

References

[1] Pupko, Tal, et al. "A fast algorithm for joint reconstruction of ancestral amino acid sequences." Molecular Biology and Evolution 17.6 (2000): 890-896.

[2] Yang, Ziheng. "PAML 4: phylogenetic analysis by maximum likelihood." Molecular biology and evolution 24.8 (2007): 1586-1591.

[3] Felsenstein, Joseph. "Maximum likelihood and minimum-steps methods for estimating evolutionary trees from data on discrete characters." Systematic Biology 22.3 (1973): 240-249.

[4] Flannery, Brian P., et al. "Numerical recipes in C." Press Syndicate of the University of Cambridge, New York 24 (1992).

[5] Hasegawa, Masami, Hirohisa Kishino, and Taka-aki Yano. "Dating of the human-ape splitting by a molecular clock of mitochondrial DNA." Journal of molecular evolution 22.2 (1985): 160-174.

[6] Rambaut, Andrew, and Nicholas C. Grass. "Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees." Computer applications in the biosciences: CABIOS 13.3 (1997): 235-238.

[7] Swofford, David L., and Wayne P. Maddison. "Reconstructing ancestral character states under Wagner parsimony." Mathematical Biosciences 87.2 (1987): 199-229.

[8] Jukes, Thomas H., and Charles R. Cantor. "Evolution of protein molecules." Mammalian protein metabolism 3.21 (1969): 132.

[9] Gascuel, Olivier, and Mike Steel. "Predicting the ancestral character changes in a tree is typically easier than predicting the root state." Systematic biology 63.3 (2014): 421-435.

[10] Kass, Robert E., and Adrian E. Raftery. "Bayes factors." Journal of the american statistical association 90.430 (1995): 773-795.

[11] Brier, Glenn W. "Verification of forecasts expressed in terms of probability." Monthly weather review 78.1 (1950): 1-3.

[12] Chevenet, François, et al. "Searching for virus phylotypes." Bioinformatics 29.5 (2013): 561-570.

[13] Salemi, Marco, et al. "High-resolution molecular epidemiology and evolutionary history of HIV-1 subtypes in Albania." PloS one 3.1 (2008): e1390.

[14] Felsenstein, Joseph. "Evolutionary trees from DNA sequences: a maximum likelihood approach." Journal of molecular evolution 17.6 (1981): 368-376.