MAXIMUM LIKELIHOOD PHYLODYNAMIC ANALYSIS

Dissertation

der Matematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

PAVEL SAGULENKO

aus Sewastopol, Ukraine

Tübingen, 2017

Gedruckt mit Genehmigung der Matematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen

Tag der mündlichen Qualifikation:	29.01.2018
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Richard Neher
2. Berichterstatter:	Prof. Dr. Daniel Huson

CONTENTS

1 INTRODUCTION 1

1.1 Divergence time estimation 2

1.2 Molecular clock models 4

The Strict Clock 4

Maximum-likelihood 5

Bayesian 6

1.3 Motivation 9

1.4 An Outline of TreeTime 9

2 MAXIMUM-LIKELIHOOD METHODS IN PHYLOGENETICS 11

2.1 A Model of sequence evolution 11

2.2 Tree likelihood calculation 16

2.3 Ancestral sequences reconstruction 20

Joint reconstruction 20

Marginal reconstruction 21

2.4 Branch lengths optimization 24

3 TREETIME 27

3.1 Divergence times reconstruction 28

Core algorithm 28

Joint reconstruction 29

Marginal reconstruction 32

Tree pre-processing 34

3.2 TreeTime additional functionality 36

Efficient search for the optimal root 36

Resolving polytomies 39

Autocorrelated molecular clock 41

Inference of time reversible substitution models 42

Coalescent priors . . . 43
3.3 Case study: analysis of the 2014-2015 Ebola Virus outbreak . . . 44
4 TREETIME VALIDATION . . . 47
4.1 Objectives . . . 47
4.2 Validation on simulated data . . . 47
Divergence times and mutation rate . . . 48
Coalescent model inference . . . 49
4.3 Validation on Influenza phylogenies . . . 51
5 TREETIME MODULE . . . 53
5.1 Python package . . . 53
5.2 Source code structure . . . 53
5.3 Implementation of likelihood distributions . . . 54
5.4 Processing pipeline . . . 57
5.5 Web application . . . 59
6 INFERENCE OF GENERAL TIME REVERSIBLE MODELS . . . 62
6.1 Site-specific substitution models . . . 62
6.2 Inference scheme for site-specific GTR . . . 65
Model parametrization . . . 65
Maximizing tree likelihood . . . 67
Final equations . . . 70
6.3 GTR inference scheme validation . . . 71
Simulating sequence evolution . . . 71
Phylogeny and ancestral reconstruction . . . 73
Testing the inference scheme . . . 73
BIBLIOGRAPHY . . . 79

ZUSAMMENFASSUNG

Die Anzahl der verfügbaren Genomsequenzen für verschiedene Pathogene hat in den letzten Jahren ausserordentlich zugenommen. Bestehende traditionelle Methoden für die phylodynamische Analyse sind nicht effizient für eine große Anzahl von Sequenzen. Um mit den heute verfügbaren Datensätzen umzugehen, sind effiziente Heuristiken notwendig.

In dieser Arbeit wird ein annähender Maximum-Likelihood Ansatz zur phylodynamischen Analyse entwickelt. Der Hauptzweck dieses Ansatzes war es die Divergenzzeiten in grossen Sequenz Alignments von schnell evolvierenden Organismen zu schätzen. Ausserdem bietet er die Funktion ancestrale Zustände zu schätzen, Evolutionsmodelle abzuleiten, Bäume neu zu wurzeln, um zeitliche Signale zu maximieren, sowie um Phylogenien der molekularen Uhr und die Geschichte von Populationsgrössen abzuschätzen. Die Laufzeit der meisten entwickelten Algorithmen verhält sich dabei linear zur Grösse des Datensatzes. Grundsätzliche Anwendungsfelder für diesen Ansatz sind epidemologische Studien sowie solche, die sich mit der Evolution von Pathogenen beschäftigen. Dies beinhaltet das Datieren von Transmissionen über Speziesgrenzen hinweg, wie auch das des Eintretens in geographiche Regionen, sowie die Untersuchung von Populationsgrössen von Pathogenen.

Im zweiten Teil dieser Arbeit stelle ich die Interferenzschemata der Evolutionsmodelle vor, die sich in der Substitutionrate ihrer Sites unterscheiden. Diese Art von Modell kann nicht nur bessere Ergebnisse bezüglich der Annäherung der phylogenetischen Rekonstruktion hervorbringen, sondern auch die evolutionären Kräfte vorhersagen, die auf Protein- oder DNA-Sequenzen einwirken.

ABSTRACT

The number of genome sequences available for different pathogens has increased dramatically over the last couple of years. Existing traditional methods for phylodynamic analysis scale poorly with the number of sequences. Therefore, efficient heuristics are needed to cope with the growing data sets available today.

In this work, an approximate maximum-likelihood framework for phylodynamic analysis is developed. Its main purpose has been to estimate divergence times in large sequence alignments of rapidly evolving organisms. In addition, it provides a functionality to estimate ancestral states, infer evolution models, re-root trees to maximize temporal signals, and estimate molecular clock phylogenies and population size histories. The run time for most of the developed algorithms scales linearly with dataset size. The basic application fields for the framework are studies for epidemiology and pathogen evolution, including dating cross-species transmissions, dating introductions into geographic regions, and studying the time course of pathogen population sizes.

In the second part of this work, I present an inference scheme for evolutionary models with substitution rate heterogeneity among sites. These types of models can not only result in a better approximation of the phylogenetic reconstruction, but also predict the evolutionary forces acting along protein or DNA sequences.

ACKNOWLEDGMENTS

I express my deepest appreciation to all people who supported me during my doctorate. Firstly I thank my supervisor, Prof. Richard Neher. He is one of the brightest people I have ever met. I was happy to work with him for almost 4 years.

I also thank my collaborators: Fabio Zanini, Vadim Puller, Emmanuel Bénard and Taylor Kessinger. They spent a lot of their time teaching me basics of population genetics, programming, linear algebra. I have learned a lot from our frequent discussions. It was a great pleasure to work with them. I hope I could learn some qualities, which make them outstanding scientists.

Thomas Musielak, Daniel Slane, Vadim Puller, Ole Herud, Csaba Veraszto commented on the preliminary versions of theis thesis and helped very much to improve it. My special appreciation to Thomas Musielak, Ole Herud and Milan Graf for improving the contents and the style of this work.

My co-supervisor Daniel Huson, the members of my Thesis Advisory Committee, my lab members, my colleagues and mentors at the Max-Planck Institute and University of Tübingen were very helpful and dedicated to advancing research first and foremost.

Personally, I would like to mention my family members, friends, and loved ones for their unexhausted support and affection. Thank you.

1

INTRODUCTION

The biodiversity observed on Earth today is the result of evolution. The evolution at the molecular level (molecular evolution) is a process of constant changes in DNA (or sometimes RNA) sequences across generations. The main source of molecular evolution are mutations. They appear as errors of DNA replications. Since they are a result of a chemical process, the mutations are stochastic in nature and typically occur randomly across genomes. Therefore, they introduce genetic diversity into populations. Different mutations may have different effects on the phenotypes of individuals. Those that increase an individual's fitness spread through a population by natural selection. "Neutral" mutations might spread in a population by genetic drift. They are also likely to be found in descendant generations. Mutations that decrease fitness will eventually disappear from the genome. So, beneficial and neutral mutations that once appeared in a genome are transferred to future generations. They can be observed in DNA samples taken from population genomes.

The sampled mutations can be used as genetic markers to reconstruct the evolutionary history of populations (*phylogeny*). Phylogeny reconstruction usually starts with the sampling and sequencing of the DNA from populations. DNA sequences are then aligned to form a *multiple sequence alignment*. Phylogenetic algorithms usually operate on multiple sequence alignments. The history of an alignment can then be represented as a *phylogenetic tree*. It is a model that explains how the observed sequences evolved from a single common ancestor. It shows their phylogenetic relationships and, therefore, how they came to be what they are today. An example of a simple reconstruction of an evolutionary history, i. e. the building of the phylogenetic

tree, is presented in figure 1. It shows the evolutionary relationships among sequences and suggests the evolutionary times at which different mutations occurred. Despite many biological processes, such as horizontal gene transfer [Ochman et al., 2000; Keeling and Palmer, 2008] or recombination [Whitehouse, 1982] make the evolutionary process look more like a network, the phylogenetic tree has always been at the basis of evolutionary reconstructions. For many purposes, trees make a very good approximation for the process of molecular evolution.

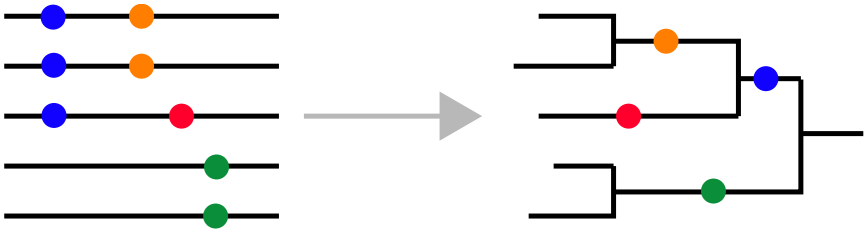

Figure 1: A simple example of reconstructing the phylogeny from a multiple sequence alignment. The sequences are represented as lines, the mutations are shown as dots of different colors. The more abundant the mutation in the population, the earlier in history it appeared.

1.1 DIVERGENCE TIME ESTIMATION

Assuming the evolutionary history of an alignment to be a tree, one can make the trivial observation that any two sequences from the alignment have a common ancestor. The most recent time when the ancestor existed corresponds to a tree branching event, in which the two lineages for the sequences were split apart. This type of evolutionary events is referred to as a *divergence* event, and the time of the event is referred to as the *divergence time*. The estimation of the divergence times in large samples is the central topic of this work.

The issue of divergence times estimation has been addressed since the very beginning of evolutionary studies. A well-known example is the analysis of lineages divergence in apes [Hasegawa et al., 1985; Moorjani et al.,

2016]. The time scale of the events being studied varies dramatically — from the divergence times of species from different kingdoms with a divergence time scale of hundreds of millions of years [Doolittle et al., 1996], to real-time studies in viral populations, with typical divergence times of tens to hundreds of years [Leitner and Albert, 1999; Suzuki and Nei, 2002]. A good example to illustrate the results of the enormous number of studies is the Time Tree project [Hedges et al., 2006; Kumar et al., 2017], which does a unique job of great importance to map and time-stamp the whole biodiversity on Earth.

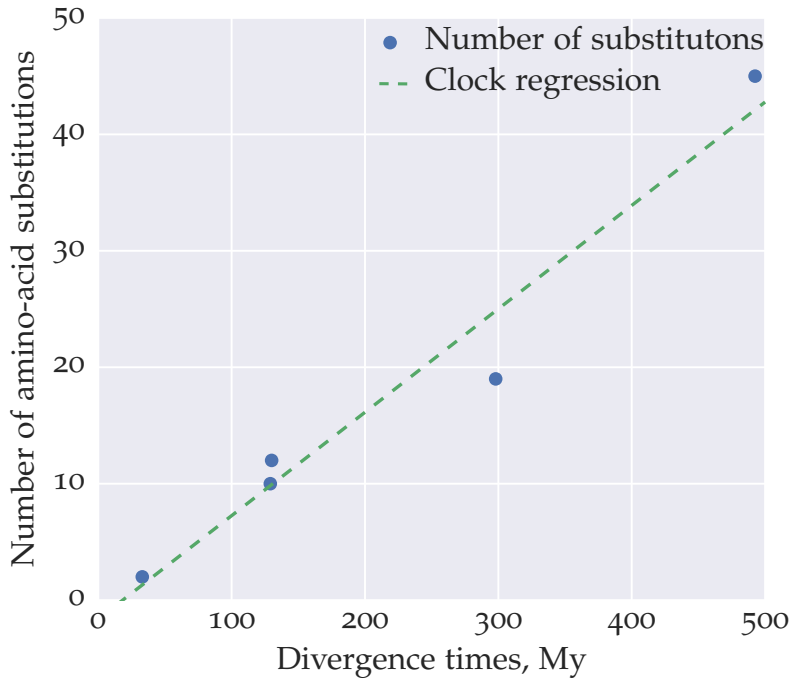

Figure 2: An illustration of the molecular clock. The figure represents the number of substitutions in a cytochrome C sequence in different species versus the species divergence times. Data adapted from [Margoliash, 1963], Table 1. The number of substitutions increases almost linearly with the divergence times, which inspired the molecular clock hypothesis in 1963.

According to observations performed in the beginning of the 1960s by different authors [Margoliash, 1963; Zuckerkandl and Pauling, 1965], the number of substitutions in proteins from different species increases almost linearly with times passed since the divergence of these species. One such

measurement is illustrated in figure 2. The figure was built from the data of Margoliash [1963], where the divergence times and the number of substitutions in cytochrome C were given for human, horse, rabbit, pig, tuna, chicken, and yeast. This and similar observations led to the conclusion that the number of substitutions in protein (and later in DNA) sequences accumulate linearly with time. Therefore, the mutations observed in these molecules can be used to estimate divergence times in phylogenetic trees. This idea gave rise to the *Molecular clock* hypothesis, which states that the substitutions accumulate steadily in time, like the ticking of a clock. Since then, the molecular clock has become a routine instrument to determine the divergence times at different timescales [Yoder and Yang, 2000]. Another conclusion that can be drawn from figure 2 (and other findings) is that the *molecular clock* is not a perfect timepiece, but rather a stochastic clock, in which the substitutions accumulation is a random process.

1.2 MOLECULAR CLOCK MODELS

The Strict Clock

The first molecular clock methods assumed a constant and universal substitution rate in all species. Fossil records were used to determine the rate and therefore to calibrate the known phylogenetic trees to the time scale. This type of models is referred to as the "Strict-clock" model, which has only one parameter: the rate of evolution. The strict clock model is calibrated using the known dates from the fossil records and then applied to the unknown dates by using linear regression [Doolittle et al., 1996]. Later tests of molecular clocks [Langley and Fitch, 1974; Felsenstein, 1981] showed that this strong assumption is often violated. The substitution rate is constant only in closely related species, such as apes. Furthermore, it became ev-

ident that substitution rates can vary among different parts of a tree and among sites along a sequence. Several refinements of the original molecular clock have been developed [Kumar and Hedges, 2016] to account for the observed phenomena. Nevertheless, the strict clock model is still used as a null model for testing for the presence of rate heterogeneities.

Maximum-likelihood

Likelihood-based approaches to infer divergence times started from a work of [Felsenstein, 1981]. As follows from its name, the method uses the *maximum-likelihood* criterion to choose the best reconstruction. The basis of the method is to estimate the tree likelihood provided the data. The data in this case is the alignment and the calibration dates for some nodes in a phylogenetic tree.

The tree likelihood is the probability to observe the data (D) on a particular tree T:

$$\mathcal{L} = \Pr(D|T). \tag{1}$$

The likelihood is related to the probability of data observation via the Bayes theorem. Namely, the probability to observe the tree and the data together is defined by:

$$\Pr(D, T) = \Pr(D|T)\Pr(T) = \Pr(T|D)\Pr(D), \tag{2}$$

where the first term in the multiplication is the tree likelihood. Note that contrary to intuition and the common language meaning, the tree likelihood is not the same as the probability of the tree, but rather the probability of observing the *data* given a certain *tree*.

The maximum-likelihood method (ML) considers the probability of each tree explaining the given data based on a model of evolution. The tree with the highest probability of explaining the data is chosen over others. In other words, it compares how the observed data is predicted by different trees and chooses the most suitable one from among all the trees.

This classical implementation of the ML is very computationally intensive, as it attempts to infer phylogenies and divergence times as a joint optimization problem. Therefore, it needs to explore a large subset of tree topologies from the tree space.

The main advantage of the ML is that it uses probabilistic models of sequence evolution, which take into account nucleotide substitutions and substitution rates. The described classic implementation of the ML is still widely used by some phylogenetic analysis tools [Rambaut, 2000; Sanderson, 2003]. However, at the end of the 1990s and the beginning of the 2000s, the Bayesian methods took over.

Bayesian

The Bayesian methods for divergence times estimation were introduced by the works of [Thorne et al., 1998; Kishino et al., 2001]. A Bayesian method is similar to the ML. It also uses the probabilistic criterion to search for the best tree. However, unlike the ML, the criterion to be maximized is the probability of a *tree* conditional on the *data*. According to the Bayes theorem, this probability is:

$$\Pr(T|D) = \frac{\Pr(D|T)\Pr(T)}{\Pr(D)}. \tag{3}$$

The denominator is the probability of observing the data, which can be represented as the marginalization over the tree priors: $\Pr(D) = \sum_i \Pr(D|T_i)\Pr(T_i)$. Since it does not depend on the choice of a tree, it can be omitted from the

optimization problem. Therefore, the Bayesian reconstruction finds an ensemble of trees that maximize the expression: $Pr(T|D) \propto Pr(D|T)\,Pr(T) \rightarrow \max$. This is the main difference between the Bayesian and the ML methods: the ML discards the prior probability $Pr(T)$ and maximizes the likelihood, whereas the Bayesian converts the prior to the posterior and maximizes the latter. To enable an efficient search in the tree space, the Bayesian usually uses the Markov Chain Monte Carlo methods (MCMC). The core idea of the MCMC is to sample from the posterior distribution of the hypothesis (in this case, the trees). If the number of samples taken is big enough, it becomes possible to make probability statements about the true tree. For example, if 90% of the samples from the posterior distribution have the {Human, Chimp} split, then we can say that the probability of this split being in a true tree is 90%. Obviously, if the uncertainty in such a prediction goes down as the number of samples increase.

The three main advantages of the Bayes approach over the ML in phylogeny reconstruction are that (i) it allows the inclusion of prior knowledge on a trees distribution, (ii) it is more computationally effective through the realization of MCMC methods, and (iii) it samples an ensemble of trees rather than search for a single phylogeny. The Bayesian methods allow for greater flexibility in accounting for uncertainty for the substitution rates, as well as for "relaxing" the strict molecular clock. It also accounts for the non-idealities in the reconstructed tree topologies [Drummond et al., 2012].

Among the software packages for molecular clock analysis, BEAST is one of the most sophisticated tools [Drummond et al., 2012]. BEAST samples many possible histories to evaluate posterior distributions of divergence times, evolutionary rates, and many other parameters. BEAST implements a large number of different phylogenetic and phylo-geographic models.

A note on phylogeny reconstruction

Most of the modern approaches [Chor et al., 2006; Drummond et al., 2012; Ho and Duchêne, 2014; dos Reis et al., 2015] aim to reconstruct tree topology along with the molecular clock. The solution of this joint problem indeed results in a phylogeny that is at the global optimum with respect to the input data and the criterion used. This global optimization problem consists of two major aspects: to reconstruct tree topology, and to optimize branch lengths of the tree to satisfy the molecular clock. Reconstructing tree topologies alone is, however, a mathematically and computationally complicated problem. There are several factors that define its complexity:

First, the evolutionary history of an alignment may be described by several phylogenies. Moreover, there is no way of inferring the real phylogeny. Therefore, some criterion is needed to choose one phylogeny from among all the possible ones. In other words, there should be a way of saying that, for a particular alignment, a particular tree is better than another. Different reconstruction algorithms are based on different criteria to compare the trees thus resulting in different phylogenies. The most frequently used criteria are the minimum evolution, the maximum parsimony, and maximum likelihood.

Second, the size of the tree space is exponential on the number of leaves, which makes the brute-force search over all trees computationally prohibitive. Therefore, efficient algorithms to search the tree space are needed. Indeed, the tree space increases exponentially with the number of nodes and hence it is unfeasible to apply the brute-force search to find the most appropriate tree. Namely, there are $\frac{(2n-3)!}{2^{n-2}(n-2)!}$ different binary tree topologies with n leaves.

These factors result in the exponential average run-time for most of the modern phylogeny packages based on maximum likelihood [Chor and Snir,

2004; Chor et al., 2006] or the Bayesian approach [Drummond et al., 2012]. Exponential run-time complexity results in run-times of days to weeks for moderately large data sets of a few hundred sequences.

This makes them impractical to be applied to large data sets, which nowadays grow very fast thanks to the next-generation sequencing. For instance, during the recent outbreaks of EBOV and the Zika virus, hundreds of sequences were generated and needed to be analyzed in near real time to inform containment efforts. Similarly, the GISRS network for the surveillance of seasonal influenza virus sequences produces hundreds of viral genomes per month. Doing a timely analysis of the data with the Bayesian methods such as BEAST is unfeasible.

1.3 MOTIVATION

Efficient heuristics are needed to cope with the growing data sets available today. The goal of my research has been to develop a fast and robust method for divergence times analysis in large alignments of homologous sequences, where other modern methods become impractical or computationally prohibitive. The primary goal is to study the viral evolution, which is observed in real-time and produces large amounts of data.

1.4 AN OUTLINE OF TREETIME

This work presents an approach to inferring the divergence times for sequence alignments with known evolutionary history. I have developed a new framework called `TreeTime`, which combines efficient heuristics with probabilistic sequence-based inference.

`TreeTime` infers maximum-likelihood *time trees* with a few thousand tips within a few minutes. `TreeTime` was designed for application in molecular

epidemiology and the analysis of rapidly evolving heterochronous viral sequences. It is already in use as an integral component of the real-time time outbreak tracking tool-kit next-strain [Neher and Bedford, 2015]. The main applications of `TreeTime` are ancestral state inference, evolutionary model inference, and time tree estimation.

Since `TreeTime` is an ML-based framework, I discuss the theoretical aspects of the maximum-likelihood methods in Chapter 2. The `TreeTime` core algorithms, its function, and implementation will follow in the subsequent chapters.

2

MAXIMUM-LIKELIHOOD METHODS IN PHYLOGENETICS

In this chapter, I describe the maximum-likelihood tools that constitute the theoretical basis of the `TreeTime` algorithms. In the introduction, I already mentioned the ML method. It scans the tree space and chooses a tree that meets the *maximum-likelihood* criterion. In other words, this method searches for the tree that maximizes the likelihood function:

$$\Pr(D|T) \rightarrow \max$$

The central part is how to calculate the tree likelihood. The calculation requires a model for sequence evolution. Therefore, I describe one of such models first. Then, I provide the mathematical description of the algorithm to compute the tree likelihood. In the following part, I also provide ML algorithms for ancestral sequences reconstruction and branch lengths optimization. Most of the theory from this chapter has been thoroughly developed by J. Felsenstein and other authors in the period from the late 1980's till the beginning of 2000's.

2.1 A MODEL OF SEQUENCE EVOLUTION

To estimate the likelihood of a given phylogenetic reconstruction, one needs a model that describes the possibility to realize a current evolutionary scenario. Such models usually operate on DNA or protein sequences as on strings of characters. Each character (the site of a DNA or a protein sequence) can be in one of the pre-defined states. For DNA sequences, there

are are four possible states: A, C, G, and T. For protein sequences, there are 20 possible states, which correspond to the one-letter codes for the amino-acids. Substitution models describe relative transition probabilities between the possible states in the process of evolution. The transition probabilities between the character states are usually written in the form of a matrix $P_{ij}(t)$, which denotes the transition from state j to state i in a certain period of time t. This study is based on the class of time-reversible models. These models assume that the character concentrations in the genome are in equilibrium at each point in time. Therefore, the fluxes between different character states are balanced in time, which is usually expressed as the detailed balanced condition. Given that the character concentrations are in equilibrium, denote these concentrations as π_i. Then, the detailed balance is written as

$$\pi_i \Pr(j|i,t) = \pi_j \Pr(i|j,t). \tag{4}$$

Another important assumption for sequence evolution models is that the substitution process (transition from one character to another) occurs randomly and independently. Furthermore, the constant substitution probability along tree branches is assumed. The class of the evolution models developed under these assumptions is referred to as *General-Time-Reversible* models (GTR models) [Yang, 2006]

GTR models describe the transition process as the time-homogeneous Markov process. According to the process, each site in the sequence is treated as a random variable, which can be in a finite discrete number of states. The Markov process specifies the transition probabilities from one state into another in a certain period of time t. These probabilities are col-

lected in the transition probability matrix $P_{ij}(t)$. To ensure the probabilistic nature of the P_{ij}, its rows should sum to 1:

$$\sum_{j=1}^{n} P_{ij}(t) = 1, \ \forall i \in \{1..n\}$$

and $P_{ij}(t) > 0 \ \forall t > 0$. It should also fulfill the Chapman-Kolmogorov equation: $P(t+s) = P(t)P(s)$, and the initial conditions: $P_{ij}(0) = I$, where I is the identity matrix. For small values of t, the $P_{ij}(t)$ can be expanded into the Taylor series up to the first derivative: $P(t) = P(0) + Qt$, where $Q = \frac{dP}{dt}$. From this expansion, we can write (using the Chapman-Kolmogorov equation):

$$P(t+dt) = P(t)P(dt) = P(t)(P(0) + Qdt)$$

According to the initial conditions, $P(0) = I$. Therefore, after trivial mathematical transformations, we obtain the differential equation for the transition probability between character states in a sequence over time t:

$$\frac{dP}{dt} = P(t)Q,$$

which is solved to

$$P_{ij}(t) = e^{Qt} \tag{5}$$

where Q_{ij} is the matrix that denotes the transition probabilities between states $j \to i$ *per unit of time*.

The eq. 5 is the central part of the GTR model. It provides a straightforward way to compute the transition probabilities between character states separated by time t once the Q matrix is known. The assumption of independent evolution between sites makes the transitions occurring at different sites to be independent probabilistic events. Therefore, this assumption

leads to a simple way to compute the transition probability from sequence S_1 to sequence S_2 over time t:

$$\Pr(S_2|S_1, t) = \prod_{\alpha} \left(e^{Qt}\right)_{i_\alpha j_\alpha} \tag{6}$$

where the product is taken over all sequence sites α.

The solution of the eqs. 5, 6 requires the exponentiation of the transition matrix Q_{ij}, which is the computationally expensive problem for an arbitrary matrix. However, given the time reversibility assumption, the Q_{ij} matrix can be diagonalized by applying the spectral decomposition:

$$Q = U\Lambda_{ij}(t)U^{-1},$$

where

$$\Lambda_{ij}(t) = \begin{pmatrix} \lambda_1 t & \dots & & \\ \vdots & \lambda_2 t & & O \\ & & \ddots & \vdots \\ O & & \dots & \lambda_n t \end{pmatrix}$$

and λ_i is the i^{th} eigenvalue of the matrix Q. Given this decomposition, the computation of the matrix exponent in the eq. 5 is trivial:

$$e^{Qt} = \sum_j \frac{(Qt)^k}{k!} = Ue^{\Lambda_{ij}(t)}U^{-1}.$$

The decomposition makes it possible to compute the transition probabilities over time t analytically.

The parametrization of the Q matrix

The transition rates matrix has a form ensuring the probability conservation:

$$\sum_i P_i = 1 \Leftrightarrow \sum_i Q_{ij} = 0, \tag{7}$$

and the detailed balance:

$$Q_{ij}\pi_j = Q_{ji}\pi_i, \tag{8}$$

where π_i are the stationary populations, obtained by solving $\partial P_i(t) = 0$. It is now easy to show [Felsenstein, 2003] that the rate matrix can be decomposed in terms of equilibrium state populations and a symmetric attempt matrix $W_{ij} = W_{ji}$ as

$$Q_{ij} = \pi_i W_{ij} \text{ for } i \neq j \tag{9}$$

$$Q_{ii} = -\sum_{j \neq i} Q_{ji}. \tag{10}$$

For short times ($t << 1$), the transition from state j to state i can be obtained by using the Taylor expansion of the matrix exponent:

$$P_{ij}(t) = \left(e^{Qt}\right)_{ij} \approx (1 + Qt)_{ij} = \delta_{i,j} + Q_{ij}t = \delta_{i,j} + \pi_i W_{ij} t, \tag{11}$$

where the expansion implies that time t is very short in the scale of the mutation rate, i. e. of the inverse eigenvalues of the GTR matrix. In particular:

$$P_{ii}(t) \approx 1 - \sum_{j \neq i} \pi_j W_{ji} t, \text{ and} \tag{12}$$

$$P_{ij}(t) = \pi_i W_{ij} t \text{ for } i \neq j. \tag{13}$$

2.2 TREE LIKELIHOOD CALCULATION

The GTR essentially models the probabilities of a sequence to evolve into another sequence in over a certain period of time. This provides a direct way to calculate the likelihood for a particular phylogenetic tree reconstruction. Recapitulate, that the tree likelihood is the probability to observe data on a certain tree:

$$\mathcal{L} = \Pr(D \mid M),$$

The following computation of the tree likelihood repeats the logic and assumptions from [Felsenstein, 2003]. The mathematical derivation is partially based on the work of [Pupko et al., 2000]. The likelihood $\mathcal{L}$ is usually calculated under assumptions that

(i) The evolution is independent in different lineages

(ii) The evolution in different sites in sequence is independent

As before, the second assumption allows to decompose the likelihood of a sequence evolution into the product of likelihoods for character evolution:

$$\mathcal{L} = \Pr(D|\, T) = \prod_{\alpha} \Pr(D_\alpha|\, T), \tag{14}$$

where D_α is the data at α's site. Therefore, one can compute the likelihood for each single character first, and then multiply the results to get the full likelihood.

In figure 3, a small example for likelihood computation is shown. The tree likelihood is the probability to observe the given sequence states at the tree leaves, given the tree:

$$\mathcal{L} = \Pr(D_\alpha|\, T) = \sum_{x} \sum_{y} \Pr(C, C, A, x, y|\, T), \tag{15}$$

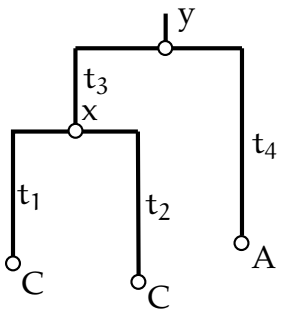

Figure 3: Illustration to the tree likelihood computations. The likelihood of the tree is the joint probability of observing the character states at the leaves of the tree, given the topology and branch lengths.

where the summation is performed over all possible character states x, y of the internal node in the tree. Given the assumption of the independent evolution in different lineages, the evolution in different tree branches is independent. So, the probability of observing all tree states simultaneously can be decomposed into the product of probabilities:

$$\Pr(C, C, A, x, y|\, T) = \Pr(A|\, y, t_4)\Pr(x|\, y, t_3)\Pr(C|\, x, t_1)\Pr(C|\, x, t_2)$$

Using the decomposition above, the tree likelihood can be computed using the dynamic programming algorithm. The algorithm is based on the iterative computations of the likelihoods of subtrees of the given tree. For the example in figure 3, the likelihood of the subtree rooted at node x is then $\Pr(D_x|x) = \Pr(C|\, x, t_1)\Pr(C|\, x, t_2)$, which denotes the probability of everything below or at the node x (the subtree of node x), conditional on the state x. D_x is the data of the x subtree, i.e. the sequences observed at the leaves of the x subtree. The likelihoods of the parent subtrees can be recursively expressed through that of the child subtrees. For example, for a node s, which has children l and m with branch lengths between parent and chil-

dren t_l and t_m, and pre-computed likelihoods $\mathcal{L}_l$ and $\mathcal{L}_m$ (see figure 4), the likelihood is given by the expression:

$$\mathcal{L}_k(s|p, D_s, t_s) = \Pr(s|p, t_s) \left(\sum_x \mathcal{L}_l(x|s) \right) \left(\sum_y \mathcal{L}_m(y|s) \right) \tag{16}$$

This likelihood is the likelihood of the s subtree given the subtree data D_s, and the state of the parent node p. The quantity $\Pr(s \mid p)$ is the probability of observing the transition from the parent state p into the child state s, which is given by the GTR model eq. 5:

$$\Pr(s|p, t_s) = \left(e^{Q_{ij} t} \right)_{i_\alpha j_\alpha},$$

where α denotes the site in the sequence, which the likelihood is computed for. i_α and j_α are the character states on both sides of the tree branch at the site α. It is convenient to represent the subtree likelihood as a vector,

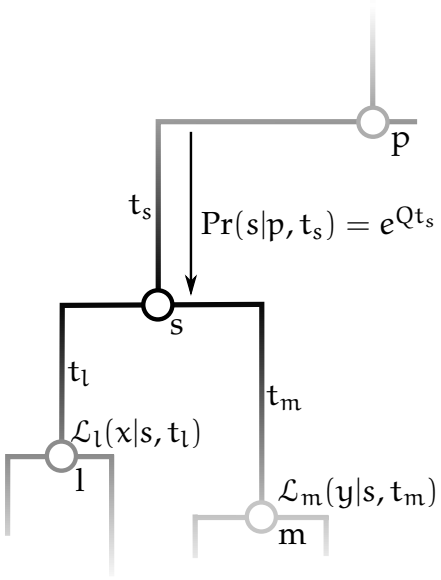

Figure 4: Recursive computation for the likelihood of a subtree s, if the likelihoods of the children is known.

which length is the number of possible states. The i^{th} element of the vector represent the likelihood of the subtree conditional on the parent state i:

$\vec{\mathcal{L}_k} = |\mathcal{L}_k(1), \mathcal{L}_k(2), ..., \mathcal{L}_k(n)|$. For instance, for nucleotide sequence, there are four possible states: A,C,G,T, which likelihoods are represented as elements of a four-dimensional vector: $\vec{\mathcal{L}_k} = |\mathcal{L}_k(A), \mathcal{L}_k(C), \mathcal{L}_k(G), \mathcal{L}_k(T)|$. From the above, it is now clear how to construct an algorithm, which computes the likelihood of the tree. It should start from the tips of the tree and gradually calculate the likelihoods of the internal nodes, visiting them in post-order. At each step, the likelihood of an internal node is computed from the data, obtained at previous steps. At the first iteration step, the initial values of the likelihood vector are defined as the probabilities to observe substitution from states p to the observed states i: $\mathcal{L}_p = |\Pr(i|p)|$, where s is the state of the leaf node. For example, if the state of the node is $A = 1$, then this vector contains the probability of mutation from any state to A: $\mathcal{L}_i = |\Pr(A|A), \Pr(A|C), \Pr(A|G), \Pr(A|T)|$. After the likelihoods of the tree leaves are defined in this way, the likelihood of the internal nodes is computed iteratively according to the procedure described. At the last step, the root state should be corrected to the stationary concentrations of the character states to eliminate the "sampling bias" and thus maintain the time-reversibility of the solution. The root likelihoods should therefore be multiplied by the stationary concentrations π_i defined in GTR model. In the end, the total tree likelihood, which accounts for a single character evolution, is the sum over all possible root states:

$$\mathcal{L}(D_\alpha|\, T) = \sum_x \pi_x \mathcal{L}_{\alpha,r}(x) \tag{17}$$

To get the total tree likelihood, one needs to compute the $\mathcal{L}(D_\alpha|\, T)$ for each character independently, and then multiply all single-character likelihoods together, according to the eq. 14. Thus constructed algorithm comprises the classic dynamic programming approach. This particular implementation requires one tree traversal per each character, and hence it has linear complexity on the number of nodes N and the sequence length L: $T \propto \mathcal{O}(NL)$

2.3 ANCESTRAL SEQUENCES RECONSTRUCTION

The algorithm described beforehand gives the straightforward way to estimate likelihoods of different phylogenies. Resulting likelihoods are marginalized over all possible states of the internal nodes, i.e. they take into account every possible combination of ancestral states.

Another fundamental problem, which is approached by the ML methods, is ancestral states reconstruction. In the scope of maximum-likelihood methods, this problem is formulated as follows: to maximize the tree likelihood in respect to states of the internal nodes. Essentially, it means that instead of summing over all possible internal node states, one should maximize the likelihood function in respect to those states:

$$\mathcal{L}(\{x_i\}) = \Pr(D_\alpha|\ T) \to \max, \tag{18}$$

where $\{x_i\}$ denote the possible states of all internal nodes.

Joint reconstruction

The joint reconstruction is accomplished using the same logic as for the tree likelihood computation. The likelihood of an internal node is determined by taking maximum over all possible states rather than by summation. This modifies the eq. 16 to:

$$\mathcal{L}_k(s|p, t_s) = \Pr(s|p, t_s) \left(\max_x \mathcal{L}_l(x|s)\right) \left(\max_y \mathcal{L}_m(y|s)\right) \tag{19}$$

The eq. 19 defines the maximum likelihood of a subtree conditional on the parent node state, and assumes that all child likelihoods are maximized. The state of the root is defined as one, which defines the maximum of the root likelihood function. This maximum is iin turn, the likelihood of the

particular ancestral states reconstruction. To reconstruct states of all other internal nodes, one more tree traversal is needed. The second iteration starts at the root node, which state is defined by

$$\mathcal{L}_R = \max_x \pi_x \mathcal{L}_r(x). \tag{20}$$

The value x, which corresponds to the maximum of $\mathcal{L}$ function is assigned to be the state of the root node. Then, all other internal nodes in the tree are visited in pre-order (parents first). At each step of the second iteration, the states of the internal nodes are reconstructed from the parent node state i and the likelihood of the node's subtree conditional on the parent: $\vec{\mathcal{L}_k} = |\mathcal{L}_k(1), \mathcal{L}_k(2), ..., \mathcal{L}_k(n)|$. The latter expression is the likelihood vector described above. The i^{th} position of this vector is the likelihood of the subtree given the parent state i, so the subtree likelihood is reconstructed by simply choosing the i^{th} element of this vector. Obviously, the character state of the internal node, which defines the value of this reconstructed likelihood should be also reconstructed. It may be stored as a separate vector C_k along with the $\mathcal{L}_k$ vector. The algorithm describes the the single-character reconstruction. So, to accomplish the full sequence reconstruction, each character should be reconstructed independently, and then the full likelihood of the reconstruction is obtained by multiplication over the character likelihoods. The complexity of the algorithm is obviously the same as for the tree likelihood computation, which is $\mathcal{O}(NL)$.

Marginal reconstruction

The joint reconstruction assigns maximum-likelihood states to all nodes at once. There is however, another way to find the maximum-likelihood states for the ancestral sequences. That is, for each internal node, find its maximum likelihood sequence marginal over all possible states of the other in-

ternal node sequences. The result of such reconstruction is the maximum likelihood assignment of internal node states conditional *only* on the leaf sequences. This type of ancestral sequence reconstruction is referred to as marginal reconstruction. The algorithm is similar to that for the joint reconstruction. It also requires the two tree traversals. The first tree traversal is similar to that of the tree likelihood determination with only difference that the likelihoods from all children nodes should be stored for the second traversal. The likelihood vector is also marginalized over all possible states of the child nodes, as shown in eq. 16. Note that at each iteration step, the likelihood is conditional on the leaves of the subtree of the particular internal node. All states of the intermediate nodes are marginalized. The likelihood is yet unconditional on the nodes of the complementary subtree, except for the root node, which subtree corresponds to the full tree. The root sequence is therefore determined straightforward, as given by eq. 20. Reconstruction of the other internal nodes is however, more complicated, because account for the complementary subtree data is required. This reconstruction is made in the second (pre-order) tree traversal. At each iteration step of this second tree traversal, the sequence likelihood conditional on the all leaves data is restored. This likelihood consists of the three parts. First, the likelihood of the node conditional on the parent state, and the states of all leaves of the node n subtree (D_n): $\mathcal{L}_n(n|p, t_n, D_n)$. It has been computed in the first tree traversal. This likelihood is conditional on the parent state. Ton resolve this condition, the two other likelihood inputs have to be taken into account: (i) the likelihood of the parent node conditional on the "upstream" subtree data (D_p), and (ii) the likelihood of the parent node conditional on the sibling node data (D_s). The three ingredients of the likelihood of an internal node n are sketched in figure 5. The likelihood of the node n is obviously

$$\mathcal{L}_n(n|D) = \mathcal{L}_n(n|p, D_n) \cdot \mathcal{L}_p(p|D_p \cup D_s),$$

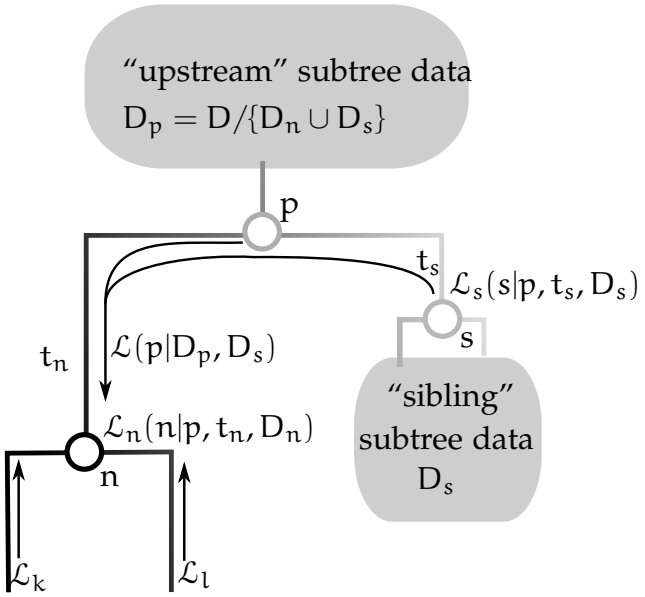

Figure 5: Reconstructing the node n ancestral state conditional on the leaf states D. The n state likelihood conditional on the node's subtree data D_n has been computed in pre-order iteration. The conditions of the complementary subtree are to be accounted for on the post-order iteration. The data of the complementary subtree is comprised of the two parts: (i) the sibling node s subtree data D_s and (ii) the "upstream" tree data $D_p = D/D_n \cup D_s$. Under the assumptions made, the $\mathcal{L}(n|D) = \mathcal{L}(n|P, D_n)\mathcal{L}(p|D_p, D_s)$

where the first term is the likelihood calculated in the first tree traversal, the the second term is the likelihood of the parent node given the complementary leaves:

$$\mathcal{L}_p(p|D_p \cup D_s) = \mathcal{L}_p(p|D_p) \cdot \mathcal{L}_p(p|s, D_s),$$

Using the time reversibility, the second multiplier is transformed into:

$$\mathcal{L}_p(p|D_s) = \mathcal{L}_s(s|p, D_s),$$

which is the same as the likelihood of the node n being computed in the first tree traversal. Finally, the node likelihood is:

$$\mathcal{L}_n(n_i|D = \{D_n \cup D_s \cup D_p\}) = \mathcal{L}_n(p_i|D_n) \cdot \mathcal{L}_s(s_i|p, D_s) \cdot \mathcal{L}_p(p_i|D_p) \quad (21)$$

where $\mathcal{L}_n(n_i)$ is the likelihood of the character at node n to be in i^{th} state. The states of the parent (p_i) and sibling (s_i) node are defined similarly. So, the final likelihood shown in eq. 21 is defined as a vector, which elements define the likelihood of a character to be in the state n_i. The maximum-likelihood state of the node n is then reconstructed by choosing the state i, which defines the maximal value of the $\mathcal{L}_n(n_i)$. The algorithm requires some complications compared to the joint reconstruction.

First, during the pre-order traversal, all likelihoods from left and right subtrees should be stored for the pre-order traversal. Second, the pre-order tree traversal requires additional computations rather than simple reconstruction of the sequence states. The algorithm's run-time complexity is nevertheless $\mathcal{O}(NL)$, which is the same as of other ML methods discussed so far.

2.4 BRANCH LENGTHS OPTIMIZATION

GTR models define the probability of two characters to be separated by time t. These probabilities can be calculated for every possible time thus leading to the probability distribution. For a single character, the analytical solution is trivial to find — it is defined by the eq. 5. In case of the equal characters, the distribution is a simple exponential decay. For non-equal characters, this distribution looks like $1 - e^{-t}$ function. (see figure 6, left panel). In case of multi-character sequence, and under the assumption of the independent substitution across sites, the probability of a branch to have length t is defined by multiplication of eq. 5 over all sites:

$$\Pr(t) = \prod_{\alpha} \left(e^{Qt}\right)_{i_\alpha j_\alpha}, \tag{22}$$

where for each site α transition from state j_α to state i_α is observed. The multiplication over many functions like on the left panel in the figure 6 results in a bell-shaped function, similar to one shown in the right panel. The prob-

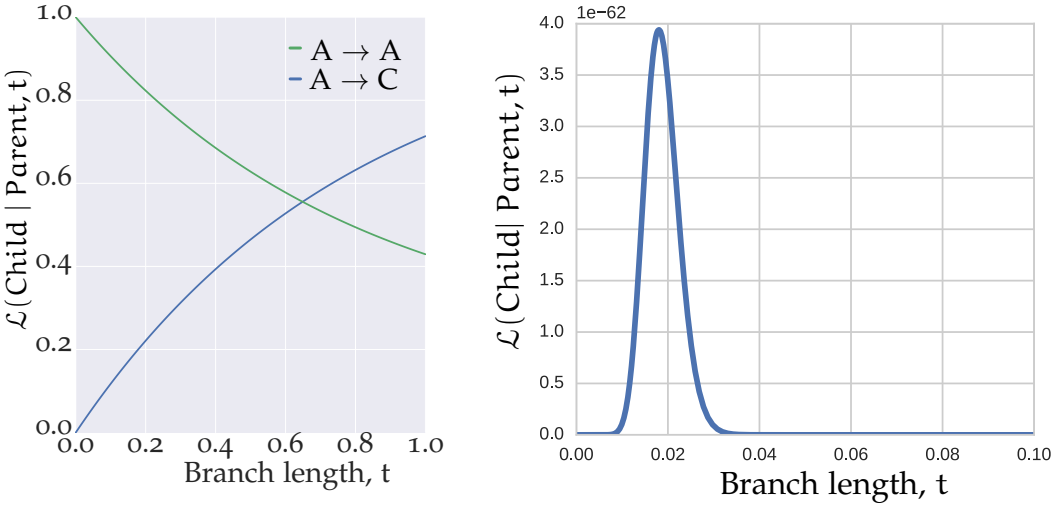

Figure 6: The likelihood distributions to observe the child given the parent and the branch length t.
Left panel shows the likelihood distributions for single character sequences computed from Jukes-Cantor model [Jukes and Cantor, 1969] for equal and different character states.
Right panel shows the branch length likelihood distribution for multi-character sequences. This distribution is multiplication over the relevant single character distributions. The example presented is the branch length distribution for two Influenza H3N2 sequences of the NA segment. The sequences are 1407 nucleotides long, the distance between them are 25 substitutions.

lem to optimize a branch length is to find the length t, which corresponds to the maximum value of the distribution 22.

The problem to find maximum-likelihood lengths for all tree branches is the coupled to the problem of ancestral sequence reconstruction. Indeed, the ancestral sequence reconstruction uses the GTR model to calculate the probability of each two sequences being separated by some time t. The time in this case is the branch length, therefore, the ancestral reconstruction relies on the branch lengths. On the other hand, the branch length optimization requires knowledge of the ancestral sequences to calculate the branch length

distributions. Traditionally, this problem is solved as the optimization problem with branch lengths and sequences together as free parameters:

$$\mathcal{L}(D|T) \to \max_{\{t_i\}}, \tag{23}$$

where optimization is made in respect to all branch lengths $\{t_i\}$, and marginalized over all possible internal node sequences. Given that the likelihood in eq. 23 is the continuous function on the branch lengths, its solution is usually searched as:

$$d\mathcal{L}(D|T) = 0.$$

For the methods in this work, we, however, designed a different approach. In our algorithms, we decouple the ancestral sequence reconstruction from the branch length optimization to conquer them separately, and optimize iteratively. At the beginning, the approximate ancestral sequences are reconstructed with or without usage of the initial branch lengths. The latter is accomplished using the Fitch parsimony algorithm [Fitch, 1971]. Then, the branch lengths are optimized independently followed by the reconstruction of the ancestral sequences. This procedure repeats iteratively until the ancestral sequences converge to their stationary values.

3

TREETIME

The maximum-likelihood methods described in the previous chapters provide us with the theoretical basis to discuss the algorithms that we have developed to build *time trees*. The algorithms are combined in a single package, which we refer to as `TreeTime`. `TreeTime` was developed with large heterochronous viral sequence alignments in mind. Currently, it is already used as the core component of the `nextstrain` real-time phylogenetic pipeline [Nextstrain, 2017].

Compared to other methods recently developed for rapid estimations of *time trees* [Britton et al., 2007; Tamura et al., 2012; To et al., 2016], treetime uses GTR models, thus allowing inference of ancestral sequences and coalescent models. `TreeTime` tries to strike a useful compromise between inflexible but fast heuristics and computationally expensive Bayesian approaches, which require extensive sampling from the tree space. The overarching algorithmic strategy is iterative optimization of efficiently solvable subproblems to arrive at a consistent approximation of the global optimum. While this strategy is approximate and often assumes short branch lengths, it converges fast for many applications. Trees with thousands of tips can be analyzed in a few minutes. The time tree inference and dating are typically faster than the estimation of the tree topology.

3.1 DIVERGENCE TIMES RECONSTRUCTION

Core algorithm

The core problem that is approached by the `TreeTime` is to find the maximum-likelihood divergence times in a given phylogenetic tree. The data used as input for the ML optimization are the multiple sequence alignment and its phylogeny. Sequences of the alignment are time-stamped (i. e. the sampling dates are known), thus providing information to build a time tree by introducing constraints for ML optimization. In other words, the `TreeTime` solves the following optimization problem:

$$LH(D|n_i, M) \rightarrow \max, \tag{24}$$

where D is the data, which comprises the alignment, the sampling dates and the tree topology. n_i are the internal node positions, and the M is the chosen model.

The core idea behind the ML inference of the divergence times is illustrated in figure 7. Knowing the sequences of the nodes n, m and the parent node p, the probability distributions for the branch lengths τ_n, τ_m are calculated from the GTR model. Given that the positions of the nodes n, m are fixed by their sampling times t_n and t_m, the branch length distributions define the likelihood of the parent node time t_p. The independence of the parallel lineage evolution leads to the parent node likelihood $\mathcal{L}(t_p|t_n, t_m)$ to be the multiplication of the children branch length distributions. The likelihood computed in this way is conditional on the positions of the nodes n and m only.

`TreeTime` implements the described procedure to calculate divergence times in a dynamic programming manner. The tree is traversed from children to parents. At each iteration step, the times of nodes are calculated

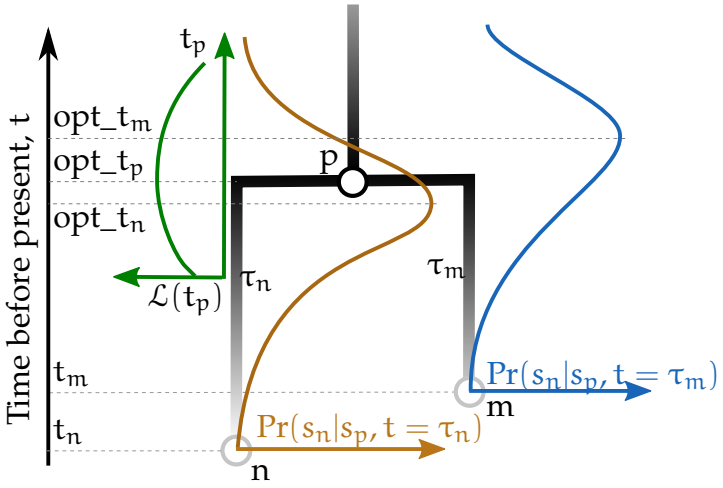

Figure 7: An illustration of the process to infer the ML divergence time of the internal node p, whose children sampling dates t_n and t_m are known. The GTR model defines the probability distributions for the branch lengths τ_n and τ_m, which, in turn, allows one to calculate the likelihood distribution $\mathcal{L}(t_p)$ of the parent node p time t_p. The resulting optimal position of the parent node opt_t_p is a "trade-off" between the optimal positions reported by the children: opt_t_n and opt_t_m.

based on the constraints introduced by the child nodes. The logic of the algorithm is similar to that of the ML ancestral sequence reconstruction. The key difference from the approach in Chapter 2 is that the present algorithm should account for the infinite number of possible node positions, whereas the former approach deals with the finite number of possible character states.

The algorithm requires two tree traversals. The first traversal is in post-order to build the subtree likelihoods conditional on the parent position. Then the root position is fixed followed by the tree traversal in pre-order to reconstruct the maximum-likelihood times of the other internal nodes.

Joint reconstruction

The joint reconstruction starts with the tree traversal in post-order. Terminal nodes are visited first. For these nodes, priors on the sampling dates are

determined. If the exact sampling date for a node is known, the prior is the delta-function distribution $\delta(t)$. Otherwise, an appropriate prior is built to account for known information of the sampling date, e. g., for cases when only the sampling year is known. In case no sampling date information is provided, the terminal node time is defined by its branch length distribution on the second tree traversal.

After the terminal node distributions are set, internal nodes likelihood distributions are built gradually based on the data from children. At each iteration step, the likelihood distribution of the internal node time is computed depending on the subtree data and the position of the parent node. The example of such computation is shown in figure 8. Post-order traversal ensures that the current node n is visited only after its children c and c_2. So, at the time n is visited, the likelihood distributions for c, c_2 are known. These distributions are conditional on the n's time t_n. Therefore, the joint likelihood of a subtree n conditional on the t_n is as follows:

$$\mathcal{L}_n(t_n) = \prod_c \mathcal{L}(t_c|D_c, t_n) \tag{25}$$

where the product is taken over all children. To propagate the likelihood to the parent, the condition on t_n is changed to the condition on the parent node time t_p:

$$\mathcal{L}(t_n|t_p, \{D_c\}) = \max_{\tau_n} \left[\Pr(\tau_n = t_p - t_n) \prod_c \mathcal{L}(t_c|D_c, t_n = t_p - \tau_n) \right] \tag{26}$$

The first term in the eq. 26 is the probability that the sequence of the parent node p is evolved into the sequence of the child node n over the time τ_n. This probability was defined by the GTR model (see eq. 6). The maximum is taken over all possible values of the branch length τ_n. Thus, eq. 26 defines the maximum likelihood distribution of the n time conditional on the time

of parent node time t_p. The likelihood distribution eq. 26 implies also that all subtrees of node n are assigned with the maximum likelihood times.

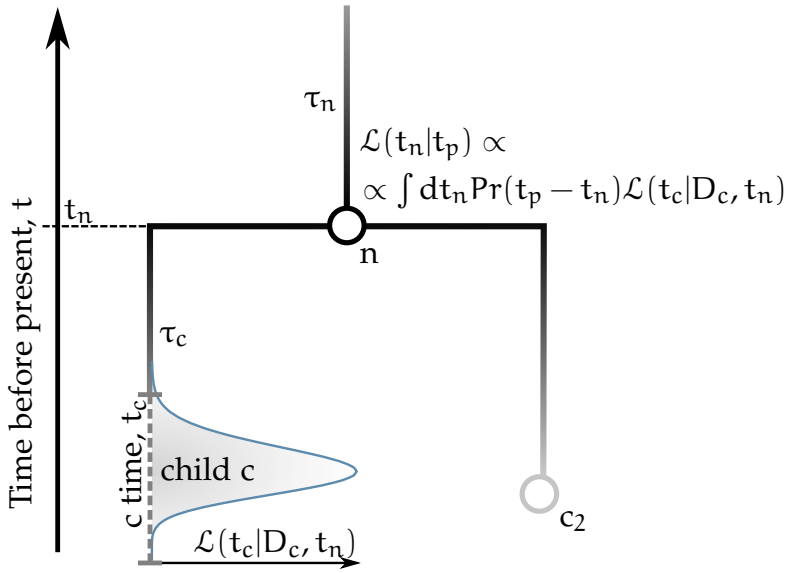

Figure 8: To the calculation of the internal node times. The likelihood of the node n is computed conditional on the parent node location t_p and hence the branch length τ_n. The subtree likelihoods of the children have been computed on the previous iteration step.

In the last step of the post-order traversal, the tree root is visited. Its time is assigned to maximize the subtree likelihoods of the root clades:

$$\mathcal{L}(t_n|t_p, \{D_c\}) = \max_{t_n} \left[\prod_c \mathcal{L}(t_c|D_c, t_n = t_p - \tau) \right] \tag{27}$$

The eq. 27 determines the joint likelihood of a given tree conditional on the alignment, and all given sampling dates of the tree leaves. The value of t_n, which defines the maximum of the distribution in eq. 27, is the maximum likelihood time for the tree root. This is the time of the most recent common ancestor of the alignment (T_{mrca}).

To reconstruct all other divergence times, the pre-order tree traversal is needed. In each step of the second traversal, the internal node times are reconstructed using (i) the pre-computed likelihood distributions and (ii) the position of the parent node t_p. The former defines the likelihood distribu-

tions for node time t_n conditional on the parent node time t_p: $\mathcal{L}_n(t_n|t_p)$. So, using the position of the parent node p, the likelihood and the position of the child node n are easily reconstructed. In this way, the times of the internal nodes are reconstructed starting at the root node and finishing at the tree leaves.

Marginal reconstruction

Marginal reconstruction of the divergence times provides likelihood distributions conditional on the sampling dates of the leaves and marginalized over all other internal node positions. In contrast to the joint reconstruction, it provides the likelihood distributions for each internal node position, which can be used, among other things, for error rate or confidence intervals estimation (note that the joint reconstruction provides the maximum likelihood position only). The algorithm requires more thorough computations to be performed, though. The logic of the algorithm implementation is similar to that of the joint reconstruction. It also requires two tree traversals. The first one is in post-order starting to build the likelihood distributions of the subtrees conditional on the parent. The second is in pre-order to reconstruct the internal nodes likelihood distributions from parent node times. On the post-order tree traversal, the likelihoods of internal nodes are constructed as follows. The likelihood for node n is marginalized over all possible times of the internal nodes in the subtree. This marginalization modifies the equation 26 to the following expression:

$$\mathcal{L}(t_n|t_p,\{D_c\}) = \int_\tau d\tau \left[\Pr(\tau = t_p - t_n) \prod_c \mathcal{L}(t_c|D_c, t_n = t_p - \tau)\right] \quad (28)$$

where, as before, the $\Pr(\tau = t_p - t_n)$ is the probability of the parent sequence evolved into the child sequence over time τ. Note that the marginalization of

the node position is performed only on the internal nodes of the current subtree. The additional marginalization, on the complementary subtree, should be made in the second (pre-order) tree traversal. At the last step of the post-order traversal, the root node time distribution is defined as

$$\mathcal{L}_R(t_R) = \prod_c \mathcal{L}(t_c|t_R)$$

The time of the root node is then the maximum of the above distribution. Similarly to the marginal reconstruction of ancestral sequences, the distributions from the left and right subtrees should be stored in order to provide the "messages" from the complementary subtrees (see figure 5 and the explanation in the text for details). In order to complete the marginalization and propagate the condition from the parent to the leaves, the data from the "upstream" tree, as well as that of the "complementary" subtree, should be taken into account. This is accomplished by combining the likelihood distributions from the "upstream" tree and from the complementary subtree into one likelihood distribution

$$\mathcal{L}_p(t_p|D_p \cup D_s) = \mathcal{L}_s(t_s|t_p, D_s) \cdot \mathcal{L}_p(t_p|D_p)$$

Note that to account for the conditions on the missing data only, one should track the subtrees that have not yet contributed to the likelihood distribution of the particular node time. Thus, for the root left child, the "upstream tree" is the right subtree and, for the left child, the "upstream tree" is the right subtree. Finally, to compute the likelihood distribution of an internal time t_n, it should be marginalized over all possible positions of the parent

node (conditional on data D_s and D_p). Therefore, the final likelihood of the internal node is

$$\mathcal{L}_n\left(t_n|\{D_n \cup D_p \cup D_s\}\right) = \int d\tau \mathcal{L}_n(\tau|t_p, D_n) \cdot \mathcal{L}_p(t_p = t_n + \tau|D_s \cup D_p) \quad (29)$$

Two things to note in the above equation are: (i) the first term under the integration is the function of τ, because the $\mathcal{L}_n$ is conditional on the parent node time t_p through the branch length; and (ii) the inverse direction of time in the pre-order traversal changed the sign $t_p - \tau$ to $t_p + \tau$, which converted the convolution function in eq. 28 to convolution-like integral in eq. 29. The integral eq. 29 defines the final likelihood distribution of the node n. This distribution is built for each internal node by traversing the tree in pre-order, starting from the root and finishing at the leaves. In this way, all distributions for the internal node positions are reconstructed, which completes the description of the core algorithm of the `TreeTime`.

Tree pre-processing

In the description of the `TreeTime` core algorithm, I assumed that the probability distributions for all branch lengths are known. Moreover, I made some implicit assumptions without explanation. In this paragraph, the missing discussion is presented.

Before divergence times can be reconstructed, a few tree preparations should be done. First, as noted above, the branch length distributions should be calculated. This, in turn, requires the knowledge of the ancestral sequences. This is done by the iterative inferring of the ancestral sequence coupled to the branch length optimizations as shown in Chapter 2. The resulting tree has the maximum-likelihood branch lengths and ancestral sequences. All branches of the tree are in the units of the substitu-

tion probabilities. The `TreeTime`, however, is designed to infer the dates of the internal nodes given the sampling dates of (some) leaves. The sampling dates are usually provided in some human-readable calendar format. Therefore, a conversion between the branch length units and the calendar dates is needed. The natural conversion of this sort is the molecular substitution rate. Assuming the *molecular clock*, the number of substitutions should increase linearly with time. Hence, the substitution rate is simply the regression coefficient between the evolutionary distance from root and the sampling dates. Thus defined, the substitution rate can be inferred from the input tree. An example for substitution rates assessment is shown in figure 9.

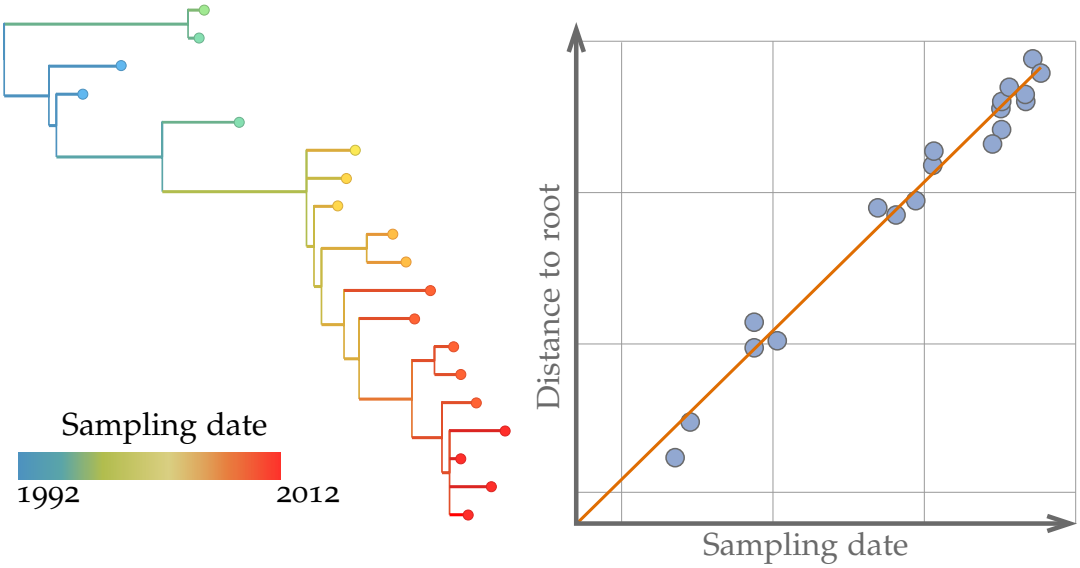

Figure 9: The molecular clock for Influenza H3N2, an HA segment. Twenty sequences were randomly chosen from a bigger alignment.

One should note that, to infer the substitution rate by the described procedure, the sequence samples should be taken at various times including those close to the tree root. For viral samples, this is usually not a problem, because of the rapid evolution, which causes 1% divergence accumulation within just a few years. For other organisms, however, the inference might not be that straightforward. For these cases, `TreeTime` provides the possibil-

ity to omit the substitution rate inference and allows instead for providing a substitution rate obtained elsewhere.

3.2 TREETIME ADDITIONAL FUNCTIONALITY

Efficient search for the optimal root

As has been shown above (see figure 9), the molecular clock for a given tree is built by relating the root-to-tip distance of the tree leaves to the sampling dates. This molecular clock regression can be used to find a better root position in the tree. The best root position is searched my maximizing the molecular clock correlation coefficient. The correlation coefficient is given by

$$r^2 = \left(\frac{N \sum_i t_i d_i - \sum_i t_i \sum_i d_i}{\sqrt{N \sum_i d_i^2 - (\sum_i d_i)^2} \sqrt{N \sum_i t_i^2 - (\sum_k t_i)^2}} \right)^2 \tag{30}$$

where the sum runs over all tips i of the tree and t_i and d_i are the sampling date and the distance from the root to node i, respectively. The regression and r^2 depend on the choice of root via the d_i. The naive implementation of the maximization of the r^2 takes $\mathcal{O}(N^2)$ time to compute: for each of (approximately) $2N - 1$ internal nodes, N leaves should be scanned to determine the d_i for the current internal node. This implementation is usually used even in popular phylogenetic software such as TempEst [Rambaut et al., 2016] or LSD [To et al., 2016]. However, the optimization problem can be solved in $\mathcal{O}(N)$ time using the dynamic programming approach that I have developed for `TreeTime`. It requires two tree traversals: one in post-order to compute the d_i and other auxiliary values, and one in pre-order to

reconstruct the r^2 for each node based on the pre-computed values. Denote the values:

$$\Theta_i = \sum_i d_i, \quad \gamma_i = \sum_i d_i t_i, \quad \delta_i = \sum_i d_i^2, \quad \tau_i = \sum_i t_i \tag{31}$$

During the post-order tree traversal, the following auxiliary values are computed iteratively, using the result of the computation for the child nodes:

$$\begin{aligned}
\Theta_{i,Nst} &= \sum_{children} (\Theta_{i,cst} + n_c L_c) \\
\gamma_{i,Nst} &= \sum_{children} (\gamma_{i,cst} + \tau_{i,cst} L_c) \\
\delta_{i,Nst} &= \sum_{children} (\delta_{i,cst} + 2 L_c \Theta_{i,cst} + n_c L_c),
\end{aligned}$$

where subscript Nst denotes that the values computed for node N take into account only the data from the subtree of N, while subscript cst stands for the values pre-computed for the child node subtree. n_c is the number of the leaves in the child node subtree and L_c is the branch length between child node c and node N. In the last step of the post-order tree traversal, the values for the root node are computed. Given that the subtree of the root node is the complete tree, the st subscript can be omitted for the root node and therefore all data for computing the regression for the root node is obtained. To get the same data for all the other internal nodes in the tree, a second tree traversal is performed. During the second tree traversal, each node is visited in pre-order. The following values are computed for each node N:

$$\begin{aligned}
\Theta_{i,N} &= \Theta_{i,p} + (n_{up} - n_{down}) L \\
\gamma_{i,N} &= \gamma_{i,p} + L (\tau_i - 2\tau_{i,Nst}) \\
\delta_{i,N} &= \delta_{i,p} + 2L\Theta_{i,p} - 4 (L\Theta_{i,Nst} + L n_{down}) + n L^2
\end{aligned}$$

where the N subscript shows that the corresponding values are computed for node N and they account for all the leaves of the subtree. The p subscript stands for the parent node and the Nst subscript shows where values computed for the subtree of node N are used. n is the total number of leaves in the tree, n_{down} is the number of leaves in the N subtree, and n_{up} is the number of leaves in the subtree complementary to N: $n_{up} = n - n_{down}$. L is the branch length between node N and its parent. Given the computed values, the regression coefficient for each node N is calculated along the second tree traversal, to get the values r^2_N. To take into account that the best root can be in a branch between existing nodes, the above expressions are adapted using the following reasoning. If the root is assigned to node N_p, it will result in the correlation coefficient $r^2_{N_p}$. By moving the root along the branch by length L, the new root would end up at node N and the regression will be r^2_N. If the root is moved by an intermediate value x, then the r^2 will result in a continuous function $r^2(x)$, defined separately for each branch in the tree. Therefore, by finding the maximum for the $r^2(x)$ on the closed segment $x \in [0, L]$, the optimal root position is obtained for each branch separately. The r^2 is the rational function of x, defined as

$$r^2(x) = \mathrm{Const}\frac{\alpha x^2 + \beta x + \gamma}{\mu x^2 + \nu x + \delta}$$

with the coefficients expressed through the values of $\theta_i, \gamma_i, \delta_i$. The points of the function extreme are defined by solving the equation

$$\frac{d}{dx} r^2(x) = 0,$$

which leads to the quadratic equation with roots given by the following expression:

$$x_{1,2} = \frac{-(\alpha\delta - \mu\gamma) \pm \sqrt{(\alpha\delta - \mu\gamma)^2 - (\alpha\nu - \beta\mu)(\beta\delta - \nu\gamma)}}{\alpha\nu - \beta\mu}.$$

The solutions are checked to (i) exist in $\mathbb{R}$ and (ii) to belong in the interval $x \in [0, L]$. If any of the solutions belong in the valid interval $x_0 = x_{1,2} \in [0, L]$, the value of the function at the extreme point x_0 is compared to the function values at the ends of the interval: $r^2_{max} = \max\left\{r^2(0) = r^2_{N_p}, r^2(L) = r^2_N, r^2(x_0)\right\}$. The maximum value shows the best regression coefficient for the local root position. The chosen value of x is then the best local position of the root node on the branch between N and N_p. The procedure to find the best local root for every branch is repeated iteratively for each internal node in the tree. The global solution for the best root is also performed during this iteration. The global value of the best regression coefficient is stored as a separate variable. Each optimal local regression coefficient is then compared to the global optimum and, in case the local regression is better than the current value of the global r^2, the latter is overridden by the new optimal value, and the position of the best root is updated. In the end, the tree is re-rooted to the new best root. A new clade is inserted in the middle of a branch if needed.

Resolving polytomies

Phylogenetic trees of many very similar sequences are often poorly resolved and contain multifurcating nodes also known as polytomies. Tree building software often randomly resolves these polytomies into a series of bifurcations, because the sequences themselves have no information on the order they should be joined together. The order of such randomly inserted bifurcations is often inconsistent with the temporal structure of the tree resulting in poor approximations. To overcome this problem, `TreeTime` makes an attempt to use the additional information from sequence time stamps and to resolve the polytomies in a manner consistent with the sampling dates.

It first prunes all branches of length zero. Then, for each pair of nodes, `TreeTime` estimates by how much the likelihood would increase when grouping this pair of nodes into a new clade of size two. Then, the polytomies are resolved iteratively by merging the pairs resulting in the highest likelihood gain.

Since the `TreeTime` algorithm finds the maximum-likelihood positions for all nodes, the position for a root of a polytomic clade will result in the "trade-off" in the branch lengths of its children. Some of them will end up having branches longer than the optimal values ("stretched branches"), and some will have branches shorter than the optimal length ("compressed branches"). The procedure to merge the stretched branch lengths is shown in figure 10. Merging two stretched nodes with introduction of a new binomial node, results in the highest likelihood gain, which consists of the following parts: (i) the gain due to making the branch lengths values closer to their optimal values and (ii) loss in likelihood because of the introduction of the new branch with zero optimal length. This estimate is not exact because polytomies resolution result in change of the parent node position. The likelihood gain by merging the "compressed" nodes, however, is not as significant. Assuming the fixed position for the parent node, merging the "compressed" nodes does not gain anything, because they remain compressed, just introducing a new clade with the zero-length branch. In real cases, sometimes a slight decrease in the overall tree likelihood in observed after resolving the compressed nodes. This effect is due to increasing the entropy of the tree (the order of merging the "compressed" nodes is arbitrary and hence all possible tree variants are equal).

Because the merging procedure requires to build the likelihood gains for each node, and repeat this procedure n times, the computation complexity of the merging is $O(n^3)$, where n is the number of polytomies in the tree. However, this complexity is local to the multifurcating clade. For all practical cases, the number of polytomies in the tree is small relative to the total

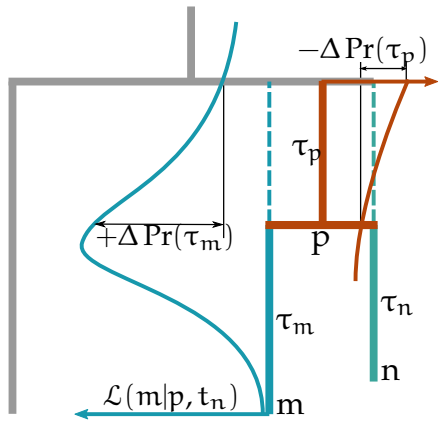

Figure 10: Estimation for likelihood change by merging two stretched branches. The estimation does not take into account possible changing of the polytomies root. The likelihood increases because of relaxing the existing branch lengths to suboptimal values ($\Delta \Pr(\tau_m)$, $\Delta \Pr(\tau_m)$, the latter is not shown). The newly inserted branch decreases the likelihood gain by the value $-\Delta \Pr(\tau_p)$, because its optimal length is zero.

number of nodes. Therefore, the polytomies resolution does not affect the overall `TreeTime` computation complexity.

Autocorrelated molecular clock

Substitution rates can vary across the tree and models that assume constant clock rates may give inaccurate inference. Models that allow for clock-rate variation have been proposed [Hasegawa et al., 1989; Yoder and Yang, 2000; Drummond et al., 2006]. These models typically regularize the clock-rate through a prior and penalize rapid changes of the rate by coupling the rate along branches – known as autocorrelated or local molecular clock [Thorne et al., 1998; Aris-Brosou et al., 2002]. `TreeTime` implements an autocorrelated molecular with a normal prior on variation in clock rates. The rate variation is implemented in `TreeTime` by assigning to each branch a mutation rate factor γ, so the local mutation rate is $\gamma \langle \mu \rangle$, where $\langle \mu \rangle$ is the average

mutation rate given by the GTR model. The rate variation is determined by maximizing the expression

$$\sum_i (\gamma_i L_i - L_{i,opt})^2 + \left[\alpha \sum_i (\gamma - 1)^2 + \beta \sum_{i,j} (\gamma_i - \gamma_j)^2 \right] \rightarrow \max \qquad (32)$$

where the summation is taken over all nodes. L_i is the observed branch length, $L_{i,opt}$ is the optimal branch length. The first term allows to relax the mutation rate to its optimal value. The second term restricts the the mutation rate deviation from the average value, ("stiffness"), and the third term restrict the rate variation across sibling nodes ("coupling"). The solution of the above optimization problem is found in linear time similar to the e.g. forward/backward trace algorithm used for the inference of internal nodes. It involves two tree traversals: in post-order to assign the values conditional to parent and given that all downstream are set to optimal followed by the pre-order traversal to reconstruct the optimal values.

Inference of time reversible substitution models

Large phylogenies typically contain 100s of substitutions and thus provide enough information to infer substitution models from the data. `TreeTime` implements an iterative algorithm to infer general time reversible substitution models [Felsenstein, 2003] parameterized by equilibrium state frequencies π_i and a symmetric substitution matrix W_{ij}. The instantaneous rate from state $j \rightarrow i$ is $Q_{ij} = \pi_i W_{ij}$. The model is inferred by first counting the time spend in different states across the tree T_i and the number of substitutions between

n_{ij} in a joint maximum likelihood assignment using a simple substitution model. Then, π and W are determined by iterating

$$W_{ij} = \frac{n_{ij} + n_{ji} + 2p_c}{\pi_i T_j + \pi_j T_i + 2p_c} \tag{33}$$

$$\pi_i = \frac{\sum_j n_{ij} + p_c + m_i}{\sum_j W_{ij} T_j + \sum_j (m_j + p_c)}, \tag{34}$$

where p_c is a small pseudo-count driving the estimate towards a flat Jukes-Cantor model in absence of data, and m_i are the number times state i is observed in the sequence of the root. In each iteration, the π is normalized to one, the diagonal of W_{ij} is set to $-\pi_i^{-1} \sum_{j \neq i} W_{ij}\pi_j$, and W_{ij} is rescaled such that the total expected substitution rate $-\sum \pi_i W_{ii} \pi_i$ equals one. The rescaling of π and W_{ij} can be absorbed into an overall rate μ. This algorithm typically converges in a few iterations.

Coalescent priors

The genealogical tree of individuals within a species depends on the size of the population, its geographic structure, and fitness variation in the population [Kingman, 1982; Nordborg, 1997; Neher, 2013]. In the simplest case of a panmictic population without fitness variation, the genealogies are described by a Kingman coalescent [Kingman, 1982], possibly with a population size that changes over time. Within the Kingman coalescent, any two lineages merge at random with a rate $\lambda(t)$ that depends on the population size $N(t)$ and the current number of lineages $k(t)$.

$$\lambda(t) = \frac{k(t)(k(t)-1)}{2N(t)} \tag{35}$$

The rate at which a given lineage merges with any of the other lineages is $\kappa(t) = (k(t)-1)/2T_c(t)$. Here, the population size $N(t)$ defines a time scale

measured in units of generation time and we will more generally refer to this time scale by $T_c(t)$ and measure it in units of the inverse clock rate.

The contribution of a branch between time points t_0 (child) and t_1 (parent) in the tree to the likelihood is then given by

$$p(t_0, t_1) = e^{-\int_{t_0}^{t_1} dt \kappa(t)}, \tag{36}$$

where a merger at time t contributes with rate $\lambda(t)$

`TreeTime` adds the contribution of each branch to the coalescent likelihood the branch likelihood object, which are then parameterized by the starting and end point of the branch, $b_n(t_n, t_n + \tau)$. The total coalescent likelihood given a tree can be evaluated in one tree traversal such that T_c can be optimized efficiently. In addition to a constant T_c, `TreeTime` can model T_c as a piecewise linear function. Such piecewise functions are known as "skyline" [Strimmer and Pybus, 2001] and can be optimized by `TreeTime` as well.

3.3 CASE STUDY: ANALYSIS OF THE 2014-2015 EBOLA VIRUS OUTBREAK

In 2014, West Africa experienced the largest known outbreak of Ebola Virus (EBOV) in humans. The genomic epidemiology has been studied intensively by multiple groups [Dudas et al., 2017]. Here, we reanalyzed a subset of 350 EBOV sequences sampled throughout the outbreak from 2014-2016. Due to the dense sampling, the maximum likelihood phylogeny has many unresolved nodes and `TreeTime` was used to resolve polytomies using temporal information. After automatic rooting and GTR model inference, `TreeTime`

produced the time tree shown in figure 11. The GTR model inferred from the tree was

$$\pi = \begin{array}{ll} A: & 0.32 \\ C: & 0.21 \\ G: & 0.195 \\ T: & 0.275 \end{array} \qquad W = \begin{array}{c|cccc} & A & C & G & T \\ \hline A & \cdot & 0.45 & 2.7 & 0.28 \\ C & 0.45 & \cdot & 0.25 & 3.7 \\ G & 2.7 & 0.25 & \cdot & 0.45 \\ T & 0.28 & 3.7 & 0.45 & \cdot \end{array} \tag{37}$$

`TreeTime` ran 4min on a regular laptop to complete this analysis. In addition to inferring a time tree, `TreeTime` estimated the time course of the coalescent population size shown in the lower panel of figure 11. The estimated population size closely mirrors the case counts reported by the WHO throughout this period.

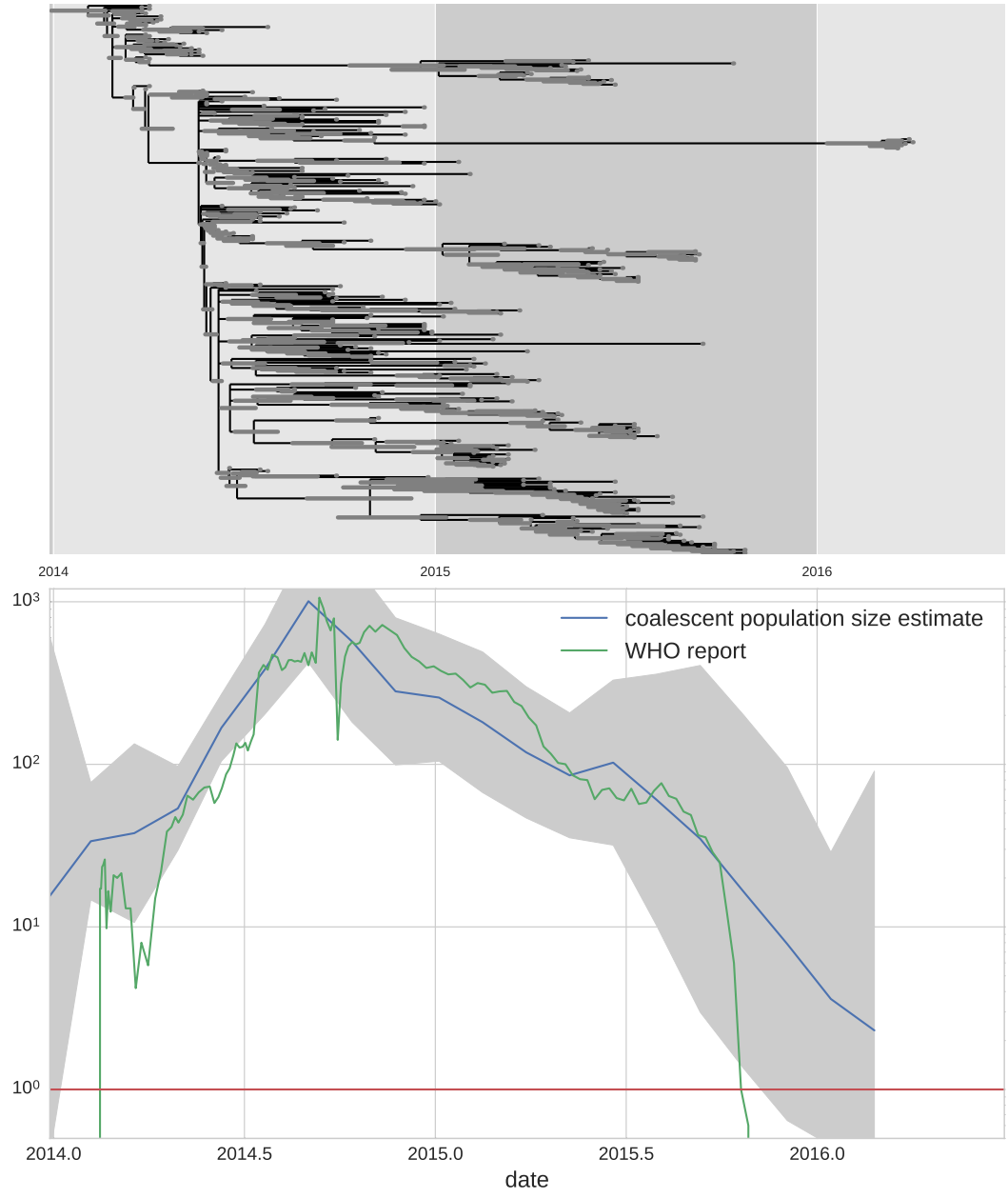

Figure 11: **EBOV phylodynamic analysis.** The top panel shows a molecular clock phylogeny of EBOV sequences obtained over from 2014-2016 in West Africa. The lower panel shows the estimate of the coalescent population size along with its confidence intervals. The estimate suggest an exponential increase until late 2014 followed by a gradual decrease leading to almost complete eradication by 2016. Ebola case counts, as reported by the [WHO, 2016] agree quantitatively with the estimate.

4

TREETIME VALIDATION

4.1 OBJECTIVES

`TreeTime` was tested predominantly on mildly diverged sequences from viruses. The iterative optimization procedures are not expected to be accurate for trees were the many sites are saturated. In such scenarios with extensive uncertainty of ancestral states and tree topology, convergence of the iterative steps can not be guaranteed. While in many cases `TreeTime` might still give approximate branch length and ancestral assignments and time tree estimates, these need to be checked for plausibility. In general global optimization and sampling of the posterior can not be avoided.

4.2 VALIDATION ON SIMULATED DATA

To assess the accuracy of date reconstructions of treetime and to compare its performance to existing tools such as BEAST and LST [Drummond et al., 2012; To et al., 2016], we generated toy data using the FFPopSim forward simulation library [Zanini and Neher, 2012]. We simulated population of size $N = 100$ and used a range evolutionary rates $\mu = 10^{-5}, \ldots, 0.002$ resulting in expected genetic diversity from 0.001 to 0.2. Sequences were sampled every 10, 20, or 50 generations. The length of the simulated sequences was $L = 1000$.

Divergence times and mutation rate

figure 12 shows the error in the estimates of the clock rate for `TreeTime`, LSD, and BEAST as a function of the evolutionary rate. `TreeTime` and LSD estimates of the clock rate are very accurate for small rates but tend to underestimate the rates at when diversity exceeds a few percent. This is expected, as maximum likelihood inference underestimates branch lengths. BEAST tends to overestimate small rates and is accurate when branches become long. By sampling trees, BEAST does not suffer from the atypical maximum likelihood assignments.

In a similar manner, `TreeTime`, LSD, and BEAST estimate the time of the most recent common ancestor to within 10% accuracy (relative to the coalescence time) across the range of simulated data.

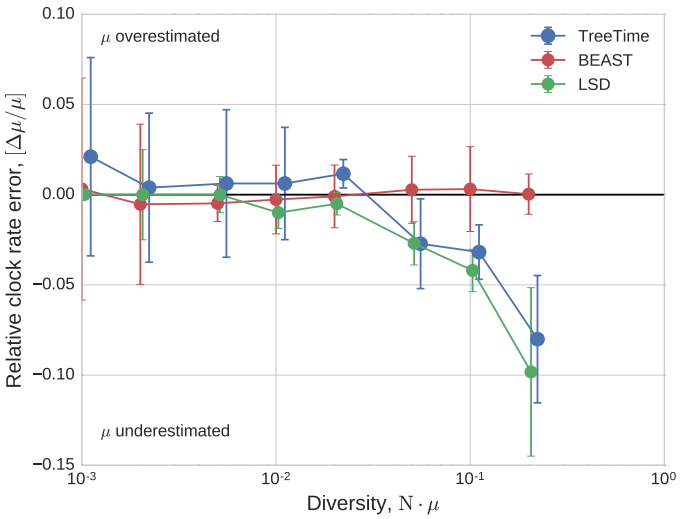

Figure 12: Estimation of the evolutionary rate from simulated data by `TreeTime`, LSD, and BEAST. `TreeTime` and LSD (after tree reconstruction using FastTree) underestimate the rate when branch length are long. BEAST tends to overestimate the rate at small rates. The error bars denote ± one standard deviation.

We also ran `TreeTime` on simulated data provided by [To et al., 2016] and compared it to the results reported by [To et al., 2016] for LSD, BEAST and

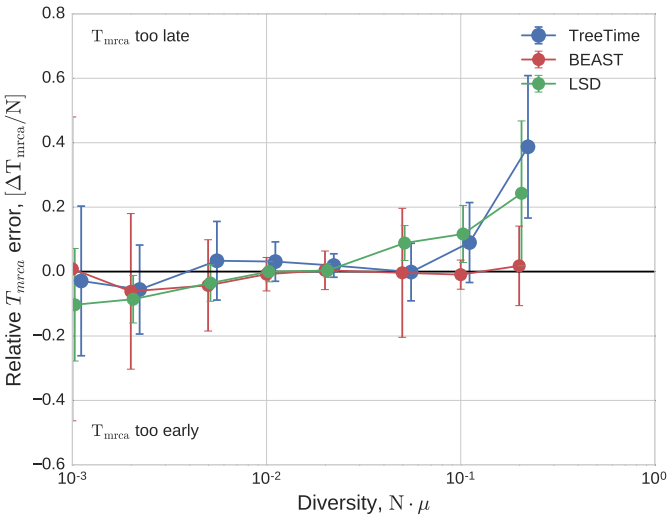

Figure 13: Estimation of TMRCA from simulated data by `TreeTime`, LSD, and BEAST. All three programs estimate the time of the MRCA with 10% accuracy, except for the very long branches when `TreeTime` tends to overestimate the age of the root. Error bars show one standard deviation.

a number of other methods. figure 14 compares the accuracy of T_{MRCA} and clock rate estimates, showing that `TreeTime` achieves similar or better accuracy than other methods.

Coalescent model inference

Population bottlenecks, selective sweeps, or population structure, affect the rate of coalescence in an often time variable way. BEAST can infer a history of effective population size (inverse coalescent rate) from a tree – often known as skyline. `TreeTime` can do a similar inference by maximizing the coalescence likelihood with respect to the pivots of a piecewise line approximation of the coalescence rate history $T_c(t)$. To test the power and accuracy of this inference, we simulated sinusoidal population size histories of different amplitude and period, uniformly sampled sequences through time, and used these data to estimate the coalescent rate history. Comparisons of true and estimated histories are shown in figure 15.

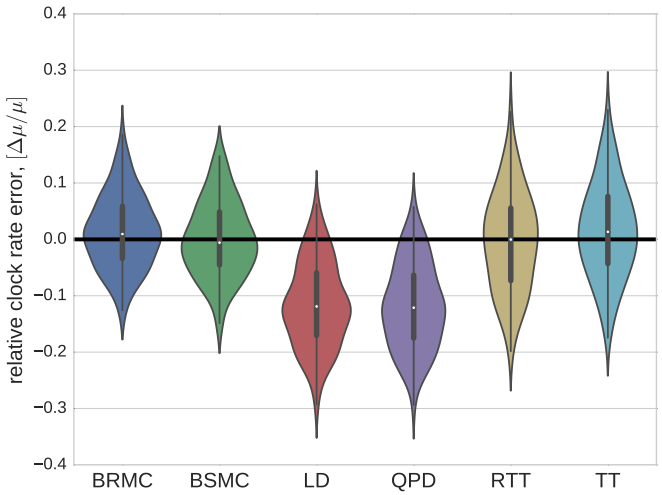

Figure 14: **LSD test data.** `TreeTime` has comparable or better accuracy as BEAST (BSMC), LSD (LD, QPD), or root-to-tip regression (RTT) when run on simulated data provided by [To et al., 2016]. Both panel use the tree set 750_3_25, the top and bottom panel show runs on alignments generated with a strict and relaxed molecular clock, respectively.

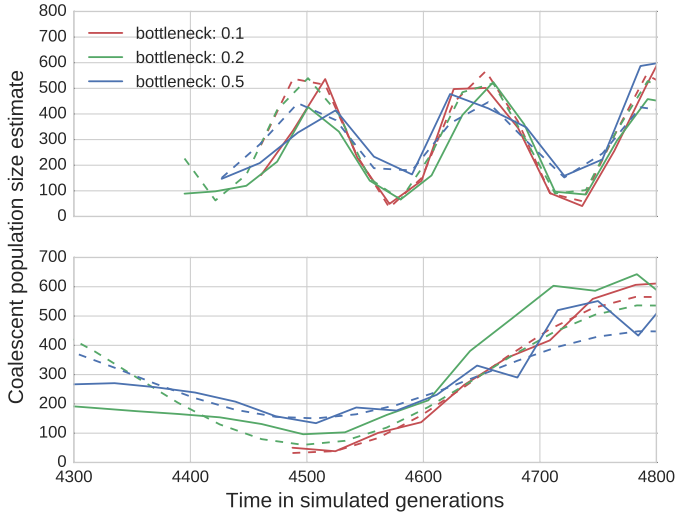

Figure 15: Reconstruction of fluctuating population sizes by `TreeTime`. The graph shows simulated population size trajectories (dashed lines) and the inference by `TreeTime` as solid lines of the same color. Different lines vary in the bottleneck sizes of 10%(red), 20%(green) and 50%(blue) of the average population size. The top panel shows data for fluctuations with period 0.5N, the bottom panel 2N. The average population size is $N = 300$.

4.3 VALIDATION ON INFLUENZA PHYLOGENIES

The dense sampling of influenza A virus sequences over many decades makes this virus an ideal test case to evaluate the sensitivity of time tree estimation to sampling depth. We estimated the clock rate and the time of the most recent common ancestor of influenza A H3N2 HA sequences sampled from 2011 to 2013 for sets of sequences varying from 30 to 3000, see figure 16. `TreeTime` estimates are stable across this range, while estimates by LSD tend to drift with lower rates and older MRCAs for larger samples. Estimates by BEAST are generally concordant with `TreeTime`.

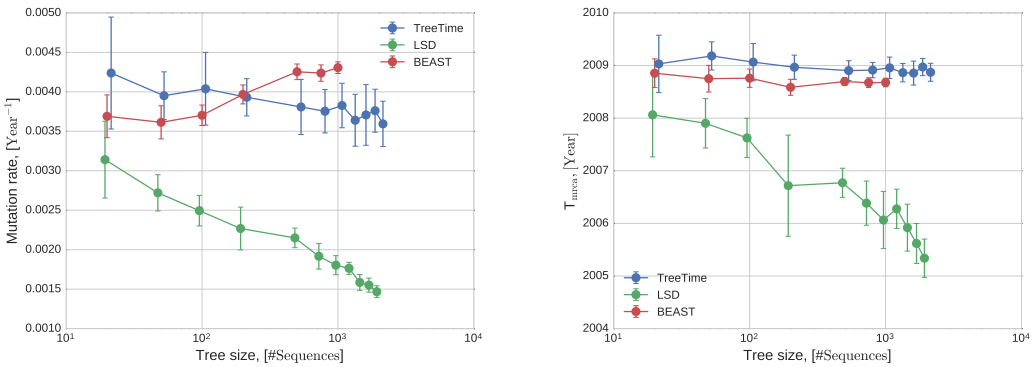

Figure 16: Variation of the estimate of the rate of evolution of H3N2 and the for different sensities of sampling.

Next, we tested how accurately `TreeTime` infers dates of tips when only a fraction of tips have dates assigned. Every tip in `TreeTime` can either be assigned a precise date, an interval within which the date is assumed to be uniformly distributed, or no constraint at all. `TreeTime` will then determine the probability distribution of the date of the node based on the distribution of the ancestor and the substitutions that occurred since the ancestor. We tested the accuracy at which missing dates can be inferred in an influenza phylogeny by erasing date information of a fraction (5% to 95%) of all nodes, see figure 17.

In summary, on data sets with short branches but fairly unambiguous topologies, timetrees inferred by `TreeTime` have similar accuracy to those inferred by BEAST but results are obtained in a fraction of the time.

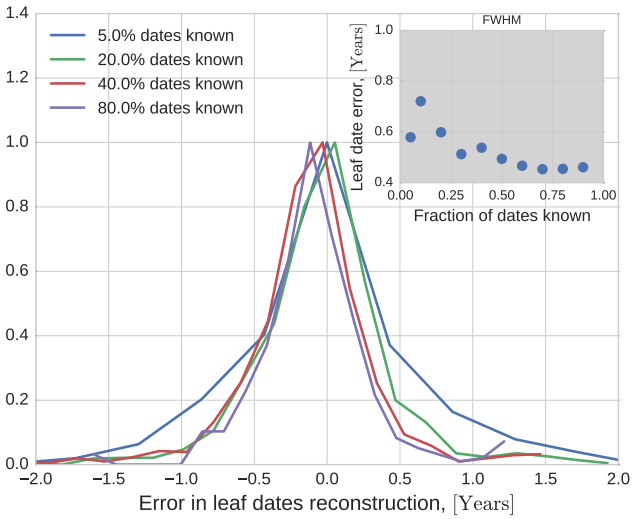

Figure 17: Tip dating and sensitivity to missing information. A) The inter-quartile range of the error of estimated tip dates decreases from 0.7 years to 0.5 years as the fraction of known dates increases from 5% to 90% (see inset).

5

TREETIME MODULE

5.1 PYTHON PACKAGE

`TreeTime` code has been developed with the usability and extensibility in mind. To facilitate the user-interactions and provide easy-to-use API, we chose to write the core code in Python-2.7 programming language. All algorithms of `TreeTime` are published open-source and are distributed as `TreeTime` package under MIT license. The source code can be found on [GitHub, b]. The complete set of the validation scripts is also available on-line [GitHub, c]. These scripts also present thorough examples of the `TreeTime` usage.

The `TreeTime` algorithms and classes can be used in larger phylogenetic analysis in python scripts. This is the most flexible way to use `TreeTime`. All the different analysis steps can be combined in custom ways with parameters. In addition, the command-line scripts are provided for typical recurring tasks such as ancestral state reconstruction, re-rooting to maximize temporal order, and time tree inference.

5.2 SOURCE CODE STRUCTURE

The lower layer is implemented as `TreeAnc` class, which purpose is to perform the standard operations and to provide a user with the basic standard algorithms, such as ancestral sequence reconstruction and inference substitution models.

The middle layer, presented by the `ClockTree` class implements the basic functionality to build time trees. It realizes the core algorithms presented in the chapter 3.

The top-most layer, which is in the `TreeTime` class is to provide the additional functionality and, more important, to define the computational pipeline, to split the global optimization problem into iteration levels, and to implement iterative divide-and-conquer approach.

The core complication in implementing the algorithms in code has been to properly deal with the likelihood distributions for branch lengths and node dates as well as to perform the integration and interpolation and transformations of the distributions. The distributions are implemented as `Distribution`, `NodeInterpolator`, and `BranchLengthInterpolator` classes, which encapsulate all necessary mathematical operations. The general time-reversible model is implemented through the `GTR` class, which provides a set of the most popular standard models for nucleotide and amino acid evolution. In addition, the possibilities to define random and user-specific models are implemented.

5.3 IMPLEMENTATION OF LIKELIHOOD DISTRIBUTIONS

The central part in implementing the mathematics is to properly discretize the likelihood distributions and to implement mathematical operations (integration, convolution, multiplication, and others) on the discrete functions. Another complication is that despite computing the exact values from the analytical expressions is possible, it is impractical due to its complexity, so the approximations should be introduced where needed.

Branch lengths

The basic functions are the likelihood distributions for the branch lengths. All other distributions for the `TreeTime` analysis are built on the basis of the branch length distributions. The branch length distributions can be evaluated from their analytical expressions in eq. 22. These evaluations are used to determine the maximum likelihood branch length (refer to as "mutation length"). The determination of the maximum likelihood is done using the Brent optimization algorithm via its standard implementation of the `SciPy` python library. The maximum likelihood branch lengths are then used to (i) optimize the tree as described in the pre-processing section of chapter 3 and (ii) to properly interpolate the branch length likelihood distribution.

To interpolate the branch length distributions, first the grid is constructed. The construction the grid for the branch length distributions consists of the two cases: (i) grid for the branch with no substitutions and (ii) for the branch where one or more substitutions occurred. In the former case, the branch length distribution is just an exponentially decaying function, which is a straight line in log-scale. The grid construction for the latter case is made by concatenating the three independent grids for the following regions of the branch lengths: $x \in [0, x_0), x \in [x_0, 5 * x_0), x \in [5 * x_0, \infty)$, where x_0 is the mutation length of the branch. In the first two regions, the grid constructed is linearly spaced, whilst in the third region, the space between the grid points is increased exponentially. For the infinity, an arbitrary big number has been chosen. In addition, several points were placed around zero branch length in order to increase the precision of the branch length evaluation around zero length.

Given a grid constructed in this way, a branch length probability is evaluated in the node points of the grid followed by the linear interpolation of the branch length distribution. Further grid refinement using the algo-

rithms described below are also used, where the interpolation quality is not enough.

Node positions

Discretizing branch length distributions is a trivial problem due to the presence of the natural scale on the t-axis, and the knowledge of the function properties (location of the maximum, curvatures, definition area). For the derivative distributions, however, there is no such prior information. Therefore, the problem of constructing the grid for these distributions arise. The solution of that problem is shown below for the example of the convolution between the branch length distribution $g(\tau)$ and another distribution $f(t)$. $f(t)$ which can be computed previously, or alternatively, it can given as a prior of a leaf date. Both input distribution are understood as interpolated discrete functions. The problem is to find an interpolated function $F(t) = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau$ for $t \in (-\infty, \infty)$ with the precision not worse than the precision of interpolating f, g. The range of t is understood as "any practical value of t". To address the problem, a special grid $\{t_i\}$ should be constructed, and the values $F(t_i)$ should be evaluated followed by the construction of the interpolated function. To evaluate the error of the linear interpolation between points x_1, x_2 , the standard expression is used: $R \leqslant \frac{Mh^2}{8}$, where $M = max_{x\in[x_1,x_2]} f''(x)$, and h is the grid step. Given the expression for the interpolation error, the grid construction is as follows. First, the rough position of the maximum for the function $F(t)$ is determined. This position is calculated by simply shifting the peak position of $f(t)$ by the value of the peak position of $g(\tau)$. Due to the following refinement procedure, the precision for the peak determination is enough. Then, a small grid is constructed around the peak position. The typical number of points for this preliminary grid is 50–100, the number has been found empirically to

provide a good trade-off between the computation cost and the initial precision. Then, the error of the function determination is estimated between the interpolation points, and additional, uniformly spaced, points are inserted where needed until the function error is lower than the given error rate. The procedure used in practice to detect the segment where grid refinement is needed is comparing the error rate with some tolerance rate:

$$F''(t_i) \geqslant 0.01 \cdot \left(1 + \frac{y_{max} - y_i}{10}\right)^4,$$

where the 0.01 is the overall tolerance coefficient, and the multiplier is to attenuate the tolerance for regions far from the distribution maximum. The typical values for the coefficients and exponent are found empirically to gain the best trade-off between the grid density (and hence the computation cost) and the interpolation precision. In practice, the equation above is used to estimate roughly the number of the points to be inserted in the selected region:

$$N_{points} = \frac{F''(t_i)}{0.01 \cdot \left(1 + \frac{y_{max} - y_i}{10}\right)^4}$$

The grid is iteratively refined this way until the function is determined with the required precision in the whole interpolation region. The procedure described allows to define grids specifically for each function, with the points density correlating with the function curvature so that the interpolation precision is always higher than the pre-defined tolerance.

5.4 PROCESSING PIPELINE

`TreeTime` solves several coupled optimization problems. For instance, optimization of the branch lengths, inferring the ancestral sequences, and inferring the GTR model are all coupled problems and therefore should be optimized simultaneously. In addition, making a time tree would perturb

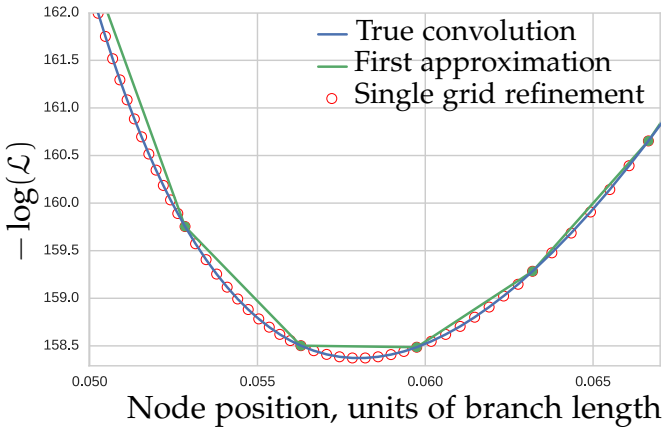

Figure 18: caption

the branch lengths and thus may influence the ancestral sequences and the resulting GTR model. Therefore, it should also be included in this global optimization problem. The same consideration may be extended to the other parts of the `TreeTime` functionality.

To maintain the simplicity, and linear scaling of the `TreeTime` run-time, we split the global optimization problem into sub-problems. These sub-problems are conquered iteratively. The solution for each subproblem is conditioned on the optimal solutions of the other ones. This approach allows split the global problem into sub-problems, to conquer these sub-problems in the iterative manner, and finally obtain the global optimum solution. The iteration is used on multiple levels, converging the joint solutions of the subproblems to the global optimum.

The simplified pipeline of the `TreeTime` run, which illustrates the iterative approach described, is shown in figure 19.

Such an iterative procedure typically converges quickly when the branch lengths of the tree are short such that ancestral state inference has little ambiguity.

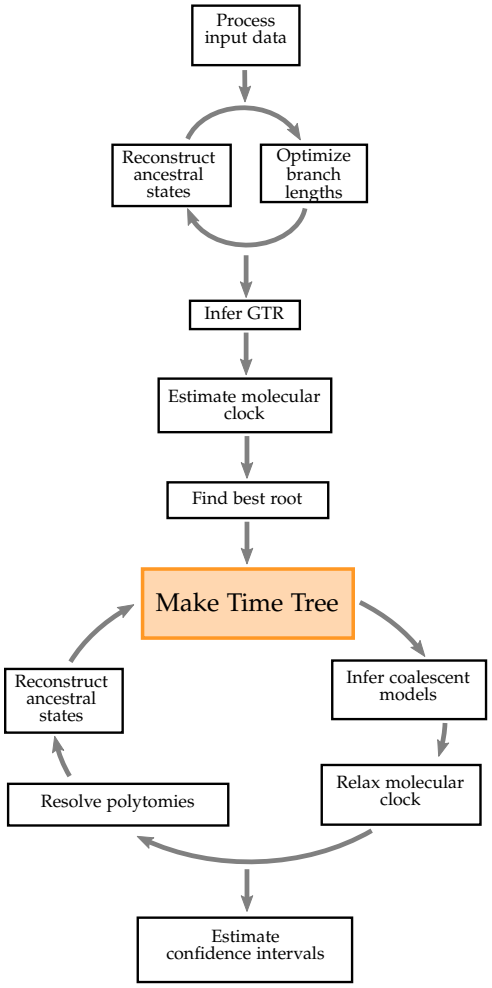

Figure 19: Main pipeline of the `TreeTime` framework. Inference of time tree, which is the core functionality of the `TreeTime`, is highlighted in pale orange. Our approach is to split the complex optimization problem into subproblems, and then iteratively solve each of them. The previously obtained solutions for sub-problems are re-calculated where needed.

5.5 WEB APPLICATION

We have also implemented a web-application for the `TreeTime`. It allows exploration and analysis of heterochronous alignments in browsers without the need to use the command-line. Another virtue of using the server version of `TreeTime` is that it provides the computational power of our servers to end users. It aims to facilitate the usage of `TreeTime` and to broaden the audience using its algorithms.

The server version of `TreeTime` implements only the standard well-tested functionality though. So, for any type of a custom analysis, the command-line version is still necessary. The web application is located in the server of Basel University [treetime.ch].

The server part of the `TreeTime` web has been written in python to provide natural access to the `TreeTime` algorithms. The server is based on the `Flask` Python library. To enable `TreeTime` and server interaction, a small wrapper class has been written. Its main purpose is to convert configurations from the server format to the `TreeTime` format, and to report computation status back to the server. It also saves all computation results, explicit logs and temporary information in json format so that these files can be accessed by the server and visualized on the client side.

The client side is implemented as dynamically created web pages, which are rendered using client machine resources. Current functionality is limited to creation of `TreeTime` run configurations, and to basic visualization of the computation results. The former is accomplished by providing a web-form. The latter renders phylogeny, plots the molecular clock and likelihood distributions (see example in figure 20).

Despite its current functionality is limited, I have developed architecture so that it can be easily extended and adapted to various usage scenarios. The web pages design is based on the popular open-source React-JS library [GitHub, a], written in JSX format. To enable support for all browsers, JSX files are compiled in plain javascript by using webpack utility. The plain javascript is then inserted in html template pages, where the scripts represent the only element. This approach showed to be very easy to develop and maintain as well as to extend to add new functionality. For instance, one of the extension direction has been to create a standalone tree viewer, which would allow to render trees in json format. The trees in turn may carry arbitrary amount of meta-data.

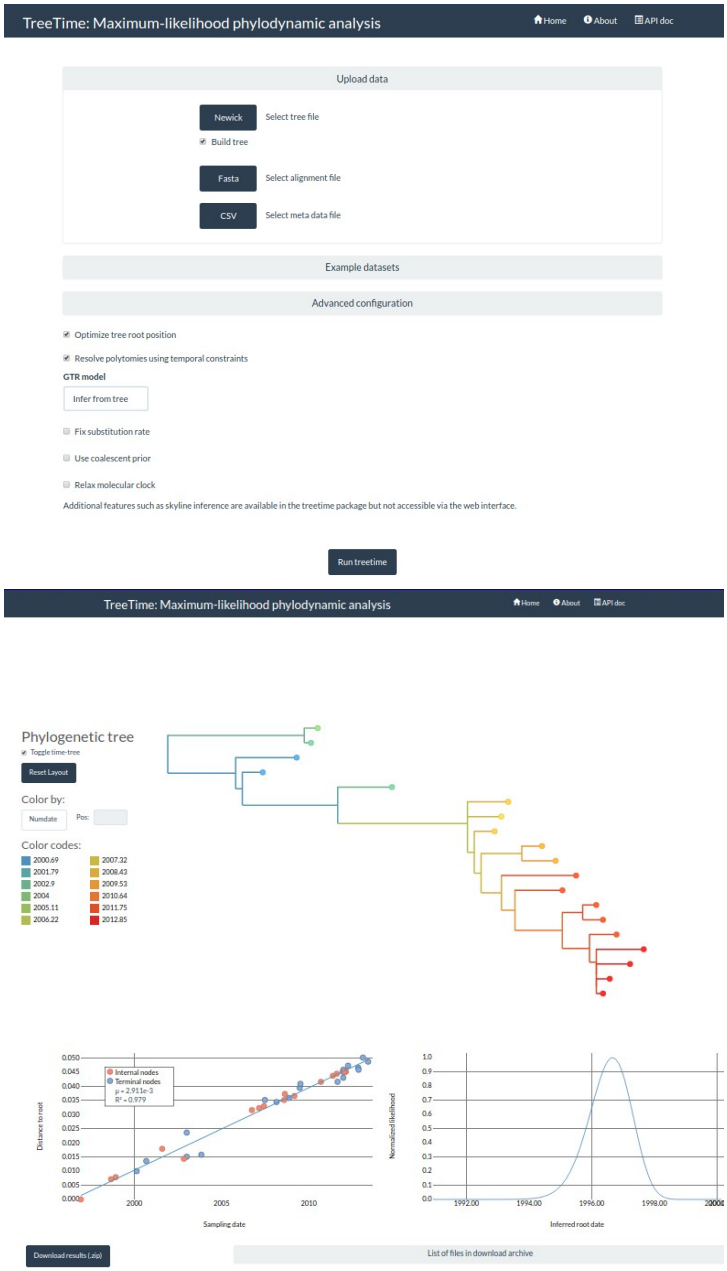

Figure 20: Screenshots for the `TreeTime` web application.
Upper panel: The `TreeTime` welcome page. `TreeTime` on the server requires to upload the data files. Alternatively, a preloaded example dataset can be used. The server version of `TreeTime` provided limited configuration options.
Lower panel: The results page. An optimized tree with basic options for coloring, zooming and navigation is shown on top. The molecular clock estimation and the likelihood distributions for each internal nodes are displayed in the lower panel. All data can be downloaded as a single .zip archive.

6

INFERENCE OF GENERAL TIME REVERSIBLE MODELS

6.1 SITE-SPECIFIC SUBSTITUTION MODELS

The theory for the GTR models in the phylogenetic inference has been described in the chapter 2. One of the fundamental assumptions, which has been made implicitly in that chapter is that each site in a sequence evolves under similarly. This is expressed by the fact that the transitions for all cites are described by the same evolution matrix Q_{ij}. The assumption made is however, almost never observed in the reality [Pagel et al., 2004]. In DNA sequences, first, second, and third codon positions, for example, tend to evolve at different rates. In addition, they might have different substitution patterns. A well-known case in which heterogeneity across sites in the pattern of evolution is predicted is in the stems and loops of ribosomal sequences [Schöniger and Von Haeseler, 1994]. If the data are nucleotides from a coding region or the amino acids of a protein sequence, then the natural selection may constrain variability at some sites more than at others (so-called purifying selection). Therefore, different sites will exhibit different rates of evolution. The heterogeneity in the evolutionary rates becomes very clear from the the substitution patterns in protein sequences. For example, figure 21 illustrates the rate heterogeneity in HIV-1 protease. The evolutionary rates were obtained from the the phylogenetic tree of approximately 10^4 sequences, which provides enough data to evaluate transition matrix for every site. The relative substitution rates were estimated as the ratio between the number of transition to the target state, and the time spent by a site in the particular source state (valine (Val) in this case). Note that both the relative

mutability of sites, and the evolution matrices are different. The former is the overall substitution rate to all possible amino acids (the net height of the bars of a single color), and the latter is the ratio between the transition rates to different target states (the ratio of the heights of the bars of different colors). In this chapter, we aim to develop the better approximation for

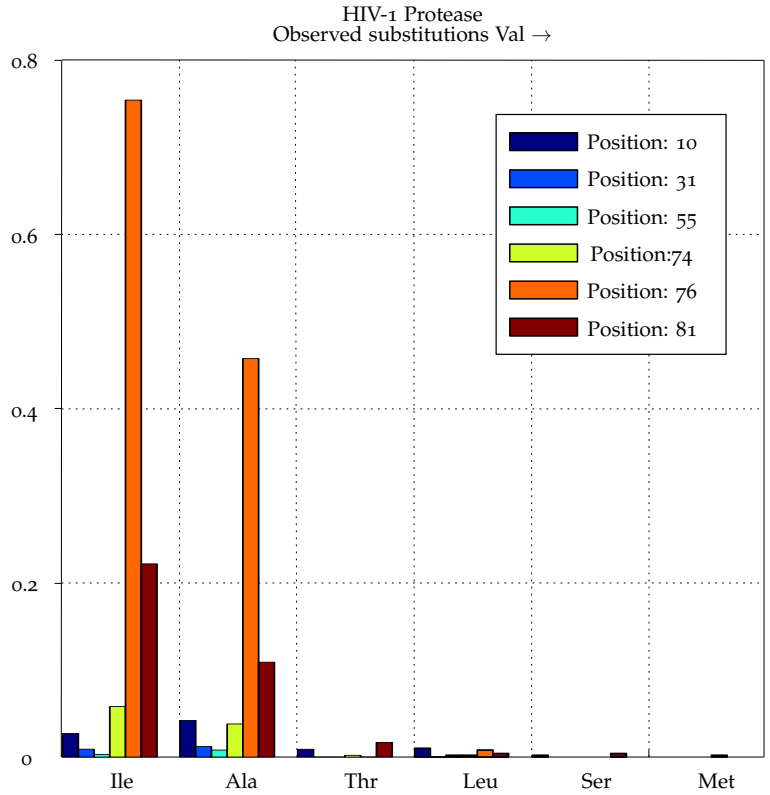

Figure 21: The illustration of the substitution rate heterogeneity among sites of the HIV-1 Protease. The relative substitution rate was calculated from the tree for approximately 10.000 sequences as the ratio of time a site spent in *Val* state to the number of transitions to the other states.

the GTR model, which accounts for the possible heterogeneity in mutation rates and in the evolution matrix. The interest in site-specific GTR models is not only to provide better approximation to the phylogeny inference, but also to detect the selection constraints on protein evolution [Fay and Wu, 2003]. Such constraints originate, e.g., from the need to preserve protein functionality.

Although in some cases the residue conservation can be inferred directly from the protein alignment [Weigt et al., 2009], this is generally not correct, since the observed residue frequency in the alignment is influenced by its state in the most recent common ancestor. Thus one usually starts with reconstruction of a protein phylogeny and and the ancestral sequences at each of its nodes. While simple estimates can be done by simply going over all branches on the tree and counting the number of mutations, most modern approaches make explicit use of the Markovian models of nucleotide evolution [Yang, 2006]. The main difficulty that one encounters when trying to infer a GTR model in this way is maximizing the likelihood of the tree in respect to the model parameters. Although there exist semi-empirical approaches to achieve this goal [Waddell and Steel, 1997; Thyagarajan and Bloom, 2014; Reis et al., 2009], the commonly used method is Markov chain Monte-Carlo, which allows combining the GTR model inference with inference of the tree topology and the ancestral residue states [Lartillot and Philippe, 2004; Shapiro et al., 2006; Minin and Suchard, 2008]. Indeed, the tree topology and the ancestral states are themselves inferred using a GTR model, and therefore can, in principle, change when the model is modified. Another sort of issues is related to the statistical validity of the site-specific inference [Pond et al., 2005]. Indeed, using site-specific model risks overfitting the data due to excessively large number of parameters, sometimes known as *extensive parametrization* [Rodrigue, 2013]. This problem is overcome either by using sufficiently big alignments (and phylogenetic trees) or by splitting residues into groups that would be described by the same GTR model (known as CAT approach, [Lartillot and Philippe, 2004; Blanquart and Lartillot, 2008]). We have developed a novel method for inferring the GTR model by maximizing tree likelihood. The method relies on the assumption of the short tree branches, which allows us to obtain a set of iterative equations for inferring the model parameters corresponding to the likelihood maximum. While this assumption is not always full-filled, it al-

lows to address the selection pressures in fast evolving organisms, such as viruses. Indeed, while the next generation sequencing has made available a wealth of data about such organisms (see, e.g., [Zanini et al., 2015; HIV-DataBase, 2017; Flu-DataBase, 2017]), applying Monte-Carlo methods to such big alignments is inefficient and time-consuming.

From the statistical viewpoint, our approach is particularly powerful when the GTR matrix is decomposed into an *attempt matrix*, common to all sites, and the site-specific nucleotide/residue frequencies [Lartillot and Philippe, 2004], which correspond to the nucleotide frequencies in the *equilibrated alignment* (i.e., the sequence alignment that one would have, if the organism was allowed to evolve for very long time under the same conditions.) Finally, we demonstrate that our approach can be modified to accommodate possible changes in reconstructed ancestral states (induced by improved GTR model inference) and how to expand its range of validity to trees with longer branches.

6.2 INFERENCE SCHEME FOR SITE-SPECIFIC GTR

Model parametrization

We describe the substitution process using site-specific GTR (general time-reversible) model, parametrized in a standard way as:

$$Q_{ij,\alpha} = \pi_{i,\alpha} W_{ij,\alpha}, \text{ for } i \neq j \tag{38}$$

$$Q_{ii,\alpha} = -\sum_{j} Q_{ij,\alpha}, \; W_{ij,\alpha} = W_{ji,\alpha}, \tag{39}$$

where index i designates a state of the nucleotide or amino-acid residue, whereas α is the site index, i.e. it describes the location of the nucleotide/residue in the sequence. In this general form, a site-specific GTR model for an alphabet of size q has $q(q+1)/2-1$ parameters per site. The expression for

the diagonal element of the *substitution matrix*, $Q_{ii,\alpha}$, guarantees probability conservation, whilst the symmetry of the *attempt matrix*, $W_{ij,\alpha}$, ensures the time-reversibility of the substitution process:

$$Q_{ij,\alpha}\pi_{j,\alpha} = Q_{ji,\alpha}\pi_{i,\alpha}.$$

The site-specific frequencies, $\pi_{i,\alpha}$, satisfy normalization condition:$\sum_i \pi_{i,\alpha} = 1$. The procedure to infer the parameters of the model (39) relies on the assumption that that the attempt matrix can be decomposed into a product of the site-specific mutation rate and a constant (non-site-specific) matrix:

$$W_{ij,\alpha} = \mu_\alpha W_{ij}. \tag{40}$$

The scale of the mutation rate is then fixed by normalization of the attempt matrix to sum to unity: $\sum_{i,j\neq i} W_{ij} = 1$. eq. 40 constitutes the central assumption for the present approach for the site-specific GTR model inference. It is justified by the fact that nucleotide mutations are mainly governed by cellular chemistry and there is no reason to assume any site-specificity for them, whereas the stationary populations, $\pi_{i,\alpha}$ are mainly determined by the selection pressures acting on the nucleotide sequence (i.e. the necessity to encode a viable protein.) Finally, μ_α may be site-dependent due to peculiarities of the transcription process. For the nucleotide/amino-acid alphabet of size q and sequence length L, decomposition eq. 40 reduces the number of parameters from $(q-1)L + q(q-1)L/2 - 1$ to $qL + q(q-1)/2$, thus reducing the risk of the over-parametrization. Given the phylogeny, the substitution model and the sequence states at every node, we can construct for every sequence site the likelihood to observe the particular realization of the nucleotide/residue states as

$$\mathcal{L}\left(\pi_i, W_{ij}\right) = \pi_{i_0} \prod_k \Pr(i_k \mid j_k, t_k), \tag{41}$$

where the product is over all tree branches, i_k, j_k are the child and parent nucleotide/residue states corresponding to this branch, t_k is the branch length, i_0 is the root state. The probability to observe child state i given the parent state j and the branch length t is given by the GTR evolution equation:

$$P_{ij}(t) = \left(e^{\hat{Q}t}\right)_{ij}. \tag{42}$$

Finally, to calculate the likelihood of the whole tree, the eq. 42 should be supplied to the general eq. 41 for each sequence site, followed by multiplication over all tree branches and all sequence sites.

Maximizing tree likelihood

The likelihood eq. (41) differs from the likelihoods used to construct phylogenetic trees [Yang, 2006; Felsenstein, 2003] by the lack of summation over the internal nucleotide/residue states. The goal is to maximize this likelihood in respect to the parameters of the GTR model. Due to the large number of parameters involved, this is usually done using Markov chain Monte-Carlo simulations (see, e.g., [Rodrigue, 2013]), which is a time consuming procedure. However, when the available sequences are known to have diverged from the common ancestor relatively recently, the tree branches are going to be short. In this case one can expand the exponent in eq. 13 and maximize the likelihood analytically.

For a single site the likelihood has the following form:

$$\mathcal{L}\left(\pi_i, W_{ij}\right) = \pi_{i_0} \prod_k \Pr(i_k \mid j_k, t_k) = \pi_{i_0} \prod_{i,j \in \mathcal{A}} \prod_k^{n_{ij}} P_{ij}\left[t_k^{(ij)}\right], \tag{43}$$

where the product is over all branches of the tree, i_k, j_k specify the nucleotide values for the child and parent sequences corresponding to branch k, t_k is the length of this branch, and i_0 is the site state at n the root node. In-

cluding the root value ensures that the tree is time-reversible, and therefore insensitive to the choice of the root. This choice turns out to be important to provide the pseudocounts, ensuring convergence of the iterative procedure described below. In the second equality in eq. 43, the product has been rearranged into classes corresponding to the state transitions with different combinations of parent and child states, so that index k now runs over all branches where the transitions is from j to i, n_{ij} is the number of branches with such transitions and $t_k^{(ij)}$ are the corresponding branch lengths. (The k index now runs over the branches within the class of branches corresponding to the same type of transition i, j.)

The likelihood for the whole tree is obtained by taking the product of the likelihoods for every site in the nucleotide sequence,

$$\mathcal{L} = \prod_{\alpha} \mathcal{L}_{\alpha}, \tag{44}$$

where α is the site index. In order to make the calculations site-specific the parameters of the GTR matrix has been made dependent on the site index: $\pi_i \to \pi_{i\alpha}$, $i_0 \to i_\alpha$. However, the attempt matrix is assumed to be identical up to the constant factor:

$$W_{ij,\alpha} = \mu_\alpha W_{ij}, \tag{45}$$

and the scale of the attempt matrix factor is fixed so that the elements of W_{ij} sum up to unity:

$$\sum_{i,j\neq i} W_{ij} = 1. \tag{46}$$

(In practice it is convenient to treat W_{ij} as a matrix with zero diagonal elements. In the following we will always deal only with the non-diagonal elements of W_{ij}, unless stated otherwise.)

Using the expressions for transition probabilities of eq. 13, and assuming the branches are short, so that the exponent can be linearized, the tree likelihood is written then as

$$\mathcal{L} = \prod_{\alpha} \left[\prod_{i} \prod_{k=1}^{n_{ii,\alpha}} \left(1 - \sum_{j \neq i} \pi_{j\alpha} \mu_\alpha W_{ji} t_k^{(ii,\alpha)} \right) \times \prod_{i,j \neq i} \prod_{k=1}^{n_{ij,\alpha}} \pi_{i\alpha} \mu_\alpha W_{ij} t_k^{(ij,\alpha)} \times \pi_{i_\alpha,\alpha} \right],$$

where the products over the branches without and with mutations have been explicitly separated. (The mutation counts and branch lengths is also supplied with the site index). The log-likelihood is then (after some simple algebraic transformations):

$$\begin{aligned} \log \mathcal{L} = \sum_{\alpha} \Big[& - \sum_{i,j \neq i} \pi_{j\alpha} \mu_\alpha W_{ji} T_{i\alpha} + \\ & + \sum_{i,j \neq i} n_{ij,\alpha} \big(\log \pi_{i\alpha} + \log \mu_\alpha + \log W_{ij} \big) + \\ & + \sum_{i,j \neq i} \sum_{k=1}^{n_{ij,\alpha}} \log t_k^{(ij,\alpha)} + \log \pi_{i_\alpha,\alpha} \Big], \end{aligned} \tag{47}$$

where $T_{i\alpha}$ denotes $\sum_{k=1}^{n_{ii,\alpha}} t_k^{(ii,\alpha)}$, which is approximately the time on the tree that site α spends in the state i.

The assumptions of the short branches (on the scale of the mutation rate) makes it possible maximizing the tree likelihood analytically. In particular, to find the maximum-likelihood values for the evolutionary model, the following expression should be maximized:

$$\log \mathcal{L} - \sum_{\alpha} \lambda_\alpha \left(\sum_{i} \pi_{i,\alpha} - 1 \right) \to \max \tag{48}$$

in respect to $\pi_{i\alpha}$, μ_α, W_{ij}. The Lagrange multipliers λ_α are introduced to ensure the correct normalization of the equilibrium nucleotide probabilities.

The resulting equations to determine the GTR parameters for the extremum position of the likelihood function are:

$$\begin{aligned}
\sum_{j\neq i} n_{ij,\alpha} + \delta_{i,i_\alpha} &= \mu_\alpha \pi_{i\alpha} \sum_{j\neq i} W_{ij} T_{j\alpha} + \lambda_\alpha \pi_{i\alpha}, \\
\sum_{i,j\neq i} n_{ij,\alpha} &= \mu_\alpha \sum_{i,j\neq i} \pi_{i\alpha} W_{ij} T_{j\alpha}, \\
\sum_{\alpha} \left(n_{ij,\alpha} + n_{ji,\alpha}\right) &= \sum_{\alpha} \left(\mu_\alpha \pi_{i\alpha} W_{ij} T_{j\alpha} + \mu_\alpha \pi_{j\alpha} W_{ij} T_{i\alpha}\right) \\
\sum_{i} \pi_{i\alpha} &= 1. \qquad (49)
\end{aligned}$$

(In this derivation one should keep in mind that, due to the symmetry of matrix W_{ij}, $\partial W_{ij}/\partial W_{rs} = \delta_{i,r}\delta_{j,s} + \delta_{i,s}\delta_{j,r}$.)

Final equations

Summing the first of eqs. 49 over i and using the second and the last equations immediately produces the value of the Lagrange multipliers: $\lambda_\alpha = 1$. Given that, the eqs. 49 can be reformulate in the form suitable for iterative solution:

$$W_{ij} = \frac{\sum_\alpha \left(n_{ij,\alpha} + n_{ji,\alpha}\right)}{\sum_\alpha \left(\mu_\alpha \pi_{i\alpha} T_{j\alpha} + \mu_\alpha \pi_{j\alpha} T_{i\alpha}\right)}, \qquad (50)$$

$$\mu_\alpha = \frac{\sum_{i,j\neq i} n_{ij,\alpha}}{\sum_{i,j\neq i} \pi_{i\alpha} W_{ij} T_{j\alpha}},$$

$$\pi_{i\alpha} = \frac{\sum_{j\neq i} n_{ij,\alpha} + \delta_{i,i_\alpha}}{\mu_\alpha \sum_{j\neq i} W_{ij} T_{j\alpha} + 1}, \qquad (51)$$

which readily ensures the proper normalization for the stationary probabilities. From technical viewpoint, eqs. (49) define the expressions to find the maximum-likelihood parameters of the GTR model under the assumptions taken.

Area of applicability

The rate of mutations at a particular site can be characterized by quantity:

$$\Gamma_\alpha = \mu_\alpha \sum_{i,j \neq i} \pi_{i,\alpha} W_{i,j} \pi_{j,\alpha} \tag{52}$$

The branch length averaged over the tree is $t = \overline{t_k}$.

The applicability of the reconstruction algorithm requires that

$$\Gamma \overline{t_k} \ll 1 \tag{53}$$

The reasonable signal-to-noise ratio requires that there is more than one mutation at every site, although they are still rare on the tree, i.e.

$$\Gamma T = \Gamma N_{br} \overline{t_k} \gg 1, \tag{54}$$

where $T = N_{br} \overline{t_k}$ is the total tree lengths, i.e. the number of branches times the mean branch length.

Finally, to ensure reliable tree reconstruction, the average number of the mutations per branch should be more than one, i.e.

$$\Gamma \overline{t_k} L \gg 1, \tag{55}$$

where L is the number of sites in the sequence.

6.3 GTR INFERENCE SCHEME VALIDATION

Simulating sequence evolution

The GTR inference scheme has been tested on the simulated data. The creation of the simulated dataset included the tree topology simulations, cre-

ation of a random GTR model, generations of the ancestral sequence at the root node, evolving the root sequence in the simulated tree using the GTR model created. The evolved sequences from the tips of the tree comprised the multiple sequence alignment, which used as an input alignment for the GTR model reconstruction. The tree topologies were generated according to specified coalescent models, using the existing software, previously developed by our group [Neher et al., 2013].

The parameters of the GTR model were chosen as follows. The attempt matrix had the Jukes-Cantor [Jukes and Cantor, 1969] form, i.e. all of its non-diagonal elements were identical. The mutation rates, μ_α were chosen to be either uniform or selected from a Gamma distribution with specified average μ. The nucleotide/residue frequencies for every site were generated randomly in such a way that they had a uniform distribution on simplex boundary $\sum_i \pi_{i\alpha} = 1$. This is easily done by generating a set of exponentially distributed random numbers $\{x_i\}$ and dividing them by their sum. (The product of exponential distributions is a uniform distribution on any surface $y = \sum_i x_i$.)

The root sequence was then generated according to the probabilities $\pi_{i\alpha}$.

The probabilities of the nucleotide states of the root descendants were then calculated using the (site-specific) transition probability Eq. (42). The sequences of the descendants were then "measured", i.e. chosen according to these probabilities and the procedure was repeated till we reached the tips of the tree. Finally, the sequences of the tips were taken as the alignment, which was used to reconstruct phylogeny, ancestral sequences and the GTR model.

Phylogeny and ancestral reconstruction

The phylogenetic tree reconstruction was accomplished using FastTree package [Price et al., 2009, 2010], which uses neighbor joining algorithm followed by the maximum likelihood (ML) optimization. The FastTree maximum-likelihood optimization is performed using a standard GTR mode. For the nucleotide sequence it is Jukes-Cantor model, for amino acid, it is either WAG01 [Whelan and Goldman, 2001], or the evolution model based on BLOSUM [Henikoff and Henikoff, 1993] matrices. The model used is identical for the whole genome. The ancestral sequences reconstruction for every node of the tree was accomplished using the previously described `TreeTime` software [Sagulenko et al., 2017].

Testing the inference scheme

In order to validate the inference scheme and determine its limits of its validity, the sequence evolution has been simulated according to known (predefined) phylogeny and GTR model for different range of input parameters. We adopt the following simple measure of the reconstruction quality:

$$\chi^2_\pi = \frac{1}{L}\sum_{i,\alpha}\left(\pi^{(0)}_{i,\alpha} - \pi_{i,\alpha}\right)^2, \tag{56}$$

where the sum runs over all nucleotide/residue alphabet states and all sequence sites, $\pi^{(0)}_{i,\alpha}$ and $\pi_{i,\alpha}$ are the frequency of the states for the input and the inferred models respectively, L is the length of the sequence.

In practice one would usually start with the alignment of sequences corresponding to the tips of the phylogenetic tree. The residue frequencies computed from this alignment are likely to be very different from those of the underlying GTR model. One then has to (i) reconstruct the phylogeny,

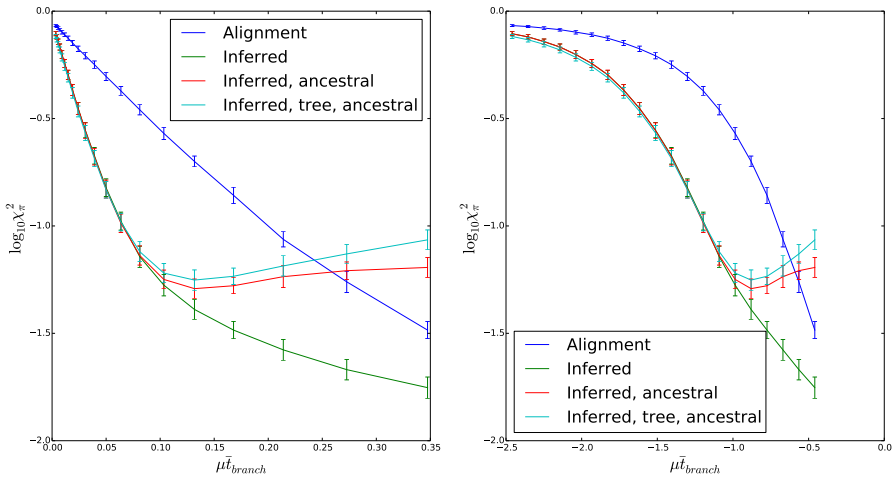

Figure 22: Distance between the model allele frequencies and the allele frequencies calculated from the tree tips alignment and GTR reconstruction. (Bootstrapped over 100 simulations.)

(ii) reconstruct the ancestral sequences for every node of the phylogenetic tree, and (iii) infer the GTR model. Each of the above mentioned steps may introduce errors. The errors introduced on different steps were identified, using the knowledge of the phylogenetic tree, and the actual internal node sequences, as shown below.

The results of GTR reconstruction for ACGT alphabet are shown in figure 22. For this simulation the mutation rate was assumed to be uniform for all sites, whereas the underlying phylogeny corresponded to the Kingman coalescent. As expected, the nucleotide frequencies in the alignment significantly differ from those of the model, but approach them as the mutation rate increases, i.e. as the alignment becomes more equilibrated. This improvement with mutation rate is roughly linear on semi-logarithmic scale, which corresponds to the exponential in eq. 42.

The frequency inferences based in GTR reconstruction procedures significantly outperform the alignment average, even though they perform just as bad at very small values of the mutation rate, where one simply doesn't have enough statistical data (i.e. enough mutation events) for reliable in-

ference. When the mutation rate is of a moderate value, the number of mutations can be approximately considered as a Poisson variable, i.e. the error squared in GTR reconstruction should decrease inversely proportional to the number of the mutations counted on the tree, i.e. inversely proportional to the mutation rate. This is indeed seen in the log-log plot in the right panel of figure 22, where the dependence of χ^2_π on the mutation rate is almost a straight line with slope 1.

The highest mutation rates used in this simulation are still sufficiently small for the assumptions of our GTR inference procedure to be valid. Correspondingly, the curve in figure 22, corresponding to the reconstruction using known topology and ancestral sequence, is monotonously decreasing, showing the improved quality of reconstruction with increasing number of mutation events. The necessity of reconstructing phylogeny and ancestral sequences however imposes a limitation, as seen from the other curves in the same figure. Indeed, as the mutation rate increases, we have a higher probability of recurring mutation at the same sequence site, which cannot be detected by any algorithms for phylogenetic and ancestral reconstruction.

The last assertion can be verified by looking at the reconstructed mutation rate, shown in figure 23. Here the error bars correspond not to different realizations of sequence evolution, but to the distribution of the mutation rates along the sequence. The saturation of the mutation rate when performing ancestral reconstruction reflects the impossibility of dealing with sites where multiple mutations have occurred. In figure 24 we show how the divergence of eq. 56 scales with the sequence length. The reconstruction procedure itself can depend on the sequence length only via non-site-specific matrix W_{ij}, which is inferred more precisely for longer sequences. Longer sequences however allow for more precise phylogenetic inference, which is seen by downward shift of the line corresponding to the calculation involving phylogenetic reconstruction.

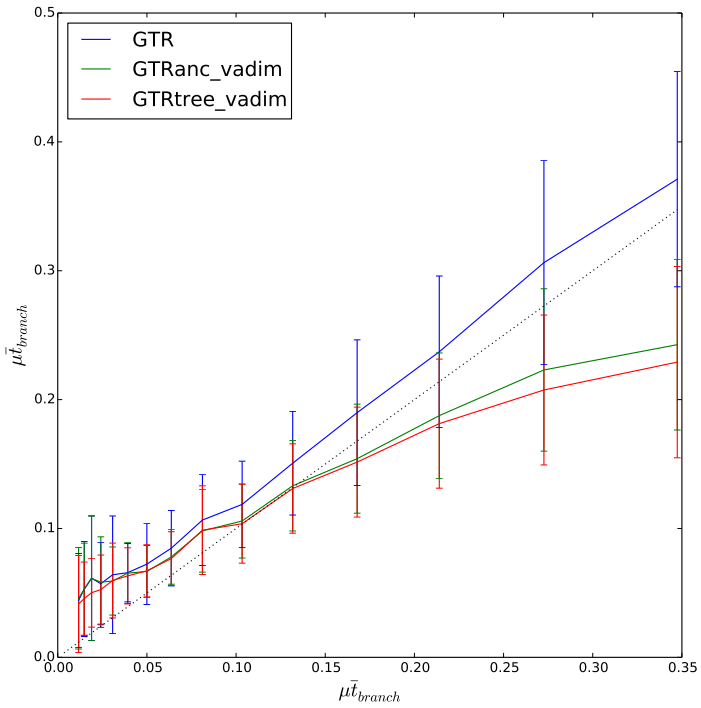

Figure 23: Reconstructed mutation rate vs. the model mutation rate. (Sequence length L = 100, averaged over sites.)

We finally address the issue of the number of parameters of the GTR model. Increasing the number of parameters, e.g., by making the model site-specific, necessarily improves the fit of the data (i.e. increases likelihood). However, too many parameters may result in over-fitting, i.e. one may have insufficient data to infer reliably the model parameters. This is a particular risk when inferring site-specific GTR models: since the sequences in the alignment are related via phylogenetic relations, adding more and more sequences does not necessarily improve the parameter inference.

One can compare the informativeness of different models by using, e.g., Akaike information criterion (AIC) [Akaike, 2011], which balances the num-

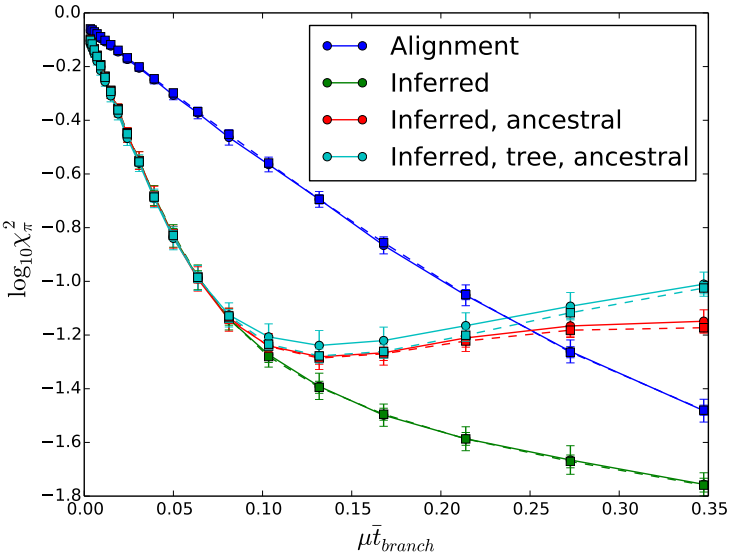

Figure 24: Scaling of the distance between the reconstructed and the model nucleotide frequencies with sequence length: solid lines with circles correspond to $L = 100$, dashed lines with squares to $L = 400$. (Boot-strapped over 100 simulations.)

ber of parameters, k, with the maximum value of the likelihood, $\log \mathcal{L}$ as

$$AIC = k - \log \mathcal{L}. \tag{57}$$

The optimal model choice thus results in smaller AIC values.

In figure 25 we show the values of the likelihoods when using three different models: a) the model with site-specific nucleotide frequencies and mutation rates, but a single for all sites attempt matrix, which has $qL + q(q-1)/2 - 1$ parameters (this is the model that has been used throughout this paper), b) the model describing all sites by the same GTR matrix, with $q - 1 + q(q-1)/2$ parameters, and c) the fully site-specific model with $(q-1)L + q(q-1)L$ parameters. The site specific models have higher likelihoods than when using a single model to fit all sites. The fully site-specific model however only slightly out-performs the model with the attempt matrix common for all sites, and when the number of parameters is incorpo-

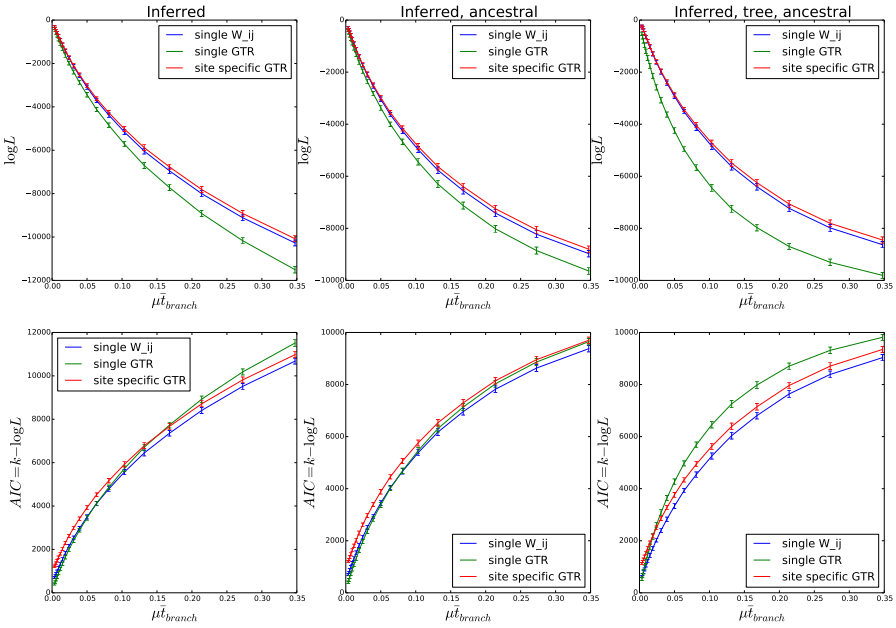

Figure 25: Maximum values of likelihood ($\log L$) and Akaike information criterion for inference with a) model with site-specific $\pi_{i\alpha}$ and μ_α but single W_{ij} for all sites, b) single GTR model for all sites; c) fully site-specific GTR model (i.e. with site-specific $W_{ij,\alpha}$.) $L = 100$, bootstrapped over 100 simulations.

rated via AIC the latter outperforms the former. We thus conclude that specifying a fully unique GTR model for every site is likely to result in over-parametrization.

BIBLIOGRAPHY

Hirotugu Akaike. Akaike's Information Criterion. In Miodrag Lovric, editor, *International Encyclopedia of Statistical Science*, pages 25–25. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-04897-5 978-3-642-04898-2. URL http://link.springer.com/referenceworkentry/10.1007/978-3-642-04898-2_110. DOI: 10.1007/978-3-642-04898-2_110.

Stéphane Aris-Brosou, Ziheng Yang, and John Huelsenbeck. Effects of Models of Rate Evolution on Estimation of Divergence Dates with Special Reference to the Metazoan 18s Ribosomal RNA Phylogeny. *Systematic Biology*, 51(5):703–714, September 2002. ISSN 1076-836X, 1063-5157. doi: 10.1080/10635150290102375. URL http://academic.oup.com/sysbio/article/51/5/703/1678442/Effects-of-Models-of-Rate-Evolution-on-Estimation.

Samuel Blanquart and Nicolas Lartillot. A Site- and Time-Heterogeneous Model of Amino Acid Replacement. *Mol Biol Evol*, 25(5):842–858, May 2008. ISSN 0737-4038. doi: 10.1093/molbev/msn018. URL https://academic.oup.com/mbe/article/25/5/842/1196978/A-Site-and-Time-Heterogeneous-Model-of-Amino-Acid.

Tom Britton, Cajsa Lisa Anderson, David Jacquet, Samuel Lundqvist, Kåre Bremer, and Frank Anderson. Estimating Divergence Times in Large Phylogenetic Trees. *Systematic Biology*, 56(5):741–752, October 2007. ISSN 1076-836X, 1063-5157. doi: 10.1080/10635150701613783. URL http://academic.oup.com/sysbio/article/56/5/741/1694998/Estimating-Divergence-Times-in-Large-Phylogenetic.

Benny Chor and Sagi Snir. Molecular Clock Fork Phylogenies: Closed Form Analytic Maximum Likelihood Solutions. *Systematic Biology*, 53(6):963–967, 2004. ISSN 1063-5157. URL http://www.jstor.org/stable/4135381.

Benny Chor, Amit Khetan, and Sagi Snir. Maximum Likelihood Molecular Clock Comb: Analytic Solutions. *Journal of Computational Biology*, 13(3): 819–837, April 2006. doi: 10.1089/cmb.2006.13.819. URL http://online.liebertpub.com/doi/abs/10.1089/cmb.2006.13.819.

Russell F Doolittle, Da-Fei Feng, Simon Tsang, Glen Cho, Elizabeth Little, and others. Determining divergence times of the major kingdoms of living organisms with a protein clock. *Science*, 271(5248):470–476, 1996.

Mario dos Reis, Philip C. J. Donoghue, and Ziheng Yang. Bayesian molecular clock dating of species divergences in the genomics era. *Nature Reviews Genetics*, 17(2):71–80, December 2015. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg.2015.8. URL http://www.nature.com/doifinder/10.1038/nrg.2015.8.

Alexei J Drummond, Simon Y. W Ho, Matthew J Phillips, and Andrew Rambaut. Relaxed Phylogenetics and Dating with Confidence. *PLoS Biology*, 4(5):e88, March 2006. ISSN 1545-7885. doi: 10.1371/journal.pbio.0040088. URL http://dx.plos.org/10.1371/journal.pbio.0040088.

Alexei J. Drummond, Marc A. Suchard, Dong Xie, and Andrew Rambaut. Bayesian Phylogenetics with BEAUti and the BEAST 1.7. *Molecular Biology and Evolution*, 29(8):1969–1973, August 2012. ISSN 1537-1719, 0737-4038. doi: 10.1093/molbev/mss075. URL https://academic.oup.com/mbe/article-lookup/doi/10.1093/molbev/mss075.

Gytis Dudas, Luiz Max Carvalho, Trevor Bedford, Andrew J. Tatem, Guy Baele, Nuno R. Faria, Daniel J. Park, Jason T. Ladner, Armando Arias, Danny Asogun, Filip Bielejec, Sarah L. Caddy, Matthew Cotten, Jonathan

D'Ambrozio, Simon Dellicour, Antonino Di Caro, Joseph W. Diclaro, Sophie Duraffour, Michael J. Elmore, Lawrence S. Fakoli, Ousmane Faye, Merle L. Gilbert, Sahr M. Gevao, Stephen Gire, Adrianne Gladden-Young, Andreas Gnirke, Augustine Goba, Donald S. Grant, Bart L. Haagmans, Julian A. Hiscox, Umaru Jah, Jeffrey R. Kugelman, Di Liu, Jia Lu, Christine M. Malboeuf, Suzanne Mate, David A. Matthews, Christian B. Matranga, Luke W. Meredith, James Qu, Joshua Quick, Suzan D. Pas, My V. T. Phan, Georgios Pollakis, Chantal B. Reusken, Mariano Sanchez-Lockhart, Stephen F. Schaffner, John S. Schieffelin, Rachel S. Sealfon, Etienne Simon-Loriere, Saskia L. Smits, Kilian Stoecker, Lucy Thorne, Ekaete Alice Tobin, Mohamed A. Vandi, Simon J. Watson, Kendra West, Shannon Whitmer, Michael R. Wiley, Sarah M. Winnicki, Shirlee Wohl, Roman Wölfel, Nathan L. Yozwiak, Kristian G. Andersen, Sylvia O. Blyden, Fatorma Bolay, Miles W. Carroll, Bernice Dahn, Boubacar Diallo, Pierre Formenty, Christophe Fraser, George F. Gao, Robert F. Garry, Ian Goodfellow, Stephan Günther, Christian T. Happi, Edward C. Holmes, Brima Kargbo, Sakoba Keïta, Paul Kellam, Marion P. G. Koopmans, Jens H. Kuhn, Nicholas J. Loman, N'Faly Magassouba, Dhamari Naidoo, Stuart T. Nichol, Tolbert Nyenswah, Gustavo Palacios, Oliver G. Pybus, Pardis C. Sabeti, Amadou Sall, Ute Ströher, Isatta Wurie, Marc A. Suchard, Philippe Lemey, and Andrew Rambaut. Virus genomes reveal factors that spread and sustained the Ebola epidemic. *Nature*, 544(7650):309–315, April 2017. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature22040. URL http://www.nature.com/doifinder/10.1038/nature22040.

Justin C. Fay and Chung-I. Wu. Sequence Divergence, Functional Constraint, and Selection in Protein Evolution. *Annual Review of Genomics and Human Genetics*, 4(1):213–235, 2003. doi: 10.1146/annurev.genom.4.020303.162528. URL https://doi.org/10.1146/annurev.genom.4.020303.162528.

J. Felsenstein. *Inferring Phylogenies*. Sinauer, 2003. ISBN 978-0-87893-177-4. URL https://books.google.de/books?id=GI6PQgAACAAJ.

Joseph Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *J Mol Evol*, 17(6):368–376, November 1981. ISSN 0022-2844, 1432-1432. doi: 10.1007/BF01734359. URL https://link.springer.com/article/10.1007/BF01734359.

Walter M. Fitch. Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. *Systematic Zoology*, 20(4):406, December 1971. ISSN 00397989. doi: 10.2307/2412116. URL http://www.jstor.org/stable/2412116?origin=crossref.

Flu-DataBase. Influenza research database, 2017. URL https://www.fludb.org/brc/home.spg?decorator=influenza.

GitHub. React-js source code, a. URL https://facebook.github.io/react/.

GitHub. Treetime source code, b. URL https://github.com/neherlab/treetime.

GitHub. Treetime validation scripts, c. URL https://github.com/neherlab/treetime_validation.

M. Hasegawa, H. Kishino, and T. Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J Mol Evol*, 22(2):160–174, 1985. ISSN 0022-2844 0022-2844.

Masami Hasegawa, Hirohisa Kishino, and Taka-aki Yano. Estimation of branching dates among primates by molecular clocks of nuclear DNA which slowed down in Hominoidea. *Journal of Human Evolution*, 18 (5):461–476, August 1989. ISSN 0047-2484. doi: 10.1016/0047-2484(89)90075-4. URL http://www.sciencedirect.com/science/article/pii/0047248489900754.

S. B. Hedges, J. Dudley, and S. Kumar. TimeTree: a public knowledge-base of divergence times among organisms. *Bioinformatics*, 22(23):2971–2972, December 2006. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btl505. URL https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btl505.

Steven Henikoff and Jorja G. Henikoff. Performance evaluation of amino acid substitution matrices. *Proteins*, 17(1):49–61, September 1993. ISSN 1097-0134. doi: 10.1002/prot.340170108. URL http://onlinelibrary.wiley.com/doi/10.1002/prot.340170108/abstract.

HIV-DataBase. Hiv sequence databases, 2017. URL https://www.hiv.lanl.gov/content/index.

Simon Y. W. Ho and Sebastián Duchêne. Molecular-clock methods for estimating evolutionary rates and timescales. *Mol Ecol*, 23(24):5947–5965, December 2014. ISSN 1365-294X. doi: 10.1111/mec.12953. URL http://onlinelibrary.wiley.com/doi/10.1111/mec.12953/abstract.

Thomas H. Jukes and Charles Cantor. Evolution of Protein Molecules. In H. N. Munro, editor, *Mammalian Protein Metabolism*. Elsevier, 1969. ISBN 978-1-4832-7290-0. Google-Books-ID: FDHLBAAAQBAJ.

Patrick J. Keeling and Jeffrey D. Palmer. Horizontal gene transfer in eukaryotic evolution. *Nature Reviews Genetics*, 9(8):605–618, August 2008. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg2386. URL http://www.nature.com/doifinder/10.1038/nrg2386.

J. F. C. Kingman. The coalescent. *Stochastic Processes and their Applications*, 13 (3):235–248, September 1982. ISSN 0304-4149. doi: 10.1016/0304-4149(82)90011-4. URL http://www.sciencedirect.com/science/article/pii/0304414982900114.

Hirohisa Kishino, Jeffrey L. Thorne, and William J. Bruno. Performance of a Divergence Time Estimation Method under a Probabilistic Model of Rate Evolution. *Mol Biol Evol*, 18(3):352–361, March 2001. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a003811. URL https://academic.oup.com/mbe/article/18/3/352/1073229/Performance-of-a-Divergence-Time-Estimation-Method.

Sudhir Kumar and S. Blair Hedges. Advances in Time Estimation Methods for Molecular Data. *Molecular Biology and Evolution*, 33(4):863–869, April 2016. ISSN 0737-4038, 1537-1719. doi: 10.1093/molbev/msw026. URL https://academic.oup.com/mbe/article-lookup/doi/10.1093/molbev/msw026.

Sudhir Kumar, Glen Stecher, Michael Suleski, and S. Blair Hedges. TimeTree: A Resource for Timelines, Timetrees, and Divergence Times. *Molecular Biology and Evolution*, 34(7):1812–1819, July 2017. ISSN 0737-4038, 1537-1719. doi: 10.1093/molbev/msx116. URL https://academic.oup.com/mbe/article-lookup/doi/10.1093/molbev/msx116.

Charles H. Langley and Walter M. Fitch. An examination of the constancy of the rate of molecular evolution. *J Mol Evol*, 3(3):161–177, September 1974. ISSN 0022-2844, 1432-1432. doi: 10.1007/BF01797451. URL https://link.springer.com/article/10.1007/BF01797451.

Nicolas Lartillot and Hervé Philippe. A Bayesian Mixture Model for Across-Site Heterogeneities in the Amino-Acid Replacement Process. *Molecular Biology and Evolution*, 21(6):1095–1109, June 2004. ISSN 1537-1719, 0737-4038. doi: 10.1093/molbev/msh112. URL https://academic.oup.com/mbe/article-lookup/doi/10.1093/molbev/msh112.

Thomas Leitner and Jan Albert. The molecular clock of HIV-1 unveiled through analysis of a known transmission history. *PNAS*, 96(19):10752–

10757, September 1999. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.96.19.10752. URL http://www.pnas.org/content/96/19/10752.

E. Margoliash. Primary structure and evolution of cytochrome C. *Proceedings of the National Academy of Sciences*, 50:672–679, 1963. URL http://www.pnas.org/content/50/4/672.full.pdf.

Vladimir N. Minin and Marc A. Suchard. Counting labeled transitions in continuous-time Markov models of evolution. *J. Math. Biol.*, 56 (3):391–412, March 2008. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-007-0120-8. URL https://link.springer.com/article/10.1007/s00285-007-0120-8.

Priya Moorjani, Carlos Eduardo G. Amorim, Peter F. Arndt, and Molly Przeworski. Variation in the molecular clock of primates. *Proceedings of the National Academy of Sciences*, 113(38):10607–10612, September 2016. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1600374113. URL http://www.pnas.org/lookup/doi/10.1073/pnas.1600374113.

Richard A. Neher. Genetic Draft, Selective Interference, and Population Genetics of Rapid Adaptation. *Annual Review of Ecology, Evolution, and Systematics*, 44(1):195–215, 2013. doi: 10.1146/annurev-ecolsys-110512-135920. URL https://doi.org/10.1146/annurev-ecolsys-110512-135920.

Richard A. Neher and Trevor Bedford. nextflu: real-time tracking of seasonal influenza virus evolution in humans. *Bioinformatics*, 31(21):3546–3548, November 2015. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btv381. URL https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btv381.

Richard A. Neher, Taylor A. Kessinger, and Boris I. Shraiman. Coalescence and genetic diversity in sexual populations under selection. *PNAS*, 110(39):15836–15841, September 2013. ISSN 0027-8424, 1091-6490. doi:

10.1073/pnas.1309697110. URL http://www.pnas.org/content/110/39/15836.

Nextstrain. Nextstrain: Real-time tracking of virus evolution., 2017. URL http://www.nextstrain.org/.

Magnus Nordborg. Structured Coalescent Processes on Different Time Scales. *Genetics*, 146(4):1501–1514, August 1997. ISSN 0016-6731, 1943-2631. URL http://www.genetics.org/content/146/4/1501.

Howard Ochman, Jeffrey G. Lawrence, and Eduardo A. Groisman. Lateral gene transfer and the nature of bacterial innovation. *Nature*, 405(6784):299–304, May 2000. URL https://search.proquest.com/docview/204485167?accountid=104721.

Mark Pagel, Andrew Meade, and Keith Crandall. A Phylogenetic Mixture Model for Detecting Pattern-Heterogeneity in Gene Sequence or Character-State Data. *Syst Biol*, 53(4):571–581, August 2004. ISSN 1063-5157. doi: 10.1080/10635150490468675. URL https://academic.oup.com/sysbio/article/53/4/571/1646012/A-Phylogenetic-Mixture-Model-for-Detecting-Pattern.

Kosakovsky Pond, Sergei L, and Simon D. W. Frost. Not So Different After All: A Comparison of Methods for Detecting Amino Acid Sites Under Selection. *Mol Biol Evol*, 22(5):1208–1222, May 2005. ISSN 0737-4038. doi: 10.1093/molbev/msi105. URL https://academic.oup.com/mbe/article/22/5/1208/1066893/Not-So-Different-After-All-A-Comparison-of-Methods.

M. N. Price, P. S. Dehal, and A. P. Arkin. FastTree: Computing Large Minimum Evolution Trees with Profiles instead of a Distance Matrix. *Molecular Biology and Evolution*, 26(7):1641–1650, July 2009. ISSN 0737-4038, 1537-

1719. doi: 10.1093/molbev/msp077. URL https://academic.oup.com/mbe/article-lookup/doi/10.1093/molbev/msp077.

Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLOS ONE*, 5(3):e9490, March 2010. doi: 10.1371/journal.pone.0009490. URL https://doi.org/10.1371/journal.pone.0009490.

Tal Pupko, Itsik Pe, Ron Shamir, and Dan Graur. A Fast Algorithm for Joint Reconstruction of Ancestral Amino Acid Sequences. *Molecular Biology and Evolution*, 17(6):890–896, June 2000. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a026369. URL http://dx.doi.org/10.1093/oxfordjournals.molbev.a026369.

Andrew Rambaut. Estimating the rate of molecular evolution: incorporating non-contemporaneous sequences into maximum likelihood phylogenies. *Bioinformatics*, 16(4):395–399, April 2000. ISSN 1367-4803. doi: 10.1093/bioinformatics/16.4.395. URL http://dx.doi.org/10.1093/bioinformatics/16.4.395.

Andrew Rambaut, Tommy T. Lam, Luiz Max Carvalho, and Oliver G. Pybus. Exploring the temporal structure of heterochronous sequences using TempEst (formerly Path-O-Gen). *Virus Evolution*, 2(1):vew007, January 2016. ISSN 2057-1577. doi: 10.1093/ve/vew007. URL https://academic.oup.com/ve/article-lookup/doi/10.1093/ve/vew007.

Mario dos Reis, Alan J. Hay, and Richard A. Goldstein. Using Non-Homogeneous Models of Nucleotide Substitution to Identify Host Shift Events: Application to the Origin of the 1918 'Spanish' Influenza Pandemic Virus. *J Mol Evol*, 69(4):333, October 2009. ISSN 0022-2844, 1432-1432. doi: 10.1007/s00239-009-9282-x. URL https://link.springer.com/article/10.1007/s00239-009-9282-x.

Nicolas Rodrigue. On the Statistical Interpretation of Site-Specific Variables in Phylogeny-Based Substitution Models. *Genetics*, 193(2):557–564, February 2013. ISSN 0016-6731, 1943-2631. doi: 10.1534/genetics.112.145722. URL http://www.genetics.org/content/193/2/557.

Pavel Sagulenko, Vadim Puller, and Richard Neher. TreeTime: maximum likelihood phylodynamic analysis. *bioRxiv*, page 153494, August 2017. doi: 10.1101/153494. URL https://www.biorxiv.org/content/early/2017/08/02/153494.

Michael J. Sanderson. r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. *Bioinformatics*, 19(2):301–302, January 2003. ISSN 1367-4803. doi: 10.1093/bioinformatics/19.2.301. URL http://dx.doi.org/10.1093/bioinformatics/19.2.301.

Michael Schöniger and Arndt Von Haeseler. A Stochastic Model for the Evolution of Autocorrelated DNA Sequences. *Molecular Phylogenetics and Evolution*, 3(3):240–247, September 1994. ISSN 1055-7903. doi: 10.1006/mpev.1994.1026. URL http://www.sciencedirect.com/science/article/pii/S1055790384710268.

Beth Shapiro, Andrew Rambaut, and Alexei J. Drummond. Choosing Appropriate Substitution Models for the Phylogenetic Analysis of Protein-Coding Sequences. *Mol Biol Evol*, 23(1):7–9, January 2006. ISSN 0737-4038. doi: 10.1093/molbev/msj021. URL https://academic.oup.com/mbe/article/23/1/7/1193608/Choosing-Appropriate-Substitution-Models-for-the.

Korbinian Strimmer and Oliver G. Pybus. Exploring the Demographic History of DNA Sequences Using the Generalized Skyline Plot. *Molecular Biology and Evolution*, 18(12):2298–2305, December 2001. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a003776. URL http://dx.doi.org/10.1093/oxfordjournals.molbev.a003776.

Yoshiyuki Suzuki and Masatoshi Nei. Origin and Evolution of Influenza Virus Hemagglutinin Genes. *Molecular Biology and Evolution*, 19(4):501–509, April 2002. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a004105. URL http://dx.doi.org/10.1093/oxfordjournals.molbev.a004105.

K. Tamura, F. U. Battistuzzi, P. Billing-Ross, O. Murillo, A. Filipski, and S. Kumar. Estimating divergence times in large molecular phylogenies. *Proceedings of the National Academy of Sciences*, 109(47):19333–19338, November 2012. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1213199109. URL http://www.pnas.org/cgi/doi/10.1073/pnas.1213199109.

J L Thorne, H Kishino, and I S Painter. Estimating the rate of evolution of the rate of molecular evolution. *Molecular Biology and Evolution*, 15(12):1647–1657, December 1998. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a025892. URL http://dx.doi.org/10.1093/oxfordjournals.molbev.a025892.

Bargavi Thyagarajan and Jesse D Bloom. The inherent mutational tolerance and antigenic evolvability of influenza hemagglutinin. *eLife*, 3, July 2014. ISSN 2050-084X. doi: 10.7554/eLife.03300. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4109307/.

Thu-Hien To, Matthieu Jung, Samantha Lycett, and Olivier Gascuel. Fast Dating Using Least-Squares Criteria and Algorithms. *Systematic Biology*, 65(1):82–97, January 2016. ISSN 1063-5157, 1076-836X. doi: 10.1093/sysbio/syv068. URL https://academic.oup.com/sysbio/article-lookup/doi/10.1093/sysbio/syv068.

treetime.ch. Treetime web server. URL http://treetime.biozentrum.unibas.ch/.

Peter J Waddell and M. A Steel. General Time-Reversible Distances with Unequal Rates across Sites: Mixing γ Invariant Sites. *Molecular Phyloge-*

netics and Evolution, 8(3):398–414, December 1997. ISSN 1055-7903. doi: 10.1006/mpev.1997.0452. URL http://www.sciencedirect.com/science/article/pii/S1055790397904528.

Martin Weigt, Robert A. White, Hendrik Szurmant, James A. Hoch, and Terence Hwa. Identification of direct residue contacts in protein–protein interaction by message passing. *PNAS*, 106(1):67–72, January 2009. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0805923106. URL http://www.pnas.org/content/106/1/67.

Simon Whelan and Nick Goldman. A General Empirical Model of Protein Evolution Derived from Multiple Protein Families Using a Maximum-Likelihood Approach. *Mol Biol Evol*, 18(5):691–699, May 2001. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a003851. URL https://academic.oup.com/mbe/article/18/5/691/1018653/A-General-Empirical-Model-of-Protein-Evolution.

H. L. K. Whitehouse. *Genetic recombination*. John Wiley & Sons, Chichester, 1982. ISBN 0-471-10205-9.

WHO. 2014 Ebola outbreak in West Africa - case counts, 2016. URL https://www.cdc.gov/vhf/ebola/csv/graph1-cumulative-reported-cases-all.xlsx.

Ziheng Yang. *Computational molecular evolution*. Oxford series in ecology and evolution. Oxford University Press, Oxford, 2006. ISBN 978-0-19-856699-1 978-0-19-856702-8. OCLC: ocm72868007.

Anne D. Yoder and Ziheng Yang. Estimation of Primate Speciation Dates Using Local Molecular Clocks. *Molecular Biology and Evolution*, 17(7):1081–1090, July 2000. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a026389. URL http://dx.doi.org/10.1093/oxfordjournals.molbev.a026389.

Fabio Zanini and Richard A. Neher. FFPopSim: an efficient forward simulation package for the evolution of large populations. *Bioinformatics*, 28(24):3332–3333, December 2012. ISSN 1460-2059, 1367-4803. doi: 10.1093/bioinformatics/bts633. URL https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/bts633.

Fabio Zanini, Johanna Brodin, Lina Thebo, Christa Lanz, Göran Bratt, Jan Albert, and Richard A. Neher. Population genomics of intrapatient HIV-1 evolution. *eLife Sciences*, 4:e11282, December 2015. ISSN 2050-084X. doi: 10.7554/eLife.11282. URL https://elifesciences.org/articles/11282.

Emil Zuckerkandl and Linus Pauling. Molecules as Documents of Evolutionary History. *Journal of Theoretical Biology*, 8:357–366, 1965.

DECLARATION

I declare that the thesis has been composed by myself and that the work has not be submitted for any other degree or professional qualification. I confirm that the work submitted is my own, except where work which has formed part of jointly-authored publications has been included. My contribution and those of the other authors to this work have been explicitly indicated below.

The `TreeTime` framework was previously published in [Sagulenko et al., 2017]. The work has been performed by me and by my supervisor, Prof. R. Neher. I have developed all algorithms for the `TreeTime` framework, except for the coalescence models part. The python code for the `TreeTime` package has been written mostly by me, with contributions from Prof. Neher. The Ebola outbreak analysis (fig. 11) has been performed by Prof. Neher. `TreeTime` validation part in Chapter 4 has been designed and performed by me, excluding the part in fig. 14. The web-server design and implementation has been done by me alone.

The evolution models inference, presented in Chapter 6 has been made in collaboration among me, Dr. V.Puller and Prof. R. Neher. It is currently being prepared to publication by Dr. Puller. I contributed to the development of the iterative scheme for GTR model inference, and to the implementation of the iterative scheme in python code. The data analysis, and the area of applicability of the GTR model inference scheme (figs. 22 — 25) has been performed entirely by Dr. Puller and Prof. Neher.

Tübingen, October 1st 2017

Pavel Sagulenko

LIST OF MY PUBLICATIONS

James Hadfield, Colin Megill, Sidney Bell, John Huddleston, Barney Potter, Charlton Callender, Pavel Sagulenko, Trevor Bedford, and Richard Neher. Nextstrain: real-time tracking of virus evolution. *(in preparation)*, 2017.

Vadim Puller, Pavel Sagulenko, and Richard Neher. Nucleotide frequency inference using gtr model on a tree. *(in preparation)*, 2017.

Pavel Sagulenko, Vadim Puller, and Richard Neher. TreeTime: maximum likelihood phylodynamic analysis. *bioRxiv*, page 153494, August 2017. doi: 10.1101/153494. URL https://www.biorxiv.org/content/early/2017/08/02/153494.