Adaptive Tree Proposals for Bayesian Phylogenetic Inference

X. Meyer^{1,*}

¹*Department of Integrative Biology, University of California, Berkeley, California 94720, U.S.A.*

**E-mail: xavier.meyer@berkeley.edu*

Abstract.—Bayesian inference of phylogenies with MCMC is without a doubt a staple in the study of evolution. Yet, this method still suffers from a practical challenge identified more than two decades ago: designing tree topology proposals that efficiently sample the tree space. In this article, I introduce the concept of tree topology proposals that adapt to the posterior distribution as it is estimated. I use this concept to elaborate two adaptive variants of existing proposals and an adaptive proposal based on a novel design philosophy in which the structure of the proposal is informed by the posterior distribution of trees. I investigate the performance of these proposals by first presenting a metric that captures the performance of each proposals within a mixture. Using this metric, I then compare the adaptive proposals performance to the performance of standard and parsimony-guided proposals on 11 empirical datasets. Using adaptive proposals led to consistent performance gains and resulted in up to 18-fold increases in mixing efficiency and 6-fold increases in converge rate without increasing the computational cost of these analyses. [Bayesian inference; Adaptive tree proposals; Markov chain Monte Carlo; phylogenetics; posterior probability distribution.]

INTRODUCTION

Studies relying on Bayesian inference of phylogenies are routinely conducted thanks to the numerous software packages designed for this purpose (Yang and Rannala 1997, 2012). These packages implement the Markov chain Monte Carlo algorithm (MCMC) to estimate the posterior distribution of parameters of a model capturing the evolutionary history (phylogeny) of taxa and their mode of evolution. Despite the pervasiveness of these analyses, estimating such posterior distributions remains a computational challenge whose complexity largely results from difficulties exploring and sampling the space of tree topologies.

The challenge of exploring the tree space has been recognized since the earliest days of Bayesian phylogenetic inference (Huelsenbeck et al. 2001). Long analyses and failure to explore the region of high posterior probability was shown to be a common occurrence that increased in frequency with the number of taxa studied (Beiko et al. 2006). Large datasets (taxa-wise) frequently resulted in rugged posterior distributions where clusters of tree topologies with high posterior probabilities were separated by low-probability valleys. A decade ago, the use of Metropolis-coupled MCMC algorithm (MC³; Altekar et al. 2004) was proposed as a solution to this major issue and, although this solution is now considered as a standard practice, the settings re-

quired for this method to perform appropriately remains a practical concern (Whidden and Matsen 2015; Brown and Thomson 2018). Using the MC³ algorithm reduces the failure rate by easing the sampling of rugged tree posterior distribution, but is not significantly improving the sampling efficiency of the key actors of the exploration of the tree space: the tree proposals.

In the last decades, only a limited number of studies have considered the challenge of defining efficient tree proposals. A thorough analysis of standard tree proposals was conducted by Lakner et al. (2008) that provided insight on their performance. Following this study, the concept of guided tree proposals was presented by Höhna and Drummond (2012). This important contribution suggested using scores (e.g., conditional clade probability or posterior probability) to bias the selection of tree alterations toward the most promising among the set of moves proposed by a traditional tree proposal. However, the practicality of the resulting guided proposals remains limited due to the additional incurring computational burden. This concept was nonetheless implemented in MrBayes under the form of parsimony-guided proposals (Ronquist et al. 2012).

Building efficient proposals for continuous parameters has been the subject of numerous studies in the field of computational statistics (e.g., Gelman et al. 1996). These studies have led to the development of adaptive proposals that are self-tuned during an MCMC run to propose

moves tailored specifically for the posterior distribution currently estimated (Haario et al. 2001, 2005; Roberts and Rosenthal 2009). The field of computational phylogenetics has employed these approaches to improve the sampling efficiency of continuous parameters by designing novel adaptive proposals (Thawornwattana et al. 2017), developing multivariate proposals that exploit the correlation existing between parameters (Baele et al. 2017; Meyer et al. 2017), or by considering the estimation of distributions approximating the probability distributions of specific parameters (i.e., branch lengths) to independently generate new parameters values (Aberer et al. 2015; Claywell et al. 2018). Most of the software for Bayesian inference of phylogeny take advantage of these methods for continuous parameters (e.g., Ronquist et al. 2012; Aberer et al. 2014; Höhna et al. 2016; Baele et al. 2017). However, as no theory is readily available from the field of computational statistics regarding the sampling of tree topologies, none of these implementations use adaptive proposals for tree topologies.

Software for Bayesian inference of phylogenies continue to mostly rely on tree proposals that naively explore the posterior distribution, as other alternatives are computationally expensive or impracticable. The performance of these proposals hinders our ability to infer large phylogenies and to consider more complex and realistic evolutionary models. In this study, I present the theoretic foundations for the development of adaptive tree proposals and use them to develop three adaptive proposals: two adaptive variants of commonly used proposals and a fully-adaptive proposal based on a novel design philosophy. I investigate the computational complexity of these proposals and define a practical performance metric that enables us to assess the efficiency of each proposal within a mixture. Using this metric, the practical performance of these proposals is then studied on simulated and empirical datasets and compared to the performance of traditional and parsimony-guided tree proposals.

MATERIALS AND METHODS

Phylogenetic Tree Proposals

I consider the problem of developing efficient proposal kernels for unrooted tree topologies to conduct Bayesian inference of phylogeny. This type of analysis requires the estimation of the posterior probability distribution:

$$p(\theta, \mathbf{v}, \tau | X) = \frac{p(\theta)p(\mathbf{v})p(\tau)p(X | \theta, \mathbf{v}, \tau)}{\sum_{\tau} \int_{\mathbf{v}} \int_{\theta} p(\theta)p(\mathbf{v})p(\tau)p(X | \theta, \mathbf{v}, \tau)d\theta d\mathbf{v}'} \quad (1)$$

with θ being the parameters of the evolutionary model, τ the unrooted tree topology, $\mathbf{v}$ the branch lengths and X the alignment. Generally, the posterior distribution is estimated using the Metropolis-Hastings algorithm in

which new parameters values are generated by a proposal kernel and accepted with probability

$$\alpha \left(\{\theta, \mathbf{v}, \tau\}, \{\theta', \mathbf{v}', \tau'\} \right) = \underbrace{\frac{p(\theta')p(\mathbf{v}')p(\tau')}{p(\theta)p(\mathbf{v})p(\tau)}}_{\text{Prior ratio}} \times \underbrace{\frac{p(X | \theta', \mathbf{v}', \tau')}{p(X | \theta, \mathbf{v}, \tau)}}_{\text{Likelihood ratio}} \times \underbrace{\frac{p(\theta, \mathbf{v}, \tau | \theta', \mathbf{v}', \tau')}{p(\theta', \mathbf{v}', \tau' | \theta, \mathbf{v}, \tau)}}_{\text{Hastings ratio}} \quad (2)$$

with $p(\theta', \mathbf{v}', \tau' | \theta, \mathbf{v}, \tau)$ defining the probability with which the kernel proposes parameters $(\theta', \mathbf{v}', \tau')$ given $(\theta, \mathbf{v}, \tau)$. In this study, I focus on proposal kernels modifying uniquely the tree topology (i.e., $\theta = \theta'$ and $\mathbf{v} = \mathbf{v}'$) and on deriving their Hastings ratios having the form $p(\tau | \tau') / p(\tau' | \tau)$.

Common proposals employed for the inference of unrooted tree topologies are the stochastic Nearest Neighbor Interchange (stNNI), extending Subtree Pruning and Regrafting (eSPR; Swofford et al. 1996) and extending Tree Bisection and Reconnection (eTBR; Huelsenbeck et al. 2008). These proposals naively explore the space of tree topology by arbitrarily altering the current tree τ using subtree swapping or pruning operations. For instance, the stNNI proposal interchanges two subtrees separated by an internal branch, while the eSPR proposal prunes a subtree, moves it along a contiguous set of branches (a path) and finally regrafts it on the last branch of the path. Choices of branches, subtrees or paths are made randomly during these proposals and therefore frequently result in tree alterations that may be detrimental (e.g., removing a branch strongly supported by the data).

To improve the quality of the generated moves, adaptive proposals for continuous parameters use summary statistics of the posterior distribution, learned during an MCMC run, to tune the proposal mechanism (Roberts and Rosenthal 2009). Using these summary statistics, parameters of a proposal kernel (e.g., the scale of the random-walk) are adapted to target an optimal acceptance rate. While the specifics of such adaptive proposals are not directly applicable to tree topology proposals, the concept of using summary statistics of the posterior distribution can still be exploited to our advantage. In this study, I consider using the estimated marginal posterior probabilities of splits, or split frequencies, to construct adaptive tree proposals. This strategy is based on two components: the estimation of the split frequencies and the design of adaptive proposals exploiting these estimates.

Split Frequencies

Each branch of a phylogenetic tree represents a unique bipartition of the set of taxa in the alignment. These

bipartitions, better known as splits, are a useful tool to summarize the posterior distribution of trees. Using samples collected during an MCMC run, the marginal posterior probability of a split can be estimated by observing the frequency with which a given split occurs within the sampled tree topologies. Split frequencies therefore provide inexpensive estimates of the support for each specific bipartition but fail to fully summarize the tree distribution by disregarding the joint distribution of splits.

Adaptive proposals require the split frequencies to be learned and made available during MCMC runs. This procedure of estimating split frequencies during a MCMC run is commonly conducted to diagnose the convergence of a set of MCMC runs (Ronquist et al. 2012). As in the post-MCMC estimation of the marginal split frequencies, this procedure is conducted by counting the occurrence of splits within the sampled tree topologies and normalizing them by the number of observed samples. This procedure guarantees that estimates of the split frequencies converge to the true posterior distribution when the MCMC algorithm is run for an infinite amount of time. However, in practice, phylogenetic inferences are not run long enough to ensure the robustness and accuracy of the estimated split frequencies. To tackle this problem, I develop a heuristic algorithm to learn the split frequencies while overcoming several potential issues (SI, Learning split frequency).

The first issue results from the bias induced by the starting parameters of the MCMC. The earliest phase of an MCMC run generally samples trees unrepresentative of the high probability region of the posterior distribution until equilibrium is reached. Therefore, in a post-processing context, split frequencies are evaluated using samples remaining after the removal of the samples collected during the burnin phase. The duration of the burnin phase is however undetermined during an MCMC run and cannot be removed leading the split frequencies to be biased by these early samples. The second issue results from the high volatility and oscillatory behavior of the estimated split frequency during the earliest phase, even when the equilibrium is reached. This oscillatory behavior depends on whether a split is present or absent in the sampled tree topology. Adaptive proposals constructed with these fluctuating estimates could induce unwanted dependencies between the proposal probabilities and the presence or absence of a split (e.g. non-reversibility of the proposals), and impact the correctness of the MCMC algorithm.

For these reasons, the heuristic learning algorithm averages the split frequencies over a fixed number of samples to reduce the volatility of the estimates, uses a relaxation mechanism that progressively reduces the impact of the earliest samples observed, and detects the convergence of the learning process when the amount of vari-

ation in split frequencies stabilizes. Upon convergence of the estimated split frequencies, the learning process is terminated to ensure that the proposals preserve the ergodicity of the MCMC algorithm.

Adaptive proposals strongly rely on the estimated split frequencies and, while this heuristic algorithm improves the robustness and accuracy of these estimates, it does not ensure a foolproof estimation procedure. Therefore, to further improve the robustness of adaptive proposals, these proposal mechanisms include a stochastic component ϵ to ensure that all trees remain accessible regardless of the accuracy or correctness of the split frequency estimates (SI, Stochastic component).

Adaptive Tree Proposals

Adaptive tree proposals rely on split frequencies to define regions of the tree that are weakly or strongly supported (i.e., having low or high split frequencies). These regions are used to define moves maintaining highly supported regions of the topology while proposing modifications to regions having weak support. For instance, applying this concept to the stNNI proposal could reduce the frequency of subtree swaps removing a split with strong support while increasing the frequency of swaps acting on a split with weak support.

While this concept can be applied to build adaptive versions of naive proposals (e.g., stNNI or eSPR), two limitations to this approach must be accounted for. First, an adaptive version of a proposal always has a larger computational overhead compared to its naive version. To be efficient, an adaptive version must therefore remain computationally inexpensive with respect to the likelihood evaluation induced by a move.

The second key limitation results from a more conceptual consideration: the structure of moves generated by a tree proposal. Naive tree proposals produce moves with a specific structure. For instance, the stNNI proposal interchanges two subtrees separated by a branch while the eTBR proposal generates moves that remove a branch (bisection) and consider a different re-connection of the two resulting subtrees. This design philosophy was fundamental to broaden the range of strategies used to explore tree distributions because each dataset benefits differently from each specific structure of move (Lakner et al. 2008). Ironically, imposing specific structure to moves may prove a hindrance for adaptive tree proposals for the exact same reasons: specific structures of moves do not benefit all datasets.

I propose here a different design philosophy where adaptive tree proposals generate moves having an *adaptive* structure. The information contained in split frequencies represents more than a guiding mechanism for naive proposals. It identifies weakly and strongly supported regions of a tree and their respective struc-

Table 1: Mathematical notation.

Variables	Interpretation
ϵ	Stochastic component
$\tau = (V, E)$	Tree topology with edges $e_i \in E$ and vertices $v_j \in V$
s_j	A unique split identified by j in the set of all possible splits
$s_j = S(e_i)$	Operator returning the split identified by edge e_k
$\pi(s_j) = \pi(S(e_i)) = \pi(e_i)$	Operator returning the split frequency of s_j
$\pi^c(s_j) = 1 - \pi(s_j)$	Complement of the split frequency
$e_i = \{v_j, v_k\} = \{v_k, v_j\}$	Undirected edges
$\vec{e}_i = \{v_j, v_k\} \neq \vec{e}_i = \{v_k, v_j\}$	Directed edges
$\nu(\vec{e}_i) = \nu(\{v_j, v_k\})$	Internal edges contiguous to $\vec{e}_i$ (neighboring edges)
$I(E)$	Set of non-terminal edges
$\rho = \{\vec{e}_{x_1}, \vec{e}_{x_2}, \dots, \vec{e}_{x_m}\}$	Contiguous path in the tree
$\text{mcf}(\vec{e}_i)$	Min. split frequency in clade identified by edge $\vec{e}_i$

ture. These structures provide insight on the structure of moves (e.g., subtrees swap, or pruning and regrafting) that would be appropriate to alter a tree topology. By using the split frequencies to define the structure of moves, adaptive tree proposals have the potential to propose tree alterations specifically tailored to fit the posterior distribution sampled.

In this study, we consider both design philosophies of designing adaptive proposals: proposals producing specific type of moves and proposals using adaptive structure of moves. I first define two adaptive variants of existing naive proposals (stNNI and eSPR) that produce specific type of moves. Then, I present a novel proposal that uses the split frequencies to define moves having an appropriate structure.

Mathematical notation.—I use the following mathematical notation to describe adaptive proposals (summarized in Table 1): a tree topology τ is defined by a set of vertices, V , and a set of edges, or branches, E . The set of edges, $I(E)$, identifies the set of internal edges. Each edge e_i identifies a split $s_j = S(e_i)$ whose frequency is estimated by the function $\pi(s_j)$. A split s_j identifies a unique bipartition of the set of taxa and can therefore be identified differently in several tree topologies. For instance, two edges in different trees (e.g., e_i in τ and e_j in τ') can identify the same split s_j .

I use a flexible definition of splits in the sense that a split can be used to identify a bipartition and also to build new partitions. For instance, if a directed edge $\vec{e}_i$ is considered in the context of a move, then the split $s_j = S(\vec{e}_i)$ identifies the taxa in the clade subtended by edge $\vec{e}_i$. Using this definition, a new split can then be constructed when a clade is moved by considering the union of two splits: for instance, $S(\vec{e}_i) \cup S(\vec{e}_j)$ would identify a bipartition segregating the taxa in the clade subtending edges $\vec{e}_i$ and $\vec{e}_j$ from all the other taxa.

Figure 1: Steps of the A-stNNI proposal. In the first step edge e_r is selected according to Eq. (3). Then, the two subtrees to swap are selected according to Eq. (4).

Edges contiguous to edge $\vec{e}_k$ are identified by the operator $\nu(\vec{e}_k)$ that returns the next edges according to the direction of $\vec{e}_k$. As most of the adaptive proposals considered act on internal edges, the $\nu(\cdot)$ operator only returns edges contained in $I(E)$. Regions of the tree topology are identified by contiguous paths ρ composed of a set of contiguous undirected or directed edges (e.g., $\rho_x = \{\vec{e}_{x_1}, \vec{e}_{x_2}, \dots, \vec{e}_{x_m}\}$). Lastly, the operator $\text{mcf}(\vec{e}_k)$ returns the smallest split frequency of the internal edges existing in the clade subtended by edge $\vec{e}_k$.

Adaptive stNNI

The adaptive stNNI (A-stNNI) proposal uses the split frequencies to guide the selection of the central edge e_r (Fig. 1) whose split $s_r = S(e_r)$ will be altered by the interchange of two subtrees located on each extremity of e_r . The choice of the interchange to apply, among the two possible outcomes, is also guided by the split frequencies.

The selection of the central edge e_r is biased toward edges identifying weakly supported splits. The central

Figure 2: Steps of the A-2SPR proposal. In the first step the edges (e_r, e_s) , along which a subtree will be moved, are selected according to Eq. (7). Then, a move among the 8 possible pruning and regrafting locations along edges (e_r, e_s) is selected according to Eq. (8).

edge e_r is therefore selected with probability

$$p(e_r | \tau) = \frac{\epsilon + \pi^c(e_r)}{\sum_{e_i \in I(E)} \epsilon + \pi^c(e_i)}. \quad (3)$$

The new split s_u replacing s_r is determined by one of the two possible outcomes of subtrees interchange. The edge $\bar{e}_x$ subtending the first subtree to swap is arbitrarily chosen among the four edges contiguous to edge e_r (i.e., with probability $p(e_x | e_r, \tau) = 1/4$). The second subtree is selected among the two subtree on the opposite side of e_r that are identified by edges e_c and e_d , respectively (Fig. 1). The new split s_u segregates the taxa identified by the edge e_x and either edge e_c or e_d from the others, resulting in split $S(e_x) \cup S(e_c)$ or $S(e_x) \cup S(e_d)$ respectively. To favor the interchange leading to the tree with the strongest support, the edge e_y identifying the second subtree is therefore selected with probability proportional to $\pi(s_u)$:

$$p(e_y | e_r, e_x, \tau) = \frac{\epsilon + \pi(S(e_x) \cup S(e_y))}{\sum_{e_i \in \{e_c, e_d\}} \epsilon + \pi(S(e_x) \cup S(e_i))} \quad \text{with } e_y \in \{e_c, e_d\}. \quad (4)$$

An A-stNNI move is identified by the triplet of edges (e_r, e_x, e_y) and is proposed according to probability

$$p(\tau' | \tau) = p(e_r | \tau) \times p(e_x | e_r, \tau) \times p(e_y | e_r, e_x, \tau), \quad (5)$$

and leads to the new tree topology τ' . The reverse move happens with probability

$$p(\tau | \tau') = p(e'_r | \tau') \times p(e_x | e'_r, \tau') \times p(e_y | e'_r, e_x, \tau') \quad (6)$$

with e'_r identifying the edge of the new split s_u (i.e., central edge) as in τ , and edges e_x and e_y identifies the same splits in τ and τ' . The Hastings ratio required to evaluate the acceptance probability of a A-stNNI move (Eq. (2)) is the ratio of Eqs. (6) and (5).

Adaptive 2-edges SPR

The adaptive 2-edges SPR (A-2SPR) is an adaptive version of the eSPR move. Similarly to the eSPR proposal, this adaptive proposal prunes a subtree, moves it along a path made of consecutive edges, and regrafts it (Fig. 2). The length of the path is however limited to exactly two edges, resulting in the alteration of two splits exactly. Conversely to the usual eSPR strategy, the A-2SPR first selects the path along which the subtree will be moved and then consider all the possible pruning and regrafting moves along this path. The A-2SPR can be seen as a natural extension of the A-stNNI to two edges since the path is selected to target region of the tree with weak support, while the move is selected to favor the resulting tree having the strongest support.

In the first step, I select a pair of contiguous edges (e_r, e_s) with probability inversely proportional to the product of their estimated marginal split frequencies, as defined by,

$$p(e_r, e_s | \tau) = \frac{\epsilon + \pi^c(e_r) \times \pi^c(e_s)}{\sum_{(e_i, e_j) \in I_2(E)} \epsilon + \pi^c(e_i) \times \pi^c(e_j)}, \quad (7)$$

where $I_2(E)$ identifies the set containing all pairs of contiguous internal edges in τ . The pair (e_r, e_s) has the four edges e_a, e_b, e_c and e_d at its extremities and the edge e_t in its center, which is the edge sharing a vertex with both edges e_r and e_s .

The second step enumerates all the possible moves across edges (e_r, e_s) for subtrees subtended by edges e_a, e_b, e_c and e_d . Each of those edges can be regrafted two ways after moving along edges (e_r, e_s) . For instance, assuming that e_a and e_b are adjacent, the subtree identified by edge e_a could be pruned and then regrafted on edges e_c or e_d . Assuming that edge e_c is selected as the regrafting point, the split $s_r = S(e_r)$ and $s_s = S(e_s)$ would be removed and would be replaced by splits s_p and s_q . The first split would separate taxa in subtrees identified by edges e_b and e_t from the others (i.e., $s_p = S(e_b) \cup S(e_t) = S(e_a) \cup S(e_c) \cup S(e_d)$), while the second would separate

Figure 3: Example of an A-PBJ proposal. The two eSPR-type moves defined by the stable path ρ_s (Fig. 4) and unstable paths $\rho_u^{(u)}$ and $\rho_u^{(v)}$ (Fig. 5) are applied on the example tree τ , resulting in tree τ' . The probability of an A-PBJ move is the joint probability of building the three paths (Eqs. (11) and (12)).

taxa in the clade identified by edges e_a and e_c from the others (i.e., $s_q = S(e_a) \cup S(e_c)$).

For mathematical convenience, edges e_a, e_b, e_c and e_d can be relabeled using their relative position using notation e_{p_1, p_2} (Fig. 2). Indexes $p_i \in \{r, l\}$ defines whether the edges are at the right ($p_1 = r$) or left ($p_1 = l$) extremity of edges (e_r, e_s) and whether the edges are the right or left one relatively to each other (p_2). Using this notation for the edge $e_{x,y}$ identifying the subtree to move at extremity x of edges (e_r, e_s) and the edge $e_{\bar{x},z}$ identifying the regrafting point at the opposite extremity $\bar{x}$ (e.g., $x = r \rightarrow \bar{x} = l$), the probability of selecting a move that favors the strongest supported tree is defined as

$$p(e_{x,y}, e_{\bar{x},z} \mid e_r, e_s, \tau) = \frac{\epsilon + \pi \left(S_1(x, y, z) \right) \times \pi \left(S_2(x, y, z) \right)}{\sum_{i,j,k \in \{l,r\}^3} \epsilon + \pi \left(S_1(i, j, k) \right) \times \pi \left(S_2(i, j, k) \right)}, \quad (8)$$

with

$$S_1(u, v, w) = S(e_{u,v}) \cup S(e_{\bar{u},w}) \cup S(e_{\bar{u},\bar{w}}),$$

and

$$S_2(u, v, w) = S(e_{u,v}) \cup S(e_{\bar{u},w}).$$

The move identified by the quadruplet of edges $(e_r, e_s, e_{x,y}, e_{\bar{x},z})$ is proposed according to probability,

$$p(\tau' \mid \tau) = p(e_r, e_s \mid \tau) \times p(e_{x,y}, e_{\bar{x},z} \mid e_r, e_s, \tau) \quad (9)$$

and leads to the new tree topology τ' . The reverse moves happens with probability,

$$p(\tau \mid \tau') = p(e'_r, e'_s \mid \tau) \times p(e_{x,y}, e_{\bar{x},z} \mid e'_r, e'_s, \tau) \quad (10)$$

where edges e'_r and e'_s identify the new splits. Edge $e_{x,y}$ identifies the edge originally pruned and $e_{\bar{x},z}$ its neighbor in τ that subtend to the same subtrees in τ and τ' . The ratio of Eqs. (10) and (9) defines the Hastings ratio for the

A-2SPR proposal that is used to evaluate the acceptance probability of a move (Eq. (2)).

This strategy can be generalized to build adaptive N-edges SPR moves. However, two pitfalls are inherent to this approach. First, using a proposal affecting a fixed number of edges is inconvenient and can result in non-ergodic proposals when used on their own. Second, the proposals efficiency would suffer from the increasing computational cost. The computational complexity of the enumeration of all N -edges paths grows as $O(2^N)$, while the one of estimating the move probabilities grows as $O(N)$ (SI, Computational complexity).

Adaptive Path Building and Jolting Proposal

While the A-stNNI and A-2SPR proposals are based on a design philosophy targeting the construction of specific type of moves, the adaptive path building and jolting proposal (A-PBJ) embraces the design philosophy of using the split frequencies to adaptively define the proposals structure. This approach is achieved by using the split frequencies to identify two types of structures within a tree topology: weakly and strongly supported regions. A proposal designed under this philosophy should strive to maintain the regions with strong support while proposing alteration to ones with weak support. Moves resulting from this strategy are specifically tailored to fit the posterior distribution of trees at hand.

The A-PBJ proposal implements this novel design philosophy by using contiguous paths within a tree to identify weakly and strongly supported regions: the unstable and stable paths, respectively. This proposal begins by selecting a stable path acting as the backbone of the move. Unstable paths are then constructed at both extremities of the backbone and are subject to eSPR-type moves. In two separate steps, each edge at the extremities of the stable path are pruned, moved along their relative unstable path and then regrafted (Fig. 3). The

Figure 4: Construction of a stable path ρ_s . In the first step the edge e_s is selected according to Eq. (13). Then, the path is extended on both side of edge e_s according to Eqs. (14-16). Finally, both extensions are concatenated forming path ρ_s (Eqs. 17-19)).

stable path and its splits remain unaltered by this move, while the splits of both unstable paths are replaced by new ones. Depending on the paths identified, moves produced by the A-PBJ proposal include (but not exclusively) moves produced by the stNNI, eSPR and eTBR proposals.

The path building strategies are key to the efficiency and reliability of the A-PBJ proposal. The stable paths must capture sets of splits having high frequencies that would benefit from alterations at their extremities; constructing stable paths starting and ending at terminal nodes would not enable any moves. The unstable paths must capture set of low-frequencies split of variable size and, in this sense, act as a generalization of the mechanisms previously employed in the A-stNNI and A-2SPR. This generalization must however avoid the expensive enumeration of all possible N-edges paths to remain computationally competitive.

The strategies used to construct such paths and their resulting probabilities are defined in the following sections. The construction of a stable path ρ_s with probability $p(\rho_s | \tau)$ is illustrated in Fig. 4. The unstable paths $\rho_u^{(u)}$ and $\rho_u^{(v)}$ are built at each extremities of the stable path ρ_s identified by edges $\tilde{e}_l$ and $\tilde{e}_r$. The constructions of those unstable paths have probabilities $p(\rho_u^{(u)} | \rho_s, \tau)$ and $p(\rho_u^{(v)} | \rho_s, \tau)$, respectively, and are illustrated in Fig. 5.

The probability of an A-PBJ move is defined using the joint probability of building its composing paths and is given as

$$p(\tau' | \tau) = p(\rho_s | \tau) \times p(\rho_u^{(u)} | \tilde{e}_l, \tau) \times p(\rho_u^{(v)} | \tilde{e}_r, \tau). \quad (11)$$

The reverse move happens with probability

$$p(\tau | \tau') = p(\rho'_s | \tau') \times p(\rho_u^{(a)} | \tilde{e}'_l, \tau') \times p(\rho_u^{(b)} | \tilde{e}'_r, \tau'), \quad (12)$$

where ρ'_s identifies the path including edges identifying the same splits as ρ_s . Paths $\rho_u^{(a)}$ and $\rho_u^{(b)}$ are inverse versions of path $\rho_u^{(u)}$ and $\rho_u^{(v)}$, respectively, that identify the reverse eSPR-type moves.

The ratio of Eqs. (12) and (11) defines the Hastings ratio for the A-PBJ proposal required to evaluate the acceptance proposal of a move (Eq. (2)). The different terms involved in Eqs. (11) and (12) are detailed in the next sections.

Stable path.—The construction of a stable path ρ_s consists first in the selection of an edge with probability

$$p(e_s | \tau) = \frac{\epsilon + \pi(e_s)}{\sum_{e_i \in I(E)} \epsilon + \pi(e_i)}, \quad (13)$$

that favors the selection of internal edges identifying high-frequency splits. The path ρ_s is then built by step-wise extension of its extremities. Starting from edge e_s the path is extended in both directions, namely the $\tilde{e}_s$ and $\tilde{e}_s$ directions, respectively (Fig. 4). After including edge e_s in a partial path (e.g., $\rho_s^{(r)}$), the extension mechanism iterates over two steps: first, the termination condition of the extension, and second (if not terminated), the extension continues by choosing the new edge to add to the path.

Assuming an initial direction $\tilde{e}_s$ and the initial partial path $\rho_s^{(r)} = (\tilde{e}_s = \tilde{e}_{r1})$, the first iteration ($i = 1$) begins by controlling the termination condition that occurs with probability

$$p(\text{stop} | \tilde{e}_{r1}, \tau) = (\epsilon + \min_{e_j \in v(\tilde{e}_{r1})} \pi^c(e_j)) \times (\epsilon + 1)^{-1}. \quad (14)$$

This probability favors the termination of the extension mechanism whenever a split identified by a neighboring internal edge $e_j \in v(\tilde{e}_{r1})$ could benefit from being altered.

If the termination condition is not met (which happens with probability $1 - p(\text{stop} | \tilde{e}_{r1}, \tau)$), the path continues its extension by selecting the next edge to add, if any. Non-terminal edges identifying a split with high frequency and leading to a clade containing low-frequency structures represent good candidates for extension and are se-

Figure 5: Construction of the unstable paths $\rho_u^{(u)}$ and $\rho_u^{(v)}$ at the extremities of ρ_s (Fig. 4). In the two separate building phase (1 and 2), the paths are constructed by extension (1a and 2a) according to Eqs. ((20)-(22)). Then, the last edges identifying the re-graft points (1b and 2b) are selected according to Eq. (23). The probability of building paths $\rho_u^{(u)}$ and $\rho_u^{(v)}$ is the joint probability of the extensions and re-graft selections (Eq. (24)).

lected with probability

$$p(\vec{e}_{r_{i+1}} | \vec{e}_{r_i}, \tau) = \frac{\epsilon + \pi(\vec{e}_{r_{i+1}}) \times (1 - MC^3 F(\vec{e}_{r_{i+1}}))}{\sum_{\vec{e}_k \in v(\vec{e}_{r_i})} \epsilon + \pi(\vec{e}_k) \times (1 - MC^3 F(\vec{e}_k))}. \quad (15)$$

The selected edge $\vec{e}_{r_{i+1}}$ is then added to the path (i.e., $\rho_s^{(r)} = (\vec{e}_{r_1}, \vec{e}_{r_2})$) and a new iteration begins ($i = i + 1$).

This process continues until the termination event occurs or until the path reaches an endpoint (i.e., $v(\vec{e}_{r_i}) = \emptyset$). The probability of having extended the partial stable path $\rho_s^{(r)} = (\vec{e}_{r_1}, \dots, \vec{e}_{r_i}, \dots, \vec{e}_{r_m})$ in direction $\vec{e}_s$ is then given as

$$p(\rho_s^{(r)} | \tau, \vec{e}_s) = \prod_{i=1}^{m-1} \left[\left(1 - p(\text{stop} | \tau, \vec{e}_{r_i}) \right) \times p(\vec{e}_{r_{i+1}} | \tau, \vec{e}_{r_i}) \right] \times p(\text{stop} | \tau, \vec{e}_{r_m}). \quad (16)$$

The extension mechanism is repeated in the opposite direction (i.e., $\vec{e}_s$) resulting in partial stable path $\rho_s^{(l)} = (\vec{e}_{l_1}, \dots, \vec{e}_{l_j}, \dots, \vec{e}_{l_k})$. Both partial paths $\rho_s^{(r)}$ and $\rho_s^{(l)}$ are then concatenated to form the stable path

$$\rho_s = (\vec{e}_{l_k}, \dots, \vec{e}_{l_j}, \dots, \vec{e}_{l_1} = e_s = \vec{e}_{r_1}, \dots, \vec{e}_{r_i}, \dots, \vec{e}_{r_m}). \quad (17)$$

The construction of this path is conditional on the selection of edge e_s as starting point and is built with probability

$$p(\rho_s | e_s, \tau) = p(e_s | \tau) \times p(\rho_s^{(r)} | \vec{e}_s, \tau) \times p(\rho_s^{(l)} | \vec{e}_s, \tau). \quad (18)$$

Given that this exact path may be built starting from any edge $e_i \in \rho_s$, the probability of building path ρ_s must

be marginalized over all potential starting edge and is defined as

$$p(\rho_s | \tau) = \sum_{e_i \in \rho_s} p(\rho_s | e_i, \tau). \quad (19)$$

Unstable paths.—The mechanism used to build unstable paths consists of extension phases starting from each of the extremities of the stable path ρ_s , each identified by edges $\vec{e}_{l_k}$ and $\vec{e}_{r_m}$, respectively. An unstable path fully identifies an eSPR-type move (Fig. 5). For instance, the unstable path $\rho_u^{(v)} = (\vec{e}_{v_1}, \dots, \vec{e}_{v_i}, \dots, \vec{e}_{v_m})$ starting from edge $\vec{e}_{v_1} = \vec{e}_{r_m}$ includes the edge to prune ($\vec{e}_{v_1}$) that identifies the moving clade C, the edge that identifies the direction of the move ($\vec{e}_{v_2}$), the edges traversed by clade C and the regrafting point of clade C ($\vec{e}_{v_m}$). As in the extension of the stable path, an unstable path $\rho_u^{(v)}$ is built by iterating over two steps: the extension termination and edge selection steps.

The extension phase is preferably terminated when the unstable path would risk begin extended by an edge identifying a split with a high-frequency in the next step. The termination event occurs then with probability

$$p(\text{stop} | \vec{e}_{v_i}, \tau) = \frac{1}{\epsilon + 1} \times \left(\epsilon + \sum_{\vec{e}_k \in v(\vec{e}_{v_i})} p(\vec{e}_k | \vec{e}_{v_i}, \tau) \times \pi(\vec{e}_k) \right), \quad (20)$$

where $p(\vec{e}_k | \tau, \vec{e}_{v_i})$ identifies the probability of extending path $\rho_u^{(v)}$ with edge $\vec{e}_k$ (Eqs. (21-23)).

If the termination does not occur, the next edge is selected according to probabilities defined by its role in the eSPR move (i.e., direction, traversed or regrafting edge). The first edge selected identifies the direction

along which clade C will move and is selected with probability

$$p(\vec{e}_{v_{i+1}} | \vec{e}_{v_i}, \tau) = \frac{\epsilon + \pi^c(\vec{e}_{v_{i+1}})}{\sum_{\vec{e}_k \in \nu(\vec{e}_{v_i})} \epsilon + \pi^c(\vec{e}_k)}, \quad i = 1, \quad (21)$$

that favors the removal of edges identifying low-frequency splits.

Each edge traversed by clade C is then selected with probability

$$p(\vec{e}_{v_{i+1}} | \vec{e}_{v_i}, \tau) = \frac{\epsilon + \pi^c(\vec{e}_{v_{i+1}}) \times \pi(s_C \cup S(\vec{e}_{v_{i+1}}))}{\sum_{\vec{e}_k \in \nu(\vec{e}_{v_i})} \epsilon + \pi^c(\vec{e}_k) \times \pi(s_C \cup S(\vec{e}_k))}, \quad 1 < i < m - 1, \quad (22)$$

where s_C identifies the split containing the taxa of clade C . This probability accounts for the removal of the current split $S(\vec{e}_{v_{i+1}})$ and the addition of the new split $(s_C \cup S(\vec{e}_{v_{i+1}}))$.

The last edge, selected after the termination of the extension phase, identifies the regrafting edge for clade C and is selected with probability

$$p(\vec{e}_{v_{i+1}} | \vec{e}_{v_i}, \tau) = \frac{\epsilon + \pi(s_C \cup S(\vec{e}_{v_{i+1}}))}{\sum_{\vec{e}_k \in \nu(\vec{e}_{v_i})} \epsilon + \pi(s_C \cup S(\vec{e}_k))}, \quad i = m - 1, \quad (23)$$

that is, proportionally to the frequency of the last split added.

Building an unstable path $\rho_u^{(v)}$ is the outcome of the selection of the direction, the extension of the traversed path and the choice of the regrafting edge. The probability of building an unstable path is therefore defined as,

$$p(\rho_u^{(v)} | \vec{e}_{v_1}, \tau) = p(\vec{e}_{v_2} | \tau, \vec{e}_{v_1}) \times \prod_{i=2}^{m-2} \left[\left(1 - p(\text{stop} | \tau, \vec{e}_{v_i}) \right) \times p(\vec{e}_{v_{i+1}} | \tau, \vec{e}_{v_i}) \right] \times p(\text{stop} | \tau, \vec{e}_{v_{m-1}}) \times p(\vec{e}_{v_m} | \tau, \vec{e}_{v_{m-1}}). \quad (24)$$

The probability of an A-PBJ move (Eqs (11) and (11)) is defined as the joint probability of building a stable path (Eq. 19) and the two unstable paths starting at its extremities (Eq. (24)).

Parsimony-Guided stNNI and eSPR

I implemented Parsimony-guided stNNI and eSPR proposals based on the concepts presented in (Höhna and Drummond 2012) to compare adaptive proposals with other strategies for guiding tree proposals (SI, Parsimony

score transformation). Two different strategies were considered: exhaustive guided stNNI (G-stNNI) and guided N-edges eSPR (G-NSPR) proposals. The mechanism of these proposals consists of defining a set of potential moves and drawing one of them proportionally to the parsimony score of the resulting trees. Each proposal differs in the strategy used to build the set of moves. The G-stNNI proposal enumerates all possible stNNI moves for the current tree τ , while the G-NSPR proposal randomly choose a subtree to prune, then enumerates all eSPR moves altering at most N -edges. Using $N = 1$ has a similar effect to a guided stNNI that would randomly choose the central edge and then use the parsimony score to guide the subtree interchange.

Theoretical Computational Complexity of Proposals

I defined the theoretical computational complexity of the different proposals and compared them to the cost of a partial likelihood (SI, Computational complexity). These complexities and the condition under which a proposal is computationally efficient with respect to a partial likelihood evaluation are summarized in Table 2. This theoretical analysis indicates that parsimony-guided tree proposals have performance improvements limited by their dependencies to the number of sites m . Conversely, adaptive tree proposals should have negligible computational cost as long as the number of taxa n is smaller than m .

The parsimony-guided proposal G-stNNI could exceed the computational complexity of a partial likelihood evaluation under the condition that the number of taxa n exceed the product of the number of symbols c (e.g., $c = 4$ for nucleotides) and the number of rate categories k under a discrete-Gamma rate model (Yang 1994). Such scenarios would happen even when using moderately complex models as the GTR+ Γ substitution model with $k = 4$ rates categories. The G-NSPR proposal seems more competitive as its efficiency condition is reached when $2^s \ll ck$. Even if this condition is not reached, a G-NSPR can be executed at a fraction $2^s/(ck)$ of a partial likelihood evaluation, which represents a reasonable computational overhead for small values of s .

While the overhead cost of adaptive proposals is non-negligible compared to naive proposals, their computational overhead remains negligible with respect to partial likelihood evaluations as long as $2n \ll cm\hat{n}k^2$, where $\hat{n}$ identifies the number of edges included in a partial likelihood evaluation. This efficiency condition is met even for the simple models (e.g., nucleotide substitutions model without rate heterogeneity, $c = 1, k = 4$) as long as $n \leq m$ (which is a prerequisite to obtain accurate inferences of phylogeny).

Table 2: Computational complexity of the proposals and likelihood operations. Alignments have m sites for n sequences defined over an alphabet of k characters (e.g., $k = 4$ for nucleotide sequences). The Gamma model for rate heterogeneity has a number c of rate categories (Yang 1994), while a partial likelihood requires $\hat{n} < n$ operations instead of n . Finally, s represents the number of splits altered by a move.

Operation	Complexity	Efficiency cond.
Full lik.	$O(cm\hat{n}k^2)$	-
Partial lik.	$O(cm\hat{n}k^2)$	-
Partial pars.	$O(m\hat{n}k)$	-
stNNI	$O(1)$	$1 \ll cm\hat{n}k^2$
eSPR	$O(s)$	$s \ll cm\hat{n}k^2$
eTBR	$O(2s)$	$2s \ll cm\hat{n}k^2$
G-stNNI	$O(m\hat{n}k)$	$n \ll ck$
G-NSPR	$O(2^s m\hat{n}k)$	$2^s \ll ck$
A-stNNI	$O(n)$	$n \ll cm\hat{n}k^2$
A-2SPR	$O(s + 2^s n)$	$s + 2^s n \ll cm\hat{n}k^2$
A-PBJ	$O(s_1 + s_2 + s_3^2 + n)$ $\approx O(2n)$	$2n \ll cm\hat{n}k^2$

Assessing the Performance of Tree Proposals

Diagnosing the behavior of an MCMC algorithm is generally achieved by monitoring its sampling performance (e.g., effective sample size) on different parameters. This task is particularly difficult when monitoring the performance of MCMC algorithms to estimate the posterior distribution of trees due to the discrete nature of this parameter. Nonetheless, two different characteristics are usually monitored. First, the time to convergence of the MCMC algorithm, that is the number of iterations required until the Markov chain reaches its equilibrium. Once at equilibrium, samples obtained from the Markov chain are representative of the true posterior distribution. The second characteristic is the mixing efficiency, that is the propensity of the MCMC algorithm to mimic the process of directly drawing samples from the true posterior distribution. By exploring the parameter space in a step-wise manner, random-walk MCMC algorithm have the tendency to generate autocorrelated samples that are more predictable than draws from the posterior distribution.

Few robust and practical procedures exist to measure these two characteristics when phylogenetic trees are involved and none are able to separately monitor the behavior of tree proposals within a mixture. In the next sections, I first enumerate the existing procedures and define how I used them, and then I present a novel metric that is able to isolate and assess the performance of tree proposals contained in a mixture of proposals.

Existing performance metrics.—The standard metric to assess the convergence of MCMC runs estimating the pos-

terior distribution of trees is the average standard deviation of split frequencies (ASDSF). This metric captures the variance of the posterior distribution of split frequencies among several independent MCMC runs. In previous studies, the efficiency of tree proposals was assessed using a convergence threshold based on the ASDSF (Lakner et al. 2008; Höhna and Drummond 2012). After a thorough investigation of this procedure, Whidden and Matsen (2015) suggested that the number of replicates plays a key role in the accuracy of this metric. Instead of the ASDSF, graph metrics such as the mean round trip cover time (MRT) were used to assess the behavior of the MCMC runs. Alas, these metrics could not be applied on diffuse tree distributions and required MCMC runs consisting of an enormous number of samples.

Using large amount of replicates or replicates having enormous numbers of samples to assess the behavior of MCMC runs is impractical when the behavior of many tree proposals are compared on several datasets. Therefore, I used multiple long and independent runs with MrBayes to obtain accurate estimates of the posterior distribution of split frequencies $p(s|X)$. These reference split frequencies were used to assess the convergence of each one of the runs under two scenarios: the overall convergence of the MCMC run (before burnin removal) and convergence after removal of the burnin phase. The first metric provides an indication on the number of iterations required to reach convergence, while the second indicates the efficiency of the proposals at estimating the split frequencies. I defined these convergence criteria as satisfied when the Euclidean distance between the reference $p(s|X)$ and the estimated $\hat{p}(s|X)$ marginal distribution of split frequencies, for all splits s_i with $p(s_i|X) > 0.1$, was lower than a fixed threshold. I defined these thresholds to represent an average error of 2% and 1% per split for the before and after-burnin removal convergence metrics, respectively.

Assessing the contribution of moves in a mixture of proposals.—The aforementioned convergence metrics capture the behavior of MCMC runs without providing information on the performance of the different tree proposals forming a mixture. With the exception of the acceptance rate of each tree proposal, no metrics are readily available to measure each the relative contribution of each proposal. Furthermore, while the acceptance rate provides some information on the proposal behavior, it fails to capture the amount of trees that the proposal can access or alter: a proposal that would efficiently visit only a small subset of the 95% highest posterior density (HPD) would have a high acceptance rate. Additionally, measuring the convergence metrics on finite MCMC runs using a single tree proposal at a time could result in failure

Figure 6: Example for the visit-cycle metrics. Panels a) displays the reference split frequencies. Figure b) shows the ideal C-V frequencies. Figure c) to e) represents the observed C-V frequencies for the stNNI, eSPR and A-stNNI, respectively. Figures f) and g) indicate the resulting C-V efficiency metric and C-V distance from the ideal proposal.

to reach convergence for some proposals. Failure to reach convergence is unrepresentative of the mixing efficiency of tree proposals and could therefore lead us to discard proposals that efficiently sample the posterior distribution of trees once convergence is reached.

I therefore considered an alternative approach to characterize the performance of each tree proposal in a mixture. This strategy considers as reference the expected behavior of an idealized proposal: the joint posterior distribution of parameters $\theta, \mathbf{v}$ and trees τ . Instead of basing these metrics on a hypothetical sequence of trees produced by this proposal, I investigate the use of summary statistics on the number of iterations between different visits of a given split. A cycle corresponding to the visit of a split begins with its appearance in the trees sampled by the MCMC process. It then includes the number of subsequent iterations when the split is still sampled, plus the number of iterations when the split is not sampled, until its first re-appearance. When trees are drawn directly from the posterior distribution, the frequency at which a split is visited depends entirely on its posterior probability and can be derived by considering the sum of two geometric random variables (Sen and Balakrishnan 1999). These two geometric distributions characterize the number of samples before the disappearance of the split and then the number of iterations before its reappearance in the trees sampled. The probability of the period k_i of a visit-cycle given the split posterior probability $p(s_i|X)$ is given as,

$$p(k_i | q = p(s_i | X), \bar{q} = 1 - p(s_i | X)) = q\bar{q}^{(k-1)} \times \frac{\bar{q}}{\bar{q} - q} \times \bar{q}q^{(k-1)} \times \frac{q}{q - \bar{q}}. \quad (25)$$

Assuming that accurate estimates of the split frequencies are available (e.g., reference runs), I can summarize

the characteristics of the ideal tree proposal with a split-wise approach by considering the expected period $E(k_i)$ of splits s_i (Fig. 6a and b). This reference behavior can then be compared to the one observed from adaptive proposals, or mixture of proposals, under the condition that their history is tracked by logging the iteration at which they are applied and the resulting effect on the tree sampled (i.e., move accepted or rejected). From this information, the observed average cycle-visit (C-V) period $\hat{E}(k_i)$ per proposal and per split can be summarized (Fig. 6c to e).

I use the C-V period for each split s_i with $p(s_i|X) > 0.1$, or more exactly their C-V frequency ($1/E(k_i)$), to assess the performance of each tree proposal applied within an MCMC run according to two metrics: their efficiency and split coverage. The C-V efficiency is estimated as the sum of the C-V frequencies over all splits and represents the average number of splits visited per moves (Fig. 6f). For proposals altering a single split (i.e., stNNI) this measure is equivalent to the acceptance rate. The second metric is estimated as the difference between the relative C-V frequency of a proposal and the one expected under the ideal proposal. This metric, the C-V, distance is defined as

$$d = \sqrt{\sum_{k_i \in S} \left(\frac{E[k_i]}{\sum_{k_j \in S} E[k_j]} \right)^{-1} - \left(\frac{\hat{E}[k_i]}{\sum_{k_j \in S} \hat{E}[k_j]} \right)^{-1}}, \quad (26)$$

where S identifies all cycle-period k_i of splits s_i having a posterior frequency $p(s_i|X) > 0.1$ (Fig. 6g). This distance represents the departure from the expected relative C-V frequencies: proposals visiting mostly a small subset of splits, for instance, would have a behavior distant to the one of the ideal proposal. To simplify the interpretation of the C-V distance in upcoming benchmarks, I will con-

sider the inverse of this metric (d^{-1}): the C-V accuracy. This transformed metric conveniently increases from 0 to 1 as the proposal behavior approaches the one of the ideal proposal.

RESULTS

This study is decomposed in two separate experiments. First, I assessed the performance of each proposal separately and validated the new efficiency metrics on datasets simulated with the INDELible software (Fletcher and Yang 2009). Trees were simulated under a birth-death model, while the alignments were simulated under the GTR model (SI, Dataset simulation settings). Then, I assessed the performance of a proposals mixture containing adaptive proposals by comparing it to the traditional mixture of naive proposals. This comparison was conducted using 11 empirical datasets commonly used to evaluate tree proposals (SI, Table 4; Lakner et al. 2008; Höhna and Drummond 2012; Whidden and Matsen 2015).

Set-up of the Analyses

I analyzed each datasets with MrBayes (Ronquist et al. 2012) to obtain accurate estimates of the tree and split posterior distributions. Simulated and empirical datasets were analyzed under the GTR model and GTR+ Γ model with 4 categories, respectively (SI, priors settings). Analyses of at least 50 million iterations with 4 Metropolis-coupled chains were replicated 3 times and used to define the reference split frequencies. Each analysis reached an ASDSF value smaller than 0.005, suggesting that runs converged properly.

The adaptive and parsimony-guided proposals are implemented in the CoevRJ software (Meyer et al. 2019) that I therefore used to assess their performance. Analyses of 10 million iterations with 10 independent chains were conducted for each proposal mixture and dataset with the same models and prior setting used to build the reference split frequencies. Additional runs with 3 Metropolis-coupled chains were conducted for the empirical datasets (using the same settings, iterations and replicas number as their MCMC equivalent).

Mixtures of Proposals

Five different proposal mixtures were considered for these experiments (SI, Table 3). The reference mixture is the *naive* mixture composed of the stNNI, eSPR and eTBR proposals applied each at equal frequency (as in MrBayes), with a probability $p = 0.6$ of extending a path for eSPR and eTBR. All subsequent mixtures applied stNNI and eSPR proposals at low frequency as a baseline.

The *adaptive* and *guided* mixtures were composed of all adaptive (A-stNNI, A-2SPR and A-PBJ) and guided (G-stNNI, G-1SPR, G-2SPR) proposals, respectively, each applied at equal frequencies. The *mixed* mixture applied all the adaptive and guided proposals with equal frequencies, with the exception of the G-stNNI that was removed due to its expensive computational cost. Finally, the *best* mixture identified a version of the *mixed* mixture with the proposal weights tuned for performance (based on empirical observations).

Performance of Proposal Mixtures on Simulated Datasets

Proposal performance.—On the four simulated data sets, the adaptive proposals achieved the best mixing efficiency according to the C-V efficiency with consistent 2 to 8-fold performance improvements over the stNNI proposal (Fig 7a). These increases in C-V efficiency were mirrored by increased acceptance rates (Fig 7c) while being of slightly lesser extent for proposals modifying multiple splits per move (e.g., A-PBJ). The ability of proposals to visit all splits according to the theoretical expectation, when compared to the stNNI proposal, generally decreased at the exception of the A-stNNI (Fig 7b). This decrease in C-V accuracy was linked with the number of edges altered (e.g., 2-edges moves only for the A-2SPR). Lastly, the measured computational cost for each proposal reflected the theoretical complexity, and particularly affected the G-stNNI proposal with a nearly 10-fold increase in computational cost (Fig 7d).

The performance of the naive proposals worsened proportionally to the amount of splits modified by a move, as shown by the decreases in all the metrics (Fig 7). Regardless of the proposal's performance, using moves that alter multiple splits remains a mandatory feature for the estimation of tree distributions to ensure the proper exploration of the tree space. For instance, eSPR-type moves were able to visit tree topologies containing splits hardly reachable by stNNI-type proposals on dataset DS3; this advantageous feature was captured by the C-V accuracy (Fig. 7b for dataset DS3). The parsimony-guided proposals, as implemented in this study, performed poorly. The G-stNNI proposal was the only parsimony-guided proposal to improve the C-V efficiency by up to 2.5-fold factor, but at the cost of a 10-fold increase in computational cost.

The A-stNNI proposal revealed to be the best overall proposal by achieving consistent increases in C-V efficiency and C-V accuracy without leading to significant increases in computational cost. The adaptive N-edges proposals, the A-2SPR and A-PBJ presented different but complementary behaviors. The A-2SPR was very efficient at visiting a subset of the splits (specialist proposal), while the A-PBJ acted as a slightly less efficient N-edges version of the A-stNNI. These two adaptive proposals

Figure 7: Performance improvements of proposals on simulated datasets using stNNI as reference. Figures a), b) and c) shows the relative increase in C-V efficiency, accuracy (1/distance) and acceptance rate, respectively. Figure d) represents the acceleration in computational time for the proposal (light colors) and for the whole MCMC iteration (proposal and likelihood evaluation; dark colors).

outperformed all their N-edges counterparts and therefore represent the best alternatives to the eSPR and eTBR proposals.

Mixtures performance and metric behaviors.—The *mixed* and *best* mixtures converged toward the reference split frequencies as fast, if not faster, than the *naïve* mixture (Fig. 8a). These mixtures, containing both type of proposals (guided and adaptive), were more consistent with respect to MCMC convergence than the mixtures composed of a single proposal type (e.g., *adaptive*). After removing the burnin, mixtures containing adaptive proposals accurately estimated the split frequencies 3- to 12-time faster than the *naïve* and *guided* mixtures (Fig. 8b).

According to the C-V efficiency and the MRT metrics, the *adaptive* and *best* mixtures were up to 12 times more efficient at sampling the posterior distribution of trees (Fig. 8c and e). Their performance was closely followed by the *mixed* mixture but remained unequaled by the *guided* mixture. The C-V accuracy metric indicated that the *guided* mixture behaved similarly to the reference and that the *best* mixture was the only one to consistently perform as well or better than the *naïve* mixture (Fig. 8d).

Most of the mixtures of proposals took slightly more computational time to reach 10 million iterations than the reference mixture. The *best* mixture was the only one to have a similar computational cost (Fig. 8f). The decreases in runtime observed on dataset DS2 and DS3 were due to a significant increase in acceptance rate of the adaptive proposals. The rejection of a proposal by the MCMC process induces computational procedures that restore the previous MCMC state. These procedures involve

costly operations (i.e., memory backups or re-evaluations of likelihoods) that are avoided whenever a move is accepted. Lastly, the only mixture that significantly underperformed in term of runtime was the *guided* mixture due to the expensive computational cost of the G-stNNI proposal.

In addition to assessing the performance of the different proposal mixtures on simulated datasets, this experiment indicated that the C-V and MRT metric were significantly more stable than the convergence metrics. Convergence metrics are more likely to be impacted by the starting tree topology and stochastic behavior of the MCMC algorithm. More importantly, this experiment highlighted that the C-V efficiency and the MRT metrics were consistently measuring improvements having the same order of magnitude for all datasets and proposals.

Proposal Mixture Performance on Empirical Datasets

The *best* proposal mixture consistently outperformed the *naïve* mixture when considering the sampling of the posterior distribution of trees, regardless of whether MC³ was used (Fig. 9c and d). On challenging datasets, the use of the *best* mixture without MC³ increased the amount of successful MCMC runs but did not guarantee convergence within the imposed 10 million iterations (Fig. 9a). The *best* mixture with MC³ was the only setting under which convergence was consistently achieved. Overall, the *best* mixture without MC³ led to runs converging as fast as the *naïve* mixture with MC³ and surpassed its performance when MC³ was not used (Fig. 9b). These improvements came at no significant additional computa-

Figure 8: Performance improvements of proposal mixtures on simulated datasets using the naive mixture as reference. Figure a) and b) summarize the acceleration in convergence time before and after removal of the burnin phase, respectively. Convergence thresholds were fixed to a 2% and 1% average error per split. Figures c) and d) represent increases in C-V efficiency and accuracy (1/distance). Figures e) and f) illustrate the improvement in MRT and total runtime.

tional cost: the variation in computational costs reached at most 11% of the reference cost, depending on the datasets and mixtures (SI Fig. 13).

Using the *best* mixture led on average to a 6-fold increase in C-V efficiency (with and without MC^3) when compared to the reference mixture (*naive* plus MC^3). The magnitude of the observed improvements were different depending on the datasets analyzed: the performance of the *best* mixture was directly correlated with the amount of information exploitable by the adaptive proposals in the distribution of split frequencies (SI, Fig. 11). Analyses on datasets with diffuse posterior tree distributions (i.e., DS5, DS9 and DS11) had only limited improvements in C-V efficiency that ranged from 2 to 2.7-fold.

Removing duplicated sequences from alignments DS9 and DS11 significantly increased their performance improvements (from 2.2 to 3.2 and 2 to 5.9-fold increases, respectively; SI, Fig. 11). For datasets with strong phylogenetic signal (e.g., DS3), the performance improvements reached up to an 18-fold efficiency increase. These observations could not be confirmed with the MRT metric due to its difficult application on diffuse tree topology distributions (Whidden and Matsen 2015). However, the observed trends of improvements and their magnitude concurred with the improvements in convergence speed after burnin removal (Fig. 9c).

Using the MC^3 algorithm had a positive effect on the convergence of Markov chains regardless of the proposal mixture: all the runs converged with the *best* mixture and only a few failed with the *naive* mixture. These convergence failures happened on the three datasets (i.e., DS1, DS2, DS4) with the highest C-V distances (SI Fig. 12), indicating that many of their splits were difficult to visit. Nonetheless, when MC^3 was used, convergence was reached on average 4.2 and 2 times faster before and af-

ter burnin removal, respectively. These improvements regarding the convergence were not captured by the C-V efficiency metric, indicating that the cold chain sampling efficiency remained mostly unaffected by MC^3 . The convergence improvements and small but systematic improvements in C-V accuracy resulting from the use of MC^3 were however consistent with the effect of accepting bolder moves that could enable to explore different peaks of the posterior distribution (Fig. 9e).

When MC^3 was not used, the C-V accuracy was improved by 22% on average by using the *best* mixture rather than the *naive* mixture. When MC^3 was used, these performance improvements in C-V accuracy were however significantly different among the datasets and the proposals mixtures, and were not apparently correlated with the improvements in mixing efficiency (e.g., DS3). While adaptive proposals slightly improved the C-V accuracy, these observations suggest that their dominant effect was to increase the frequency at which appropriate moves were proposed, rather than enabling the visit of splits unreachable by naive proposal (e.g., multiple peaks).

DISCUSSION

In this study, I have presented the concept of adaptive proposals for unrooted tree topologies and developed a family of adaptive tree proposals. These adaptive tree proposals generate moves more likely to adequately sample the posterior distribution of trees by exploiting an estimate of the marginal split frequencies. This concept was applied to two standard proposals (i.e., stNNI and eSPR) generating specific type of moves. Additionally, I presented a novel philosophy for the design of tree proposals enabled by the concept of adaptive proposals.

Figure 9: Performance improvements on empirical datasets using the *naive* mixture with MC³ as reference. Figure a) shows the percentage of runs having succeeded to converge within 10 million iterations. Figures b) and c) summarize the acceleration in convergence time before and after burnin removal, respectively. Convergence thresholds were fixed to a 2% and 1% average error per split. Figures d) and e) represent increases in C-V efficiency and accuracy.

This philosophy was used to design a proposal defining the most appropriate structure of moves by identifying strongly and weakly supported region of the phylogeny using a path-building mechanism. I showed that, while being more computationally expensive than standard proposal, the theoretic computational complexity of adaptive proposals was significantly lower than the complexity of parsimony-guided proposals and likelihood evaluations.

The performance of these adaptive tree proposals was assessed on simulated and empirical datasets. Using performance metrics designed for these experiments, I showed that adaptive proposals consistently outperformed their counterparts on simulated dataset. Using an empirically tuned proposal mixture to analyze 11 empirical datasets resulted in 2 to 18-fold improvements in mixing efficiency and up to 6 times faster convergence of MCMC and MC³ runs when compared to a standard proposal mixture composed of stNNI, eSPR and eTBR proposals. These performance improvements came without significant increase in computational cost and were correlated with the amount of phylogenetic signal in the alignments.

Adaptive proposals revealed to be superior to the naive and parsimony-guided proposals according to all metrics. The first key advantage enabling adaptive proposals to outperform other proposals is their ability to locate regions of a tree topology subject to uncertainties

that could benefit from being modified. Non-adaptive proposals always start by arbitrarily choosing a region of the tree topology to modify. After this first arbitrary choice, naive proposals continue to apply randomized topological modification, while guided proposals use a score (e.g., parsimony or posterior probability [Höhna and Drummond 2012](#))) to select the most promising resulting tree from a finite set of moves. Even if the guiding mechanism identifies good alterations, these proposals remain limited by the arbitrary choice made at first. I investigated an alternative approach for parsimony-guided proposals that involved an exhaustive exploration of all possible stNNI moves for a given tree. This proposal's mixing efficiency was better than naive proposals but worse than adaptive proposals. Furthermore, its computational complexity exceeded the cost of a likelihood evaluation by an order of magnitude.

The second key advantage of adaptive proposals is their ability to exploit the *shape* of the posterior distribution of trees by cheaply approximating the split frequencies during an MCMC run. Contrary to guided proposals using proxy scores to approximate the posterior distribution (e.g., parsimony), adaptive proposals are not affected by the substitution models used for the analyses. However, this second key advantage is also the potential pitfall because the adaptive proposals strongly rely on accurate estimates of the split frequencies. Inaccurate estimates could lead adaptive proposals to have worse per-

formance than their naive counterparts. When MC³ was used, this outcome was not observed on the 11 empirical datasets analyzed in this study, despite the ruggedness of their posterior distribution of trees (Whidden and Matsen 2015).

The concept of adaptive proposals as presented in this study is not limited to the three adaptive proposals that I developed, but opens new avenues toward the development of other tree proposals. For instance, the design philosophy used for the A-PBJ proposal that consists in designing proposals adaptively defining the move structure using the marginal split frequencies, could lead to other novel proposals by considering other path-building or structure-building strategies. Another avenue for improvements would be to estimate the joint distribution of split frequencies and exploit this information to consider bolder and more complex topological alterations. While the use of adaptive proposals could already benefit other types of phylogenetic inferences, these additional developments would be particularly beneficial to models having stronger constraints, such as clock-constrained or the inference of gene-trees within species trees (Rannala and Yang 2017).

In conclusion, the three adaptive tree topology proposals defined in this study represent a practical improvement for Bayesian inference of phylogeny, regardless of the substitution model considered. The concepts developed offer a fresh perspective on the design of tree proposals that should be advantageous to more challenging types of phylogenetic inferences and could therefore bring a new outlook to a challenging limitation existing since the early days of Bayesian inference of phylogenies.

ACKNOWLEDGMENTS

X.M. received funding from the Swiss National Science Foundation (P2GEP2_178032). I thank M. May and J. Huelsenbeck for discussion and feedback on this work.

REFERENCES

- Aberer, A. J., Kobert, K., and Stamatakis, A. (2014). Exabayes: massively parallel bayesian tree inference for the whole-genome era. *Molecular biology and evolution*, 31(10):2553–2556.
- Aberer, A. J., Stamatakis, A., and Ronquist, F. (2015). An Efficient Independence Sampler for Updating Branches in Bayesian Markov chain Monte Carlo Sampling of Phylogenetic Trees. *Systematic Biology*, page syv051.
- Altekar, G., Dwarkadas, S., Huelsenbeck, J. P., and Ronquist, F. (2004). Parallel Metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. *Bioinformatics*, 20(3):407–415.
- Baele, G., Lemey, P., Rambaut, A., and Suchard, M. A. (2017). Adaptive MCMC in Bayesian phylogenetics: an application to analyzing partitioned data in BEAST. *Bioinformatics*, 33(12):1798–1805.
- Beiko, R. G., Keith, J. M., Harlow, T. J., and Ragan, M. A. (2006). Searching for Convergence in Phylogenetic Markov Chain Monte Carlo. *Systematic Biology*, 55(4):553–565.
- Brown, J. M. and Thomson, R. C. (2018). The Behavior of Metropolis-Coupled Markov Chains When Sampling Rugged Phylogenetic Distributions. *Systematic Biology*, 67(4):729–734.
- Claywell, B. C., Dinh, V., Fourment, M., McCoy, C. O., and Matsen IV, F. A. (2018). A Surrogate Function for One-Dimensional Phylogenetic Likelihoods. *Molecular Biology and Evolution*, 35(1):242–246.
- Fletcher, W. and Yang, Z. (2009). INDELible: A Flexible Simulator of Biological Sequence Evolution. *Molecular Biology and Evolution*, 26(8):1879–1888.
- Garey, J. R., Near, T. J., Nonnemacher, M. R., and Nadler, S. A. (1996). Molecular evidence for acanthocephala as a subtaxon of rotifera. *Journal of Molecular Evolution*, 43(3):287–292.
- Gelman, A., Roberts, G. O., Gilks, W. R., et al. (1996). Efficient metropolis jumping rules. *Bayesian statistics*, 5(599-608):42.
- Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242. Mathematical Reviews number (MathSciNet) MR1828504, Zentralblatt MATH identifier 0989.65004.
- Haario, H., Saksman, E., and Tamminen, J. (2005). Componentwise adaptation for high dimensional MCMC. *Computational Statistics*, 20(2):265–273.
- Hedges, S. B., Moberg, K. D., and Maxson, L. R. (1990). Tetrapod phylogeny inferred from 18s and 28s ribosomal rna sequences and a review of the evidence for amniote relationships. *Molecular Biology and Evolution*, 7(6):607–633.
- Henk, D. A., Weir, A., and Blackwell, M. (2003). Laboulbeniopsis termittarius, an ectoparasite of termites newly recognized as a member of the laboulbeniomyces. *Mycologia*, 95(4):561–564.
- Huelsenbeck, J. P., ANe, C., Larget, B., and Ronquist, F. (2008). A bayesian perspective on a non-parsimonious parsimony model. *Systematic biology*, 57(3):406–419.
- Huelsenbeck, J. P., Ronquist, F., Nielsen, R., and Bollback, J. P. (2001). Bayesian inference of phylogeny and its impact on evolutionary biology. *science*, 294(5550):2310–2314.
- Höhna, S. and Drummond, A. J. (2012). Guided Tree Topology Proposals for Bayesian Phylogenetic Inference. *Systematic Biology*, 61(1):1–11.
- Höhna, S., Landis, M. J., Heath, T. A., Boussau, B., Lartillot, N., Moore, B. R., Huelsenbeck, J. P., and Ronquist, F. (2016). RevBayes: Bayesian Phylogenetic Inference Using Graphical Models and an Interactive Model-Specification Language. *Systematic Biology*, page syw021.
- Ingram, A. L. and Doyle, J. J. (2004). Is eragrostis (poaceae) monophyletic? insights from nuclear and plastid sequence data. *Systematic Botany*, 29(3):545–552.
- Kroken, S. and Taylor, J. W. (2000). Phylogenetic species, reproductive mode, and specificity of the green alga trebouxia forming lichens with the fungal genus letharia. *Bryologist*, pages 645–660.
- Lakner, C., van der Mark, P., Huelsenbeck, J. P., Larget, B., and Ronquist, F. (2008). Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. *Systematic Biology*, 57(1):86–103.
- Meyer, X., Chopard, B., and Salamin, N. (2017). Accelerating Bayesian inference for evolutionary biology models. *Bioinformatics*, 33(5):669–676.
- Meyer, X., Dib, L., Silvestro, D., and Salamin, N. (2019). Simultaneous bayesian inference of phylogeny and molecular coevolution. *Proceedings of the National Academy of Sciences*, page 201813836.
- Rannala, B. and Yang, Z. (2017). Efficient Bayesian Species Tree Inference under the Multispecies Coalescent. *Systematic Biology*, 66(5):823–842.
- Roberts, G. O. and Rosenthal, J. S. (2009). Examples of Adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2):349–367.
- Ronquist, F. (1998). Fast fitch-parsimony algorithms for large data sets. *Cladistics*, 14(4):387–400.
- Ronquist, F., Teslenko, M., van der Mark, P., Ayres, D. L., Darling, A., Höhna, S., Larget, B., Liu, L., Suchard, M. A., and Huelsenbeck, J. P. (2012). MrBayes 3.2: Efficient Bayesian Phylogenetic Inference and Model Choice Across a Large Model Space. *Systematic Biology*, 61(3):539–542. WOS:000303336200013.

- Rossman, A. Y., McKemy, J. M., Pardo-Schultheiss, R. A., and Schroers, H.-J. (2001). Molecular studies of the bionectriaceae using large sub-unit rDNA sequences. *Mycologia*, 93(1):100–110.
- Sen, A. and Balakrishnan, N. (1999). Convolution of geometrics and a reliability problem. *Statistics & Probability Letters*, 43(4):421–426.
- Suh, S.-O. and Blackwell, M. (1999). Molecular phylogeny of the cleistothecial fungi placed in cephalothecaceae and pseudoeurotiaceae. *Mycologia*, 91(5):836–848.
- Swofford, D. L., Olsen, G. J., and Waddell, P. J. (1996). Phylogenetic inference. in “molecular systematics”(dm hillis, c. moritz, and bk mable, eds). *Sinauer, Sunderland, Massachusetts*, 14:407–5.
- Thawornwattana, Y., Dalquen, D., and Yang, Z. (2017). Designing Simple and Efficient Markov Chain Monte Carlo Proposal Kernels. *Bayesian Analysis*.
- Whidden, C. and Matsen, F. A. (2015). Quantifying MCMC Exploration of Phylogenetic Tree Space. *Systematic Biology*, 64(3):472–491.
- Yang, Z. (1994). Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *Journal of Molecular Evolution*, 39(3):306–314.
- Yang, Z. and Rannala, B. (1997). Bayesian phylogenetic inference using dna sequences: a markov chain monte carlo method. *Molecular biology and evolution*, 14(7):717–724.
- Yang, Z. and Rannala, B. (2012). Molecular phylogenetics: principles and practice. *Nature reviews genetics*, 13(5):303.
- Yang, Z. and Yoder, A. D. (2003). Comparison of likelihood and bayesian methods for estimating divergence times using multiple gene loci and calibration points, with application to a radiation of cute-looking mouse lemur species. *Systematic biology*, 52(5):705–716.
- Yoder, A. D. and Yang, Z. (2004). Divergence dates for malagasy lemurs estimated from multiple gene loci: geological and evolutionary context. *Molecular Ecology*, 13(4):757–773.
- Zhang, N. and Blackwell, M. (2001). Molecular phylogeny of dogwood anthracnose fungus (*discula destructiva*) and the diaporthales. *Mycologia*, 93(2):355–365.

SUPPLEMENTARY MATERIALS

Learning Split Frequencies

To improve the learning of split frequencies, I used two frequencies per split: the internal and visible frequencies. The internal frequency is updated each $k_1 = 25$ samples and is only internally used by the learning process. The visible frequency is the one used by the adaptive proposals and reflects the value of the internal frequency after $k_2 = 10$ updates. Change of the split frequencies, as seen by the adaptive proposals, represents the behavior of the MCMC process averaged over a span of $k_1 \times k_2 = 250$ iterations.

To improve the rate at which split frequencies can vary, I use a relaxation mechanism that can be triggered two ways: upon an increase of 15% in the total number of splits encountered during the run or upon a significant change in the estimated marginal split frequencies. Changes in split frequencies are assessed by monitoring the trends of change of each split frequency over the last $k_3 = 1000$ updates. The trend of change is estimated as the ratio between the absolute value of the observed change over k_3 iterations and the maximum expected change over the same span of iterations. A relaxation is triggered whenever the average observed trend of change (averaged over all splits) exceeds 10% or if the maximum observed trend of change exceeds 30%. A trigger of the relaxation mechanism enables the estimates of the split frequency to adapt faster by multiplying the splits counts $cnt(e_i)$ and the total count cnt by a factor $\max(0.9, k_4/cnt)$. To ensure stability, a relaxation can only be triggered once every $k_5 = 20 \times k_1$ updates and can not reduce cnt under $k_6 = 2000$.

After a significant period of stability in split frequencies, the learning process is terminated and the frequencies are fixed. The convergence is detected once cnt reach $k_7 = 10000$. The convergence is therefore achieved once there has been no relaxation for a significant amount of iterations, implying that the number of new splits is not subject to significant increase and that the scale of change in split frequencies is low.

Figure 10 illustrates the effect of the relaxation mechanism and the relative convergence detection on a simulated dataset. The relaxation mechanism when active maintains the scale of change per update until the trends of change per split frequency reach their thresholds (Fig. 10b, c and d). The relative convergence termination enables then the split frequencies to continue their convergence for several iterations before halting the learning process. Runs without the relaxation mechanism reveal two problematic behaviors: the run with fixed convergence threshold terminate while split frequencies are still being learned, while the run without convergence threshold shows that the split frequencies are still being consistently altered after 1 million iterations (Fig. 10).

Stochastic Component

The estimation of the split frequency may turn out to be unreliable, specifically during the early steps of the learning process. I therefore augment the adaptive tree proposals with a stochastic component based on a residual term ϵ drawn from an exponential distribution with parameter λ . The value of λ follows a decreasing schedule (e.g., from 5.0 to 0.01) during the earliest steps of the learning process. This strategy improves adaptive proposals in two ways. First, moves proposed when the split frequencies are unreliable are acting similarly to naive moves, while still accounting weakly for split frequencies. As these estimates stabilize, the effect of the stochastic component decrease but continues to enable the consideration of moves unsupported by the split frequencies. Proposing unsupported moves remains advantageous by enabling the MCMC to visit unexplored parts of the tree space or by limiting the effect of inaccurate split frequency estimates.

Parsimony Score Transformation

While the concept of using different scores to guide a tree proposal is well defined (Höhna and Drummond 2012), there exists to my knowledge no reference on how to convert a parsimony score into a valid probability. The transformation procedure considers proposals that, given tree τ , enumerates K moves leading to trees $\tau'_i \forall i \in \{1 \dots K\}$ and uses the function $h(\cdot)$ to compute the parsimony score of a tree (e.g. fast-Fitch algorithms (Ronquist 1998)).

I first re-scale the relative difference between the parsimony score of the current tree $h(\tau)$ and the one of each of the proposed trees $h(\tau'_i)$ to fit within the $[-1..1]$ interval according to

$$\delta_i = \frac{h(T) - h(T'_i)}{N_{site}} \quad (27)$$

The extreme values of the $[-1..1]$ interval are unrealistic to be encountered for δ_i on local moves affecting at most less than a tenth of splits. The δ_i scores are therefore truncated within the $[-0.1..0.1]$ interval and linearly transformed to

fit within the $[0..1]$ interval according to

$$\sigma_i = \frac{\min[(\max(\delta_i, -0.1), 0.1) + 0.1]}{0.1 - (-0.1)}. \quad (28)$$

The probability of proposing move τ'_i is then defined as

$$p(\tau'_i, | \tau) = \frac{\sigma_i^x}{\sum_{k \in K} \sigma_k^x} \quad (29)$$

where x can be used to tune the behavior of the move ranging from a naive behavior with $x = 0$, to a strongly guided proposal with $x \gg 0$. I set x to 8 based on empirical observations.

Computational Complexity of Proposals

The theoretical computational complexity of the different type of proposals provides a good indication of their computational overhead with respect to the evaluation of a likelihood. An alignment contains the molecular state of m sites for n taxa, where the state is defined over k characters (e.g., $k = 4$ for nucleotide sequences). The theoretical cost of computing a parsimony score is of $O(mnk)$ operations, while the one of a likelihood using the Felsenstein pruning algorithm costs $O(mnk^2)$. To account for the use of partial evaluations (i.e., refreshing computations on a subset of the tree), I use the $\hat{n}$ notation defining a subset of operations on n . Similarly, the use of the Gamma model for rate heterogeneity is accounted for using the c variable to define the number of rate categories, leading to a complexity of $O(cm\hat{n}k^2)$ for a partial likelihood evaluation.

A naive tree proposal (e.g. stNNI or eSPR) has a computational complexity proportional to the number, s , of splits it alters (e.g. $O(1)$ for the stNNI or $O(s)$ for the eSPR). Parsimony-guided proposals have a complexity proportional to the size of the set of moves generated and the cost of estimating parsimony score for each of them. The $G - NSPR$ generates $O(2^s)$ moves, where s represents the maximum number of edges, for the eSPR move, while the $G - stNNI$ estimates 2 moves for each internal edge ($O(n)$) resulting in proposal complexity of $O(2^s m \hat{n} k)$ and $O(m \hat{n} n k)$, respectively.

The complexity of adaptive tree proposals is generally proportional to the size of the tree ($\approx O(n)$). More specifically, the A-stNNI proposal requires the computation of the normalization constant for the internal edge probabilities (denominator of Eq.(3)) that costs $O(n)$. The A-2SPR proposal requires the enumeration of all s -edges contiguous paths with $s = 2$ and the normalization of their probabilities ($O(2^s n)$). The enumeration of the 8 possible moves and the estimation of the change in split frequencies over s -edges costs $O(8s)$, leading to a total complexity of $O(s + 2^s n)$.

The A-PBJ proposal requires the calculation of all internal edge probabilities ($O(n)$) and the construction of three paths. Unstable paths cost each $\approx O(s_1)$ and $\approx O(s_2)$, respectively. The stable paths cost $\approx O(s_3^2)$ due to the marginalization over all possible central edges (Eq.(19)). This proposal's total complexity is of the order of $O(s_1 + s_2 + s_3^2 + n)$. While each path has a different length s , the sum of the path sizes has for constraint that $s_1 + s_2 + s_3 < n$ given that the concatenation of all three paths represents a contiguous path in the tree. The complexity of the A-PBJ, under realistic settings, is therefore at most $O(s_1 + s_2 + s_3^2 + n) \approx O(2n)$. Extreme cases could lead this complexity to $\approx O(n + n^2)$, but would imply a stable path including nearly all internal edges of the tree. Only nearly linear bifurcating trees having each posterior probability of split frequencies close to 1 would induce this type of therefore improbable scenario.

Dataset Simulation Setting

I simulated dataset with 20 and 32 taxa. For each taxa number, I simulated alignment composed of 500 and 1000 sites. The tree topology and branch length of these 4 datasets were simulated under a birth death model with 1.9 speciation rate, 1.1 extinction rate and the tree length was rescaled such as to have an expected branch length of 0.1. Alignments were simulated under a GTR model with realistic stationary frequencies ($\pi_{ACTG} = (0.3, 0.2, 0.2, 0.3)$) and exchangeability rates ($a = 0.06, b = 0.11, c = 0.28, d = 0.06, e = 0.33, f = 0.17$) that were estimated from a 16S ribosomal RNA alignment.

Prior Settings

Simulated datasets were analyzed under a GTR model, while empirical datasets were analyzed under a GTR+Gamma model with 4 discrete rate categories. I used flat Dirichlet priors for the state frequencies and the exchangeability rates.

Exponential distributions with rates $\lambda = 10$ were assigned to branch lengths and with rate $\lambda = 1$ to the α parameter of the Gamma model.

Figure 10: Learning process of the split frequencies.

Mixture	Proposals (weight)
<i>Naive</i>	stNNI (1), eSPR (1.0), eTBR (1)
<i>Adaptive</i>	stNNI (1/3), eSPR (1/3), A-stNNI (1), A-2SPR (1), A-PBJ (1)
<i>Guided</i>	stNNI (1/3), eSPR (1/3), G-stNNI (1), G-1SPR (1), G-2SPR (1)
<i>Mixed</i>	stNNI (1/3), eSPR (1/3), A-stNNI (1), A-2SPR (1), A-PBJ (1), G-1SPR (1), G-2SPR (1)
<i>Best</i>	stNNI (1/15), eSPR (1/20), A-stNNI (1), A-2SPR (1/10), A-PBJ (1/5) G-2SPR (1/15)

Table 3: Proposal mixtures and proposal weights. The weights define the relative frequencies at which each proposals are applied.

Data	N	Cols	Type of data	Study	$P(s_i > 0.3)$
DS1	27	1949	rRNA; 18s	Hedges et al. (1990)	0.89
DS2	29	2520	rDNA; 18s	Garey et al. (1996)	0.94
DS3	36	1812	mtDNA; COII (1678); cytb (679-1812)	Yang and Yoder (2003)	0.97
DS4	41	1137	rDNA; 18s	Henk et al. (2003)	0.90
DS5	50	378	Nuclear protein coding; wingless	Lakner et al. (2008)	0.71
DS6	50	1133	rDNA; 18s	Zhang and Blackwell (2001)	0.88
DS7	59	1824	mtDNA; COII; and cytb	Yoder and Yang (2004)	M2249
DS8	64	1008	rDNA; 28s	Rossmann et al. (2001)	0.90
DS9	67	955	Plastid ribosomal protein; s16 (rps16)	Ingram and Doyle (2004)	0.67
DS9_U	43	955			0.72
DS10	67	1098	rDNA; 18s	Suh and Blackwell (1999)	0.86
DS11	71	1082	rDNA; internal transcribed spacer	Kroken and Taylor (2000)	0.59
DS11_U	53	1082			0.87

Table 4: Empirical datasets used and the *information* contained in the split frequencies. Datasets postfixed with the letter *u* have been altered by removing duplicate sequences.

Figure 11: Linear regression of the information contained in the split frequencies and the log of the C-V efficiency increase. The information contained in split frequencies is computed as the CDF(0.3, 1.0) of the distribution of split frequencies and captures the amount of strongly structure in the tree.

Figure 12: Absolute performance metrics on empirical datasets. Figure a) and b) summarize the convergence time before and after burn-in removal, respectively. Convergence thresholds were fixed to a 2% and 1% average error per split. Figures c) and d) represent the C-V efficiency and distance.

Figure 13: Overall computational overhead on empirical datasets using the naive mixture as reference. Analyses with MC³ were run using multiple processors (parallel).