

# Multi-rate Poisson Tree Processes for single-locus species delimitation under Maximum Likelihood and Markov Chain Monte Carlo.

## Supplementary Material I

Paschalia Kapli<sup>1</sup>, Sarah Lutteropp<sup>1,2</sup>, Jiajie Zhang<sup>1</sup>, Kassian Kobert<sup>1</sup>, Pavlos Pavlidis<sup>3</sup>,  
Alexandros Stamatakis<sup>1,2</sup>, and Tomáš Flouri<sup>1,2</sup>

<sup>1</sup> Heidelberg Institute for Theoretical Studies, Germany

<sup>2</sup> Karlsruhe Institute of Technology, Institute for Theoretical Informatics, Germany

<sup>3</sup> Institute of Molecular Biology and Biotechnology (IMBB), Foundation of Research and Technology, Heraklion, Greece

### 1 Asymptotic time and space complexities

Here we prove the asymptotic time complexity of the heuristic algorithm that estimates the maximum-likelihood delimitation. This supplement follows the notation introduced in the main paper.

**Theorem 1 (Complexity).** *Given a rooted binary tree of  $n+1$  leaves, the two algorithms (single- and multi-rate) have an asymptotic run-time complexity of  $\mathcal{O}(n^2)$  on average, and  $\mathcal{O}(n^3)$  in the worst-case.*

*Proof.* W.l.o.g, we first compute the asymptotic run-time complexity for updating the array of size  $m = |T_u| + 1$  at some inner node  $u$  of the binary tree  $T$  (the root of subtree  $T_u$ ). Let  $v$  and  $w$  be the two child nodes of  $u$ , and  $T_u$  the subtree rooted at  $u$ . For computing an arbitrary entry  $i$  of the array, the algorithm uses each entry  $j \in \{0, 1, \dots, i\}$  of the array at node  $v$  in combination with entry  $k = i - j$  at node  $w$ . Processing each such triplet  $i, j, k$  requires a constant time  $c = \mathcal{O}(1)$ . Therefore, the asymptotic time complexity for computing all  $m$  entries of node  $u$  is

$$\sum_{i=0}^{m-1} \sum_{j=0}^{i-2} c = \sum_{i=0}^{m-1} (i-1)c = c \left( \sum_{i=1}^{m-1} i - m \right) = c \left( \frac{m(m-1)}{2} - m \right) = \mathcal{O}(m^2). \quad (1)$$

To analyze the average case complexity of the algorithm, we first estimate the average cost for updating the entries of a node  $u$ . Since it is proportional to the size of the subtree  $T_u$ , we first estimate the average size of a subtree over all possible binary trees with  $n+1$  tips. The number of possible binary trees with  $n+1$  tips is given by the  $n$ -th Catalan number [8]

$$C_n = \frac{1}{n+1} \binom{2n}{n}$$

whose asymptotic growth is

$$C_n \sim \frac{4^n}{n^{3/2} \sqrt{\pi}} \quad (2)$$

(see [5]). The sum of all subtree sizes for all possible binary trees of  $n+1$  tip nodes is

$$2^{2n+1} - \binom{2n+1}{n+1} \sim 2^{2n+1} \left( 1 - \frac{1}{\sqrt{n\pi}} \right) \quad (3)$$

see [6] and [3], pp. 184-185. The average subtree size  $\bar{s}_n$  is the ratio of the sum of all subtree sizes for all possible binary trees with  $n + 1$  tip nodes (Equation 3) over the number of all subtrees (Equation 2 multiplied by the number of inner nodes  $n$ )

$$\bar{s}_n \sim \frac{2^{2n+1}(1 - \frac{1}{\sqrt{n\pi}})}{n \frac{4^n}{n^{3/2}\sqrt{\pi}}} \quad (4)$$

Simplifying equation 4, we obtain the asymptotic average subtree size

$$\bar{s}_n \sim \frac{2^{2n+1}(1 - \frac{1}{\sqrt{n\pi}})}{n \frac{4^n}{n^{3/2}\sqrt{\pi}}} = \frac{2^{2n+1}(\frac{\sqrt{n\pi}-1}{\sqrt{n\pi}})}{\frac{2^{2n}(2n+1)}{n\sqrt{n\pi}}} = 2 \frac{\sqrt{n\pi}-1}{\frac{2n+1}{n}} \sim \mathcal{O}(\sqrt{n}). \quad (5)$$

Now, assume  $T$  is a tree of  $n + 1$  tips, and as such, consists of exactly  $n$  inner nodes. The cost for processing an inner node  $u$  is  $\mathcal{O}(m^2)$  where  $m = |T_u| + 1$ . From Equation 5, the subtree size is  $\mathcal{O}(\sqrt{n})$  on average. Therefore, the number of operations required for the algorithm is  $n \times \mathcal{O}(\sqrt{n}^2)$  on average, which equals to  $\mathcal{O}(n^2)$ .

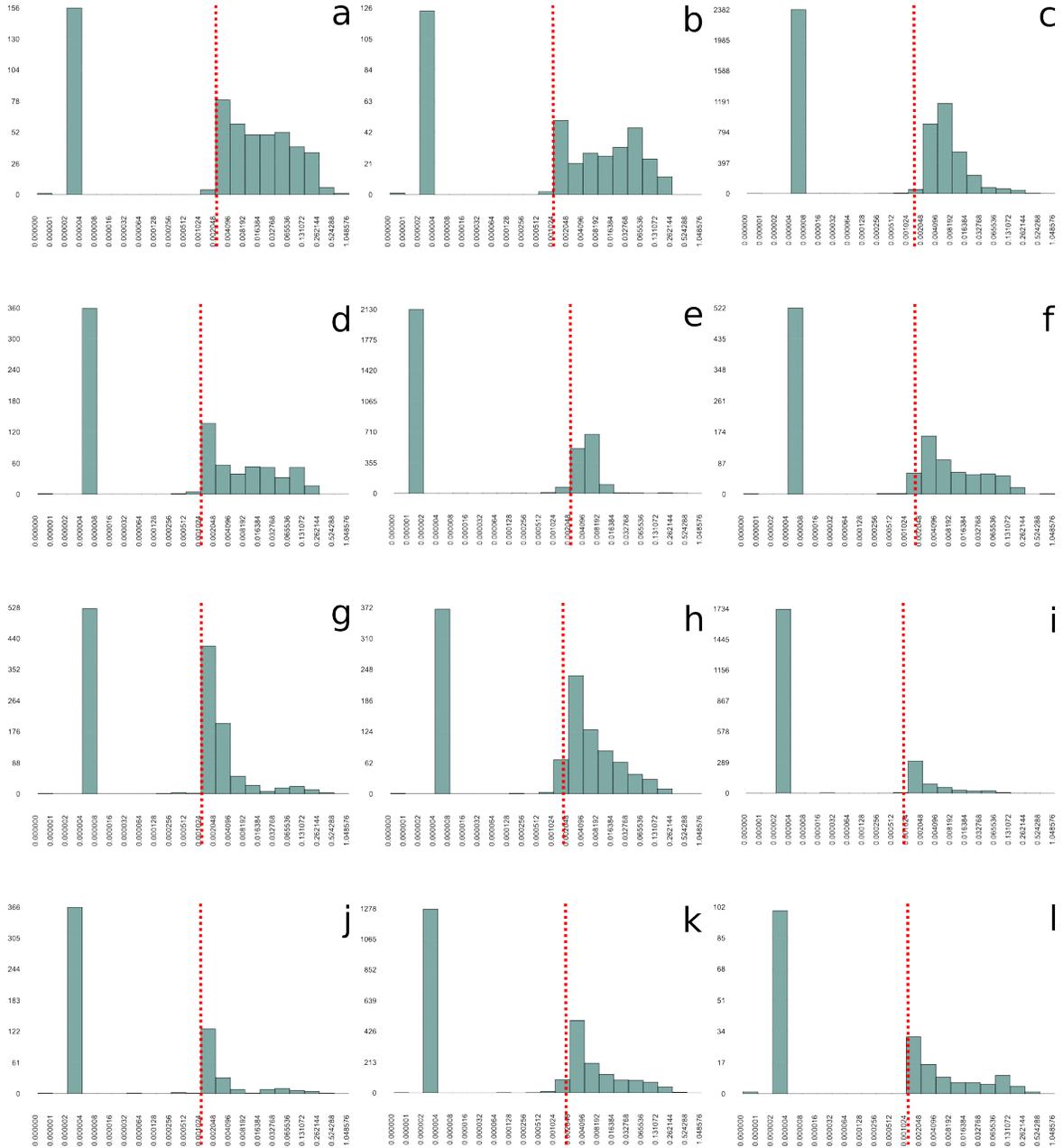
For computing the worst-case time complexity, we first compute an upper bound which trivially we know cannot be exceeded. We consider that if *all* subtrees of  $T$  were of size  $\mathcal{O}(n)$ , then the worst-case complexity would be  $\mathcal{O}(n^3)$  obtained in a similar way as the average-case time complexity. We show that this complexity can be reached with *caterpillar* trees, i.e., trees in which all nodes are within distance 1 of a central path. Consider a caterpillar tree with  $n$  tip nodes. In this case, the total amount of work needed for all inner nodes is

$$\sum_{i=1}^{n-1} (2i)^2 = 4 \frac{(n-1)n(2(n-1)+1)}{2}$$

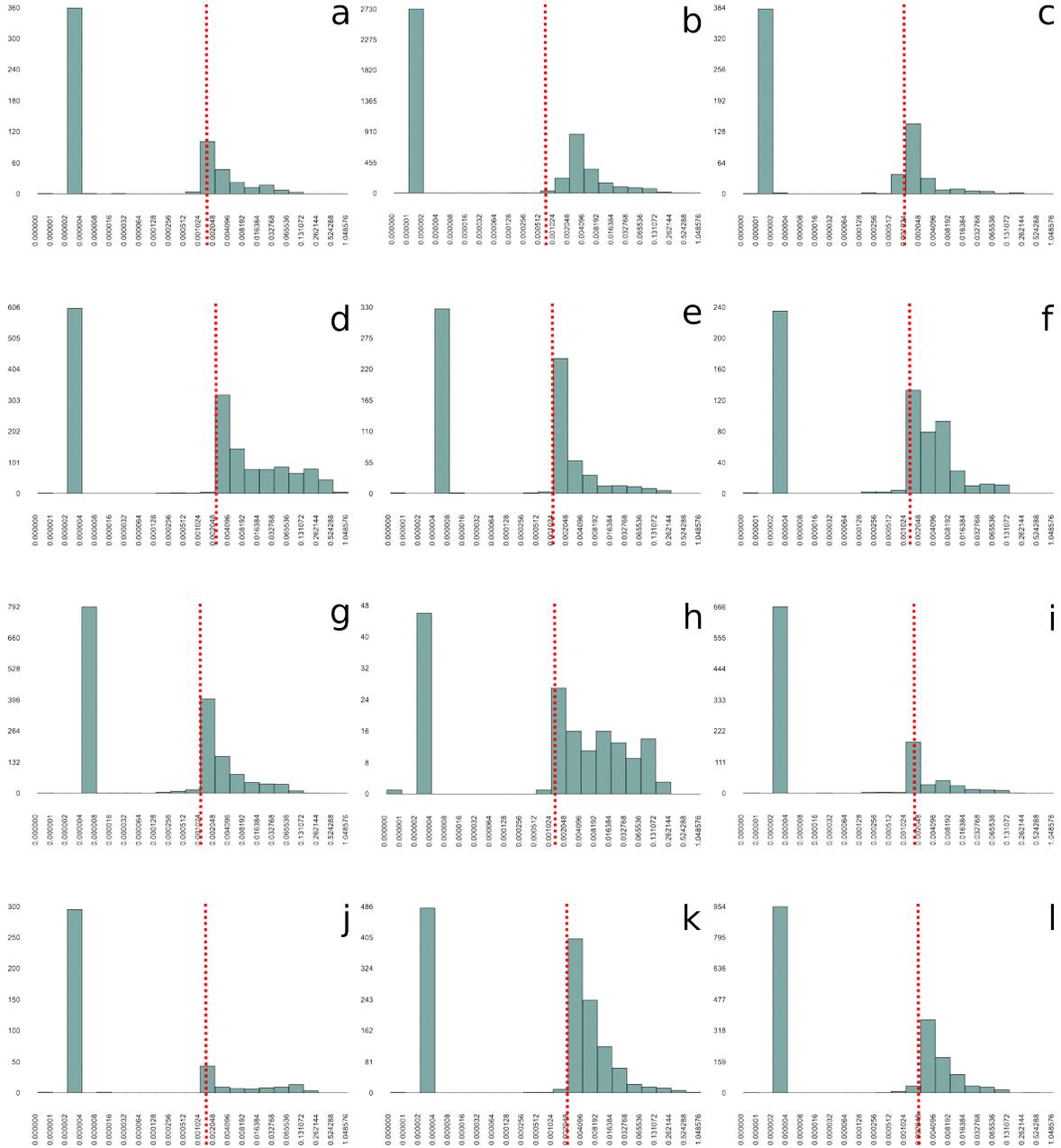
which reduces to  $\mathcal{O}(n^3)$ . □

## 2 Auto-detection of minimum branch length threshold

During phylogenetic inference, the topology of the tree is forced to remain binary at all steps of the optimization in most common implementations. Therefore, very short non-zero branch lengths are enforced among identical sequences, to retain the binary shape of the tree. These branch lengths are significantly smaller than the remaining ones, and thus, it is probable that the method falsely classifies these two groups of branch lengths into within and between species processes, respectively. One way to avoid this type of error is by preprocessing the data and keeping one representative per sequence cluster. Such a procedure is trivial when considering identical sequences (for instance, using the “-f c” option in RAxML [7]). However, it turns into an optimization problem when selecting over sequences that are of different size, contain gaps or ambiguities. The problem becomes more complex if we further consider that selecting representative sequences from such clusters (i.e. cluster of sequences that are the same but differ in size/gaps/ambiguities) might result in removing informative sites from the phylogenetic inference. To handle this complication, PTP ignores branch lengths below a certain minimum threshold (0.0001), which, based on empirical observations, were found to correspond to nonsensical branch lengths due to similar sequences in the alignment. The weakness of this approach is that such a threshold may vary among different datasets and phylogenetic inference methods. In the current implementation, we therefore use a novel approach to identify the appropriate minimum branch length threshold for any dataset. The aim of the method is to identify the subtrees of the phylogeny that correspond to sequences of zero p-distance and set the maximum branch length found in these subtrees as the threshold branch length. This approach identifies the correct threshold while it frees the user from cumbersome data preprocessing. For this step of the analysis, both, the phylogenetic tree, and the alignment are required. Subtrees exclusively comprising branch lengths smaller or equal to this threshold are subsequently ignored in the delimitation step. Figures 1 and 2, present the branch length distribution of the 24 phylogenies used in the study and the branch length threshold value that was identified by the auto-detection method.



**Fig. 1.** Branch length distribution of the empirical datasets: a) *Amyntas* b) *Anolis* c) *Anopheles* d) *Atheta* e) *Balanus* f) *Bembidion* g) *Calcinus* h) *Carabus* i) *Clubiona* j) *Coryphopterus* k) *Culicoides* l) *Cyanea*. The y-axis represents the number of sequences and the x-axis the branch lengths. The red dashed line indicates approximately the branch length value under which branch lengths correspond to sequences with a P-Distance of zero



**Fig. 2.** Branch length distribution of the empirical datasets: a) *Digrammia* b) *Drosophila* c) *Echinococcus* d) *Gammarus* e) *Holothuria* f) *Mopalia* g) *Myotis* h) *Ophiura* i) *Philodromus* j) *Phyllotreta* k) *Rhagada* l) *Xysticus*. The y-axis represents the number of sequences and the x-axis the branch lengths. The red dashed line indicates approximately the branch length value under which branch lengths correspond to sequences with a P-Distance of zero

### 3 Complexity of the Markov Chain Monte Carlo method

Given a tree  $T = (V, E)$ , the MCMC method samples the space of possible delimitations in order to draw *support values* for each node, i.e., the percentage of MCMC samples (delimitations) in which the particular node was part of the speciation process. The method explores the space of possible delimitations by proposing a new delimitation from the previous one. This is done by applying with equal probability one of the following two moves:

- **Join.** Join two coalescent processes represented by adjacent subtrees  $T_v$  and  $T_w$  into a single coalescent process, and set their parent node  $u$  (which was part of the speciation process) as the coalescent root.
- **Split.** Split a coalescent process represented by subtree  $T_u$  into two new processes represented by subtrees  $T_v$  and  $T_w$  where  $v, w$  are the two direct descendants of  $u$ . Node  $u$  is now part of the speciation process, and  $v$  and  $w$  are coalescent roots.

After applying a move, the method computes the maximum likelihood estimator of the rate parameter (rate of exponential distribution) for each coalescent process and for the single speciation process, and then calculates the likelihood of the delimitation.

In the following two subsections, we show that the calculation of the rate parameters is done in  $\mathcal{O}(1)$  time for each delimitation, except the initial one which requires  $\mathcal{O}(|T|^2)$  time on average. Assuming that the method runs for  $n$  MCMC steps, we show that the algorithm requires time  $\mathcal{O}(n + |T|^2)$  to compute support values for each node of  $T$ , i.e., all subsequent steps (apart from the first step) are independent of the tree size.

#### 3.1 Parameter estimation and computation of log-likelihood

As each possible coalescent process is represented by a subtree of  $T$ , we can pre-compute the MLE of the rate parameters for each coalescent process in a single pre-processing step. Recall that the MLE estimate of  $\lambda_i$  for some (sub-)tree  $T_i = (E_i, V_i)$  is

$$\hat{\lambda}_i = \left( \frac{1}{|E_i|} \sum_{x \in E_i} \ell(x) \right)^{-1},$$

i.e.,  $\lambda_i$  is dependent on the number of branches of subtree  $T_i$ , and the sum of their lengths. Therefore, using a single, bottom-up (postorder) traversal of the tree we can compute the rate parameters for all  $|L(T)| - 1$  possible coalescent processes (subtrees) in time  $\mathcal{O}(|T|)$ . Similarly, we may also pre-compute the likelihood function for each coalescent process (represented by subtree  $T_i$ ), i.e.,

$$\prod_{x \in E_i} \hat{\lambda}_i e^{-\hat{\lambda}_i \ell(x)}. \quad (6)$$

Due to the exponential number of possible speciation processes ( $\mathcal{O}(1.502^m)$  for a tree of  $m$  tips, see [4]), we must compute the MLE of its parameter on-the-fly each time we propose a new delimitation. However, it is possible to compute it in  $\mathcal{O}(1)$  time if we keep track of the number of branches (*count*) and the sum of their lengths (*sum*) within the speciation process when applying each of the two possible moves:

- **Split.** Sum the lengths of the two edges that were previously part of a coalescent process into *sum*, and increase *count* by two.
- **Join.** Subtract the branch lengths of the two edges that are added to the newly created coalescent process from *sum*, and decrease *count* by two.

The likelihood formula (Eq. 6) can be simplified to

$$\hat{\lambda}_i^{|E_i|} e^{-\hat{\lambda}_i \sum_{x \in E_i} \ell(x)}.$$

As it only depends on the number of branches, their sum and the rate parameter, it follows that maintaining the variables *sum* and *count* allows us to compute the likelihood function for the speciation process in  $\mathcal{O}(1)$  time.

Similarly, we compute the overall likelihood  $L'$  of the new proposed delimitation  $\theta'$  from the likelihood  $L$  of the previous delimitation  $\theta$ . As a first step, we divide  $L$  by the likelihood of the speciation process from  $\theta$  and multiply it with the likelihood of the new speciation process from  $\theta'$ . Then, depending on the move used to obtain  $\theta'$ , we perform one of the following operations:

- **Join.** Divide  $L$  by the likelihoods of the two coalescent processes from  $\theta$  (which were joined into a single process in  $\theta'$ ), and multiply it with the likelihood of the new coalescent process.
- **Split.** Divide  $L$  by the likelihood of the eradicated coalescent process, and multiply it with the likelihood of the two new coalescent processes.

The new value of  $L$  is the likelihood  $L'$  of the proposed delimitation. Clearly, it takes  $\mathcal{O}(1)$  time to compute the likelihood of each subsequent delimitation from the delimitation of the previous step.

### 3.2 Support values

The support value of a node  $u$  is the normalized sum of Akaike weights [1,2] of all sampled delimitations for which  $u$  was part of the speciation process. For simplicity, we will first assume that support values indicate the fraction of sampled delimitations in which a node was part of the speciation process, i.e., we will assume that all sampled delimitations have equal weights. Then, at the end of this section, we will show, with a simple modification, how to accommodate Akaike weights.

Assume  $count(u)$  stores the number of times (count) node  $u$  was part of the speciation process. A naive way to compute it would be to visit all nodes at each MCMC step, and increase  $count(u)$  by one for every node  $u$  that is part of the speciation process. This, however, requires  $\mathcal{O}(|T|)$  time and, assuming  $n$  MCMC steps, would increase the total run-time of the method to  $\mathcal{O}(n|T| + |T|^2)$ . We use a faster way that requires an additional variable  $start(u)$  for each node  $u$ , that indicates the MCMC step at which  $u$  became part of the speciation process. We further introduce two additional phases; an initialization phase that is executed before the MCMC method (after building the initial delimitation) and a finalization phase that is executed after the MCMC sampling ends.

**Initialization step.** We visit all nodes and set  $start(u) = 0$  for each node  $u$  that is part of the speciation process in the initial delimitation, and  $start(u) = -1$  to all nodes which are part of the coalescent process. This step requires  $\Theta(|T|)$  time.

**Step  $i$  of MCMC.** At each step  $i$ , the MCMC procedure performs one of the following two cases, depending on the type of move that led to the proposed delimitation.

- *Split.* We split a coalescent process with root  $u$  into two new coalescent processes and make  $u$  part of the speciation process. We set its two direct descendants as new coalescent roots. In case the new proposed delimitation is *accepted*, we set  $start(u) = i$ , i.e., the step from which  $u$  started to be part of the speciation process. In the opposite case (*rejected*), we increase  $count(u)$  by one to account for this one sample in which  $u$  was part of a speciation.
- *Join.* We join two coalescent processes with roots  $v$  and  $w$  into one process with root  $u$  (their parent node) making  $u$  part of the coalescent process. In case the new proposed delimitation is *rejected*, we decrease  $count(u)$  by one to account for this one sample in which  $u$  was part of a coalescent process. In the opposite case (*accepted*),  $u$  will be part of a coalescent process in the next delimitations, and hence we must add to  $count(u)$  the amount of steps  $u$  was in the speciation process since step  $start(u)$ . This is equal to exactly  $i - start(u)$ . Finally, we set  $start(u) = -1$  to indicate that the node is part of a coalescent process.

Clearly, for both cases, this step requires  $\mathcal{O}(1)$  time.

**Finalization step.** After the last step of the MCMC method is completed, we visit each node  $u$  for which  $start(u) \neq -1$  (i.e., nodes that are part of the speciation process) and add to  $count(u)$  the number of steps  $u$  was in speciation process since  $start(u)$ , i.e.,  $n - start(u)$ . This requires exactly  $\Theta(|T|)$  time.

Therefore, the total run-time of the MCMC method for tree  $T$  and  $n$  steps is  $\mathcal{O}(|T|^2 + n)$  if the maximum likelihood delimitation is used in the initial step, or  $\mathcal{O}(|T| + n)$  for a randomly generated initial delimitation.

To accommodate Akaike weights (in short “weights” from now on) in the method, we introduce an additional variable  $weights(u)$  for each node  $u$ , and a single global variable  $weightsum$ . We denote the weight of sampled delimitation  $i$  as  $w_i$ . At step  $i$ , the global variable  $weightsum$  holds the sum of all sampled delimitations so far, i.e.,  $\sum_{j=1}^i w_j$ . Each time a node  $u$  becomes part of the speciation process, we set  $weights(u)$  to  $weightsum$ , i.e. the sum of weights of all sampled delimitations so far. The first difference with the previous approach is that whenever a node  $u$  becomes part of a coalescent process, we will add to  $count(u)$  the difference  $weightsum - weights(u)$ , which accounts for the sum of weights from the step  $u$  became part of the speciation until the current step in which  $u$  becomes part of a coalescent. The second difference is that instead of adding a weight of 1 to  $count(u)$  at some step  $i$ , as described with the assumption of uniform weights, we will have to add  $w_i$ .

## 4 Supplementary Results

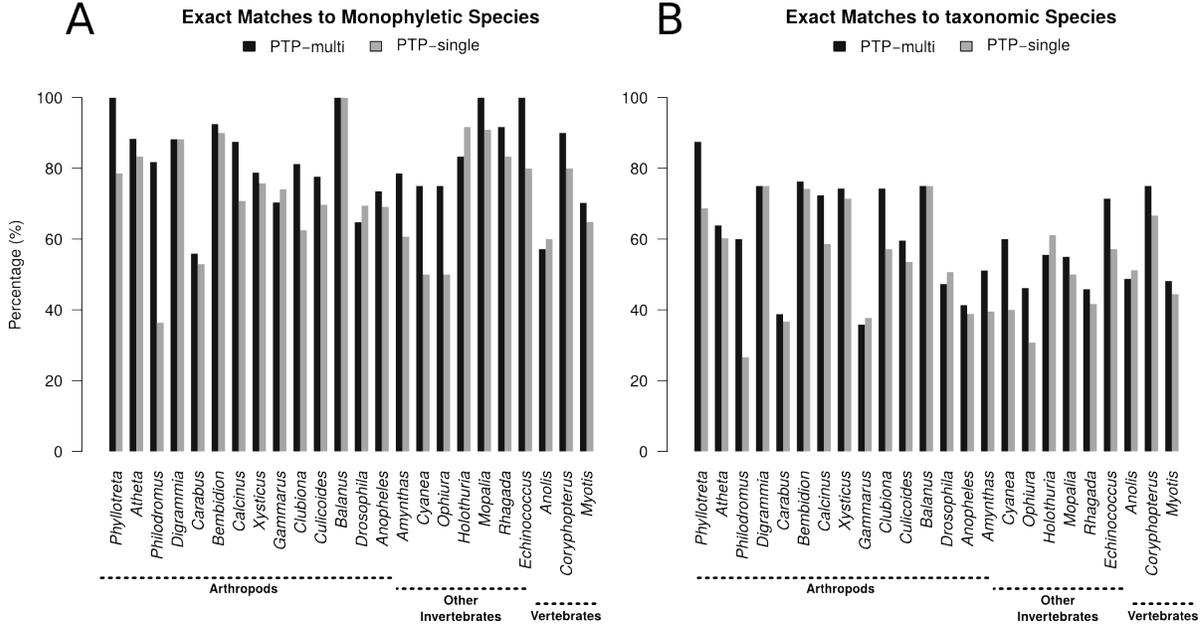
### 4.1 Results of empirical studies

Table 1, presents the efficiency of the five delimitation methods for the 24 empirical datasets used in the study. The efficiency was measured with the three parameters: i) percentage of recovered taxonomic species (RTS), ii) F-score and iii) the number of delimited species (for more details refer to the main text).

**Table 1.** Percentage of RTS, F-scores and number of delimited species for the five delimitation methods (mPTP, PTP, Uclust, Crop and ABGD) for all 24 empirical datasets.

Genus	RTS (%)					F-score					Number of species				
	mPTP	PTP	Uclust	Crop	ABGD	mPTP	PTP	Uclust	Crop	ABGD	mPTP	PTP	Uclust	Crop	ABGD
<i>Amyntas</i>	51	40	44	40	44	0.784	0.638	0.649	0.674	0.673	64	104	95	91	90
<i>Phyllotreta</i>	88	69	75	75	75	0.945	0.937	0.944	0.904	0.910	15	18	16	14	17
<i>Atheta</i>	64	60	57	55	60	0.839	0.844	0.836	0.832	0.850	85	100	95	95	91
<i>Philodromus</i>	60	27	47	53	47	0.882	0.717	0.812	0.852	0.828	21	38	26	20	24
<i>Digrammia</i>	70	75	40	50	55	0.901	0.912	0.810	0.751	0.773	26	22	25	16	16
<i>Carabus</i>	35	37	37	41	43	0.699	0.698	0.728	0.727	0.740	42	85	69	62	74
<i>Bembidion</i>	73	74	74	69	72	0.899	0.907	0.914	0.889	0.879	97	115	104	106	97
<i>Calcinus</i>	72	59	69	72	72	0.902	0.836	0.887	0.905	0.897	30	38	42	34	33
<i>Xysticus</i>	74	71	57	51	60	0.929	0.744	0.903	0.882	0.907	40	66	51	48	44
<i>Gammarus</i>	32	38	38	30	36	0.642	0.511	0.486	0.488	0.512	96	177	193	195	177
<i>Clubiona</i>	74	57	74	77	74	0.922	0.777	0.932	0.943	0.930	39	79	39	38	38
<i>Culicoides</i>	59	53	47	43	53	0.828	0.658	0.769	0.768	0.803	132	207	175	197	163
<i>Balanus</i>	75	75	50	50	100	0.964	0.964	0.637	0.935	0.964	5	5	16	7	5
<i>Drosophila</i>	47	51	41	34	53	0.728	0.559	0.747	0.729	0.765	139	444	183	217	157
<i>Anopheles</i>	41	39	30	31	40	0.787	0.704	0.730	0.681	0.730	126	218	193	154	118
<i>Anolis</i>	49	51	51	49	61	0.776	0.765	0.765	0.767	0.844	35	66	66	65	56
<i>Coryphopterus</i>	75	67	42	58	67	0.990	0.986	0.984	0.935	0.990	13	14	16	13	13
<i>Myotis</i>	52	44	48	46	48	0.796	0.793	0.776	0.805	0.811	60	81	74	66	70
<i>Cyanea</i>	60	40	80	80	60	0.763	0.741	0.762	0.781	0.887	11	23	22	17	12
<i>Ophiura</i>	38	31	38	38	38	0.617	0.678	0.667	0.696	0.707	18	22	22	22	20
<i>Holothuria</i>	56	61	67	67	67	0.852	0.858	0.908	0.881	0.883	19	22	22	21	21
<i>Mopalia</i>	55	50	50	45	50	0.932	0.923	0.904	0.854	0.932	20	21	21	23	19
<i>Rhagada</i>	46	42	50	46	42	0.553	0.589	0.445	0.414	0.574	48	71	97	132	50
<i>Echinococcus</i>	71	57	71	57	43	0.934	0.879	0.974	0.893	0.860	9	12	9	10	7

Figure 3, presents the percentage of the recovered taxonomic species by PTP and mPTP for each of the 24 empirical datasets. When we consider only the monophyletic species in each phylogeny the recovery percentage is significantly higher for both methods. The latter illustrates the dependency of the accuracy of the methods to monophyly.



**Fig. 3.** A) Both PTP models recover a high percentage of monophyletic species present in each of the empirical datasets, (80% and 72 % for mPTP and PTP on average correspondingly), B) percentage of overall recovered taxonomic species (including polyphyletic species) for each of the datasets (59% and 53% for mPTP and PTP on average correspondingly)

## 4.2 Execution times

Large biodiversity studies often involve hundreds or thousands of sequences. In the analysis of such data, computational resources can be a limiting factor, and thus, short execution times can be a crucial factor for software selection. The novel dynamic programming implementation of mPTP computes the Maximum Likelihood (ML) delimitation scheme almost instantaneously even for phylogenies involving thousands of sequences. Compared to the original PTP implementation, it is at least five orders or magnitude faster (see Table 2). Comparing execution times among the software used in this study was not as straight-forward, since they differ from (m)PTP in the preprocessing of the data [i.e. (m)PTP requires a phylogeny] while for some of them (e.g. Crop, ABGD) the execution time changes drastically with respect to the input parameters.

In Table 3 we further provide the execution times for assessing the confidence of the ML solution using MCMC sampling. For each dataset, we executed ten MCMC runs of  $2 \times 10^7$  steps each, starting from an initial random delimitation every time. We estimated the convergence of the independent runs by calculating the “Average Standard Deviation of Delimitation Support Values” (ASDDSV), and to obtain an overall support for the ML estimate, we computed the mean “Average Support Value” (ASV) over all ten independent runs. Comparing the equivalent execution times with the MCMC implementation of PTP [9] (available at <http://species.h-its.org/>) is impossible, considering the execution time for the ML optimization alone.

if one considers the corresponding execution times for the ML optimization.

Genus	Execution Time	
	PTP	mPTP
<i>Amynthas</i>	0m27.068s	0m0.013s
<i>Phyllotreta</i>	0m8.840s	0m0.007s
<i>Atheta</i>	2m10.569s	0m0.019s
<i>Philodromus</i>	4m9.498s	0m0.016s
<i>Digrammia</i>	0m57.262s	0m0.008s
<i>Carabus</i>	7m20.678s	0m0.040s
<i>Bembidion</i>	11m9.534s	0m0.029s
<i>Calcinus</i>	9m56.954s	0m0.051s
<i>Xysticus</i>	23m58.893s	0m0.055s
<i>Gammarus</i>	70m13.939s	0m0.069s
<i>Clubiona</i>	85m49.569s	0m0.036s
<i>Culicoides</i>	143m0.667s	0m0.125s
<i>Balanus</i>	198m17.230s	0m0.187s
<i>Drosophila</i>	844m32.892s	0m0.287s
<i>Anopheles</i>	2031m3.358s	0m0.733s
<i>Anolis</i>	0m8.983s	0m0.007s
<i>Coryphopterus</i>	0m30.269s	0m0.010s
<i>Myotis</i>	48m53.366s	0m0.056s
<i>Cyanea</i>	0m7.573s	0m0.003s
<i>Ophiura</i>	0m1.952s	0m0.003s
<i>Holothuria</i>	1m14.270s	0m0.016s
<i>Mopalía</i>	0m58.813s	0m0.016s
<i>Rhagada</i>	15m25.988s	0m0.066s
<i>Echinococcus</i>	1m2.249s	0m0.010s

**Table 2.** Execution times for the estimation of the inference of the ML delimitation with PTP and mPTP for each of the 24 empirical datasets

Genus	ASDDSV	ML ASV (%)	Execution Time
<i>Phyllotreta</i>	0.001991	72	0m38.588s
<i>Atheta</i>	0.000605	92	0m49.332s
<i>Philodromus</i>	0.003614	92	0m59.615s
<i>Digrammia</i>	0.001561	86	1m4.019s
<i>Carabus</i>	0.001857	86	1m3.661s
<i>Bembidion</i>	0.000867	91	0m56.494s
<i>Calcinus</i>	0.000079	99	0m55.099s
<i>Xysticus</i>	0.00742	79	0m55.573s
<i>Gammarus</i>	0.000548	91	0m56.498s
<i>Clubiona</i>	0.003026	83	0m54.260s
<i>Culicoides</i>	0.003973	94	0m59.164s
<i>Balanus</i>	0.000004	100	0m47.911s
<i>Drosophila</i>	0.048099	88	1m2.911s
<i>Anopheles</i>	0.016768	92	1m9.379s
<i>Amynthas</i>	0.000564	88	0m54.368s
<i>Cyanea</i>	0.000175	91	1m4.829s
<i>Ophiura</i>	0.000299	84	1m1.562s
<i>Holothuria</i>	0.00017	99	0m54.739s
<i>Mopalía</i>	0.002167	89	0m56.258s
<i>Rhagada</i>	0.005595	85	0m58.763s
<i>Echinococcus</i>	0.000248	95	0m54.002s
<i>Anolis</i>	0.000415	82	0m54.772s
<i>Coryphopterus</i>	0.000613	97	0m54.055s
<i>Myotis</i>	0.002721	89	0m57.154s

**Table 3.** The table shows the results for 1) the ASDDSV, 2) the (ASV) for the ML delimitation and 3) the accumulated execution time for 10 independent MCMC runs with  $2 \times 10^7$  MCMC steps per run, for each of the datasets.

## References

1. Akaike, H.: A Bayesian analysis of the minimum AIC procedure. *Annals of the Institute of Statistical Mathematics* 30(1), 9–14 (1978)
2. Burnham, K., Anderson, D.: *Model Selection and MultiModel Inference: A Practical Information-Theoretic Approach* (2002)
3. Flajolet, P., Sedgewick, R.: *Analytic Combinatorics*. Cambridge University Press (2009), <https://books.google.de/books?id=0h-4QcA1c1QC>
4. Fujisawa, T., Barraclough, T.G.: Delimiting Species Using Single-Locus Data and the Generalized Mixed Yule Coalescent Approach: A Revised Method and Evaluation on Simulated Data Sets. *Systematic Biology* 62(5), 707–724 (2013)
5. Graham, R., Knuth, D., Patashnik, O.: *Concrete Mathematics: A Foundation for Computer Science*. A foundation for computer science, Addison-Wesley (1994), <https://books.google.de/books?id=pntQAAAAMAAJ>
6. N.J.A., S., Lang, W.: Sequence A000346. <https://oeis.org/A000346>
7. Stamatakis, A.: RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* p. btu033 (2014)
8. Weisstein, E.: *CRC Concise Encyclopedia of Mathematics*, Second Edition. CRC Press (2002)
9. Zhang, J.: *Models and algorithms for phylogenetic marker analysis*. Ph.D. thesis, Lübeck, Univ., Diss., 2015 (2015)