

# Probabilistic programming: a powerful new approach to statistical phylogenetics

Fredrik Ronquist<sup>1†\*</sup>, Jan Kudlicka<sup>2†</sup>, Viktor Senderov<sup>1†</sup>, Johannes Borgström<sup>2</sup>, Nicolas Lartillot<sup>3</sup>, Daniel Lundén<sup>4</sup>, Lawrence Murray<sup>5</sup>, Thomas B. Schön<sup>2</sup>, David Broman<sup>4</sup>

<sup>1</sup>Department of Bioinformatics and Genetics, Swedish Museum of Natural History, Box 50007, SE-104 05 Stockholm, Sweden

<sup>2</sup>Department of Information Technology, Uppsala University, Box 337, SE-751 05 Uppsala, Sweden

<sup>3</sup>Laboratoire de Biométrie et Biologie Evolutive, UMR CNRS 5558, Université Claude Bernard Lyon 1, FR-69622 Villeurbanne Cedex, France

<sup>4</sup>Department of Computer Science, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden

<sup>5</sup>Uber AI, San Francisco CA 94105, United States

Statistical phylogenetic analysis currently relies on complex, dedicated software packages, making it difficult for evolutionary biologists to explore new models and inference strategies. Recent years have seen more generic solutions based on probabilistic graphical models, but this formalism can only partly express phylogenetic problems. Here we show that universal probabilistic programming languages (PPLs) solve the model expression problem, while still supporting automated generation of efficient inference algorithms. To illustrate the power of the approach, we use it to generate sequential Monte Carlo (SMC) algorithms for recent biological diversification models that have been difficult to tackle using traditional approaches. This is the first time that SMC algorithms have been available for these models, and the first time it has been possible to compare them using model testing. Leveraging these advances, we re-examine previous claims about the performance of the models. Our work opens up several related problem domains to PPL approaches, and shows that few hurdles remain before PPLs can be effectively applied to the full range of phylogenetic models.

1 In statistical phylogenetics, we are interested in learn-  
2 ing the parameters of models where evolutionary trees—  
3 phylogenies—play an important part. Such analyses have a  
4 surprisingly wide range of applications across the life sci-  
5 ences<sup>1,2,3</sup>. In fact, the research front in many disciplines is  
6 partly defined today by our ability to learn the parameters  
7 of realistic phylogenetic models.

8 Statistical problems are often analyzed using generic  
9 modeling and inference tools. Not so in phylogenetics,  
10 where empiricists are largely dependent on dedicated soft-  
11 ware developed by small teams of computational biolo-  
12 gists<sup>3</sup>. Even though these software packages have become  
13 increasingly flexible in recent years, empiricists are still  
14 limited to a large extent by predefined model spaces and  
15 inference strategies. Venturing outside these boundaries  
16 typically requires the help of skilled programmers and in-  
17 ference experts.

18 If it were possible to specify arbitrary phylogenetic mod-  
19 els in an easy and intuitive way, and then automatically  
20 learn the latent variables (the unknown parameters) in them,  
21 the full creativity of the research community could be un-  
22 leashed, significantly accelerating progress. There are two  
23 major hurdles standing in the way of such a vision. First, we  
24 must find a formalism (a language) that can express phyloge-  
25 netic models in all their complexity, while still being easy to  
26 learn for empiricists (*the model expression problem*). Sec-  
27 ond, we need to be able to generate computationally efficient  
28 inference algorithms from such model descriptions, draw-  
29 ing from the full range of techniques available today (*the*  
30 *automated inference problem*).

31 In recent years, there has been significant progress to-  
32 wards solving the model expression problem by adopting  
33 the framework of probabilistic graphical models (PGMs)<sup>4,5</sup>.

PGMs can express many components of phylogenetic mod-  
els in a structured way, so that efficient Markov chain  
Monte Carlo (MCMC) samplers—the current workhorse  
of Bayesian statistical phylogenetics—can be automatically  
generated for them. Other inference strategies are also read-  
ily applied to PGM components<sup>6,7</sup>.

Unfortunately, PGMs cannot express the core of phyloge-  
netic models: the stochastic processes that generate the tree,  
and anything dependent on those processes. This is because  
the resulting evolutionary tree has variable topology, while  
a PGM expresses a fixed topology. It is possible to express  
the tree as a single stochastic variable within the PGM, but  
then the structure of this critical component of the model is  
opaque to the inference machinery. Hiding the tree inside  
a stochastic variable also means that it becomes impossi-  
ble to describe relations between tree-generating processes  
and other model components, such as the rate of evolution,  
organism traits or biogeography.

Here, we show that the model expression problem can be  
solved using universal probabilistic programming languages  
(PPLs). PPLs have a long history in computer science<sup>8</sup>, but  
until recently they have been largely of academic interest  
because of the difficulty of generating efficient inference  
machinery when using such expressive languages. This is  
now changing rapidly thanks to improved methods of auto-  
mated inference for PPLs<sup>9,10,11,12,13,14</sup>, and the increased  
interest in more flexible approaches to statistical modeling  
and analysis.

To demonstrate the potential of PPLs in statistical phylo-  
genetics, we tackle a tough problem domain: models that ac-  
commodate variation across lineages in diversification rate.  
These include the recent ClaDS<sup>15</sup>, LSBDS<sup>16</sup> and BAMB<sup>17</sup>  
models, attracting considerable attention among evolution-  
ary biologists despite the difficulties in developing good  
inference algorithms for them<sup>18</sup>.

\*E-mail: fredrik.ronquist@nrm.se

†F.R., J.K., and V.S. contributed equally to this work.

69 Using WebPPL—an easy-to-learn PPL<sup>9</sup>—and Birch—a  
70 language with a more efficient inference machinery<sup>14</sup>—we  
71 develop an effective encoding approach, and then automat-  
72 ically generate sequential Monte Carlo (SMC) algorithms  
73 based on short model descriptions ( $\sim 100$  lines of code  
74 each). This is the first time that powerful and flexible SMC  
75 algorithms have been available for these models, and the first  
76 asymptotically exact inference machinery for BAMM. It is  
77 also the first time that it has been possible to compare the  
78 models directly using Bayes factors. We end the paper by  
79 discussing a few problems, all seemingly tractable, which  
80 remain to be solved before PPLs can be used to address  
81 the full range of phylogenetic models. Solving them would  
82 facilitate the adoption of a wide range of novel inference  
83 strategies that have seen little or no use in phylogenetics  
84 before.

## 85 Results

86 **Probabilistic programming.** Consider one of the sim-  
87 plest of all diversification models, constant rate birth-death  
88 (CRBD), in which lineages arise at a rate  $\lambda$  and die out at  
89 a rate  $\mu$ , giving rise to a phylogenetic tree  $\tau$ . Assume that  
90 we want to infer the values of  $\lambda$  and  $\mu$  given some phylo-  
91 genetic tree  $\tau_{\text{obs}}$  of extant (now living) species that we have  
92 observed (or inferred from other data). In a Bayesian anal-  
93 ysis, we would associate  $\lambda$  and  $\mu$  with prior distributions,  
94 and then learn their joint posterior probability distribution  
95 given the observed value of  $\tau$ .

96 Let us examine a PGM description of this model, say  
97 in RevBayes<sup>5</sup> (Listing 1). The first statement associates  
98 an observed tree with the variable `myTree`. The priors on  
99 `lambda` and `mu` are then specified, and it is stated that the  
100 tree variable `tau` is drawn from a birth-death process with  
101 parameters `lambda` and `mu` and generating a tree with leaves  
102 matching the taxa in `myTree`. Finally, `tau` is associated with  
103 ('clamped to') the observed value `myTree`.

### Listing 1: PGM description of the CRBD model

```
104 1 myTree = readTrees( "treefile.nex" )  
105 2  
106 3 lambda ~ dnGamma( 1, 1 )  
107 4 mu ~ dnGamma( 1, 1 )  
108 5  
109 6 tau ~ dnBirthDeath( lambda, mu, myTree.taxa )  
110 7 tau.clamp( myTree )
```

111 There is a one-to-one correspondence between these  
112 statements and elements in the PGM graph describing the  
113 conditional dependencies between the random variables in  
114 the model (Fig. 1). Given that the conditional densities  
115 `dnGamma` and `dnBirthDeath` are known analytically, along  
116 with good samplers, it is now straightforward to automat-  
117 ically generate standard inference algorithms for this prob-  
118 lem, such as MCMC.

119 Unfortunately, a PGM cannot describe from first princi-  
120 ples (elementary probability distributions) how the birth-  
121 death process produces a tree of extant species. The PGM  
122 has a fixed graph structure, while the probability of a sur-  
123 viving tree is an integral over many outcomes with varying  
124 topology. Specifically, the computation of `dnBirthDeath`

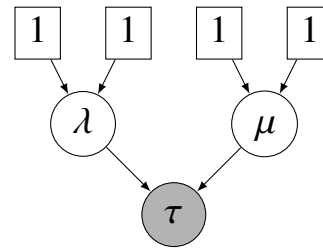


Figure 1: A probabilistic graphical model describing constant rate birth-death (CRBD). The square boxes are fixed nodes (parameters of the gamma distributions) and the circles are random variables. The shaded variable ( $\tau$ ) is observed, and  $(\lambda, \mu)$  are latent variables to be inferred.

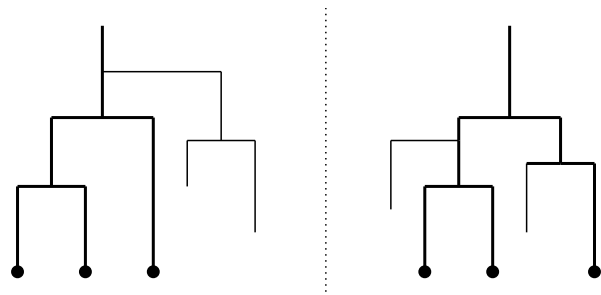


Figure 2: Two trees with extinct side branches (thin lines), each corresponding to the same observed phylogeny of extant species (thick lines). The trees illustrate just two examples of an infinite number of possible PGM expansions of the  $\tau$  node in Fig. 1.

125 requires integration over all possible ways in which the pro-  
126 cess could have generated side branches that eventually go  
127 extinct, each of these with a unique configuration of speci-  
128 ation and extinction events (Fig. 2). The integral must be  
129 computed by special-purpose code based on analytical or  
130 numerical solutions specific to the model. For the CRBD  
131 model, the integral is known analytically, but as soon as we  
132 start experimenting with more sophisticated diversification  
133 scenarios, as evolutionary biologists would want to do, com-  
134 puting the integral is likely to require dedicated numerical  
135 solvers, if it can be computed at all.

136 Universal PPLs solve the model expression problem by  
137 providing additional expressivity over PGMs. A PPL model  
138 description is essentially a simulation program (or generative  
139 model). Each time the program runs, it generates a  
140 different outcome. If it is executed an infinite number of  
141 times, we obtain a probability distribution over outcomes.  
142 The trick is to write a PPL program so that the distribu-  
143 tion over outcomes corresponds to the posterior probability  
144 distribution of interest. This is straightforward if we under-  
145 stand how to simulate from the model, and how to insert the  
146 constraints given by the observed data.

147 Assume, for instance, that we are interested in computing  
148 the probability of survival and extinction under CRBD for  
149 specific values of  $\lambda$  and  $\mu$ , given that the process started  
150 at some time  $t$  in the past. We will pretend that we do  
151 not know the analytical solution to this problem; instead we  
152 will use a PPL to solve it. WebPPL<sup>9</sup> is an easy-to-learn PPL  
153 based on JavaScript, and we will use it here for illustrating

PPL concepts. WebPPL can be run in a web browser at <http://webppl.org> or installed locally (Supplementary Section 2). Like many PPLs, WebPPL has two special constructs that we will see in the following: (1) a **sample** statement, which specifies the prior distributions from which random variables are drawn; and (2) a **condition** statement, conditioning a random variable on an observation.

In WebPPL, we define a function `goesExtinct`, which takes the values of `time`, `lambda` and `mu` (Listing 2). It returns `true` if the process does not survive until the present (that is, `goes extinct`) and `false` otherwise (survives to the present).

Listing 2: Basic birth-death model simulation in WebPPL

```

166 1 var goesExtinct = function(time, lambda, mu) {
167 2   var waitingTime = sample(
168 3     Exponential({a: lambda + mu})
169 4   )
170 5
171 6   if (waitingTime > time) { return false }
172 7
173 8   var isSpeciation = sample(
174 9     Bernoulli({p: lambda / (lambda + mu)})
175 10  )
176 11
177 12  if (isSpeciation == false) { return true }
178 13
179 14  return goesExtinct(time - waitingTime, lambda, mu)
180 15  && goesExtinct(time - waitingTime, lambda, mu)
181 16 }

```

The function starts at some `time > 0` in the past. The `waitingTime` until the next event is drawn from an exponential distribution with rate `lambda + mu` and compared with `time`. If `waitingTime > time`, the function returns `false` (the process survived). Otherwise, we flip a coin (the `Bernoulli` distribution) to determine whether the next event is a speciation or an extinction event. If it is a speciation, the process continues by calling the same function recursively for each of the daughter lineages with the updated time `time - waitingTime`. Otherwise the function returns `true` (the lineage went extinct).

If executed many times, the `goesExtinct` function defines a probability distribution on the outcome space `{ true, false }` for specific values of `t`,  $\lambda$  and  $\mu$ . To turn this into a Bayesian inference problem, let us associate  $\lambda$  and  $\mu$  with gamma priors, and then infer the posterior distribution of these parameters assuming that we have observed a group originating at time  $t = 10$  and surviving to the present. To do this, we combine the prior specifications and the conditioning on survival to the present with the `goesExtinct` function into a program that defines the distribution of interest (Listing 3).

Listing 3: CRBD model description in WebPPL

```

204 1 var model = function() {
205 2   var lambda = sample(
206 3     Gamma({shape: 1, scale: 1})
207 4   )
208 5   var mu = sample(
209 6     Gamma({shape: 1, scale: 1})
210 7   )
211 8   var t = 10
212 9
213 10  condition(goesExtinct(t, lambda, mu) == false)

```

```

11
12   return [lambda, mu]
13 }

```

Universal PPLs are by definition Turing-complete, that is, they have the same expressive power as most sophisticated programming languages used today. PGM-based systems lack expressions for stochastic branching (conditional `if-then-else` statements involving random variables) and unbounded recursion, such as the one used in the `goesExtinct` function above (Listing 2). If such constructs are provided by PGM-based software, they are only executed when the model is initiated; they are not part of the model description itself. Because of the popularity of PPLs in recent years, the term ‘probabilistic programming’ is now often used also for PGM-based languages, but here we reserve ‘probabilistic programming’ and ‘PPL’ for Turing-complete languages.

Inference in PPLs is typically supported by constructs that take a model description as input. Returning to the previous example, the joint posterior distribution is inferred by calling the built-in `Infer` function with the model, the desired inference algorithm, and the inference parameters as arguments (Listing 4).

Listing 4: Specifying inference strategy in WebPPL

```

1 Infer({model: model, method: 'SMC', particles:
10000})

```

To develop this example into a probabilistic program equivalent to the `RevBayes` model discussed previously (Listing 1), we need to describe the CRBD process along the observed tree, conditioning on all unobserved side branches going extinct (Supplementary Listings 2 and 3). The PPL specification of the CRBD inference problem is longer than the PGM specification because it does not use the analytical expression for the CRBD density. However, it exposes all the details of the diversification process, so it can be used as a template for exploring a wide variety of diversification models, while relying on the same inference machinery throughout. We will take advantage of this in the following.

**Diversification models.** The simplest model describing biological diversification is the Yule (pure birth) process<sup>19,20</sup>, in which lineages speciate at rate  $\lambda$  but never go extinct. For consistency, we will refer to it as constant rate birth (CRB). The CRBD model<sup>21</sup> discussed in the examples above adds extinction to the process, at a per-lineage rate of  $\mu$ .

An obvious extension of the CRBD model is to let the speciation and/or extinction rate vary over time instead of being constant<sup>22</sup>, referred to as the generalized birth-death process. Here, we will consider variation in birth rate over time, keeping turnover ( $\mu/\lambda$ ) constant, and we will refer to this as the time-dependent birth-death (TDBD) model, or the time-dependent birth (TDB) model when there is no extinction. Specifically, we will consider the function

$$\lambda(t) = \lambda_0 e^{z(t_0-t)},$$

where  $\lambda_0$  is the initial speciation rate at time  $t_0$ ,  $t$  is current time, and  $z$  determines the nature of the dependency. When

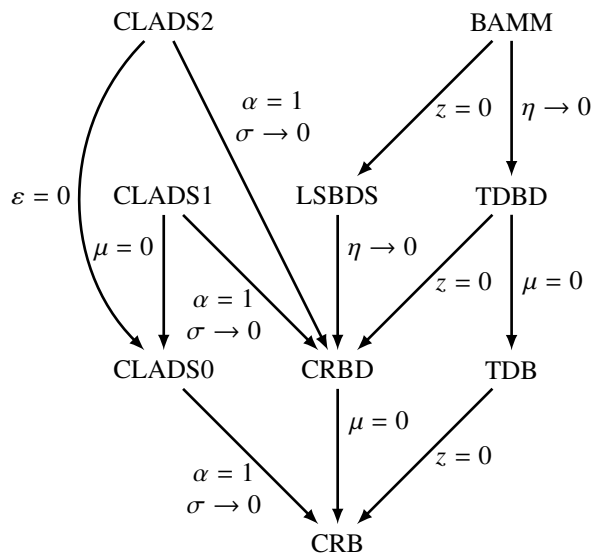


Figure 3: Relations between the diversification models considered in this paper.

269  $z > 0$ , the birth rate grows exponentially and the number of  
 270 lineages explodes. The case  $z < 0$  is more interesting bio-  
 271 logically; it corresponds to a niche-filling scenario. This is  
 272 the idea that an increasing number of lineages leads to com-  
 273 petition for resources and—all other things being equal—to  
 274 a decrease in speciation rate. Other potential causes for  
 275 slowing speciation rates over time have also been consid-  
 276 ered<sup>23</sup>.

277 The four basic diversification models—CRB, CRBD,  
 278 TDB and TDBD—are tightly linked (Fig. 3). When  $z = 0$ ,  
 279 TDBD collapses to CRBD, and TDB to CRB. Similarly,  
 280 when  $\mu = 0$ , CRBD becomes equivalent to CRB, and TDBD  
 281 to TDB.

282 In recent years, there has been a spate of work on mod-  
 283 els that allow diversification rates to vary across lineages.  
 284 Such models can accommodate diversification processes  
 285 that gradually change over time. They can also explain  
 286 sudden shifts in speciation or extinction rates, perhaps due  
 287 to the origin of new traits or other factors that are specific  
 288 to a lineage.

289 One of the first models of this kind to be proposed  
 290 was Bayesian analysis of macroevolutionary mixtures  
 291 (BAMM)<sup>17</sup>. The model is a lineage-specific, episodic  
 292 TDBD model. A group starts out evolving under some  
 293 TDBD process, with extinction ( $\mu$ ) rather than turnover ( $\epsilon$ )  
 294 being constant over time. A stochastic process running  
 295 along the tree then changes the parameters of the TDBD  
 296 process at specific points in time. Specifically,  $\lambda_0$ ,  $\mu$  and  $z$   
 297 are all redrawn from the priors at these switch points. In  
 298 the original description, the switching process was defined  
 299 in a statistically incoherent way<sup>18</sup>; here, we assume that the  
 300 switches occur according to a Poisson process with rate  $\eta$ .

301 The BAMM model has been implemented in dedicated  
 302 software using a combination of MCMC sampling and other  
 303 numerical approximation methods<sup>17,24</sup>. The implementa-  
 304 tion has been criticized because it results in severely biased  
 305 inference<sup>18</sup>. To date, it has not been possible to provide  
 306 asymptotically exact inference machinery for BAMM.

307 In a recent contribution, a simplified version of BAMM  
 308 was introduced: the lineage-specific birth-death-shift (LS-  
 309 BDS) model<sup>16</sup>. LSBDS is an episodic CRBD model, that  
 310 is, it is equivalent to BAMM when  $z = 0$ . Inference ma-  
 311 chinery for the LSBDS model has been implemented in  
 312 RevBayes<sup>5</sup> based on numerical integration over discretized  
 313 prior distributions for  $\lambda$  and  $\mu$ , combined with MCMC. The  
 314 computational complexity of this solution depends strongly  
 315 on the number of discrete categories used. If  $k$  categories  
 316 are used for both  $\lambda$  and  $\mu$ , computational complexity is mul-  
 317 tiplied by a factor  $k^2$ . Therefore, it is tempting to simplify  
 318 the model. We note that, in the empirical LSBDS examples  
 319 given so far,  $\mu$  is kept constant and only  $\lambda$  is allowed to  
 320 change at switch points<sup>16</sup>. When  $z = 0$ , BAMM collapses  
 321 to LSBDS, and when  $\eta \rightarrow 0$  it collapses to TDBD (Fig. 3).  
 322 When  $\eta \rightarrow 0$ , LSBDS collapses to CRBD.

323 A different perspective is represented by the cladogenetic  
 324 diversification rate shift (ClaDS) models<sup>15</sup>. They map di-  
 325 versification rate changes to speciation events, assuming  
 326 that diversification rates change in small steps over the en-  
 327 tire tree. After speciation, each descendant lineage inher-  
 328 its its initial speciation rate  $\lambda_i$  from the ending speciation  
 329 rate  $\lambda_a$  of its ancestor through a mechanism that includes  
 330 both a deterministic long-term trend and a stochastic effect.  
 331 Specifically,

$$\log \lambda_i \sim \mathcal{N}(\log(\alpha \lambda_a), \sigma^2).$$

332 The  $\alpha$  parameter determines the long-term trend, and its  
 333 effects are similar to the  $z$  parameter of TDBD and BAMM.  
 334 When  $\alpha < 1$ , that is,  $\log \alpha < 0$ , the speciation rate decreases  
 335 over time, corresponding to  $z < 0$ . The standard deviation  
 336  $\sigma$  determines the noise component. The larger the value,  
 337 the more stochastic fluctuation there will be in speciation  
 338 rates.  
 339

340 There are three different versions of ClaDS, characterized  
 341 by how they model  $\mu$ . In ClaDS0, there is no extinction,  
 342 that is,  $\mu = 0$ . In ClaDS1, there is a constant extinction rate  
 343  $\mu$  throughout the tree. Finally, in ClaDS2, it is the turnover  
 344 rate  $\epsilon = \mu/\lambda$  that is kept constant over the tree. All ClaDS  
 345 models collapse to CRB or CRBD models when  $\alpha = 1$  and  
 346  $\sigma \rightarrow 0$  (Fig. 3). The ClaDS models are implemented in the  
 347 R package RPANDA<sup>25</sup>, using a combination of advanced  
 348 numerical solvers and MCMC simulation<sup>15</sup>.

349 In contrast to previous work, where these models are  
 350 implemented independently in complex software packages,  
 351 We used PPL model descriptions (100 lines of code each)  
 352 to generate efficient and asymptotically correct inference  
 353 machinery for all diversification models described above.  
 354 This machinery relies on sophisticated Monte Carlo algo-  
 355 rithms which, unlike classical MCMC, can also estimate the  
 356 marginal likelihood (the normalization constant of Bayes  
 357 theorem). We then compared the performance of the dif-  
 358 ferent diversification models on empirical data by inferring  
 359 the posterior distribution over the parameters of interest and  
 360 by conducting model comparison based on the marginal  
 361 likelihood (Bayes factors). Specifically, we implemented  
 362 the CRB, CRBD, TDB, TDBD, BAMM, LSBDS, ClaDS0,  
 363 ClaDS1 and ClaDS2 models in WebPPL and Birch. The  
 364 model descriptions are provided at [github.com/phypp1/](https://github.com/phypp1/probabilistic-programming)  
 365 [probabilistic-programming](https://github.com/phypp1/probabilistic-programming). They are similar in struc-

366 ture to the CRBD program presented above.

367 **Inference strategies.** We used inference algorithms in the  
368 SMC family, an option available in both WebPPL and Birch.  
369 An SMC algorithm runs many simulations (called particles)  
370 in parallel, and stops them when some new information, like  
371 the time of a speciation event or extinction of a side lineage,  
372 becomes available. At such points, the particles are sub-  
373 jected to *resampling*, that is, sampling (with replacement)  
374 based on their likelihoods. SMC algorithms work particu-  
375 larly well when the model can be written such that the  
376 information derived from observed data can successively be  
377 brought to bear on the likelihood of a particle during the  
378 simulation. This is the case when simulating a diversifica-  
379 tion process along a tree of extant taxa, because we know  
380 that each ‘hidden’ speciation event must eventually result in  
381 extinction of the unobserved side lineage. That is, we can  
382 condition the simulation on extinction of the side branches  
383 that arise (Supplementary Listing 3). Similarly, we can con-  
384 dition the simulation on the times of the speciation events  
385 leading to extant taxa.

386 Despite this, standard SMC (the bootstrap particle filter)  
387 remains relatively inefficient for these models. Therefore,  
388 we employed three new PPL inference techniques that we  
389 developed or extended as part of this study: alignment<sup>26</sup>,  
390 delayed sampling<sup>13</sup> and the alive particle filter<sup>27</sup> (see Meth-  
391 ods).

392 **Empirical results.** To demonstrate the power of the ap-  
393 proach, we applied PPLs to compare the performance of  
394 the nine diversification models discussed above for 40 bird  
395 clades (see Methods and Supplementary Table 5). The re-  
396 sults (Supplementary Figs. 12–21) are well summarized by  
397 the four cases represented in Fig. 4. Focusing on marginal  
398 likelihoods (top row), we observe that the simplest mod-  
399 els (CRB, CRBD), without any variation through time or  
400 between lineages, provide an adequate description of the di-  
401 versification process for around 40% of the trees (Fig. 4a).  
402 In the remaining clades, there is almost universal support for  
403 slowing diversification rates over time. Occasionally, this  
404 is not accompanied by strong evidence for lineage-specific  
405 effects (Fig. 4b) but usually it is (Figs. 4c and d). In the  
406 latter case, the ClaDS models always show higher marginal  
407 likelihoods than BAMM and LSBDS, and this even for trees  
408 on which the latter do detect rate shifts (Fig. 4d). Interest-  
409 ingly, ClaDS2 rarely outperforms ClaDS0, which assumes  
410 no extinction. More generally, models assuming no ex-  
411 tinction often have a higher marginal likelihood than their  
412 counterparts allowing for it.

413 The parameter estimates (Fig. 4, rows 2–6) show the con-  
414 servative nature of the Bayes factor tests, driven by the re-  
415 latively vague priors we chose on the additional parameters  
416 of the more complex models (Supplementary Fig. 2). How-  
417 ever, even when complex models are marginally worse than  
418 simple or no-extinction models, there is evidence of the kind  
419 of variation they allow. For instance, the posterior distribu-  
420 tions on  $z$  and  $\log \alpha$  suggest that negative time-dependence  
421 is quite generally present. Similarly, more sophisticated  
422 models usually detect low levels of extinction when they are  
423 outperformed by extinction-free counterparts. For a more

extensive discussion of these and other results, see Supple-  
mentary Section 9.

## 426 Discussion

427 Universal PPLs provide Turing-complete languages for  
428 model descriptions, which guarantees that virtually all inter-  
429 esting phylogenetic models can be expressed. The expres-  
430 siveness of PPLs is liberating for empiricists but it forces  
431 statisticians and computer scientists to approach the infer-  
432 ence problem from a more abstract perspective. This can  
433 be challenging but also rewarding, as inference techniques  
434 for PPLs are so broadly applicable. Importantly, express-  
435 ing phylogenetic models as PPLs opens up the possibility  
436 to apply a wide range of inference strategies developed for  
437 scientific problems with no direct relation to phylogenetics.  
438 Another benefit is that PPLs reduce the amount of manually  
439 written code for a particular inference problem, facilitating  
440 the task and minimizing the risk of inadvertently introducing  
441 errors, biases or inaccuracies. Our verification experiments  
442 (Supplementary Section 7) suggest that the light-weight PPL  
443 implementations of ClaDS1 and ClaDS2 provide more accu-  
444 rate computation of likelihoods than the thousands of lines  
445 of code developed originally for these models.

446 Previous discussion on the relative merits of diversifica-  
447 tion models have centered around the results of simulations  
448 and arguments over biological realism<sup>17,18,29,15,16</sup>, and it has  
449 been complicated by the lack of asymptotically correct infer-  
450 ence machinery for BAMM<sup>18,29</sup>. Our most important con-  
451 tribution in this context is the refinement of PPL techniques  
452 so that it is now possible to implement correct and efficient  
453 parameter inference under a wide range of diversification  
454 models, and to compare their performance on real data us-  
455 ing rigorous model testing procedures. The PPL analyses of  
456 bird clades confirm previous claims that the ClaDS models  
457 provide a better description of lineage-specific diversifica-  
458 tion than BAMM<sup>15</sup>. Even when simpler models have higher  
459 likelihoods, the ClaDS models seem to pick up a consistent  
460 signal across clades of small, gradual changes in diversifi-  
461 cation rates. Like many previous studies<sup>30</sup>, our analyses  
462 provide little or no support for extinction rates above zero.  
463 This appears to be due in part to systematic biases in the  
464 sampling of the leaves in the observed trees<sup>31,32</sup>, a problem  
465 that could be addressed by extending our PPL model scripts  
466 (Supplementary Section 9.6). Such sampling biases may  
467 also partly explain the strong support for slowing diversifi-  
468 cation rates<sup>23</sup>. A fascinating question that is now open  
469 to investigation is whether there remains evidence of oc-  
470 casional major shifts in diversification rates once the small  
471 gradual changes have been accounted for, something that  
472 could be addressed by a model that combines ClaDS- and  
473 BAMM-like features.

474 Our results show that PPLs can already now compete suc-  
475 cessfully with dedicated special-purpose software in several  
476 phylogenetic problem domains. Separately, we show how  
477 PPLs can be applied to models where diversification rates  
478 are dependent on observable traits of organisms (so-called  
479 state-dependent speciation and extinction models)<sup>27</sup>. Other  
480 problem domains that may benefit from the PPL approach  
481 already at this point include epidemiology<sup>33</sup>, host-parasite

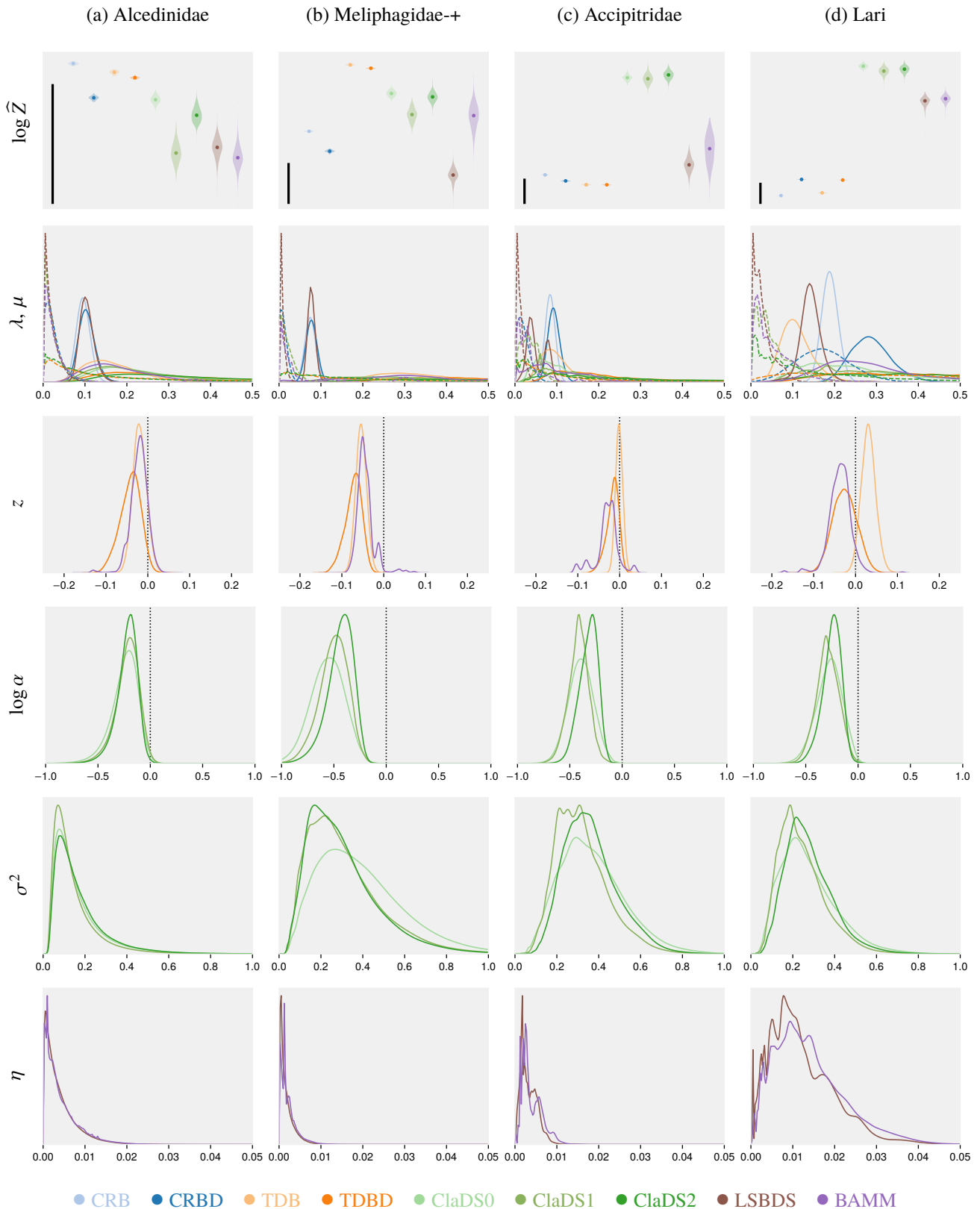


Figure 4: Comparison of diversification models for four bird clades exemplifying different patterns. **Alcedinidae**: simple models are adequate; **Meliphagidae+**: slowing diversification but no lineage-specific effects; **Accipitridae**: gradual (ClaDS) lineage-specific changes in diversification; and **Lari**: evidence for both gradual (ClaDS) and for punctuated (BAMM and LSBDS) lineage-specific changes in diversification. The upper plots show the marginal likelihoods (log scale); a difference of 5 units (scale bar) is considered strong evidence in favor of the better model<sup>28</sup>. The remaining plots show estimated posterior distributions of model parameters. The  $\mu$  distributions are shown with dashed lines.

482 co-evolution<sup>34</sup>, and biogeography<sup>35,36,37,38</sup>.

483 What is missing before it becomes possible to generate ef-  
484 ficient inference machinery for the full range of phylogenetic  
485 models from PPL descriptions? Assume, for instance, that  
486 we would like to do joint inference of phylogeny (from DNA  
487 sequence data) and diversification processes, instead of as-  
488 suming that the extant tree is observed; this would seem to  
489 touch on all the major obstacles that remain. We then need to  
490 extend our current PPL models so that they also describe the  
491 nucleotide substitution process along the tree, and condition  
492 the simulation on the observed sequences. To generate the  
493 standard MCMC machinery for sampling across trees from  
494 such descriptions, delayed sampling needs to be extended  
495 to summarize over ancestral sequences (Felsenstein's prun-  
496 ing algorithm)<sup>39</sup>, and it should be applied statically through  
497 analysis of the script before the MCMC starts rather than dy-  
498 namically. State-of-the-art MCMC algorithms for PPLs<sup>12</sup>  
499 must then be extended to generate computationally efficient  
500 tree samplers, such as stochastic nearest neighbour inter-  
501 change<sup>40</sup>. To facilitate use of PPLs, we think it will also be  
502 important to provide a domain-specific PPL that is easy to  
503 use, while supporting both automatic state-of-the-art infer-  
504 ence algorithms for phylogenetic problems as well as man-  
505 ual composition of novel inference strategies suited for this  
506 application domain. These all seem to be tractable prob-  
507 lems, which we aim to address within the TreePPL project  
508 ([treeppl.org](http://treeppl.org)). We hope this paper will inspire readers to  
509 explore PPLs, and we invite computational biologists to join  
510 us in developing languages and inference strategies support-  
511 ing this powerful new approach to statistical phylogenetics.

## 512 Methods

513 **PPL software and model scripts.** All PPL analyses de-  
514 scribed here used WebPPL version 0.9.15, Node version  
515 12.13.1<sup>9</sup> and the most recent development version of Birch  
516 (as of June 12, 2020)<sup>14</sup>. We implemented all models (CRB,  
517 CRBD, TDB, TDBD, ClaDS0, ClaDS1, ClaDS2, LSBDS  
518 and BAMM) as explicit simulation scripts that follow the  
519 structure of the CRBD example discussed in the main text  
520 (Supplementary Section 5). We also implemented compact  
521 simulations for the four simplest models (CRB, CRBD, TDB  
522 and TDBD) using the analytical equations for specific val-  
523 ues of  $\lambda$ ,  $\mu$  and  $z$  to compute the probability of the observed  
524 trees.

525 In the PPL model descriptions, we account for incom-  
526 plete sampling of the tips in the phylogeny based on the  
527  $\rho$ -sampling model<sup>41</sup>. That is, each tip is assumed to be  
528 sampled with a probability  $\rho$ , which is specified a priori. To  
529 simplify the presentation in this paper, we always set  $\rho = 1$ .  
530 Arguably, this is the relevant setting for the empirical anal-  
531 yses, as the selected trees comprise all or nearly all extant  
532 species.

533 We standardized prior distributions across models to fa-  
534 cilitate model comparisons (Supplementary Section 4, Fig.  
535 2). To simplify the scripts, we simulated outcomes on or-  
536 dered but unlabeled trees, and reweighted the particles so  
537 that the generated density was correct for labelled and un-  
538 ordered trees (Supplementary Section 3.2). We also de-  
539 veloped an efficient simulation procedure to correct for sur-

540 vivorship bias, that is, the fact that we can only observe trees  
541 that survive until the present (Supplementary Section 5.3).

**Inference strategies.** To make SMC algorithms more ef-  
542 ficient on diversification model scripts, we applied three  
543 new PPL inference techniques: alignment, delayed sam-  
544 pling, and the alive particle filter. *Alignment*<sup>26,42</sup> refers to  
545 the synchronization of resampling points across simulations  
546 (particles) in the SMC algorithm. The SMC algorithms  
547 previously used for PPLs automatically resample particles  
548 when they reach **observe** or **condition** statements. Diver-  
549 sification simulation scripts will have different numbers and  
550 placements of hidden speciation events on the surviving tree  
551 (Fig. 2), each associated with a **condition** statement in a  
552 naive script. Therefore, when particles are compared at re-  
553 sampling points, some may have processed a much larger  
554 part of the observed tree than others. Intuitively, one would  
555 expect the algorithm to perform better if the resampling  
556 points were aligned, such that the particles have processed  
557 the same portion of the tree when they are compared. This  
558 is indeed the case; alignment is particularly important for  
559 efficient inference on large trees (Supplementary Fig. 3).  
560 Alignment at code branching points (corresponding to ob-  
561 served speciation events in the diversification model scripts)  
562 can be generated automatically through static analysis of  
563 model scripts<sup>26</sup>. Here, we manually aligned the scripts by  
564 replacing the statements that normally trigger resampling  
565 with code that accumulate probabilities when they did not  
566 occur at the desired locations in the simulation (Supplemen-  
567 tary Section 6.1).

*Delayed sampling*<sup>13</sup> is a technique that uses conjugacy to  
568 avoid sampling parameter values. For instance, the gamma  
569 distribution we used for  $\lambda$  and  $\mu$  is a conjugate prior to  
570 the Poisson distribution, describing the number of births  
571 or deaths expected to occur in a given time period. This  
572 means that we can marginalize out the rate, and simulate  
573 the number of events directly from its marginal (gamma-  
574 Poisson) distribution, without having to first draw a specific  
575 value of  $\lambda$  or  $\mu$ . In this way, a single particle can cover a  
576 portion of parameter space, rather than just single values of  
577  $\lambda$  and  $\mu$ . Delayed sampling is only available in Birch; we  
578 extended it to cover all conjugacy relations relevant for the  
579 diversification models examined here.

The *alive particle filter*<sup>27</sup> is a technique for improving  
582 SMC algorithms when some particles can 'die' because  
583 their likelihood becomes zero. This happens when SMC is  
584 applied to diversification models because simulations that  
585 generate hidden side branches surviving to the present need  
586 to be discarded. The alive particle filter is a generic im-  
587 provement on SMC, and it collapses to standard SMC with  
588 negligible overhead when no particles die. This improved  
589 version of SMC, inspired by state-dependent speciation-  
590 extinction models<sup>27</sup>, is only available in Birch.

**Verification.** To verify that the model scripts and the au-  
592 tomatically generated inference algorithms are correct, we  
593 performed a series of tests focusing on the normalization  
594 constant (Supplementary Section 7). First, we checked that  
595 the model scripts for simple models (CRB(D) and TDB(D))  
596 generated normalization constant estimates that were con-  
597

598 sistent with analytically computed likelihoods for specific  
599 model parameter values (Supplementary Fig. 4). Second,  
600 we used the fact that all advanced diversification models  
601 (ClaDS0-2, LSBDS, BMM) collapse to the CRBD model  
602 under specific conditions, and verified that we obtained the  
603 correct likelihoods for a range of parameter values (Supple-  
604 mentary Fig. 5). Third, we verified for the advanced models  
605 that the independently implemented model scripts and the  
606 inference algorithms generated for them by WebPPL and  
607 Birch, respectively, estimated the same normalization con-  
608 stant for a range of model parameter values (Supplementary  
609 Fig. 6). Fourth, we checked that our normalization constant  
610 estimates were consistent with the RPANDA package<sup>25,15</sup>  
611 for ClaDS0, ClaDS1, and ClaDS2, and with RevBayes for  
612 LSBDS<sup>5,16</sup>. For these tests, we had to develop special-  
613 ized PPL scripts emulating the likelihood computations of  
614 RPANDA and RevBayes. The normalization constant esti-  
615 mates matched for LSBDS (Supplementary Fig. 8) and for  
616 ClaDS0 (Supplementary Fig. 7) but not for ClaDS1 and  
617 ClaDS2. Our best-effort interpretation at this point is that  
618 the PPL estimates for ClaDS1 and ClaDS2 are more ac-  
619 curate than those obtained from RPANDA (Supplementary  
620 Section 7.4). Finally, as there is no independent software  
621 that computes BMM likelihoods correctly yet, we checked  
622 that our BMM scripts gave the same normalization con-  
623 stant estimates as LSBDS under settings where the former  
624 collapses to the latter (Supplementary Fig. 9).

625 **Data.** We applied our PPL scripts to 40 bird clades derived  
626 from a previous analysis of divergence times and relation-  
627 ships among all bird species<sup>43</sup>. The selected clades are  
628 those with more than 50 species (range 54–316) after out-  
629 groups had been excluded (Supplementary Table 5). We  
630 followed the previous ClaDS2 analysis of these clades<sup>15</sup> in  
631 converting the time scale of the source trees to absolute time  
632 units. The clade ages range from 12.5 Ma to 66.6 Ma.

633 **Bayesian inference.** Based on JavaScript, WebPPL is  
634 comparatively slow, making it less useful for high-precision  
635 computation of normalization constants or estimation of  
636 posterior probability distributions using many particles.  
637 WebPPL is also less efficient than Birch because it does  
638 not yet support delayed sampling and the alive particle fil-  
639 ter. Delayed sampling, in particular, substantially improves  
640 the quality of the posterior estimates obtained with a given  
641 number of particles. Therefore, we focused on Birch in  
642 computing normalization constants and posterior estimates  
643 for the bird clades.

644 For each tree, we ran the programs implementing the  
645 ClaDS, BMM and LSBDS models using SMC with de-  
646 layed sampling and the alive particle filter as the inference  
647 method. We used 5000 particles for all models except  
648 BMM, for which we increased the number of particles  
649 to 20000. We ran each program 500 times and collected the  
650 estimates of  $\log \hat{Z}$  from each run together with the informa-  
651 tion needed to estimate the posterior distributions.

652 For CRB, CRBD, TDB and TDBD we exploited the  
653 closed form for the likelihood in the programs. We used  
654 sequential importance sampling with 10,000 particles as  
655 the inference method, and ran each program 50 times.

**Visualization.** Visualizations were prepared with Mat-  
plotlib<sup>44</sup>. We used the collected data from all runs to draw  
violin plots for  $\log \hat{Z}$  as well as the posterior distributions  
for  $\lambda$ ,  $\mu$  (for all models),  $z$  (for TDB, TDBD and BMM),  
 $\log \alpha$  and  $\sigma^2$  (for the ClaDS models), and  $\eta$  (for LSBDS  
and BMM). By virtue of delayed sampling, the posterior  
distributions for  $\lambda$  and  $\mu$  for all ClaDS models as well as  
BMM and LSBDS were calculated as mixtures of gamma  
distributions, the posterior distribution for  $\log \alpha$  and  $\sigma^2$   
for all ClaDS models as mixtures of normal inverse gamma  
and inverse gamma distributions, and the posterior distribution  
for  $\eta$  for BMM and LSBDS as a mixture of gamma dis-  
tributions. For the remaining model parameters, we used  
the kernel density estimation (KDE) method. Exact plot  
settings are provided in the code repository accompanying  
the paper.

**Reporting Summary** Further information on research de-  
sign is available in the Nature Research Reporting Summary  
linked to this article.

## Data availability

The data used to compare the diversification mod-  
els, together with full literature references, are  
available from <https://github.com/phypppl/probabilistic-programming/data>.

## Code availability

The WebPPL models are available from <https://github.com/phypppl/probabilistic-programming/webppl>  
and the Birch models from <https://github.com/phypppl/probabilistic-programming/birch>.

## References

1. Felsenstein, J. *Inferring Phylogenies* (Sinauer Associates, Sunderland, Massachusetts, 2003).
2. Yang, Z. *Molecular Evolution: A Statistical Approach* (Oxford University Press, Oxford, United Kingdom ; New York, NY, United States of America, 2014), 1 edition edition.
3. Nascimento, F. F., dos Reis, M. & Yang, Z. A biologist's guide to Bayesian phylogenetic analysis. *Nature Ecology & Evolution* **1**, 1446–1454 (2017).
4. Höhna, S. et al. Probabilistic graphical model representation in phylogenetics. *Systematic Biology* **63**, 753–771 (2014).
5. Höhna, S. et al. RevBayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. *Systematic Biology* **65**, 726–736 (2016).
6. Fourment, M. & Darling, A. E. Evaluating probabilistic programming and fast variational Bayesian inference in phylogenetics. *PeerJ* **7**, e2722 (2019).



- 705 7. Bouchard-Côté, A. et al. Blang: Bayesian declarative modelling of arbitrary data structures. Preprint at <https://arxiv.org/abs/1912.10396> (2019). 756
- 706 8. Kozen, D. Semantics of probabilistic programs. In *20th Annual Symposium on Foundations of Computer Science*, pages 101–114 (San Juan, Puerto Rico, USA, 1979). 757
- 707 9. Goodman, N. D. & Stuhlmüller, A. The design and implementation of probabilistic programming languages. <http://dippl.org> (2014). Accessed: 2020-5-12. 758
- 708 10. Wood, F., Meent, J. W. & Mansinghka, V. A new approach to probabilistic programming inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 1024–1032 (Reykjavik, Iceland, 2014). 759
- 709 11. Mansinghka, V., Selsam, D. & Perov, Y. Venture: a higher-order probabilistic programming platform with programmable inference. Preprint at <https://arxiv.org/abs/1404.0099> (2014). 760
- 710 12. Ritchie, D., Stuhlmüller, A. & Goodman, N. C3: Lightweight incrementalized MCMC for probabilistic programs using continuations and callsite caching. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 28–37 (Cadiz, Spain, 2016). 761
- 711 13. Murray, L. M., Lundén, D., Kudlicka, J., Broman, D. & Schön, T. B. Delayed sampling and automatic Rao–Blackwellization of probabilistic programs. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, volume 21, page 10 (Lanzarote, 2018). 762
- 712 14. Murray, L. M. & Schön, T. B. Automated learning with a probabilistic programming language: Birch. *Annual Reviews in Control* **46**, 29–43 (2018). 763
- 713 15. Maliet, O., Hartig, F. & Morlon, H. A model with many small shifts for estimating species-specific diversification rates. *Nature Ecology & Evolution* **3**, 1086–1092 (2019). 764
- 714 16. Höhna, S. et al. A Bayesian approach for estimating branch-specific speciation and extinction rates. Preprint at <https://www.biorxiv.org/content/10.1101/555805v1> (2019). 765
- 715 17. Rabosky, D. L. Automatic detection of key innovations, rate shifts, and diversity-dependence on phylogenetic trees. *PLoS ONE* **9**, e89543 (2014). 766
- 716 18. Moore, B. R., Höhna, S., May, M. R., Rannala, B. & Huelsenbeck, J. P. Critically evaluating the theory and performance of Bayesian analysis of macroevolutionary mixtures. *Proceedings of the National Academy of Sciences of the United States of America* **113**, 9569–9574 (2016). 767
- 717 19. Yule, G. U. A mathematical theory of evolution, based on the conclusions of Dr. JC Willis, FRS. *Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of a Biological Character* **213**, 21–87 (1924). 768
- 718 20. Nee, S. Birth-death models in macroevolution. *Annual Review of Ecology, Evolution and Systematics* **37**, 1–17 (2006). 769
- 719 21. Feller, W. Die Grundlagen der Volterraschen Theorie des Kampfes ums Dasein in wahrscheinlichkeitstheoretischer Behandlung. *Acta Biotheoretica* **5**, 11–40 (1939). 770
- 720 22. Kendall, D. G. On the generalized "birth-and-death" process. *The Annals of Mathematical Statistics* **19**, 1–15 (1948). 771
- 721 23. Moen, D. & Morlon, H. Why does diversification slow down? *Trends in Ecology & Evolution* **29**, 190–197 (2014). 772
- 722 24. Rabosky, D. L. et al. BAMMtools: an R package for the analysis of evolutionary dynamics on phylogenetic trees. *Methods in Ecology and Evolution* **5**, 701–707 (2014). 773
- 723 25. Morlon, H. et al. RPANDA: an R package for macroevolutionary analyses on phylogenetic trees. *Methods in Ecology and Evolution* **7**, 589–597 (2016). 774
- 724 26. Lundén, D., Broman, D., Ronquist, F. & Murray, L. M. Automatic alignment of sequential Monte Carlo inference in higher-order probabilistic programs. Preprint at <https://arxiv.org/abs/1812.07439> (2018). 775
- 725 27. Kudlicka, J., Murray, L. M., Ronquist, F. & Schön, T. B. Probabilistic programming for birth-death models of evolution using an alive particle filter with delayed sampling. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence 2019*, volume 2019, page 11 (Tel Aviv, Israel, 2019). 776
- 726 28. Jeffreys, H. *The Theory of Probability* (Oxford University Press, Oxford, 1961). 777
- 727 29. Rabosky, D. L., Mitchell, J. S. & Chang, J. Is BAMM flawed? Theoretical and practical concerns in the analysis of multi-rate diversification models. *Systematic biology* **66**, 477–498 (2017). 778
- 728 30. Pyron, R. A. & Burbrink, F. T. Phylogenetic estimates of speciation and extinction rates for testing ecological and evolutionary hypotheses. *Trends in Ecology & Evolution* **28**, 729–736 (2013). 779
- 729 31. Höhna, S., Stadler, T., Ronquist, F. & Britton, T. Inferring speciation and extinction rates under different sampling schemes. *Molecular Biology and Evolution* **28**, 2577–2589 (2011). 780
- 730 32. Rosindell, J., Cornell, S. J., Hubbell, S. P. & Etienne, R. S. Protracted speciation revitalizes the neutral theory of biodiversity: Protracted speciation and neutral theory. *Ecology Letters* **13**, 716–727 (2010). 781
- 731 782
- 732 783
- 733 784
- 734 785
- 735 786
- 736 787
- 737 788
- 738 789
- 739 790
- 740 791
- 741 792
- 742 793
- 743 794
- 744 795
- 745 796
- 746 797
- 747 798
- 748 799
- 749 800
- 750 801
- 751 802
- 752 803
- 753 804
- 754 805
- 755 806
- 807
- 808

- 809 33. Baele, G., Dellicour, S., Suchard, M. A., Lemey, P. & 859  
810 Vrancken, B. Recent advances in computational phy- 860  
811 lodynamics. *Current Opinion in Virology* **31**, 24–32 861  
812 (2018). 862
- 813 34. Braga, M. P., Landis, M. J., Nylin, S., Janz, N. & Ron- 863  
814 quist, F. Bayesian inference of ancestral host-parasite 864  
815 interactions under a phylogenetic model of host reper- 865  
816 toire evolution. *Systematic Biology* **67**, 000–000 (Ad- 866  
817 vance access) (2020).
- 818 35. Ronquist, F. & Sanmartín, I. Phylogenetic methods in 867  
819 biogeography. *Annual Review of Ecology, Evolution, 868*  
820 *and Systematics* **42**, 441–464 (2011). 869
- 821 36. Matzke, N. J. Model selection in historical biogeog- 870  
822 raphy reveals that founder-event speciation is a crucial 871  
823 process in island clades. *Systematic Biology* **63**, 951– 872  
824 970 (2014). 873
- 825 37. Landis, M. J., Matzke, N. J., Moore, B. R. & Huelsen- 874  
826 beck, J. P. Bayesian analysis of biogeography when 875  
827 the number of areas is large. *Systematic Biology* **62**, 876  
828 789–804 (2013).
- 829 38. Ree, R. H. & Sanmartín, I. Conceptual and statisti- 877  
830 cal problems with the DEC+J model of founder-event 878  
831 speciation and its comparison with DEC via model se- 879  
832 lection. *Journal of Biogeography* **45**, 741–749 (2018).
- 833 39. Felsenstein, J. Evolutionary trees from DNA sequences: 880  
834 a maximum likelihood approach. *Journal of molecular 881*  
835 *evolution* **17**, 368–376 (1981). 882
- 836 40. Lakner, C., van der Mark, P., Huelsenbeck, J. P., Larget, 883  
837 B. & Ronquist, F. Efficiency of Markov chain Monte 884  
838 Carlo tree proposals in Bayesian phylogenetics. *Sys- 885*  
839 *tematic Biology* **57**, 86–103 (2008).
- 840 41. Stadler, T. On incomplete sampling under birth-death 886  
841 models and connections to the sampling-based coales- 887  
842 cent. *Journal of Theoretical Biology* **261**, 58–66 (2009).
- 843 42. Lundén, D., Borgström, J. & Broman, D. Correctness 888  
844 of sequential Monte Carlo inference for probabilistic 889  
845 programming languages. Preprint at [https://arxiv.](https://arxiv.org/abs/2003.05191)  
846 [org/abs/2003.05191](https://arxiv.org/abs/2003.05191) (2020).
- 847 43. Jetz, W., Thomas, G. H., Joy, J. B., Hartmann, K. & 890  
848 Mooers, A. O. The global diversity of birds in space 891  
849 and time. *Nature* **491**, 444–448 (2012).
- 850 44. Hunter, J. D. Matplotlib: A 2d graphics environment. 892  
851 *Computing in Science & Engineering* **9**, 90–95 (2007).

## 852 Acknowledgements

853 We thank Lars Arvestad and Philippe Veber for their con-  
854 tributions to the early development of the ideas presented  
855 here, and Odile Maliet for extensive help with the RPANDA  
856 implementation of the ClaDS models. This work was sup-  
857 ported by grants from the Swedish Research Council to  
858 J.B. and J.K. (No 2013-4853) and to F.R. (No 2018-04620),

by the Swedish Foundation for Strategic Research (SSF) via  
the project *ASSEMBLE* (contract number: RIT15-0012) to  
T.S., D.B. and J.K., and by the European Union’s Hori-  
zon 2020 research and innovation program under the Marie  
Skłodowska-Curie grant agreement PhyPPL No 898120 to  
V.S. The computations were performed on resources pro-  
vided by the Swedish National Infrastructure for Computing  
(SNIC) at the National Supercomputer Centre (NSC).

## 867 Author contributions

868 F.R. and N.L. initiated the project. All authors contributed  
869 to the further development of concepts and algorithms. F.R.,  
870 J.K. and V.S. implemented algorithms, supported by D.L.,  
871 J.B., L.M., N.L., T.S. and D.B. Verification experiments  
872 and empirical analyses were run by J.K. and V.S., who also  
873 generated most of the illustrations assisted by D.L., F.R. and  
874 J.B. The final manuscript was a joint effort.

## 875 Competing interests

876 The authors declare no competing interests.

## 877 Additional information

878 **Supplementary information** for this paper is  
879 available at [https://github.com/phypppl/](https://github.com/phypppl/probabilistic-programming/supplementary)  
880 [probabilistic-programming/supplementary](https://github.com/phypppl/probabilistic-programming/supplementary).

881 **Correspondence and requests for materials** should be  
882 addressed to F.R., J.K. or V.S.