Chapter 5.6 A Tutorial on the Use of BPP for Species Tree Estimation and Species Delimitation

Tomáš Flouri[1]
Department of Genetics, Evolution and Environment, University College London [London WC1E 6BT, United Kingdom]
t.flouris@ucl.ac.uk
https://orcid.org/0000-0002-8474-9507

Bruce Rannala
Department of Evolution and Ecology, University of California Davis [One Shields Avenue, Davis CA USA]
brannala@ucdavis.edu
https://orcid.org/0000-0002-8355-9955

Ziheng Yang[2]
Department of Genetics, Evolution and Environment, University College London [London WC1E 6BT, United Kingdom]
z.yang@ucl.ac.uk
https://orcid.org/0000-0003-3351-7981

Abstract

BPP is a Bayesian Markov chain Monte Carlo program for analyzing multilocus sequence data under the multispecies coalescent (MSC) model with and without introgression. Among the analyses that can be conducted are estimation of population size and species divergence times, species tree estimation, species delimitation and estimation of cross-species introgression intensity. The program can also be used to simulate gene trees and sequence alignments under the MSC model with, or without, migration. In this tutorial, we illustrate the use of BPP for species tree estimation and species delimitation. We also provide practical guidelines on running BPP on multicore systems. As BPP is continuously updated, the most up-to-date version of this tutorial, as well as the data files, are available at `http://github.com/bpp/tutorial`.

How to cite: Tomáš Flouri, Bruce Rannala, and Ziheng Yang (2020). A Tutorial on the Use of BPP for Species Tree Estimation and Species Delimitation. In Scornavacca, C., Delsuc, F., and Galtier, N., editors, *Phylogenetics in the Genomic Era*, chapter No. 5.6, pp. 5.6:1–5.6:16. No commercial publisher | Authors open access book. The book is freely available at https://hal.inria.fr/PGE.

Supplement Material `https://github.com/bpp/tutorial`

1 Introduction

BPP is a Bayesian Markov Chain Monte Carlo (MCMC) program for analyzing sequence alignments from multiple loci and multiple species under the multispecies coalescent (MSC) model (Rannala and Yang, 2003; Yang, 2002) with and without introgression (Xu and Yang 2016 and Chapter 5.5 [Rannala and Yang 2020]). The program allows four types of analysis,

[1] T.F. is supported by a Biotechnology and Biological Sciences Research Council grant (BB/P006493/1).
[2] Z.Y. is supported by a Biotechnology and Biological Sciences Research Council grant (BB/P006493/1).

Phylogenetics in the genomic era.
Editors: Celine Scornavacca, Frédéric Delsuc and Nicolas Galtier; chapter No. 5.6; pp. 5.6:1–5.6:16
A book completely handled by researchers.
No publisher has been paid.

referred to as A00, A01, A10, and A11, and specified by using two variables in the control file (Yang, 2015; Flouri et al., 2018, table 1). Analysis A00 is a within-model inference and is used to estimate the parameters in the MSC or MSC with introgression (MSci) models, such as the species divergence times (τs), population sizes (θs), and the introgression probability at hybridization/introgression events (φs) (Rannala and Yang, 2003; Burgess and Yang, 2008; Flouri et al., 2019), when the species tree model is given by the user and fixed. The other three analyses are trans-model inferences, in which the Markov chain moves between different models. Analysis A01 is used for species tree inference when the assignments of sequences to species are provided by the user (Yang and Rannala, 2014; Rannala and Yang, 2017). Analysis A10 conducts species delimitation using a user-specified guide tree (Yang and Rannala, 2010), and A11 implements joint species delimitation and species tree inference or unguided species delimitation (Yang and Rannala, 2014; Rannala and Yang, 2017).

The basic parameters in the MSC model include the species divergence times (τs) and population size parameters $\theta = 4N\mu$, where N is the effective population size and μ is the mutation rate per site per generation so that θ is the average proportion of sites having different bases between two sequences sampled at random from the population. Both τs and θs are measured by the expected number of mutations per site. Given a species tree with s species, there are $s - 1$ divergence times and at most $2s - 1$ population size parameters (contemporary populations with only one sequence sampled have no θ parameter). The goal of analysis A00 is to estimate those parameters when the species tree is fixed. Analyses A01, A10, and A11 compare different models (for more information on the MSci model and a review of the MSC model, see Xu and Yang 2016; Flouri et al. 2019; Chapters 5.5 and 5.6 [Rannala and Yang 2020; Flouri et al. 2020]).

The current version (4.2.0) of BPP supports a variety of mutation/substitution models, such as JC69, K80, HKY, F81, F84, T92, TN93, and GTR for DNA sequence data. For simplicity, this tutorial focuses on the analysis of closely related species and uses the JC69 mutation mode.

BPP is written in C and can be compiled to run on the command line on any of the major operating systems including MacOS, Linux, and Windows. A basic knowledge of the command line will be needed. Here we use the Unix command line, and Windows users need to make adjustments accordingly. If you have not used the command line before, please work through one of the following short tutorials first:

- `http://abacus.gene.ucl.ac.uk/software/CommandLine.Windows.pdf`
- `http://abacus.gene.ucl.ac.uk/software/CommandLine.MACosx.pdf`

In this tutorial we begin by describing how to install and execute the BPP program. This is followed by an explanation of the basic format of three input files: the sequence alignment file, the imap file, and the control file. We go through the important variables in the control file to illustrate the specification of the priors and the settings for the MCMC algorithm. We then illustrate the A01 analysis for species tree estimation. Once the species tree is inferred, we will use Analysis A00 to estimate the divergence times (τs) and population sizes (θs) on the fixed species tree. We will show how to combine the MCMC samples from multiple runs to produce a summary of the posterior. In the second part, we will use Analysis A11 to conduct a joint species tree estimation and species delimitation.

1.1 Installation of BPP

BPP is an open-source software available for download on GitHub at `http://github.com/bpp/bpp`. The latest stable executable files for Windows and MacOS can be downloaded

Table 1 The four types of analyses implemented in BPP

speciesdelimitation	speciestree	
	0	1
0	**A00.** Estimation of parameters under the multispecies coalescent model (Yang, 2002; Rannala and Yang, 2003)	**A01.** Inference of species tree when the assignment and delimitation are given (Rannala and Yang, 2017)
1	**A10.** Species delimitation using a fixed guide tree (Yang and Rannala, 2010; Rannala and Yang, 2013)	**A11.** Joint species delimitation and species-tree inference or unguided species delimitation (Yang and Rannala, 2014)

from `http://github.com/bpp/releases`. If you want the most recent pre-release version you will need to have a C compiler and the git program installed on your computer. You can obtain the source code and compile it using the following steps:

```
mkdir software
cd software
git clone http://github.com/bpp/bpp
cd bpp/src
make
```

This creates an executable file called `bpp`, which should be copied into a folder that is included in the `PATH` environment variable. This will allow us to execute BPP without having to type the full path to the executable. For example,

```
mkdir ~/mybpp
cp bpp ~/mybpp
export PATH=$PATH:~/mybpp
```

For detailed instructions on compilation and installation please see the GitHub repository.

To run an analysis of the the example dataset, create a folder called `bpptutorial/data/` and copy the example data files in the folder from either `http://abacus.gene.ucl.ac.uk/ziheng/data.html` or `http://abacus.gene.ucl.ac.uk/ziheng/data/HornedLizardsData.tgz`. Then create another working folder `A00/r1/`, and run the program there. For example,

```
mkdir -p bpptutorial/data
cd bpptutorial/data
wget http://abacus.gene.ucl.ac.uk/ziheng/data/HornedLizardsData.tgz
tar -xvzf HornedLizardsData.tgz
mkdir -p A00/r1
cd A00/r1
bpp --cfile ../../lizards.bpp.A00.ctl
```

2 Input files

A BPP analysis requires three input files: a control file, a multiple sequence alignment file and a imap file. The control file specifies the type of analysis, sets the parameters of the prior distributions and specifies the details of the MCMC run. The sequence alignment file contains aligned sequences for one or more loci, arranged one after another. The Imap file assigns/maps individuals to populations or species.

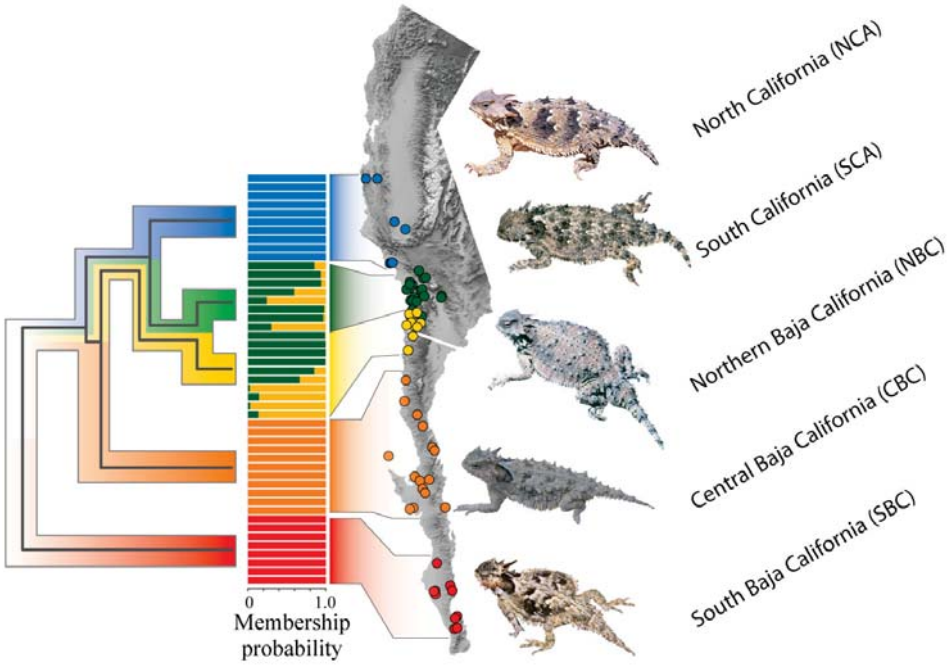

Figure 1 Geographical distributions of Coast Horned Lizards (genus *Phrynosoma*), with five phylogeographic groups arranged latitudinally: North California (NCA), South California (SCA), Northern Baja California (NBC), Central Baja California (CBC), and South Baja California (SBC). Pictures courtesy of Dr Adam Leaché.

2.1 Dataset and multiple sequence alignment file

For this tutorial we will use the Coast Horned Lizard dataset of Leaché et al. (2009), which includes two nuclear loci (*BDNF*: 132 sequences, 529bp; and *RAG-1*: 136 sequences, 1100 bp). This dataset was analyzed by Yang and Rannala (2014) using an earlier version of BPP. Assignment is based on an mtDNA phylogeny, with five phylogeographic groups arranged latitudinally: North California (NCA), South California (SCA), Northern Baja California (NBC), Central Baja California (CBC), and South Baja California (SBC) (Figure 1). Hence, there are five species or populations in the BPP analysis.

The multiple sequence alignment file is a single file that contains the sequence data for all loci in sequential PHYLIP format, with one alignment followed by another. Each sequence alignment (for one locus) is specified using the PHYLIP sequential format, with the first line for each locus specifying the number of sequences and the number of sites in the alignment (see Figure 1, left). Subsequent lines specify the sequences: each line starts with the sequence name followed by two or more whitespaces and then the sequence itself. The sequence name consists of two parts, in the format `x^y` where `x` is a sequence name while `y` is a tag for the specimen or individual. The individual is then assigned to a population or species in the Imap file. A portion of the sequence alignment file and Imap file is shown in Listing 1. Listing 2 depicts the control file we will use for the species tree estimation tutorial.

2.2 Imap file

The Imap file has two columns, separated by white spaces: the first column contains a unique label for each individual and the second column contains the population (or species) name the individual is assigned to. Each individual that occurs in the sequence alignment file must

```
phryno.txt                                          phryno5s.Imap.txt

136 1054                                            1 CBC
BCN_10a^1    ATAAAGGAAAAGCGGCAGCT...                2 CBC
BCN_10b^2    ATAAAGGAAAAGCGGCAGCT...                3 NBC
BCN_11a^3    ATAAAGGAAAAGCGGCAGCT...                4 NBC
BCN_11b^4    ATAAAGGAAAAGCGGCAGCT...                5 NBC
BCN_14a^5    ATAAAGGAAAAGCGGCAGCT...                6 NBC
BCN_14b^6    ATAAAGGAAAAGTGGCAGCT...                7 CBC
...                                                 ...
```

Listing 1 Portions of the sequence data file (`phryno.txt`) and the Imap file (`phryno5s.Imap.txt`) on the left and right, respectively. In the sequence data file each sequence is tagged (1, 2, etc). The part of the sequence name before the caret (^) is read and then ignored. In the Imap file each individual tag is assigned to a population.

be assigned to a population in the Imap file, although the Imap file may include individuals for which no sequence data are available. See Listing 1, right.

3 The control file and BPP settings

See Listing 2 for an example control file. A complete reference of options in the control file is available on the BPP GitHub wiki page at http://github.com/bpp/bpp/wiki. Here, we provide a description of those options relevant to this tutorial. The general format for most options in the control file is: `Option = value(s)`. Text of a line following a symbol '#' or '*' is considered a comment and ignored.

The four types of analysis are specified by setting two binary variables: `speciesdelimitation` and `speciestree`. Those variables take values 0 (meaning disabled) or 1 (enabled). Table 1 illustrates the combination of variables triggering the corresponding analyses.

```
          seed =  -1
       seqfile = ../phryno.txt
      Imapfile = ../phryno5s.Imap.txt
       outfile = out.txt
      mcmcfile = mcmc.txt
* speciesdelimitation = 0 * fixed species tree
* speciesdelimitation = 1 0 2 * delimitation algorithm0 finetune(e)
* speciesdelimitation = 1 1 2 0.5 * delimitation algorithm1 finetune(a m)
          speciestree = 1 * species-tree by SPR

 speciesmodelprior = 1          * 0: uniform labeled histories; 1:uniform rooted trees

  species&tree = 5  NCA SCA NBC CBC SBC
                     18  44  20  34  20
                  ((NCA, ((SCA, NBC), CBC)), SBC);

       usedata = 1     * 0: no data (prior); 1:seq like
         nloci = 2     * number of data sets in seqfile

     cleandata = 0     * remove sites with ambiguity data (1:yes, 0:no)?

    thetaprior = 3 0.004 e * Inv-Gamma(a, b) for theta
      tauprior = 3 0.004   * Inv-Gamma(a, b) for root tau & Dirichlet(a) for other tau's

      * auto (0 or 1): finetune for GBtj, GBspr, theta, tau, mix, locusrate, seqerr
      finetune = 1: .01 .0001 .005 .0005 .2 .01 .01 .01

         print = 1 0 0 0    * MCMC samples, locusrate, hereditytscalars Genetrees
        burnin = 8000
      sampfreq = 2
       nsample = 100000
       threads = 2 1 1
```

Listing 2 Sample control file `lizards.bpp.A01.ctl` for species tree estimation (with `speciesdelimitation = 0` and `speciestree = 1`. Lines starting with an asterisk are comments and the default values of `speciesdelimitation` and `speciestree` are 0.

Option `seed` should be a positive integer and sets the seed for the pseudo-random number generator. Runs with identical seeds analyzing the same data will produce identical results. Setting seed to -1 will cause BPP to use a randomly generated seed (which is recorded in the output file `Seedused`). Option `species&tree` defines the species and the starting species tree, and is typically specified in three lines with the following syntax:

```
species&tree = S  S_1 S_2 ... S_S
                  N_1 N_2 ... N_S
                 NEWICK-TREE
```

If only one population/species is specified, the last line (NEWICK-TREE) must be left empty. In the first line we define the number of species S followed by a list of the species names (S_1 to S_S) separated by whitespaces. The second line comprises S numbers, where number N_i indicates the maximum number of sequences for species S_i at any locus (the actual number of sequences at any locus must be less than or equal to this value). Finally, the last line is the Newick (`https://en.wikipedia.org/wiki/Newick_format`) representation for the starting species tree.

BPP uses a Bayesian model-selection framework to evaluate different models of species phylogenies and species delimitations. Therefore, we specify prior probabilities for the models (species trees) to be compared. The two priors for species trees implemented in BPP are specified using the `speciesmodelprior` keyword (Table 2) with Prior 0 (`speciesmodelprior = 0`) assigning equal probabilities to labeled histories (rooted trees with the internal nodes ordered by age; see Figure 2 and Chapter 5.5 [Rannala and Yang 2020]) and Prior 1 (`speciesmodelprior = 1`) assigning equal probabilities to rooted trees (Yang and Rannala, 2014).

Table 2 Four species tree/species delimitation priors implemented in BPP (using the control variable `speciesmodelprior`)

Prior	Description
0	Assigns equal probabilities to labeled histories (rooted trees with internal nodes ordered by age)
1	Assigns equal probabilities to rooted species trees
2	Assings equal probabilities for the number of delimited species (that is, $1/s$ each for $1, 2, \ldots, s$ delimited species given s populations) and divides up the probability for any specific number of species among the compatible models of species delimitation and species phylogeny in proportion to the number of compatible label histories
3	Same as Prior 2 but instead divides the probability among the compatible models uniformly

Note.— Priors 0 and 1 are used for species tree estimation (analyses A01) and species delimitation on a guide tree (analysis A10), while priors 0–3 are used for joint species delimitation and species tree estimation (analysis A11). This prior has no effect for analysis A00.

For example, there are 15 rooted trees in the case of four species, with 12 unbalanced and 3 balanced trees (Figure 2). Each unbalanced tree, e.g., $(((a, b), c), d)$, is compatible with only one labeled history as there is only one ordering of the internal nodes. Each balanced tree, e.g., $((a, b), (c, d))$, is compatible with two labeled histories, depending on whether the ancestor of a and b is older or younger than the ancestor of c and d. Prior 0 assigns the probability 1/18 to each of the unbalanced trees and 2/18 to each of the balanced trees. Prior 1 assigns the probability 1/15 to each of the 15 rooted trees. Here, we use Prior 1, which is the default.

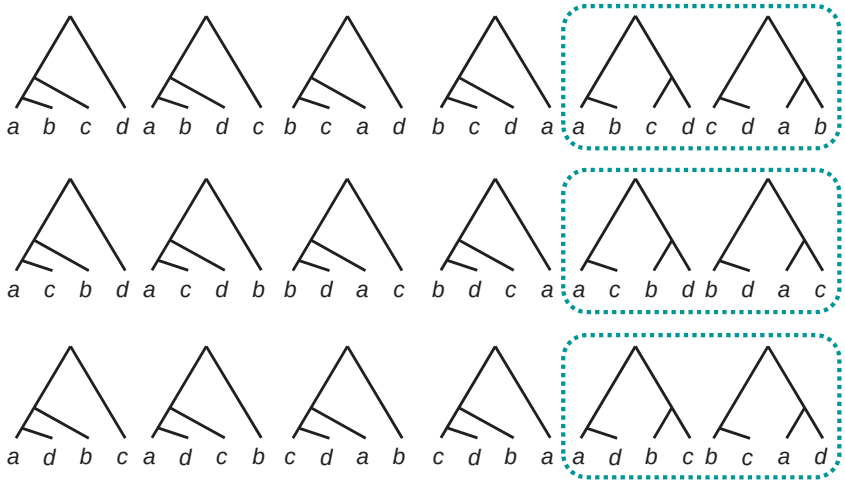

Figure 2 The 18 labeled histories (rooted trees with internal nodes ranked by age) and 15 rooted trees for four tips. Each unbalanced rooted tree is compatible with only one labeled history, but each balanced rooted tree is compatible with two labeled histories.

Within each species tree model, we assign the inverse-gamma priors $\theta \sim \text{IG}(3, 0.004)$ for all θs and $\tau \sim \text{IG}(3, 0.004)$ for the age τ_0 of the root. The inverse-gamma $\text{IG}(a, b)$ has mean $m = b/(a-1)$ if $a > 1$ and variance $s^2 = b^2/[(a-1)^2(a-2)]$ if $a > 2$. If little information is available about the parameters, you can use $a = 3$ for a diffuse prior and then adjust b so that the mean is reasonable. For example, parameter θ measures the genetic diversity (heterozygosity) in the species. This varies among species, with 0.01 (one difference per 100 bps) to be a large value while 0.001 a small value. Parameter τ_0 measures the age of the root in the rooted species tree and depends on the species included in the data set. Thus including an outgroup species will typically mean that a larger prior mean for τ_0 is appropriate.

When specifying the priors, it may be useful to plot the inverse-gamma density and calculate the 95% prior interval. The corresponding R functions are in the MCMCpack, which can be installed with the following command in the R interpreter:

```
install.packages("invgamma");
```

Then we can use the following code in R to plot the inverse-gamma density $\text{IG}(3, 0.004)$ and calculate the 95% prior interval, which is $(0.000554, 0.006465)$:

```
library("invgamma")
a=3; b=0.004;
curve(dinvgamma(x,a,b), from=0, to=0.01)
qinvgamma(c(0.025, 0.975), a, b)
```

Alternatively, a web application may be used for plotting the inverse Gamma (`rdrr.io/cran/bayesAB/man/plotInvGamma.html`). Parameters θ and τ are assigned inverse-gamma priors, using the options `thetaprior` and `tauprior`, respectively. Both options accept two parameters: α and β. In addition, `thetaprior` accepts an optional third parameter - the letter '`e`' (as in estimate). If the third parameter is not specified, BPP integrates out analytically the θ parameters, using conjugate inverse-gamma priors (Hey and Nielsen, 2007). This reduces the state of the Markov chain, resulting in slight improvement in the mixing properties of cross-model MCMC algorithms. The downside is that this approach cannot be used in conjunction with parallelization (see Parallelization).

The number of MCMC iterations is determined by three variables in the control file as: `burnin` + `nsample` × `sampfreq`, where `burnin` is the number of samples that will be discarded (not logged in the sample file) before starting to log a sample every `sampfreq` iterations. The total number of samples logged in the file `mcmc.txt` will be `nsample`.

To assess convergence, run each analysis multiple times (at least twice) using different starting seeds, which are specified in the control file (see Section 2). If the results appear different between runs, re-run the program using a larger number of samples (`nsample`) and/or larger sampling frequency (`sampfreq`). Standard tools are available for diagnosing convergence and mixing problems of MCMC algorithms (pp. 459-510 in Robert and Casella 2005; pp. 226-244 in Yang 2014). However, our experience suggests that running the same analysis multiple times with different seeds and examining the consistency of estimates across runs is the most effective method to guarantee the reliability of the results.

4 Species tree estimation (A01)

We assume the BPP executable resides in a folder which is part of `PATH` (i.e. can be executed without explicitly specifying its path location). We assume that the input files (data, imap and control file) are in a folder named `lizards`. We create two subfolders called `lizards/A01/r1` and `lizards/A01/r2`, and copy the control files to the subfolders,

```
cd lizards
mkdir -p A01/r1 A01/r2
cp lizards.bpp.A01.ctl A01/r1
cp lizards.bpp.A01.ctl A01/r2
```

Note that the `seqfile` (sequence alignment file) and `imapfile` (imap file) variables in the control file (Listing 2) specify that both files are two levels up (e.g., `../../`) in the directory hierarchy relative to the current working directory, while `outfile` and `mcmcfile` specify the current directory (either `A01/r1` or `A01/r2`). Also recall that analysis A01 is triggered by having `speciesdelimitation=0` and `speciestree=1`. We execute each run with the following commands:

```
cd A01/r1
bpp --cfile lizards.bpp.A01.ctl

# Wait until the run is finished

cd ../r2
bpp --cfile lizards.bpp.A01.ctl
cd ../../
```

For the runs we can use the option `threads = 2`. On a laptop computer with a dual-core Intel i7-7500 CPU, one run with two threads takes roughly 10 minutes.

BPP 4 currently uses the subtree pruning and regrafting (SPR) algorithm to search through the space of species tree in the MCMC (Rannala and Yang, 2017). The program collects the sampled species trees (and θs and τs) into the sample file `mcmc.txt`. The summary of the MCMC sample is shown in Listing 3 and the top three species trees (out of a total of 16 in the sample) are illustrated on Figure 3, along with their posterior probabilities. Those three trees have a total posterior probability of 0.96 and therefore consists the 95% credibility set. The majority-rule consensus tree, i.e. the tree built from clades appearing in at least 50% of the trees in the sample, is a binary (resolved) tree and is in line with the maximum a posteriori (map) tree (the tree with highest posterior probability) in the sample.

```
 -3% 0.72 0.31 0.10 0.11 0.41   0.0325   0.0016   0.0014   1467.13645  -2863.66605
Current Pjump:        0.72446   0.31473   0.09694   0.11175   0.40700
Current finetune:     0.01000   0.00010   0.00500   0.00050   0.20000
New finetune:         0.04248   0.00011   0.00151   0.00017   0.29183
 -2% 0.72 0.31 0.25 0.15 0.22   0.0335   0.0015   0.0013   1516.46431  -2860.71283
Current Pjump:        0.72493   0.31210   0.25456   0.14675   0.22400
Current finetune:     0.04248   0.00011   0.00151   0.00017   0.29183
New finetune:         0.18078   0.00011   0.00125   0.00008   0.21028
 -1% 0.73 0.31 0.28 0.28 0.38   0.0220   0.0015   0.0012   1493.60713  -2860.03028

Current Pjump:        0.72533   0.31019   0.27683   0.28025   0.38000
Current finetune:     0.18078   0.00011   0.00125   0.00008   0.21028
New finetune:         0.77068   0.00012   0.00114   0.00007   0.28047
  0% 0.73 0.30 0.29 0.29 0.25   0.0375   0.0015   0.0012   1528.87389  -2859.22283   0:22

Current Pjump:        0.72657   0.29810   0.29267   0.29100   0.25100
Current finetune:     0.77068   0.00012   0.00114   0.00007   0.28047
New finetune:         3.30232   0.00011   0.00111   0.00007   0.22902
  5% 0.72 0.30 0.31 0.28 0.35   0.0313   0.0016   0.0012   1546.63206  -2860.31352   0:53
 10% 0.72 0.30 0.31 0.30 0.35   0.0306   0.0015   0.0012   1489.25672  -2860.20919   1:28
...
 95% 0.72 0.30 0.31 0.29 0.35   0.0358   0.0016   0.0012   1525.98871  -2860.60363   11:25
100% 0.72 0.30 0.31 0.29 0.35   0.0360   0.0016   0.0012   1469.18934  -2860.57826   12:00

12:00 spent in MCMC

Species in order:
   1. NCA
   2. SCA
   3. NBC
   4. CBC
   5. SBC

(A) Best trees in the sample (16 distinct trees in all)
    63037   0.63036   0.63036 ((CBC, ((NBC, SCA), NCA)), SBC);
    19842   0.19842   0.82878 (((CBC, (NBC, SCA)), NCA), SBC);
    12596   0.12596   0.95474 (((CBC, NCA), (NBC, SCA)), SBC);
     2326   0.02326   0.97800 ((CBC, ((NBC, NCA), SCA)), SBC);
     1592   0.01592   0.99392 ((CBC, (NBC, (NCA, SCA))), SBC);
      250   0.00250   0.99642 (((CBC, (NBC, SCA)), SBC), NCA);
      162   0.00162   0.99804 ((CBC, SBC), ((NBC, SCA), NCA));
...

(B) Best splits in the sample of trees (13 splits in all)
 99393 0.993920  11110
 96078 0.960770  01100
 67204 0.672033  11100
 20123 0.201228  01110
 12638 0.126379  10010
  2331 0.023310  10100
  1592 0.015920  11000
...

(C) Majority-rule consensus tree
(((NCA, (SCA, NBC) #0.960770) #0.672033, CBC) #0.993920, SBC);

(D) Best tree (or trees from the mastertree file) with support values
((CBC, ((NBC, SCA) #0.960770, NCA) #0.672033) #0.993920, SBC);   [P = 0.630364]
```

Listing 3 Output from analysis A01 (species tree estimation). The progress indicator is negative during burnin, and BPP goes through four rounds of automatic step-length adjustments, aiming to achieve a near-optimal acceptance proportion of 30% for the parameter moves (Yang and Rodríguez, 2013). Sampling in `mcmc.txt` starts after the burn-in is over. At the end of the MCMC run, the sample is processed to calculate the posterior probabilities of the species trees, which are further summarized to calculate the posterior for splits as well as the majority-rule consensus tree.

In the next step we will run the A00 analysis with the species tree fixed at the MAP tree to estimate the parameters of the MSC model. Using the same control file but with `speciestree=0`, we conduct two runs of the A00 analysis:

```
mkdir -p A00/r1 A00/r2
cp lizards.bpp.A00.ctl A00/r1
cp lizards.bpp.A00.ctl A00/r2
cd A00/r1
bpp --cfile lizards.bpp.A00.ctl

# Wait until the run is finished

cd ../r2
bpp --cfile lizards.bpp.A00.ctl
cd ../
```

We can combine the output of the two runs to increase the number of samples from the posterior distribution and summarize the new concatenated sample independently. To do that we create a new folder and copy the MCMC sample file of the first run, and then concatenate the samples from the second run (note the `tail -n +2` command which skips the header line from the mcmc sample file):

```
mkdir combined; cd combined
cp ../r1/lizards.bpp.A00.ctl .
cp ../r1/mcmc.txt .
tail -n +2 ../r2/mcmc.txt >> mcmc.txt

# !IMPORTANT! Edit print line in control file to read print=-1
bpp --cfile lizards.bpp.A00.ctl
```

Lastly, we must change the line `print = 1 0 0 0 0` in the control file to `print = -1`. This causes BPP to only read and summarize the specified MCMC sample file rather than running a new MCMC analysis. We again run BPP and the posterior means are shown in Figure 4 along with the 95% credible interval for divergence times for each internal node.

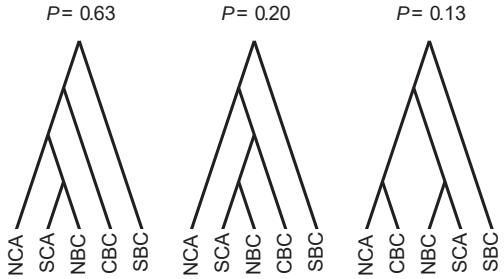

Figure 3 The top three species trees in the 95% CI and their posterior probabilities, with a total probability of 0.96

5 Species delimitation (A11)

In Analysis A11, both the species delimitation model and the species phylogeny are changing in the MCMC. We change the variables in the control file to have `speciesdelimitation = 1` and `speciestree = 1`, create the necessary directory structure and re-run BPP in the

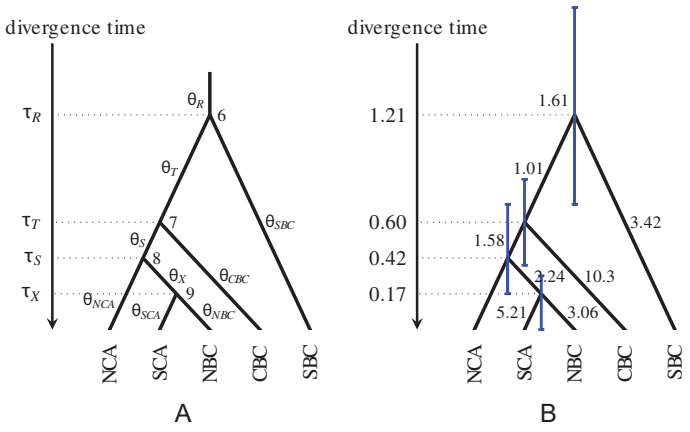

Figure 4 A species tree for five horned lizard species/populations NCA, SCA, NBC, CBC and SBC, illustrating the parameters in the multispecies coalescent model. Those include four species divergence time parameters (τ) for the three ancestral nodes, 6 (NCA, SCA, NBC, CBC, SBC), 7 (NCA, SCA, NBC, CBC), 8 (NCA, SCA, NBC) and 9 (SCA, NBC), and nine population size parameters (θs) for the nine populations on the tree. Estimates were multiplied by 10^3.

same way as in the A01 analysis. The control file differs from the A01 file only in the value of the `speciesdelimitation` variable.

```
cd lizards
mkdir -p A11/r1 A11/r2
cp lizards.bpp.A11.ctl A11/r1
cp lizards.bpp.A11.ctl A11/r2

cd A11/r1
bpp --cfile lizards.bpp.A11.ctl
cd ../r2
bpp --cfile lizards.bpp.A11.ctl
```

The algorithm explores different species delimitation models and different species phylogenies. The assignment to populations is nevertheless fixed; that is, the program attempts to merge different populations into one species but never tries to split one population into multiple species. The SPR algorithm is used to change the species tree topology (Rannala and Yang, 2017), while a reversible-jump MCMC (rjMCMC) algorithm is used to move between different species delimitation models, by either splitting one species into two or joining two species into one species (Yang and Rannala, 2010). Two alternative rjMCMC algorithms are implemented in BPP, which differ in the way that new θ parameters are proposed during the split move. They are specified through the `speciesdelimitation` option which takes one of two formats:

```
speciesdelimitation = 1 0 ε # Algorithm 0
speciesdelimitation = 1 1 a m # Algorithm 1
```

The second digit (0 or 1) distinguishes between the two rjMCMC algorithms. For Algorithm 0, we use a value of $\varepsilon = 2$ in equations 3 and 4 of Yang and Rannala (2010). Reasonable values for ε are 1, 2, 5, etc. For Algorithm 1, we set $a = 2$ and $m = 1$ in equations 6 and 7 of Yang and Rannala (2010). Reasonable values are $a = 1, 1.5, 2$ etc., and $m = 0.5, 1, 2$ etc. When the chain mixes well, the results should be the same between multiple runs and between the two algorithms.

BPP offers four priors on delimitation models for Analysis A11, specified using the variable `speciesmodelprior`, which takes the values 0, 1, 2, or 3, with Prior 1 being the default. These are outlined in Table 2. Prior 3 may be suitable when there is a large number of populations. One such scenario is when each sequence (specimen) is assigned into its own "population", so that BPP will explore different models of assignment, species delimitation and species tree estimation (Olave et al., 2014). For this tutorial we use the default Prior 1 (for more information on Priors 2 and 3 see Yang, 2015).

The runs should take 10 to 15 minutes on a modern laptop when using 2 threads. The program collects the sampled species trees (and θs and τs) into the sample file `mcmc.txt`, as well as the number of delimited species. The branch lengths of the species tree are used to distinguish collapsed populations. Once the analyses finish, we check that the runs have converged by comparing the list of best models and the posteriors on the number of species. The summary of one of the runs is shown on Listing 4. The posterior probability of 5 species (NCA, SCA, NBC, CBC, SBC) is 0.93, while that of four species (with NBC and SCA joined into one species) is 0.07. The results regarding the phylogenetic relationships among the delimited species are consistent with the results of the A01 method, with the 99% credibility set including four distinct species tree topologies, with a posterior probability of 0.58 for the best tree. The data seem to contain more information about species delimitation than about species phylogeny.

```
 -3% 0.73 0.29 0.10 0.04 0.46    4 10 0.0010 0.0000 \
 P(3)=0.4745 0.0015   0.0016   1304.39469 -2863.21212
 Current Pjump:       0.72606   0.29288   0.09978   0.04008   0.45500
 Current finetune:    0.01000   0.00010   0.00500   0.00050   0.20000
 New finetune:        0.04276   0.00010   0.00155   0.00006   0.34061
 -2% 0.70 0.30 0.26 0.43 0.20    3  7 0.0015 0.0000 \
 P(3)=0.6525 0.0014   0.0013   1270.04377 -2857.17969
 Current Pjump:       0.70270   0.30469   0.25726   0.42525   0.19600
 Current finetune:    0.04276   0.00010   0.00155   0.00006   0.34061
 New finetune:        0.16644   0.00010   0.00130   0.00010   0.21257
 -1% 0.71 0.31 0.27 0.29 0.40    5 13 0.0028 0.0095 \
 P(5)=0.5740 0.0015   0.0013   1505.49123 -2859.64272
 Current Pjump:       0.71257   0.30731   0.27124   0.28854   0.39800
 Current finetune:    0.16644   0.00010   0.00130   0.00010   0.21257
 New finetune:        0.67367   0.00010   0.00116   0.00009   0.30111
 0% 0.73 0.31 0.30 0.21 0.24    5 13 0.0000 0.0480 \
 P(5)=1.0000 0.0018   0.0011   1490.58826 -2860.30822   0:26
 Current Pjump:       0.72664   0.31391   0.30472   0.20950   0.24250
 Current finetune:    0.67367   0.00010   0.00116   0.00009   0.30111
 New finetune:        2.88751   0.00011   0.00118   0.00006   0.23667
 5% 0.72 0.30 0.30 0.33 0.34    5 13 0.0000 0.0347 \
 P(5)=1.0000 0.0018   0.0012   1472.90507 -2860.39423   1:01
 10% 0.72 0.31 0.30 0.34 0.34    5 13 0.0000 0.0379 \
 P(5)=1.0000 0.0017   0.0012   1507.73604 -2860.41155   1:36
 ...
 95% 0.72 0.31 0.29 0.34 0.33    5 13 0.0007 0.0345 \
 P(5)=0.9268 0.0016   0.0012   1542.82446 -2860.30955   11:51
 100% 0.72 0.31 0.29 0.33 0.33    5 13 0.0007 0.0353 \
 P(5)=0.9305 0.0016   0.0012   1505.18890 -2860.32580   12:27

 12:27 spent in MCMC

(A) List of best models (count postP #species SpeciesTree)
 58409 0.584090 0.584090 5  (CBC NBC NCA SBC SCA)  ((CBC, ((NBC, SCA), NCA)), SBC);
 19842 0.198420 0.782510 5  (CBC NBC NCA SBC SCA)  (((CBC, (NBC, SCA)), NCA), SBC);
 11567 0.115670 0.898180 5  (CBC NBC NCA SBC SCA)  (((CBC, NCA), (NBC, SCA)), SBC);
 5459 0.054590 0.952770 4  (CBC NBCSCA NCA SBC)  ((CBC, (NBCSCA, NCA)), SBC);
 1383 0.013830 0.966600 5  (CBC NBC NCA SBC SCA)  ((CBC, ((NBC, NCA), SCA)), SBC);
 312 0.013120 0.979720 5  (CBC NBC NCA SBC SCA)  ((CBC, (NBC, (NCA, SCA))), SBC);
 760 0.007600 0.987320 4  (CBC NBCSCA NCA SBC)  (((CBC, NCA), NBCSCA), SBC);
 729 0.007290 0.994610 4  (CBC NBCSCA NCA SBC)  (((CBC, NBCSCA), NCA), SBC);
  ...
(B) 2 species delimitations & their posterior probabilities
 93048 0.930480    5 (CBC NBC NCA SBC SCA)
 6952 0.069520    4 (CBC NBCSCA NCA SBC)

(C) 6 delimited species & their posterior probabilities
 100000 1.000000 SBC
 100000 1.000000 NCA
 100000 1.000000 CBC
 93048 0.930480 SCA
 93048 0.930480 NBC
 6952 0.069520 NBCSCA

(D) Posterior probability for # of species
 P[1] = 0.000000   prior[1] = 0.175000
 P[2] = 0.000000   prior[2] = 0.175000
 P[3] = 0.000000   prior[3] = 0.225000
 P[4] = 0.069520   prior[4] = 0.250000
 P[5] = 0.930480   prior[5] = 0.175000
```

Listing 4 Output from BPP analysis A11 (joint species delimitation and species-tree estimation).

We note that two approaches to species delimitation are implemented in BPP. The first is the approach of Bayesian model comparison, as illustrated above in the A11 analysis (Yang and Rannala, 2010; Leaché et al., 2019). The second is to use the estimates of parameters in the MSC or MSci models and rely on heuristic criteria such as the genealogical divergence index (*gdi*) of Jackson et al. (2017). This approaches relies on the A00 analysis to estimate parameters – given the parameter values, calculation of heuristic index is straightforward. Leaché et al. (2019) suggested a recursive procedure to apply *gdi* when the data include more than two populations. See Chapter 5.5 (Rannala and Yang 2020) for more details.

6 Parallelization

BPP implements two levels of parallelization at the moment: instruction-level and intra-node (or multithreading) parallelism.

6.1 Instruction-level parallelism

Instruction-level parallelism (also known as *vectorization*) is achieved through single-instruction, multiple-data (SIMD) instruction set extensions to the x86 architecture. Currently BPP utilises code for three such instruction sets: *S*treaming SIMD Extensions (SSE), *A*dvanced Vector eXtensions (AVX) and AVX-2. SIMD instructions make use of *vector registers* (storage space within the processor) with a length of 128 (SSE), 256 (AVX and AVX-2) and 512 (AVX-512; not yet supported by BPP) bits. Those registers can hold multiple, independent data values of smaller size (e.g., a 256-bit register can hold four 64-bit double-precision floating-point values). For instance, if two registers contain four values each, the software can perform element-wise multiplications of the vector elements using a single instruction, instead of performing four separate multiplications as in the traditional x86 instruction set. Those instruction sets can significantly speed-up computation in matrix manipulations.

BPP automatically detects the best instruction set available on the computer it is executed on, and uses optimized code for that particular instruction set. On modern hardware, auto-detection works well. However, one can force a specific instruction set using the `arch` option in the control file, which takes four possible values: `CPU`, `SSE`, `AVX`, and `AVX2`. For example, to disable vectorization completely add the following line to the control file:

```
arch = CPU
```

6.2 Multithreading and NUMA architecture

BPP implements multithreading via pthreads and currently supports *medium-grained* parallelization across loci. Loci are distributed evenly to threads, and each thread handles its assigned loci sequentially for each move in an MCMC iteration. A move here is a collection of MCMC proposals of similar nature: for instance, the gene tree node age move cycles through all nodes on the gene tree for a locus and proposes a change to the age of each node. Communication between threads is reduced to one synchronization barrier at the end of each move. The number of threads used cannot exceed the number of loci in the dataset.

Furthermore, modern multiprocessor computers typically utilize the non-uniform memory access (NUMA) architecture, in which memory access time depends on the memory location relative to the processor. Figure 5 depicts the memory layout of a NUMA multiprocessor system with four CPUs. Each CPU may comprise several cores, and has its own local memory which is faster to access than the local memory of another processor. Currently BPP does

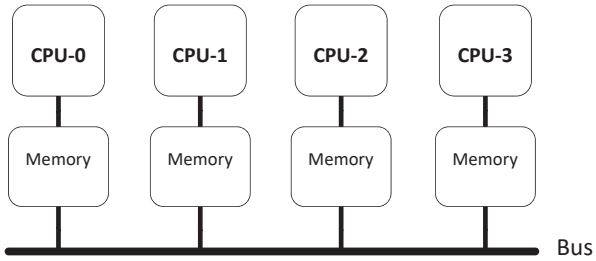

Figure 5 Non-uniform memory access (NUMA) is standard in modern multiprocessing computer architecture, in which memory access is faster if the memory is local to the processor.

not take full advantage of the NUMA memory layout, as all memory accessed throughout the run of the program is allocated locally to the processor on which the first (master) thread is running. Therefore, to achieve optimal performance it is important to ensure that all threads are allocated on cores of the same processor. Given that the typical strategy followed by operating systems is to distribute the processing workload equally across processors, the performance of BPP can degrade substantially when increasing numbers of processors (not to be confused with cores) are involved in the computation. To alleviate this issue, we have implemented core pinning, i.e. each thread is pinned to a particular CPU core.

Multithreading is enabled by specifying the `threads` variable in the control file which has the format `threads = N A B`, where N is the number of threads to be used, A is the starting core/thread number, and B is the stride, so the N threads will be assigned to cores $A, A+B, \ldots, A+(N-1)B$. Parameters A and B are optional, and their default values are 1.

```
threads = 4       * equivalent to threads = 4 1 1
```

The `lscpu` program is available on most GNU/Linux and MacOS distributions and can be used to see the topology of the system, such as the number of processors, cores and threads. For example, the following shows the output of the `lscpu` command for a quad-processor Lenovo ThinkSystem SR850 with four Intel Xeon Gold 6154 CPUs.

```
lscpu | egrep 'NUMA|Thread|Core'
```

and observe the output:

```
Thread(s) per core:    2
Core(s) per socket:    18
NUMA node(s):          4
NUMA node0 CPU(s):     0-17,72-89
NUMA node1 CPU(s):     18-35,90-107
NUMA node2 CPU(s):     36-53,108-125
NUMA node3 CPU(s):     54-71,126-143
```

There are four processors (NUMA nodes), and each processor comprises 18 physical cores, each of which can execute two threads (hyperthreading). Cores 1-18 are part of CPU1, 19-36 of CPU2, 37-54 of CPU3 and 55-72 of CPU4. The remaining 72 cores are hyperthreaded. (Note that the `lscpu` output starts from 0 while we start from 1 in the `threads` option in BPP.) Thus the following uses all 18 cores of the second processor:

```
threads = 18 19
```

or equivalently

```
threads = 18 19 1
```

A second example is for a dual-processor Dell PowerEdge T640 with two Intel Xeon Gold 5118 CPUs.

```
lscpu | egrep 'NUMA|Thread|Core'

Thread(s) per core:  2
Core(s) per socket:  12
NUMA node(s):        2
NUMA node0 CPU(s):   0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46
NUMA node1 CPU(s):   1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43,45,47
```

In this case, the cores of a processor are not enumerated sequentially, but are interleaved. Then `threads = 4 1 2` will specify the first four cores of CPU 1. The option `threads = 4 1 1` would use the first two cores of CPU 1 and the first two cores of CPU 2, and should be avoided.

Note that using more cores or threads, although always taking more computing resources, may not always reduce the running time. For many datasets involving sequence data from closely related species, we found that using 4 and 8 threads on the same processor gave near optimal performance. A good strategy is to execute a short run with low numbers for `burnin`, `sampfreq` and `nsample` (so that the run finishes in a few minutes) and experiment with different numbers of threads (with `threads` = 1, 2, 4, or 8, say), recording the running time to determine the optimal choice.

7 Discussion

This chapter has outlined the basic features of the BPP program and provided examples of simple analyses aimed at either species tree inference or species delimitation. Our goal has been to provide practical instruction on the use of the program. More detailed information regarding the underlying models implemented in BPP may be found in Chapters 3.3 and 5.5 (Rannala et al. 2020; Rannala and Yang 2020).

Detailed BPP documentation describing all the features and options of the program is available on the GitHub wiki at `https://github.com/bpp/bpp/wiki` and as a PDF manual. User support is available on the BPP Google group at `https://groups.google.com/forum/#!forum/bpp-discussion-group`. A web application for preparing BPP control and map files is available at `https://brannala.github.io/bpps/`.

References

Burgess, R. and Yang, Z. (2008). Estimation of hominoid ancestral population sizes under Bayesian coalescent models incorporating mutation rate variation and sequencing errors. *Mol. Biol. Evol.*, 25:1979–1994.

Flouri, T., Jiao, X., Rannala, B., and Yang, Z. (2018). Species tree inference with BPP using genomic sequences and the multispecies coalescent. *Mol. Biol. Evol.*, 35(10):2585–2593.

Flouri, T., Jiao, X., Rannala, B., and Yang, Z. (2019). A Bayesian implementation of the multispecies coalescent model with introgression for phylogenomic analysis. *Mol. Biol. Evol.* msz296.

Flouri, T., Rannala, B., and Yang, Z. (2020). A tutorial on the use of bpp for species tree estimation and species delimitation. In Scornavacca, C., Delsuc, F., and Galtier, N., editors, *Phylogenetics in the Genomic Era*, chapter 5.6, pages 5.6:1–5.6:16. No commercial publisher | Authors open access book.

Hey, J. and Nielsen, R. (2007). Integration within the Felsenstein equation for improved Markov chain Monte Carlo methods in population genetics. *Proc. Natl. Acad. Sci. U.S.A.*, 104(8):2785–2790.

Jackson, N. D., Carstens, B. C., Morales, A. E., and O'Meara, B. C. (2017). Species delimitation with gene flow. *Syst. Biol.*, 66(5):799–812.

Leaché, A. D., Koo, M. S., Spencer, C. L., Papenfuss, T. J., Fisher, R. N., and McGuire, J. A. (2009). Quantifying ecological, morphological, and genetic variation to delimit species in the coast horned lizard species complex (*Phrynosoma*). *Proc. Natl. Acad. Sci. U.S.A.*, 106:12418–12423.

Leaché, A. D., Zhu, T., Rannala, B., and Yang, Z. (2019). The spectre of too many species. *Syst. Biol.*, 68(1):168–181.

Olave, M., Solà, E., and Knowles, L. L. (2014). Upstream analyses create problems with DNA-based species delimitation. *Systematic Biology*, 63(2):263–271.

Rannala, B., Edwards, S. V., Leaché, A., and Yang, Z. (2020). The multi-species coalescent model and species tree inference. In Scornavacca, C., Delsuc, F., and Galtier, N., editors, *Phylogenetics in the Genomic Era*, chapter 3.3, pages 3.3:1–3.3:20. No commercial publisher | Authors open access book.

Rannala, B. and Yang, Z. (2003). Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. *Genetics*, 164:1645–1656.

Rannala, B. and Yang, Z. (2013). Improved reversible jump algorithms for Bayesian species delimitation. *Genetics*, 194(1):245–253.

Rannala, B. and Yang, Z. (2017). Efficient Bayesian species tree inference under the multispecies coalescent. *Syst. Biol.*, 66:823–842.

Rannala, B. and Yang, Z. (2020). Species delimitation. In Scornavacca, C., Delsuc, F., and Galtier, N., editors, *Phylogenetics in the Genomic Era*, chapter 5.5, pages 5.5:1–5.5:17. No commercial publisher | Authors open access book.

Robert, C. P. and Casella, G. (2005). *Monte Carlo Statistical Methods*. Springer-Verlag, Berlin, Heidelberg.

Xu, B. and Yang, Z. (2016). Challenges in species tree estimation under the multispecies coalescent model. *Genetics*, 204:1353–1368. doi: 10.1534/genetics.116.190173.

Yang, Z. (2002). Likelihood and Bayes estimation of ancestral population sizes in hominoids using data from multiple loci. *Genetics*, 162(4):1811–1823.

Yang, Z. (2014). *Molecular Evolution: A Statistical Approach*. Oxford University Press, Oxford, England.

Yang, Z. (2015). The BPP program for species tree estimation and species delimitation. *Curr. Zool.*, 61(5):854–865. http://dx.doi.org/10.1093/czoolo/61.5.854.

Yang, Z. and Rannala, B. (2010). Bayesian species delimitation using multilocus sequence data. *Proc. Natl. Acad. Sci. U.S.A.*, 107:9264–9269.

Yang, Z. and Rannala, B. (2014). Unguided species delimitation using DNA sequence data from multiple loci. *Mol. Biol. Evol.*, 31(12):3125–3135.

Yang, Z. and Rodríguez, C. E. (2013). Searching for efficient Markov chain Monte Carlo proposal kernels. *Proc. Natl. Acad. Sci. U.S.A.*, 110(48):19307–19312.