BP&P

VERSION 3.1 (March 2015)

The program is provided "as is" without warranty of any kind.
The program is provided free of charge for academic use only.

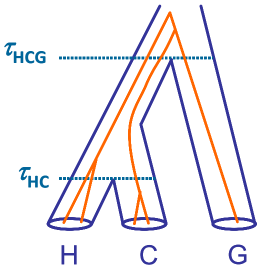


Table of Contents

0. Introduction

BPP is a Bayesian Markov chain Monte Carlo (MCMC) program for analyzing DNA sequence alignments from multiple loci and multiple closely-related species under the multispecies coalescent (MSC) model (Rannala and Yang, 2003; Yang, 2002). The program can be used to conduct four different analyses, specified using two variables in the control file:

A00 (**`speciesdelimitation = 0, speciestree = 0`**): estimation of the parameters of species divergence times and population sizes under the MSC model when the species phylogeny is given (Rannala and Yang, 2003);

A01 (**`speciesdelimitation = 0, speciestree = 1`**): inference of the species tree when the assignments are given by the user (Rannala and Yang, in preparation);

A10 (**`speciesdelimitation = 1, speciestree = 0`**): species delimitation using a user-specified guide tree (Yang and Rannala, 2010; Rannala and Yang, 2013);

A11 (**`speciesdelimitation = 1, speciestree = 1`**): joint species delimitation and species tree inference of unguided species delimitation (Yang and Rannala, 2014).

Underlying all those analyses is the MSC model, which specifies the probabilistic distribution of gene trees given the species tree (Rannala and Yang, 2003; see also Takahata et al., 1995; Yang, 2002). The basic parameters in the MSC model include the species divergence times (τs), measured by the expected number of mutations per site, 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 different sites between two sequences sampled at random from the population. For a species tree with s species, there are $s - 1$ divergence times (τs) and at most $2s - 2$ population size parameters (θs). Analysis A00 is to estimate those parameters when the species delimitation and species tree is fixed. Analyses A01, A10, and A11 compare different MSC models.

See Yang (2014: Chapter 9) for a discussion of Bayesian inference under the MSC model. Chapters 7 and 8 of the same book describes Bayesian MCMC algorithms. Please also look at the BPP tutorial which illustrates the four analyses (Yang, 2015).

Standard assumptions made in the program include no recombination within a locus, free recombination between loci, no migration (gene flow) between species, and neutral evolution. The JC69 mutation model (Jukes and Cantor, 1969) is assumed to accommodate multiple hits. The sequences are supposed to be close, so that JC69 is deemed adequate.

How to cite the program. You can cite the BPP tutorial and the original papers that described the methods you used (see above). Describe the priors you used since they are necessary for reproducibility. If you conduct a joint analysis of species delimitation and species tree inference, your method description may look like the following (replace the numbers in green with those you used):

"Joint Bayesian species delimitation and species tree estimation was conducted using the program BPP (Yang, 2015). The method uses the multispecies coalescent model to compare different models of species delimitation and species phylogeny in a Bayesian framework, accounting for incomplete lineage sorting due to ancestral polymorphism and gene tree-species tree conflicts (Yang and Rannala, 2010; Rannala and Yang, 2013; Yang and Rannala, 2014). The population size parameters (θs) are assigned the gamma prior G(2, 1000), with mean 2/2000 = 0.001. The divergence time at the root of the species tree (τ_0) is assigned the gamma prior G(2, 1000), while the other divergence time parameters are assigned the Dirichlet prior (Yang and Rannala, 2010: equation 2). Each analysis is run at least twice to confirm consistency between runs."

1. Getting started

1.1 Compiling the program

The BPP program is written in ANSI C. Win32 executables are included in the archive. If you use UNIX /linux or Mac osx, remove the .exe files and re-compile the program. You need to do this only once. You can use gcc or any ANSI C-compatible compiler. Some source files are from my PAML package (paml.h, tools.c, treesub.c). Look at the README.txt file, or try one of the following:

```
cc -o bpp -fast bpp.c tools.c -lm
cc -o bpp -O4 bpp.c tools.c
gcc -o bpp -O3 bpp.c tools.c -lm
cl -O2 -Ot bpp.c tools.c   (MS VC++)
```

The -o flag specifies the name of the resulting executable file; the -O3 , -O4 and -fast flags are for optimizing the code; and -lm is to link to the math library. You may have to change some of the optimization flags (-fast, -O4, -O3, etc.), and the -lm flag is not needed on some systems.

The same source code can be compiled into a simulation program (MCcoal). See the section *The simulation program (MCcoal)* later in this document for details.

1.2 Trial run

Run the program from a command box (rather than double-clicking the executable) so that you will see the error messages. In the bpp/ folder, run the program by typing the following command

```
bpp bpp 5s.bpp.ctl
./bpp bpp 5s.bpp.ctl
```

Or move to the examples/ folder, and run the example analysis

```
cd examples
../bpp yu2001.bpp.ctl
../bpp ChenLi2001.bpp.ctl
../bpp lizard.bpp.ctl
```

You may use the graphical interface BPPX, written by Bo Xu. This documentation assumes that you run BPP from the command line. If you have not used the command line before, here are simple tutorials that you should go through first:
http://abacus.gene.ucl.ac.uk/software/CommandLine.Windows.pdf
http://abacus.gene.ucl.ac.uk/software/CommandLine.MACosx.pdf

2. File formats

2.1 Sequence data file and individual map (Imap) file

The sequence file. The sequence alignments are in the phylip/paml format, with one alignment following the other, all in one file. The sequence name should be separated from the sequence by a line break or by at least two spaces. Have a look at the sequence files 5stest.txt and ChenLi2001.txt. An alignment is also called a locus. Every locus must have at least 2 sequences, different loci can have different numbers of sequences, and some species can be missing at some loci.

By default (cleandata = 0), alignment gaps and ambiguity nucleotides are used in the likelihood calculation, with gaps treated as question marks (see Yang, 2006, pp. 107-108). If cleandata = 1, all columns with gaps or ambiguity characters are removed before analysis.

The Imap file. It is assumed that each sequence is from an individual and each individual is from a species. Each sequence name has a tag indicated by ^. The tag is a short string and is the individual (specimen) ID, such as MBt03 (for Mallet Basement Topshelf #3). For instance, in the data file 5stest.txt, the sequence name A3^3 means sequence A3 is from individual 3. We then use a map file to map the individuals to the species (which are specified in the control file bpp.ctl). For example, 5s.Imap.txt for the 5stest.txt dataset is shown below, which maps individual 3 to species A, say.

```
1 A
2 A
3 A
4 B
5 C
6 C
7 D
8 E
```

Our model and program uses only the information about which species each sequence is from. It does not use the information about which individual each sequence is from. Note that the model assumptions are quite different from cluster algorithms (such as STRUCTURE) which use information from multi-locus genotypes. BPP reads sequence names, use the individual ID tags to construct the species ID for each sequence and then forget the sequence names. It does not bother the program if ten sequences at a locus have the same individual ID. We could have used a species ID in each sequence name and got rid of the Imap file. We are using the current design because we imagined that one might want to analyze the same sequence data on different guide trees with different assignments, in which case the current format requires changes to the small Imap file but not to the much larger sequence data file.

2.2 Control file

2.2.1 The control variables

The default control file name is bpp.ctl. Lines beginning with an asterisk are comments. Most often the order of the lines is unimportant. Below I will use ChenLi2001.bpp.ctl to explain the variables in the control file. This is for analysis A00: parameter estimation under the MSC using multiple-loci multiple-species data on a fixed species tree (Rannala and Yang, 2003;

Burgess and Yang, 2008). I will then comment on the other analyses (A01, A10, and A11). Please also read the section on the example datasets later in this document.

Below is a copy of the control file ChenLi2001.bpp.ctl. Note that if there are *s* species in the species tree, the model will involve the following parameters: (s – 1) species divergence times (τs) and (s – 1) ancestral θs. If a species has at least two sequences at any loci, a θ for that species will be used as well. If there is one species, the model will involve one parameter only, θ for that species. The parameters are ordered as follows: θs for the extant species, θs for the ancestral species (s – 1 of them), and the divergence times τs for the ancestral nodes (s – 1 of them).

```
          seed =  -1

       seqfile = ChenLi2001.txt
      Imapfile = ChenLi2001.Imap.txt
       outfile = out.txt
      mcmcfile = mcmc.txt

speciesdelimitation = 0         * fixed species delimitation
*speciesdelimitation = 1 0 2    * rjMCMC speciesdelimitation algorithm0(e)
*speciesdelimitation = 1 1 2 1  * rjMCMC speciesdelimitation algorithm1(a m)
 speciestree = 0        * species tree fixed
*speciestree = 1        * NNI over species/guide trees

*speciesmodelprior = 1  * 0: uniform LH; 1:uniform rooted trees; 2: uniformSLH; 3: uniformSRooted

  species&tree = 4  H  C  G  O
                    1  1  1  1
                 (((H, C), G), O);

       usedata = 1    * 0: no data (prior); 1:seq like
         nloci = 53    * number of datasets in seqfile

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

    thetaprior =  2 2000   # gamma(a, b) for theta
      tauprior = 14 1000   # gamma(a, b) for root tau & Dirichlet(a) for other tau's

*     locusrate = 0 2.0   # (0: No variation, 1: estimate, 2: from file) & a_Dirichlet (if 1)
*      heredity = 0    # (0: No variation, 1: estimate, 2: from file) & a_gamma b_gamma (if 1)
*      heredity = 1 4 4   # (0: No variation, 1: estimate, 2: from file) & a_gamma b_gamma (if 1)
*      heredity = 2 heredity.txt   # (0: No variation, 1: estimate, 2: from file) & a_gamma b_gamma (if 1)

* sequenceerror = #  model of sequencing errors has changed, to be described later

       finetune =  0: 0.5 0.0015 0.0006  0.0004 0.06 0.2 1.0  # auto adjustment: finetune steplengths for
GBtj, GBspr, theta, tau, mix, locusrate, seqerr

         print = 1 0 0 * print MCMC samples, locusrate, heredityscalars
        burnin = 2000
      sampfreq = 2
       nsample = 20000
```

seed is the random number seed. If you use a positive integer, the program will produce identical results in different runs. This is useful for debugging. If you use -1, the program will use the wall clock to generate a seed, and different runs will produce different results. It is recommended that you run the same analysis at least twice using different seeds to confirm that the results are stable across runs.

seqfile is the name of the sequence alignment file, while
Imapfile is the individual map file. The Imapfile is not needed if the data contain only one species. These two files are input.

```
speciesdelimitation = 0         * fixed species delimitation
speciestree = 0        * no change to species tree or guide trees
```

speciesdelimitation = 0 and speciestree = 0 specify analysis A00: estimation of parameters under the MSC model. For analyses A01, A10, and A11, those two variables may be specified the value 1. See the next subsection below. If those two variables are missing or their lines are commented out, the default value of 0 is assumed. The variable **speciesmodelprior** is used in analyses A01, A10, and A11, and has no effect for analysis A00.

```
species&tree = 4  H  C  G  O
                  1  1  1  1
              (((H, C), G), O);
```

The above block specifies 4 species in the data, which are H (human), C (chimp), G (gorilla), and O (orang). The maximum number of sequences at any locus is 1 for H, 1 for C, 1 for G, and 1 for O. These numbers serve two purposes. First, they are used to determine which θ parameters are involved in the model and should be estimated. Second, they specify the maximum number of sequences for all species at a locus. The model always use a θ and a τ (age) for every interior (ancestral) node on the species tree. The model also uses a θ for each extant species if and only if that species has more than one sequence at some loci. In the example here, the parameters are the three ancestral θs and three node ages (τs).

The species tree is in the familiar parenthesis format, ending with a semicolon (;). The tree is fixed if **speciesdelimitation = 0** and is used as the guide tree in the rjMCMC run for species delimitation if **speciesdelimitation** = 1 (see below).

usedata = 0 is for running the MCMC without sequence data to generate the prior, while **usedata = 1** is for generating the posterior.

nloci specifies the number of loci (alignments). If you have 200 loci in the data file and choose **nloci = 2**, BPP will read the first two loci only.

cleandata = 1 causes the program to remove all columns in the alignment which have gaps or ambiguity characters. **cleandata** = 0 means that those will be used in the likelihood calculation.

thetaprior = 2 2000 specifies the gamma prior $G(\alpha, \beta)$ for the θ parameters, with the mean to be α/β. In the example, the mean is 2/2000 = 0.001 (one difference per kb). Note that all θ parameters in the MSC model (for both modern species and extinct ancestral species) are assigned the gamma prior with the same parameters.

tauprior = 14 1000 specifies the gamma prior $G(\alpha, \beta)$ for τ_0, the divergence time parameter for the root in the species tree. Other divergence times are generated from the Dirichlet distribution (Yang and Rannala, 2010: equation 2). In the example, the mean is 14/1000 = 0.014 (which means 1.4% of sequence divergence). If the mutation rate is 10^{-9} mutations/site/year, this distance will translate to a human-orangutan divergence time of 14MY.

```
locusrate = 0                    #  (0: No variation)
locusrate = 1 2.0                 # (1: estimate, & a_Dirichlet)
locusrate = 2 LocusRateFileName     # (2: from file)
```

locusrate = 0 (default) means that all loci have the same mutation rate. **locusrate = 1** or **2** specifies two models for variable mutation rates among loci (Burgess and Yang, 2008). **locusrate = 1** specifies the random-rates model of Burgess & Yang (2008: equation 4). The average rate for all loci is fixed at 1, while the rates among loci are assumed to generated from the Dirichlet distribution $D(\alpha)$. Parameter α is inversely related to rate variation, with a large α meaning similar rates among loci. If all loci are noncoding, the rates are probably similar, so $\alpha = 10$ or 20 may be reasonable, while $\alpha = 2$ or 1 may be too small. The MCMC generates the posterior for rates at loci.

locusrate = 2 LocusRateFileName specifies the fixed-rates model of Burgess & Yang (2008). This is the strategy used by Yang (2002), with the relative rates estimated by the distance to an outgroup species. The relative locus rates are listed in the file: there should be as many numbers in the file, separately by spaces or line returns, as the number of loci or **nloci**. The program re-scales those rates so that the average among all loci is 1 and then use those relative rates as fixed constants.

Note. The model of variable rates among loci implemented here has some differences from a similar model implemented in the IMa program (Hey and Nielsen, 2004). The biggest difference appears to be the parametrization. BPP defines mutation rate on a per-nucleotide basis, so the prior specifies that the expectation of the mutation rate per site is constant among loci. IM defines mutation rate on a per-locus basis, so its prior specifies that the expectation of the mutation rate per locus is the same among loci. If locus one has 100 sites and locus two has 1000 sites, then IM assumes that the per-site rate for locus one is 10 times for locus two, while BPP assumes the same per-site rate. Also IM constrains the geometric mean of rates across loci to be one, while BPP constrains their arithmetic mean to be one. The IM assumptions do not appear to me to be realistic.

```
heredity = 0      # (0: No variation)
heredity = 1 4 4  # (1: estimate, & a_gamma b_gamma)
heredity = 2 heredity.txt     # (2: from file)
```

heredity = 0 is the default and means that θ is the same for all loci. **heredity = 1** or **2** specifies two models that allow θ to vary among loci, which may be useful for combined analysis of data from autosomal, mitochondrial, X and Y loci. With such mixed data, the effective population sizes are different among loci, so that a heredity multiplier (inheritance scalars, Hey and Nielsen, 2004) should be applied. Other factors such as natural selection may also cause θ to deviate from the neutral expectation. BPP implements two options for this. The first option (**heredity = 1**) is to estimate the multipliers from the data, using a gamma prior with parameters α and β specified by the user. In the example above, a gamma prior $G(4, 4)$, with mean 4/4 = 1, is specified for the multiplier for each locus. The MCMC should then generate a posterior for the multiplier for each locus. The second option (**heredity = 2**) is for the user to specify the multipliers in a file, and the multipliers will then be used as fixed constants in the MCMC run. The file simply contains as many numerical values as the number of loci, separated by spaces or line breaks.

Genome	Heredity scalar
Nuclear autosome	1
X chromosome	0.75
Y chromosome	0.25
Mitochondrial	0.25

Note. The effect of the locus-specific mutation rates and the locus-specific heredity multipliers are different. A locus rate is used to multiply all θs and τs for the locus, while a heredity multiplier is used to multiply all θ parameters for the locus but not the τs. Nevertheless, those parameters are quite likely to be strongly correlated, especially when the species tree is small.

Ziheng note on 3 March 2011.
The model of sequence errors is rewritten, so that the old option is unavailable. Use

```
sequenceerror = 0               #  (0: default: No error)
```

```
finetune = 0: 0.5 0.0015 0.0006 0.0004 0.06 0.2 1.0  # auto (0 or 1): finetune for GBtj, GBspr, theta, tau, mix, locusrate, seqerr
```

```
finetune = 1: 0.01 0.01 0.01 0.01 0.01 0.01 0.01  # auto (0 or 1): finetune for GBtj, GBspr, theta, tau, mix, locusrate, seqerr
```

This is about the step lengths used in the proposals in the MCMC algorithm. The first value, before the colon, is a switch, with 0 meaning no automatic adjustments by the program and 1 meaning automatic adjustments by the program. Following the colon are the step lengths for the proposals used in the program. If you choose to let the program adjust the step lengths, burnin has to be >200, and then the step lengths specified here will be the initial step lengths, and the program will try to adjust them using the information collected during the burnin step. This option appears to work fine. Some notes about manually adjusting those finetune step lengths are provided below in section 3.2.

```
print = 1 0 0 0 * print MCMC samples, locusrate, heredityscalars, GeneTrees
burnin = 2000
sampfreq = 2
nsample = 20000
```

print = 0 means that no MCMC samples are written into the file, which may be useful if you need the screen output only. print = 1 generates a file mcmc.txt containing the MCMC samples. The next two flags on the same line are for printing locus rates and locus heredity scalars if those are estimated from the data using gamma priors. I suspect there may be trouble if you have many thousands of loci. I think the option of printing and processing gene trees for loci are not working, so choose 0.

MCMC samples are taken after the burnin, and in this example, are taken every 2 iterations, with a total of 20,000 samples taken. The total number of MCMC iterations is burnin + output × nsample. The resulting file can be large. For analysis A00 (speciesdelimitation = 0, speciestree = 0), this file is readable from R or Andrew Rambaut's TRACER. For other analyses (A01, A10, and A11), the sample file is not readable by R or TRACER.

`print = -1` means that BPP will bypass the MCMC. Instead it will read the MCMC sample and summarize the results. Thus with `print = 1`, the mcmc.txt file will be output, but with `print = -1`, it will be the input. Take care to avoid useful files being overwritten.

2.2.1 The four analyses (A00, A01, A10, and A11)

Here we describe the specifics of the four different analyses.

A00 (`speciesdelimitation = 0, speciestree = 0`), for estimation of the parameters of species divergence times and population sizes (τs and θs) under the MSC model when the species phylogeny is given (Rannala and Yang, 2003), has been explained in detail above using the control file ChenLi2001.bpp.ctl as an example.

If there is only one species, the MSC model will become the single-population coalescent (Kingman, 1982). Take a look at examples/yu2001.bpp.ctl, which is for analyzing a sample of 61 human sequences from Yu et al. (2001) to estimate the single parameter $\theta = 4N\mu$. There is no need for the Imap file, or the need to tag the sequence names in the sequence file (yu2001.txt): the sequence names are read and then ignored. Multiple loci may be included in the sequence file. There is no need for a species tree, so the block for specifying species names and species tree looks like this:

```
species&tree = 1  H
                 100  * max number of sequences
```

In the single-species analysis, the printout on the monitor includes the posterior mean of θ, and posterior means of μt_{MRCA} for the loci, calculated up to that point in the MCMC run. If you have many loci, only the first few μt_{MRCA} are printed on the monitor.

The mcmc sample file lists $1 + g + 1$ columns for g loci: θ, μt_{MRCA} for the g loci, and the log likelihood. The μt_{MRCA} are not parameters, but are sometimes of interest as well.

Common features of analyses A01, A10, and A11. Before we describe analyses A01, A10, and A11, it may be fitting to describe some common features they share. Note that A00 is a within-model inference: there is one well-specified model (the MSC model on a fixed species tree) and the parameters in the model are all well defined. The objective of the analysis is to estimate those parameters.

In contrast A01, A10, and A11 are all trans-model inferences. They move between different models, and the main objective is to calculate the posterior probabilities for those models. Each of those models is an instance of the MSC model, but the species delimitation (the number and nature of the species) and/or the species phylogeny may differ between models. In analyses A01, A10, and A11, the gamma prior specified using thetaprior = 2 2000 applies to all θ parameters in all models: in other words, there may be thousands of θ parameters across the models, and each one is assigned the G(2, 2000) prior. Similarly each of those MSC models (if they specify two or more delimited species) may have a parameter τ_0 for the divergence time of the root. The specification tauprior = 2 1000 then means that all those τ_0 parameters will be assigned the gamma prior G(2, 1000).

Another difference is that the sample file mcmc.txt is readable in R or by TRACER for analysis A00, but not for analyses A01, A10, and A11.

A01 means species tree estimation when the assignments and delimitation are fixed.

```
speciesdelimitation = 0    *
        speciestree = 1    * NNI/SPR over species trees
  speciesmodelprior = 1  * 0: uniform LH; 1:uniform rooted trees; 2: uniformSLH; 3: uniformSRooted
```

This invokes the NNI or SPR algorithm to change the species tree topology, while species delimitation is fixed (so that the number of species and the assignment of individuals to species are fixed).

A10, for species delimitation using a user-specified guide tree (Yang and Rannala, 2010; Rannala and Yang, 2013), is specified using

```
speciesdelimitation = 1 0 2   * speciesdelimitation algorithm0 and finetune(e)
speciesdelimitation = 1 1 2 1 * speciesdelimitation algorithm1 finetune (a m)
  speciesmodelprior = 1  * 0: uniform LH; 1:uniform rooted trees
```

The first line specifies rjMCMC algorithm 0, with $\varepsilon = 2$ in equations 3 and 4 of Yang & Rannala (2010). Reasonable values for ε are 1, 2, 5, etc.
The second line specifies rjMCMC algorithm 1, with $\alpha = 2$ and $m = 1$ in equations 6 and 7 of Yang & Rannala (2010). Reasonable values are $\alpha = 1, 1.5, 2$, etc. and $m = 0.5, 1, 2$, etc.

The two algorithms in theory should produce identical results. The variable speciesmodelprior specifies Priors 0 and 1. Prior 0 means equal probabilities for labeled histories (which are rooted trees with internal nodes ordered by their age). This is the prior used by Yang & Rannala (2010: equation 2). Prior 1 means equal probabilities for rooted trees. This is now the default. The prior with the user specified probabilities for nodes described by Rannala and Yang (2013) is deleted in the current version. You will have to use version 2.2 for that.

A11, for joint species delimitation and species tree inference or for unguided species delimitation (Yang and Rannala, 2014), is specified as follows.

```
speciesdelimitation = 1 0 2    * rjMCMC speciesdelimitation algorithm0(e)
*speciesdelimitation = 1 1 2 1  * rjMCMC speciesdelimitation algorithm1(a m)
       speciestree = 1    * NNI over species trees
 speciesmodelprior = 1  * 0: uniform LH; 1:uniform rooted trees; 2: uniformSLH; 3: uniformSRooted
```

In this case, BPP will use the rjMCMC algorithm (either algorithm 0 or algorithm 1 of Yang and Rannala, 2010) to change the species delimitation model and the NNI/SPR move to change the species tree topology.

For A11, **speciesmodelprior** can take the four values 0, 1, 2, 3, which mean Priors 0, 1, 2, 3, respectively. As mentioned above, Prior 1 (which is the default) assigns equal probabilities to the rooted species trees, while Prior 0 means equal probabilities for the labeled histories (rooted trees with the internal nodes ordered by age). Priors 2 and 3 assign equal probabilities for the numbers of species (1/*s* each for 1, 2, ..., *s* species given *s* populations) and then divided up the probability for any specific number of species among the compatible models (of species delimitation and species phylogeny) either uniformly [Prior 3] or in proportion to the labeled histories [Prior 2]. Priors 2 and 3 are mentioned by Yang and Rannala (2014) and implemented by Yang (2015). Prior 3 may be suitable when there is a large number of populations.

3. Screen and file outputs

3.1 A00 screen output

First we consider the simple analysis under the MSC model with the species tree fixed (A00: `speciesdelimitation` = 0, `speciestree` = 0). We use this case to explain the acceptance proportions of MCMC moves. The screen outputs for analyses A01, A10, and A11 will have differences, which will be described later.

Make the window wider, with 100 or 120 columns, say, before you run the program. (On Windows, you right-click the window title bar and choose Properties – Layout and change Window Size Width.) Pay attention to screen outputs, especially at the start of the run, to make sure that the control file and sequence data file are read correctly by the program, and that the acceptance proportions are reasonable. Use Ctrl-C to terminate the run, if needed.

Here is an outline of the steps taken by the program. The sample output is from analyzing the example dataset ChenLi2001.txt, on a fixed species tree. The differences for the species delimitation analysis are discussed later. The program first prints out the species tree, as well as a population-population table, which describes the descendant-ancestor relationship between populations and which you may ignore. It then defines the θ and τ parameters involved in the model.

The program then reads and processes the sequence data file.

It then generates the initial values for parameters θs and τs by using the gamma priors and the initial gene trees and coalescent times by sampling from the prior. The program prints out the initial θs and τs, as well as the initial log likelihood lnL0, and starts the MCMC.

```
Starting MCMC...
prior theta ~ G(2.000, 2000.000)
prior tau   ~ G(14.000, 1000.000)

Initial parameters, np = 6 (gene trees generated from the prior):
  0.00091  0.00111  0.00100  0.01374  0.00306  0.00036
lnL0 =  -43347.191
  0%  0.28 0.42 0.37 0.38 0.38  0.00192 0.00344 0.00240  0.01370 0.00593 0.00477 -42852.55  0:05
  5%  0.28 0.39 0.35 0.37 0.36  0.00198 0.00342 0.00182  0.01403 0.00602 0.00483 -42845.57  0:22
 10%  0.28 0.38 0.34 0.37 0.36  0.00186 0.00349 0.00174  0.01408 0.00596 0.00486 -42837.31  0:35
```

Then on the same line, it prints out a percentage progress indicator (with negative values indicating burnin), followed by the acceptance proportions for the MCMC moves (highlighted in red in the sample output), and by the posterior means of the parameters (in this example there are three θ parameters and three τ parameters). The last number before the time used is the current log likelihood. You may adjust the finetune variables in the control file so that the acceptance proportions are close to 0.3 or lie in the interval (0.15, 0.7). See below.

The program will then read and process the file mcmc.txt to calculate the mean, min, max, median, and percentiles, and histogram information. This may take quite some time if many samples (say, 1M) are collected in the file.

The mcmc.txt sample file generated in analysis A00 can be read in R or TRACER.

3.2 Adjusting step lengths for MCMC moves (finetune)

You can use the automatic adjustment of the finetune variable (MCMC step lengths), which appears to be reliable, but make sure that the acceptance proportions are neither too small nor too large. Below are notes for manual adjustments of the step lengths. Often I use automatic adjustments to generate good step lengths and then copy them into the control file.

Below are some notes about adjusting the step lengths manually (`finetune = 0`).

First note the line like the following in the control file ChenLi2001.bpp.ctl:

```
finetune =  0: 0.5 0.005 0.0006  0.0004 0.06 0.2 1.0  # finetune for GBtj, GBspr, theta, tau, mix, locusrate, seqerr
```

There are seven finetune parameters here. They are in a fixed order and always read by the program even if the concerned proposal is not used. The first five of them are ε_1, ε_2, ε_3, ε_4, ε_5, described in Rannala & Yang (2003). These are the step lengths used in the MCMC proposals that (1) change internal node ages in the gene tree, (2) prune and re-graft nodes in the gene tree, (3) update θs, (4) update τs using the rubber-band algorithm, and (5) implements the mixing step. The 6th and 7th are for the proposals that change the locus rates or heredity multipliers and that change the sequencing errors, respectively. If the model assumes the same rate for all loci and does not use heredity multipliers, the 6th proposal step is not used. If the model assumes no sequencing errors, the 7th step is not used. The acceptance proportions for the first five proposals are always printed out on the screen, but those for the 6th and 7th are printed out only if the concerned proposal is used in the model. In the example above, only the first five proposals are used in the algorithm and we will change the step lengths so that the acceptance proportions become close to 30%. If the acceptance proportion is too small (say, <0.10)), decrease the corresponding finetune parameter. If the acceptance proportion is too large (say, >0.80), increase the finetune parameter.

Run the program for a small number of iterations and look at the screen output for the acceptance proportions.

```
lnL0 =  -42908.975
  0%  0.28 0.13 0.34 0.37 0.37  0.00189 0.00356 0.00176  0.01408 0.00589 0.00478 -42857.29  0:06
  5%  0.28 0.15 0.39 0.36 0.38  0.00184 0.00347 0.00282  0.01349 0.00594 0.00483 -42828.80  0:18
```

Here the second acceptance proportion, at 0.13 or 0.15, is somewhat too small, which means that the corresponding finetune parameter (0.005 above) is too large. Terminate the run (Ctrl-C) and decrease the value in the control file. Then run the program again (use the up ↑ and down ↓ arrow keys to retrieve past commands). Repeat this process a few times until every acceptance proportion is neither too small nor too large. In this example, changing 0.005 to 0.002 brings the acceptance proportion to 34%, which is good.

Those MCMC proposals are used in all four analyses (A00, A01, A10, A11), so that the description here applies to all of them. Note that the finetune parameters affect the efficiency of the MCMC or how fast one can obtain reliable results. In theory they do not change the results if all runs using different finetune parameters are long enough to generate reliable results.

3.4 A01: species tree estimation

Suppose we use the control file bpp.4s.ctl to run analysis A01: species tree estimation with species delimitation and assignment fixed (speciesdelimitation = 0, speciestree = 1).

The screen output will look like this:

```
  5% 0.25 0.30 0.29 0.31 0.25  0.127  0.0586 0.1857   54.33 -3367.37  0:14
 10% 0.25 0.30 0.29 0.32 0.26  0.099  0.0579 0.1873   57.86 -3361.23  0:22
 15% 0.25 0.30 0.29 0.32 0.26  0.103  0.0586 0.1861   50.06 -3352.25  0:31
    ^^ Pjump for MCMC moves ^^  PNNI   theta  tau      lnprior  lnL
```

Here the five Pjump values are the acceptance proportions for the five conventional MCMC moves, as discussed above. PNNI is the acceptance proportion for the NNI or SPR moves that change the species phylogeny. The next two numbers are posterior means of θ for the root population and τ_0 for the root age.

The MCMC sample of species trees is collected in the file mcmc.txt. Below are two lines from that file. The numbers after : are the branch lengths (τs), while those after # are θs.

```
(C#0.081857: 0.161553, (D#0.01373: 0.161516, (B#0.0462185: 0.0853056, A#0.0278149:
0.0853056)#0.00334327: 0.0762101)#0.00167966: 3.69996e-005)#0.0885696;

(C#0.081857: 0.161553, (D#0.0261241: 0.161516, (B#0.0462185: 0.0853056, A#0.0171918:
0.0853056)#0.00334327: 0.0762101)#0.00167966: 3.69996e-005)#0.0632965;
```

The BPP summary of the sample will look like the following.

```
Read tree sample, count trees & splits
tree 100000  (D, ((B, A), C));
100000 trees read, 3 distinct trees.

Species in order:
 1. A
 2. B
 3. C
 4. D

(A) Best trees in the sample (3 distinct trees in all)
  74922  0.74922  0.74922  (D, (C, (A, B)));
  20016  0.20016  0.94938  (C, (D, (A, B)));
   5062  0.05062  1.00000  ((C, D), (A, B));

(B) Best splits in the sample of trees (4 splits in all)
 100000   1.00000  1100
  74922   0.74922  1110
  20016   0.20016  1101
   5062   0.05062  0011

(C) Majority-rule consensus tree
(D, (C, (A, B)#1)#0.74922);
```

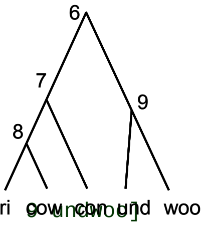
Section (A) lists the species trees in decreasing order of posterior probabilities. From this you can easily identify the 95% or 99% credibility set of species trees. Section (B) lists the splits (or bipartitions) and their posterior probabilities; for example the split 1110, which means ABC-D or the presence of the ABC clade, has posterior probability 0.74922. The split 0011, which means the CD clade, has the probability 0.05062. Note that the splits here take into account the location of the root, and may be different from the splits for unrooted trees. Section (C) prints the majority-rule consensus tree, with posterior probabilities for nodes.

3.3 A10: species delimitation using rjMCMC

We apply rjMCMC algorithm to the lizard data (A10: speciesdelimitation = 1, speciestree = 0).

```
   cd examples
   ..\bpp lizard.bpp.ctl
```

The screen output will look like the following. The species delimitation models that can be generated from the fixed guide tree are listed, together with their prior probabilities calculated by BPP. (As a check, if you use usedata = 0, the MCMC should be sampling from this prior distribution.) The species delimitation model is represented using four 0-1flags for the four interior nodes 6, 7, 8, 9 in the guide tree, with 0 for 'collapsed' and 1 for 'resolved'. Note that the tips in the guide tree are numbered 1, 2, …, *s* for *s* potential species, while the interior (ancestral) nodes are numbered *s* + 1, *s* + 2, …, 2*s* – 1, with *s* + 1 to be the root of the guide tree. The numbering is through a tree-traversal algorithm, fixed by the program. This same order is used to specify the divergence time parameters (τs), so you can work out the order by looking at the list of nodes in the screen output (look at the "population by population table", "# species divergence times in the order:", etc.).



```
Number of species tree models =  7
        delimitation model   1: 0000  prior  0.14286
        delimitation model   2: 1000  prior  0.14286
        delimitation model   3: 1001  prior  0.14286
        delimitation model   4: 1100  prior  0.14286
        delimitation model   5: 1101  prior  0.14286
        delimitation model   6: 1110  prior  0.14286
        delimitation model   7: 1111  prior  0.14286

[Note: Ancestral nodes in order:   6 tricowconundwoo  7 tricowcon  8 tricow  9 undwoo]

Initial parameters, np = 9 (gene trees are generated from the prior):
  0.00235  0.00222  0.00272  0.00176  0.00177  0.00164  0.00189  0.00183  0.00158
lnL0 =   -1987.887
-15% 0.59 0.01 0.14 0.25 0.97  12 0.0853 1111 P[7]=0.861  0.0033 0.0021  167.82 -1786.18  0:01

(nsteps = 5)
Current Pjump:      0.59231  0.00826  0.13569  0.24842  0.97500
Current finetune:   0.01000  0.01000  0.01000  0.01000  0.01000
New     finetune:   0.02634  0.00025  0.00425  0.00807  0.49952

-10% 0.59 0.43 0.27 0.27 0.13  12 0.0512 1111 P[7]=0.916  0.0036 0.0017  153.10 -1782.15  0:02
...
 -5% 0.60 0.37 0.25 0.20 0.49   6 0.0249 1001 P[3]=0.559  0.0032 0.0023  158.61 -1787.85  0:04
...
  0% 0.61 0.32 0.27 0.14 0.26   6 0.0519 1001 P[3]=0.802  0.0038 0.0021  157.65 -1795.81  0:05
  5% 0.60 0.33 0.29 0.12 0.33   6 0.0174 1001 P[3]=0.867  0.0028 0.0026  156.63 -1787.23  0:07
 10% 0.60 0.32 0.29 0.11 0.33   6 0.0151 1001 P[3]=0.923  0.0028 0.0026  161.77 -1793.64  0:08
 15% 0.60 0.32 0.30 0.11 0.34   9 0.0225 1101 P[3]=0.834  0.0030 0.0024  167.09 -1791.24  0:10
...
100% 0.60 0.32 0.30 0.15 0.33   6 0.0312 1001 P[3]=0.632  0.0031 0.0024  169.88 -1804.09  0:43

   ^^ Pjump for MCMC moves ^^  Prj       P[model 3]  theta  tau lnprior  lnL
```

The starting species delimitation model is generated by choosing at random an interior node for collapsing.

After the chain has started, the five ratios after the % sign are the acceptance proportions for the conventional MCMC moves discussed above.

After the acceptance proportions for the MCMC moves, there are three numbers related to the rjMCMC move, highlighted in red above. The rest of the line shows the posterior mean for θ for the root (which is a parameter shared by all species-tree models) and the current log likelihood lnL.

The three numbers related to the rjMCMC move, "`12 0.0853 1111`" in the example, mean that the current model is 1111 (the full model of 5 species), and it has 12 parameters, and the rjMCMC move has the acceptance proportion 0.0853. In general the larger this proportion, the more efficient the rjMCMC algorithm is. However there is no optimal acceptance proportion for the rjMCMC move, and a value close to 0 may not necessarily mean a problem. If one model has posterior probability close to 1, the acceptance proportion should be near 0 as well. Thus both poor mixing of the rjMCMC algorithm and extreme posterior model probabilities can cause the acceptance proportion for the rjMCMC to be close to 0. It has been noted that if the rjMCMC algorithm is suffering from poor mixing, different starting species trees often lead to different results.

Next the posterior probability for the best species-tree model (the most frequently visited tree model up to now) is printed. In the example, "**`P[7]=0.861`**" means tree model 7 (1111) is the most favoured model, with the posterior at 0.861.

After the MCMC is finished, the program will summarize the sample. The output looks like the following. The seven delimitation models are listed again, together with their posterior and prior probabilities. The “Guide tree with posterior probability for presence of nodes” can be copied into TreeView.

```
Summarizing the species-delimitation sample in file mcmc.txt

Number of species-delimitation models =  7
      model    prior  posterior
    1  0000   0.14286   0.00000
    2  1000   0.14286   0.01520
    3  1001   0.14286   0.63180
    4  1100   0.14286   0.00250
    5  1101   0.14286   0.13410
    6  1110   0.14286   0.00990
    7  1111   0.14286   0.20650

[Note: Ancestral nodes in order:   6 tricowconundwoo  7 tricowcon  8 tricow  9 undwoo]

Guide tree with posterior probability for presence of nodes
(((tri, cow)#0.2164, con)#0.353, (und, woo)#0.9724)#1;
```

The MCMC sample file mcmc.txt. As the number of parameters changes when the rjMCMC moves between models, the mcmc sample file may not be very useful, so you can ignore it. Right now the header line is generated using the starting species tree and should be ignored. After the header line, each line of output has the following numbers, separated by TABs: iteration number, the number of parameters, the tree, the sampled parameter values, and lnL.

```
20 6 1001 0.000538 0.006272 0.013239 0.005056 0.001706 0.000075 -1027.982
```

For example, the above sample is taken when the chain in species model 1001, with 6 parameters. If you know the unix command grep, you can retrieve the lines for the same tree model to summarize the posterior for parameters in that model.

```
grep “6      1001” mcmc.txt > result.Tree1001.txt
```

In theory this should give you the same posterior as if you run analysis A00 with the species tree fixed at tree 1001. In practice I think it is simpler that you edit the Imap file and the control file to run analysis A00.

Notes about running the rjMCMC algorithms.

- The rjMCMC algorithms for species delimitation allow the chain to move from one model to another but can have mixing problems. Make sure you get very similar results from multiple runs using Algorithm0 and Algorithm1 and you get the same results whatever the starting species tree is. The starting species tree is printed on the monitor on a line like the following

  Starting species tree = 0000

  The starting tree is chosen by the program at random and will vary among runs. Make sure that some of your runs are started with the one-species model (0000, say), some from the fully resolved tree (1111, say), and some from other trees in between. If you get consistent results among runs with different starting trees and using the two algorithms, you are unlikely to have a convergence or mixing problem.
  If you have a computer with multicore, you can run those different combinations or replicates in different folders at the same time.

- You can compile a version of the program that takes the starting tree from the keyboard. Open the file bpp.c and search for the following block , which is right now commented out (inside the /* */). Remove the /* and */ so that the code becomes active. Recompile. The program will then ask for the node to collapse in the starting species tree. You then type a number at the keyboard that is between $s + 1$ and $2s$ if you have s species on the guide tree. You can then look at the starting species tree printed on the monitor to see the effect.

```
/*
is = sptree.nspecies;
printf("\nNode to collaps (a number between %d and %d, also %d)? ", sptree.nspecies+1, 2*sptree.nspecies-1, 2*sptree.nspecies);
scanf("%d", &is);
is--;
*/
```

- In medium-sized or large datasets with multiple loci, we have come across cases where the chain is stuck at the one-species model (0000, say), or have difficulty moving into the one-species model. The problem does not appear to occur in small datasets with one or two loci. If you have a similar problem, you may start the analysis with 1 locus, then 2 loci, etc. to observe how the results change (you can do this by changing nloci in the control file as there is no need to change the sequence file).
- The program used the burnin to adjust the step lengths for the proposals in the MCMC algorithm. If the rjMCMC stays in species model 0, which does not have any τ parameter, no information is collected during the burnin about the proposals to change τ and the automatically adjusted step length for τ can be very poor.
- You probably need to evaluate the impact of the priors on θs and τ. If you don't have much info you can use $\alpha = 1.5$ or 2. However you need to use sensible prior means. See the note about the gamma distribution later in this document.

3.5 A11: joint species delimitation and species tree estimation

Again, we use the control file bpp.4s.ctl for to for joint species delimitation and species tree estimation (A11: speciesdelimitation = 1, speciestree = 1).

The screen output looks like this:

```
MCMC settings: 4000 burnin, sampling every 1, 100000 samples
Approximating posterior, using sequence data
(Settings: cleandata=1 print=1 saveconP=1 moveinnode=1)

Starting rjMCMC+SPR...
PrSplit = 0.500000
rj algorithm 1: new theta from G(a=2.00, m=1.00)
...

Initial parameters, np = 7 (gene trees are generated from the prior):

lnL0 =   -3534.627
 -3% 0.26 0.49 0.41 0.25 0.01   4 10 0.0020 0.1803 P(4)=1.000  0.0676 0.1457   18.49 -2103.26
 -1% 0.25 0.31 0.28 0.38 0.17   4 10 0.0000 0.0733 P(4)=1.000  0.0489 0.1731   28.61 -2098.88
  0% 0.26 0.32 0.33 0.26 0.40   4 10 0.0000 0.4410 P(4)=1.000  0.1057 0.0880    8.60 -2096.49  0:04
  5% 0.25 0.30 0.30 0.23 0.27   4 10 0.0089 0.5016 P(4)=0.914  0.1180 0.0670   24.32 -2099.73  0:08
 10% 0.25 0.30 0.29 0.27 0.28   4 10 0.0046 0.3275 P(4)=0.957  0.0903 0.1100   13.84 -2105.25  0:13
 15% 0.26 0.30 0.28 0.28 0.28   4 10 0.0031 0.2875 P(4)=0.971  0.0830 0.1217   36.43 -2107.22  0:18

     ^^ Pjump for MCMC moves ^^ S  p   Prj    PNNI P(S=4)       theta   tau0   lnprior  lnL
```

Here S is the number of species in the current model, and p is the number of parameters in the current model. Prj is acceptance proportion for the rjMCMC moves, while PNNI is acceptance proportion for the NNI or SPR moves. P(4)=0.971 means that up to now, the number of species 4 has the highest posterior probability (compared with 1 species, 2 species, etc.) and the probability is 0.971.

The next two numbers are posterior means of θ for the root population and τ_0 for the root age. These are followed by the logarithms of the prior and the likelihood.

The MCMC sample file mcmc.txt has lines like the following.

```
((A#0.0501558: 0.0162935, (D#0.0129667: 0.00634715, B#0.0558265: 0.00634715)#0.0381929:
0.00994634)#0.0375784: 0.00372412, C#0.0199665: 0.0200176)#0.133123; 4
(((D: 0, B: 0)#0.0324587: 0.00928457, C#0.0169688: 0.00928457)#0.0476308: 0.00772763,
A#0.0426255: 0.0170122)#0.161703; 3
```

The numbers after : are branch lengths (τs) while those after # are population size parameters (θs). Zero-length branches represent collapsed nodes, meaning that the descendent populations at that node belong to the same species. In the first sampled above, there are four distinct species: A, B, C, and D, while in the second, there are three: A, BD, and C.

The summary of the sample by BPP has four sections, as follows.

```
The bpp summary of the MCMC sample file looks like this.
Summarizing the species-tree sample in file mcmc.txt
read tree 100000  (D, ((A, B), C));
(A) List of best models (count postP #species SpeciesTree)
 46601  0.46601  0.46601   4 (A B C D)   (D, (C, (A, B)));     1100 1110
 35339  0.35339  0.81940   4 (A B C D)   (C, (D, (A, B)));     1100 1101
  7749  0.07749  0.89689   4 (A B C D)   ((C, D), (A, B));     0011 1100
  2686  0.02686  0.92375   4 (A B C D)   (C, (A, (B, D)));     0101 1101
  1470  0.01470  0.93845   4 (A B C D)   (A, (C, (B, D)));     0101 0111
  1183  0.01183  0.95028   4 (A B C D)   (C, (B, (A, D)));     1001 1101
  1129  0.01129  0.96157   4 (A B C D)   ((B, D), (A, C));     0101 1010
   869  0.00869  0.97026   4 (A B C D)   ((B, C), (A, D));     0110 1001
   551  0.00551  0.97577   4 (A B C D)   (A, (D, (B, C)));     0110 0111
   484  0.00484  0.98061   4 (A B C D)   (D, (A, (B, C)));     0110 1110
   415  0.00415  0.98476   4 (A B C D)   (B, (C, (A, D)));     1001 1011
   349  0.00349  0.98825   4 (A B C D)   (A, (B, (C, D)));     0011 0111
   281  0.00281  0.99106   3 (A BC D)    (BC, (A, D));         101
   278  0.00278  0.99384   4 (A B C D)   (D, (B, (A, C)));     1010 1110
   212  0.00212  0.99596   4 (A B C D)   (B, (A, (C, D)));     0011 1011
   169  0.00169  0.99765   4 (A B C D)   (B, (D, (A, C)));     1010 1011
    76  0.00076  0.99841   3 (A BC D)    (A, (BC, D));         011

(B)  4 species delimitations & their posterior probabilities
  99484   0.99484   4 (A B C D)
    420   0.00420   3 (A BC D)
     74   0.00074   3 (A BD C)
     22   0.00022   3 (A B CD)

(C)  7 delimited species & their posterior probabilities
 100000   1.00000  A
  99904   0.99904  D
  99558   0.99558  C
  99506   0.99506  B
    420   0.00420  BC
     74   0.00074  BD
     22   0.00022  CD

(D) Posterior probability for # of species
P[ 3] =   0.00516  prior[ 3] =  0.28571
P[ 4] =   0.99484  prior[ 4] =  0.23810
```

Section (A) lists the best models in the decreasing order of the posterior probabilities. Here a model is a full MSC model that species both the species delimitation and species phylogeny. From this section, one can easily construct the 95% or 99% credibility sets of models. This section is further summarized to produce sections B, C, and D. Section (B) gives the posterior probabilities for the top few delimitations. For example, the probability (0.99484) for 4 species is a sum of the posterior probabilities for all the 15 rooted trees for the four species. Section (C) lists the delimited species and their posterior probabilities, and section

(D) lists the posterior probability for the number of species, together with the prior probabilities calculated by BPP.

4. Example data files

Below are some notes about the example datasets included in the package. Those can be used to duplicate previous results concerning .

(i) The dataset of Yu et al. (2001), with one species (yu2001.bpp.ctl and yu2001.txt). This can be used to estimate θ for a locus from human individuals. The results are in table 3 of Rannala & Yang (2003), with $\hat{\theta}_H$ = 0.00035, the μt_{MRCA} = 0.00031.

```
cd examples
../bpp yu2001.bpp.ctl
```

(ii) The hominoid data of Chen & Li (2001). This includes one sequence from each of human, chimpanzee, gorilla, and orangutan, at 53 loci. The data are used to estimate three θ parameters (θ_{HC}, θ_{HCG}, θ_{HCO}) and three species divergence time parameters (τ_{HC}, τ_{HCG}, τ_{HCGO}) on the species tree (((H, C), G), O).

```
cd examples
../bpp ChenLi2001.bpp.ctl
```

The results should be compared with those in the column "Posterior (53 loci)" in table 2 of Rannala & Yang (2003). However, the speciation times are now 'parametrized' differently. Rannala & Yang used $\tau_{\text{HCGO}} - \tau_{\text{HCG}}$, $\tau_{\text{HCG}} - \tau_{\text{HC}}$, and τ_{HC} while bpp now uses τ_{HCGO}, τ_{HCG}, and τ_{HC} in both prior specification and output. Furthermore, the specification of priors has changed as well. Rannala & Yang (2003) used a separate gamma prior for each θ parameter, while bpp now uses the same gamma prior for all θs. Also bpp assigns a gamma prior for the root age (τ for the root node) and a Dirichlet distribution prior to generate the other node ages (τs). For comparison, the results are listed below in table 1.

TABLE 1 Prior and posterior distributions of parameters in the Bayesian analysis of the 53 loci of Chen and Li (2001)

	Gamma prior	Prior	Posterior (53 loci)
Parameter	(α, β)	Mean (95% interval)	Mean (95% interval)
θ_{HC}	(2, 2000)	0.001 (0.00012, 0.00279)	0.00187 (0.00051, 0.00392)
θ_{HCG}		0.001 (0.00012, 0.00279)	0.00349 (0.00216, 0.00500)
θ_{HCGO}		0.001 (0.00012, 0.00279)	0.00199 (0.00026, 0.00450)
τ_{HCGO}	(14, 1000)	0.01415 (0.00758, 0.02227)	0.01395 (0.01230, 0.01541)
τ_{HCG}		0.00956 (0.00246, 0.01852)	0.00593 (0.00515, 0.00681)
τ_{HC}		0.00471 (0.00018, 0.01340)	0.00483 (0.00395, 0.00562)

NOTE .— Both τ and θ are measured as the expected number of mutations per site.

(iii) The dataset of North American fence lizards *Sceloporus* (Leaché, 2009) is used for species delimitation. The files are lizard.txt, lizard.Imap.txt, and lizard.bpp.ctl. This is one of the datasets analyzed by Yang & Rannala (2010). To duplicate the results, type

```
cd examples
../bpp lizard.bpp.ctl
```

The results for one locus are shown below in table 2. In the paper we used cleandata = 1. You can change the rjMCMC algorithm as well as the fine-tune parameters to see how the algorithm behaves.

TABLE 2 Posterior probabilities for different tree models for the lizard data (1 locus)

Tree	cleandata = 1	cleandata = 0
0000	0.006	0.006
1000	0.020	0.086
1001	0.088	0.413
1100	0.039	0.026
1101	0.168	0.100
1110	0.129	0.084
1111	0.549	0.284

(iv) The Coast horned lizard dataset of Leaché et al. (2009) is used in Rannala & Yang (2013) and Yang & Rannala (2014). The files are in the Cavefish folder.

(v) The North American cavefish dataset of (Niemiller et al., 2011) is used in Rannala & Yang (2013) and Yang & Rannala (2014). The files are in the Cavefish folder.

(vi) The East Asian brown frog data of Zhou et al. (2012) is used in the tutorial of Yang (2015).

5. The gamma prior

The gamma distribution is used in BPP to specify priors for parameters, partly because all parameters involved in the model are strictly positive. The gamma $G(\alpha, \beta)$ has mean $m = \alpha/\beta$ and variance $s^2 = \alpha/\beta^2$. (Note that in the literature another commonly used parametrization of the gamma gives the mean as $\alpha\beta$.) It may be easier to think of mean m and standard deviation s when constructing the prior, and then get α and β as $\alpha = (m/s)^2$ and $\beta = m/s^2$. For example, as the divergence time between humans and chimpanzees is ~5MY, and the mutation rate is ~10^{-9} per site per year, the mean of τ_{HC} should be ~0.005. To use a fairly informative prior, we may set $\alpha = 25$ and then $\beta = \alpha/m = 5000$.

If little information is available about the parameter, we may want to use a diffuse prior. We can fix m first, and then adjust s or α, with a large s or small α meaning a less informative prior. In a normal distribution, the 95% interval is given roughly as $m \pm 2s$. This rule does not apply well to the gamma distribution if α is small (say, $\alpha < 1$), but may still be used as a guide. With this reasoning, $s = m$ or $\alpha = 1$ represent a very diffuse prior while $s = m/2$ or $\alpha = 4$ is still quite diffuse. Thus values like 1.5 or 2 for α are fairly diffuse, and you can then choose β to get the mean roughly right. Do not use very small α (such as 0.1 or 0.01), as they may cause numerical problems.

For example, two random human sequences are different at ~0.06% of sites, with $\theta_{\mathrm{H}} = 0.0006 \approx 0.001$ (1 difference per kb). The values for chimpanzees and gorillas are larger. Then a reasonable prior for analysis of hominoid data may be

```
      thetaprior = 2 2000   # gamma(a, b) for theta
```

This prior is quite diffuse, in the neighborhood of 0.001. Flies and worms have larger population sizes. If their θvalues are in the neighborhood of 0.01, you can have

```
      thetaprior = 2 200   # gamma(a, b) for theta
```

6. The simulation program (MCcoal)

A simulation program MCcoal is included to simulate sequence alignments on the fixed species tree, that is, under the MSC model (Rannala and Yang, 2003). In addition, the simulation model allows migration between species/populations, even though the inference program bpp does not allow migration. Look at the README.txt file for compiling the program. To run the program, type one of the following. The default control file name is MCcoal.ctl. Always look at the screen output to confirm that the program reads the control file correctly.

```
MCcoal
MCcoal MCcoalMigration.ctl
```

6.1 Simulating without migration

```
           seed = 12345

        seqfile = MySeq.txt 0 * comment out this line if you don't want seqs
       treefile = MyTree.tre  * comment out this line if you don't want trees
       Imapfile = MyImap.txt

   species&tree = 4  A  B  C  D
                     3  2  1  1

   ((A #0.1, B #0.2) : 0.5 #.12, (C, D) :0.8 #0.34) : 1.0 #.1234;

    loci&length = 100 1000 * number of loci & number of sites at each locus
```

The example above (MCcoal.ctl) is for simulating 100 loci, each of 1000 sites, on the species tree ((A, B), (C, D)), with 3, 2, 1, 1 sequences for A, B, C, D, so that there are 7 sequences at each locus. The divergence time parameters (τs) are after ':' in the tree, while the population size parameters (θs) are after '#'. Thus we have $\theta_A = 0.1$, $\theta_B = 0.2$, $\theta_{AB} = 0.12$, $\theta_{CD} = 0.34$, $\theta_{ABCD} = 0.1234$, $\tau_{ABCD} = 1.0$, $\tau_{AB} = 0.5$, and $\tau_{CD} = 0.8$. We need θ_A and θ_B because 2 or more sequences are sampled from species A and B. Parameters θ_C and θ_D are unnecessary: if you specify them, they will be ignored by the program. (Those parameter values are too big. Replace them with more realistic values.)

By default MCcoal prints out the sequence alignment at each locus, but if you use file format 1 (seqfile = MySeq.txt 1), the program will print out site pattern counts instead. The file may then be smaller. This format is readable by bpp and 3s, but perhaps not by other programs.

6.2 Simulating with migration

```
migration = 7   * number of pops (order fixed by program)

        A     B     C     D    ABCD    AB   CD
 A      0     1.1   1.2   1.3    0     0    -1
 B      0.1   0     1.4   1.5    0     0    -1
 C      0.2   0.4   0     1.6    0     1.7   0
 D      0.3   0.5   0.6   0      0     1.8   0
 ABCD   0     0     0     0      0     0     0
 AB     0     0     0.7   0.8    0     0     1.9
 CD    -1    -1     0     0      0     0.9   0
```

The control file for simulating migration as well as coalescence includes a block as above (see the file MCcoalMigration.ctl). The line migration = 7, where 7 is the number of populations on the species tree, tells the program to simulate migrations. This number is fixed by the species tree and is used here for error checking. This is then followed by a migration matrix, of size 7 × 7. The names of the populations are read and ignored by the program, and the order of the populations is fixed. You should run the program without the migration matrix first and then use the screen output to use the correct order of populations to specify the migration matrix.

In the migration matrix, values 0 and −1 mean that migration is either impossible (for example, if the two populations did not live at the same time) or not allowed (if the two populations were contemporary but no migration between them is assumed to occur). Use 0 for both cases here. Positive values are scaled migration rates, with the element on the *i*th row *j*th column to be $M_{ij} = N_j m_{ij}$, where m_{ij} is the migration rate per generation in population *j* from population *i*, or the proportion of individuals in population *j* that are immigrants from population *i*, and where μ is the mutation rate per site per generation. In the example above, $M_{\mathrm{A}\to\mathrm{B}} = 1.1$, which means that on average 1.1 individuals are immigrants from population A to population B.

Note that the θ parameter for a modern species has to be specified in the tree file even if only one sequence is sampled from that species but migrations into that species are allowed by the migration matrix.

7. References

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

Chen, F.-C., and W.-H. Li. 2001. Genomic divergences between humans and other Hominoids and the effective population size of the common ancestor of humans and chimpanzees. Am. J. Hum. Genet. 68:444-456.

Hey, J., and R. Nielsen. 2004. Multilocus methods for estimating population sizes, migration rates and divergence time, with applications to the divergence of *Drosophila pseudoobscura* and *D. persimilis*. Genetics 167:747-760.

Jukes, T. H., and C. R. Cantor. 1969. Evolution of protein molecules. Pages 21-123 *in* Mammalian Protein Metabolism (H. N. Munro, ed.) Academic Press, New York.

Kingman, J. F. C. 1982. The coalescent. Stochastic Process Appl. 13:235-248.

Leaché, A. D. 2009. Species tree discordance traces to phylogeographic clade boundaries in North American fence lizards (sceloporus). Syst. Biol. 58:547-559.

Leaché, A. D., M. S. Koo, C. L. Spencer, T. J. Papenfuss, R. N. Fisher, and J. A. McGuire. 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.

Niemiller, M. L., T. J. Near, and B. M. Fitzpatrick. 2011. Delimiting species using multilocus data: diagnosing cryptic diversity in the southern cavefish, Typhlichthys subterraneus (Teleostei: Amblyopsidae). Evolution 66:846-866.

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

Rannala, B., and Z. Yang. 2013. Improved reversible jump algorithms for Bayesian species delimitation. Genetics 194:245-253.

Takahata, N., Y. Satta, and J. Klein. 1995. Divergence time and population size in the lineage leading to modern humans. Theor. Popul. Biol. 48:198-221.

Yang, Z. 2002. Likelihood and Bayes estimation of ancestral population sizes in Hominoids using data from multiple loci. Genetics 162:1811-1823.

Yang, Z. 2006. *Computational Molecular Evolution*. Oxford University Press, Oxford, UK.

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.

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

Yang, Z., and B. Rannala. 2014. Unguided species delimitation using DNA sequence data from multiple loci. Mol. Bio.l Evol. 31:3125-3135.

Yu, N., Z. Zhao, Y. X. Fu, N. Sambuughin, M. Ramsay, T. Jenkins, E. Leskinen, L. Patthy, L. B. Jorde, T. Kuromori, and W. H. Li. 2001. Global patterns of human DNA sequence variation in a 10-kb region on chromosome 1. Mol. Biol. Evol. 18:214-222.

Zhou, W. W., Y. Wen, J. Fu, Y. B. Xu, J. Q. Jin, L. Ding, M. S. Min, J. Che, and Y. P. Zhang. 2012. Speciation in the *Rana chensinensis* species complex and its relationship to the uplift of the Qinghai-Tibetan Plateau. Mol. Ecol. 21:960-973.