Supplementary material for 'DISSECT: an assignment-free Bayesian discovery method for species delimitation under the multispecies coalescent'

Graham Jones

2014-02-17

Contents

1 Notation

Notation	Description
	General
$\mathbf{1}_A$	the indicator function for set A.
B_n	Bell numbers.
	Evolutionary model
n	the number of individuals, and therefore the maximum possible number of species.
k	the number of species.
t	the origin time.
$f()$	the node density.
$m()$	the ‘mixin’ density.
$q()$	the prior density for t.
λ	the speciation rate.
μ	the extinction rate.
α, β	$\alpha = \lambda - \mu$ and $\beta = \mu/\lambda$.
	User-chosen speciation parameters
w	the ‘collapsing weight’.
ϵ	the ‘collapsing height’.
τ	the speciation threshold.
	Simulations, results
G	the number of loci.
T	the mutation rate in substitutions per site per generation.
Ω	the table of clusterings $Z_1, Z_2, \dots Z_z$ with corresponding posterior probabilities $p_1, p_2, \dots p_z$.
$Z^\star$	the true clustering.
M	an estimated similarity matrix.
$M^\star$	the true similarity matrix.
	Rand index definition
$S = \{o_1, \dots, o_n\}$	a set of n objects.
$X = \{X_1, \dots, X_x\}$,	a clustering of S into x subsets (clusters).
$Y = \{Y_1, \dots, Y_y\}$,	a clustering of S into y subsets (clusters).
$f_1(Z, i, j)$	defined as 1 if i and j are in the same cluster in the clustering Z and 0 otherwise.
$f_2(X, Y, j, k)$	defined as 1 1 if $f_1(X, i, j)$ is equal to $f_1(Y, i, j)$, else 0.

2 Figures and tables

These include some figures in the main text as well as extra ones, for convenient comparison. We define an alternative point estimator to be the clustering $\bar{Z}$ in Ω which minimises $D_{mat}(\bar{Z}, M(\Omega))$.

2.1 Varying epsilon

Note that the scales of the x-axes differ between Figs 2 and 3.

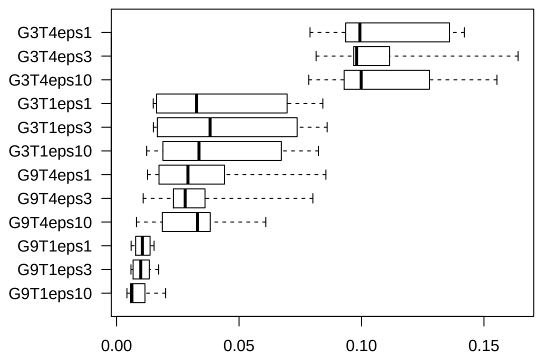

Figure 1: The boxplots show the values of the error metric $D(\Omega, Z^{\star})$ over ten replicates as G, T, and ϵ vary. In the labels, G3T4eps1 means the number of genes G is 3, the mutation rate T is 4e-8, and ϵ is 1e-5 = 0.00001. The other values of ϵ are 0.00003 and 0.0001. A beta prior with shape parameters 8 and 2 was used for w which is estimated.

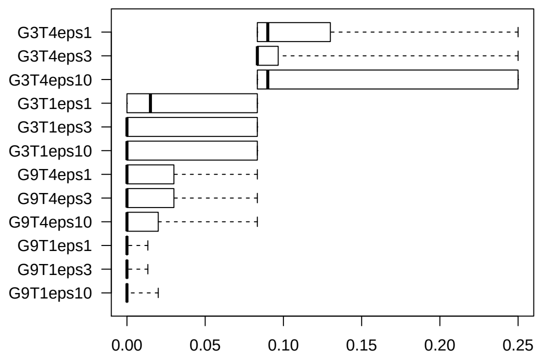

Figure 2: Rand metric between point estimate $\hat{Z}$ and true clustering, as G, T, and ϵ vary.

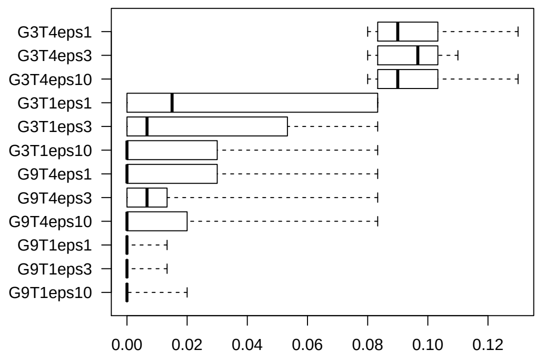

Figure 3: Rand metric between point estimate $\bar{Z}$ and true clustering, as G, T, and ϵ vary.

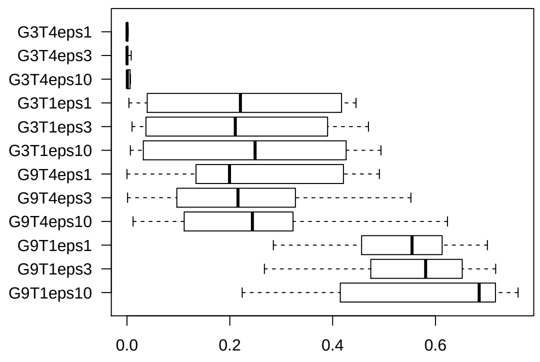

Figure 4: Posterior probability of true clustering, as G, T, and ϵ vary.

	eps10	eps3	eps1
G3T4	2	5	4
G3T1	0	0	2
G9T4	0	0	1
G9T1	0	0	0

Table 1: Number of times out of 10 that the true clustering failed to be in the 0.95 credible set from a MCMC run over 20 mlliion generations with the first 10 million discarded as burnin., as G, T, and ϵ vary.

2.2 Varying w

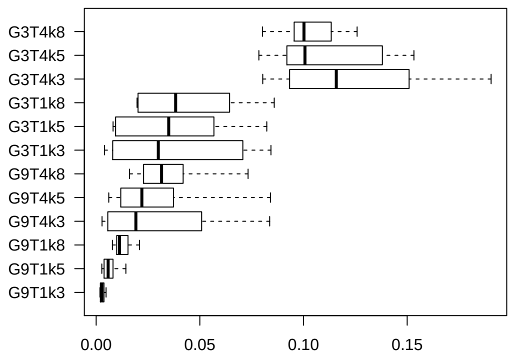

Figure 5: $D(\Omega, Z^{\star})$ as G, T, and w vary. The numbers after the 'k' in the label are the prior mean values for the number of species which are affected by the value of w The value of ϵ is 0.0001.

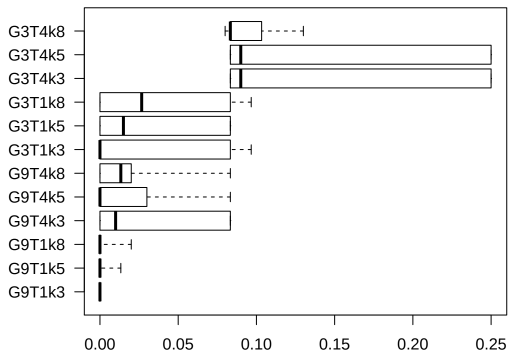

Figure 6: Rand metric between point estimate $\hat{Z}$ and true clustering, as G, T, and w vary.

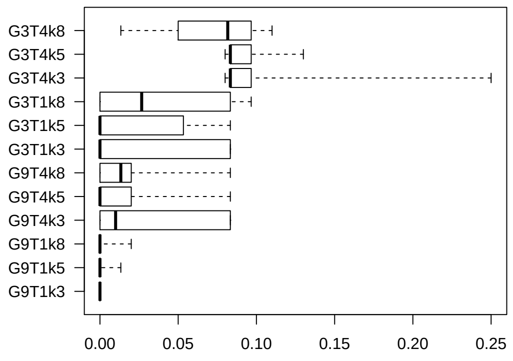

Figure 7: Rand metric between point estimate $\bar{Z}$ and true clustering, as G, T, and w vary.

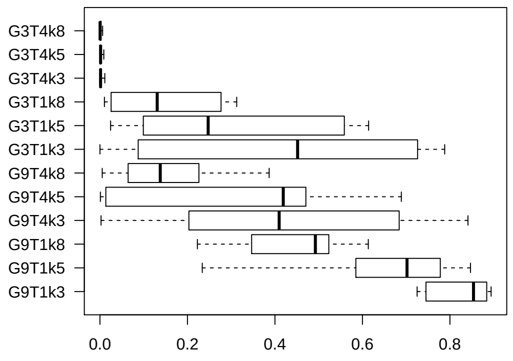

Figure 8: Posterior probability of true clustering, as G, T, and w vary.

	k3	k5	k8
G3T4	4	3	2
G3T1	2	0	0
G9T4	1	1	0
G9T1	0	0	0

Table 2: Number of times out of 10 that the true clustering failed to be in the 0.95 credible set, as G, T, and w vary.

2.3 Single species

Note that all the Rand metrics between $\hat{Z}$ and true clustering, and $\bar{Z}$ and true clustering, were zero, and the true clustering was always in the 0.95 credible set.

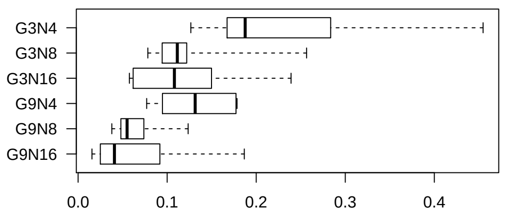

Figure 9: The true clustering is a single cluster. The plot shows $D(\Omega, Z^{\star})$ for varying number of individuals n and genes G. The value of ϵ is 3e-5. A beta prior with shape parameters n-1 and 1 was used for w which is estimated.

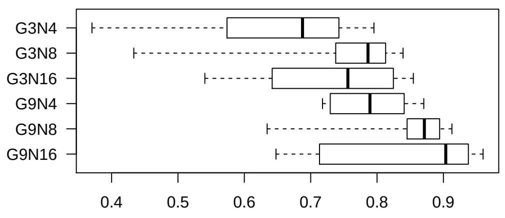

Figure 10: Posterior probability of true clustering, as n and G vary.

2.4 Thomomys data set

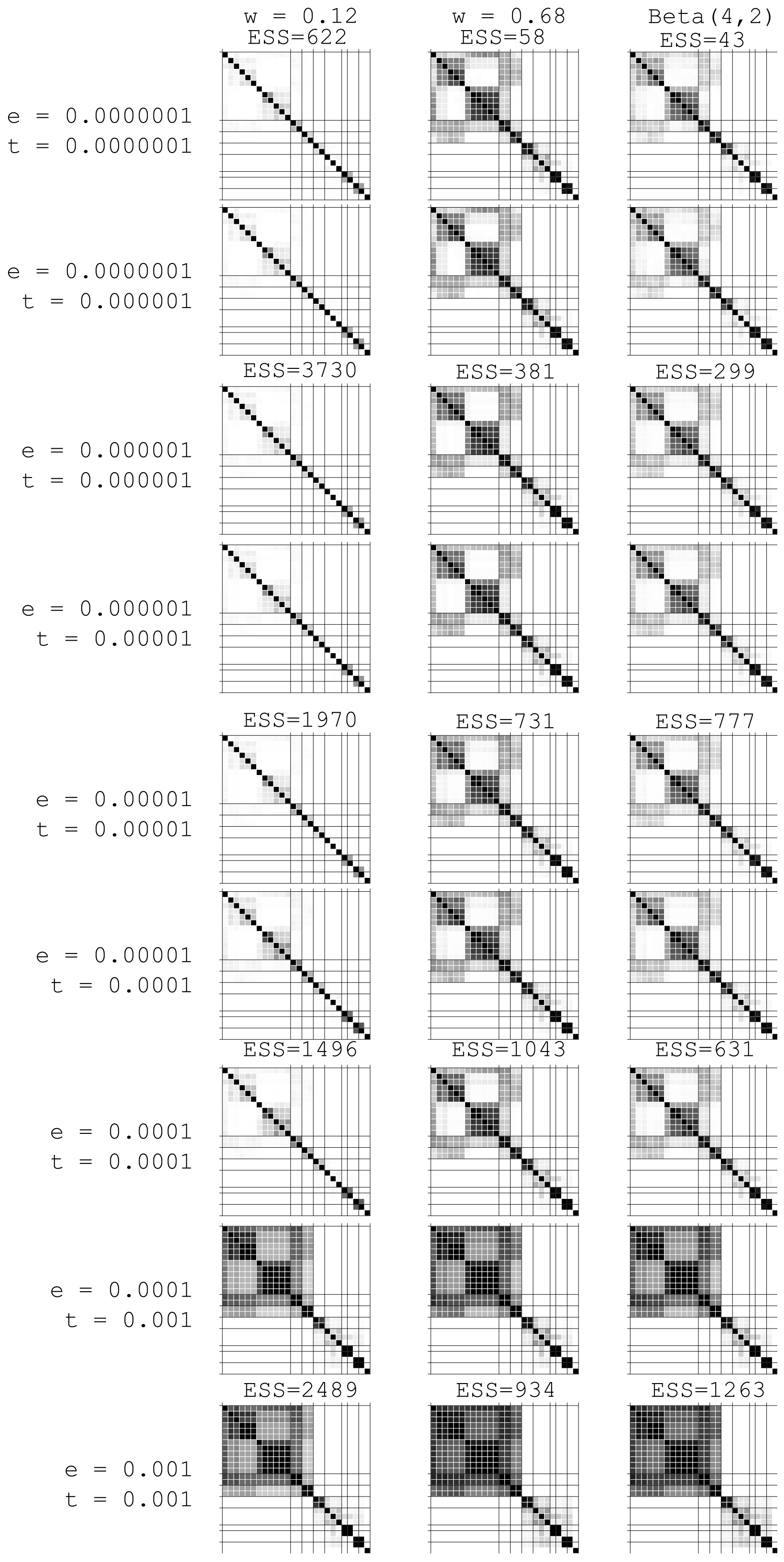

Figure 11: Similarity matrices for Thomomys data set under various ϵ (e), τ (t) and collapse weight (w) values.

3 Cluster analysis for systematists

Species delimitation is a type of cluster analysis. Cluster analysis is used in many areas of science, including machine learning and data mining, as well as various areas within bioinformatics, but it may be unfamiliar to some systematists. Different areas of science use different terminology, and one potential confusion seems worth pointing out. In most areas of science a sharp distinction is made between classification and cluster analysis. 'Classification' is used to mean *only* the assignment of new specimens to one of a fixed set of *known* classes. The assignment of a handwritten symbol to one letter in an alphabet is a typical example of classification. Cluster analysis means the discovery or invention of clusters among specimens. These clusters might later be regarded as classes into which new specimens can be classified, but this is no longer cluster analysis.

There is a huge variety of clustering algorithms, and also many ways of evaluating clusterings, and measuring the differences between them. In this paper, we use one of the simplest, the **Rand index** [6]:

> Given a set of n elements $S = \{o_1, \ldots, o_n\}$ and two partitions of S to compare, $X = \{X_1, \ldots, X_r\}$, a partition of S into r subsets, and $Y = \{Y_1, \ldots, Y_s\}$, a partition of S into s subsets, define the following:
>
> - a, the number of pairs of elements in S that are in the same set in X and in the same set in Y
> - b, the number of pairs of elements in S that are in different sets in X and in different sets in Y
> - c, the number of pairs of elements in S that are in the same set in X and in different sets in Y
> - d, the number of pairs of elements in S that are in different sets in X and in the same set in Y
>
> The Rand index, R, is:
>
> $$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$

This comes from Wikipedia where more information can be found.

3.1 Online resources

`http://en.wikipedia.org/wiki/Cluster_analysis`
`http://en.wikipedia.org/wiki/Rand_measure`
A book by Jain and Dubes [5]:
`homepages.inf.ed.ac.uk/rbf/BOOKS/JAIN/Clustering_Jain_Dubes.pdf`

4 Calculations

4.1 Node height density

Conditioned on k, t, λ, and μ the density of an unordered node height s is given by Theorem 2.5 of [3] as

$$f(s|k,t,\lambda,\mu) = \frac{(\lambda-\mu)^2 e^{-(\lambda-\mu)s}}{(\lambda-\mu e^{-(\lambda-\mu)s})^2} \frac{\lambda-\mu e^{-(\lambda-\mu)t}}{1-e^{-(\lambda-\mu)t}} \mathbf{1}_{[0,t]} \qquad (1)$$

Setting $\alpha = \lambda - \mu$ and $\beta = \mu/\lambda$ so that $\mu = \beta\lambda$ and $\lambda = \alpha/(1-\beta)$, this can be rewritten as

$$f(s|k,t,\lambda,\mu) = \frac{\alpha(1-\beta)e^{-\alpha s}}{(1-\beta e^{-\alpha s})^2}\frac{1-\beta e^{-\alpha t}}{1-e^{-\alpha t}}\mathbf{1}_{[0,t]} \tag{2}$$

where s is a node height. Note that k does not appear on the right hand side.

4.2 Origin height density

Usually an improper uniform prior on $[0,\infty)$ is assumed for the origin time t of the tree, which is then conditioned on the number of species k. This conditional density for t is shown in Theorem 3.2 of [3] to be

$$\begin{aligned} q(t|k) &= k\lambda^k(\lambda-\mu)^2\frac{(1-e^{-(\lambda-\mu)t})^{k-1}e^{-(\lambda-\mu)t}}{(\lambda-\mu e^{-(\lambda-\mu)t})^{k+1}} \\ &= k\alpha(1-\beta)e^{-\alpha t}\frac{(1-e^{-\alpha t})^{k-1}}{(1-\beta e^{-\alpha t})^{k+1}} \end{aligned} \tag{3}$$

Using the probabilities from expression (3) in the main paper (the one starting $(|C(k)|)$...), the prior density for t is

$$q(t) = \sum_{k=1}^{n}\binom{n-1}{k-1}(1-w)^{k-1}w^{n-k}q(t|k) \tag{4}$$

This can be expressed as

$$\begin{aligned} q(t) &= \sum_{j=0}^{n-1}\binom{n-1}{j}(1-w)^j w^{n-1-j}q(t|j+1) \\ &= \sum_{j=0}^{n-1}\binom{n-1}{j}(1-w)^j w^{n-1-j}(j+1)\alpha(1-\beta)e^{-\alpha t}\frac{(1-e^{-\alpha t})^j}{(1-\beta e^{-\alpha t})^{j+2}} \\ &= \alpha(1-\beta)e^{-\alpha t}(1-\beta e^{-\alpha t})^{-2}\sum_{j=0}^{n-1}(j+1)\binom{n-1}{j}w^{n-1-j}\left(\frac{(1-w)(1-e^{-\alpha t})}{1-\beta e^{-\alpha t}}\right)^j \\ &= a\sum_{j=0}^{n-1}(j+1)\binom{n-1}{j}w^{n-1-j}b^j \end{aligned}$$

where

$$a = \alpha(1-\beta)e^{-\alpha t}(1-\beta e^{-\alpha t})^{-2} \quad \text{and} \quad b = (1-w)(1-e^{-\alpha t})(1-\beta e^{-\alpha t})^{-1}. \tag{5}$$

Then

$$\begin{aligned} q(t) &= a\left(\sum_{j=0}^{n-1}\binom{n-1}{j}w^{n-1-j}b^j + \sum_{j=0}^{n-1}j\binom{n-1}{j}w^{n-1-j}b^j\right) \\ &= a\left((w+b)^{n-1} + \sum_{i=0}^{n-2}(i+1)\binom{n-1}{i+1}w^{n-2-i}b^{i+1}\right) \\ &= a\left((w+b)^{n-1} + b\sum_{i=0}^{n-2}(n-1)\binom{n-2}{i}w^{n-2-i}b^i\right) \\ &= a\left((w+b)^{n-1} + (n-1)b(w+b)^{n-2}\right) \\ &= a(w+b)^{n-2}\left((w+b)+(n-1)b\right) \\ &= a(w+b)^{n-2}(w+nb). \end{aligned} \tag{6}$$

Equations (1) and (6) provide the basis for a new class `BirthDeathCollapseModel` extending `SpeciationModel`. It contains a parameter for the origin height t as well as for α and β as in the usual birth-death model.

4.3 Distance between similarity matrices

We start with an explicit formula for the Rand index. Suppose there is a set of n objects $S = \{o_1, \ldots, o_n\}$ and two clusterings of S to compare, $X = \{X_1, \ldots, X_x\}$, a clustering of S into x clusters; and $Y = \{Y_1, \ldots, Y_y\}$, a clustering of S into y clusters. For any clustering Z of S, and $1 \leq i < j \leq n$, define $f_1(Z, i, j)$ to be 1 if i and j are in the same cluster in the clustering Z and 0 otherwise. Next define $f_2(X, Y, i, j)$ to be 1 if $f_1(X, i, j)$ is equal to $f_1(Y, i, j)$, and 0 otherwise. In words, $f_2(X, Y, i, j)$ is 1 if the two clusterings X and Y 'agree' on the pair (i, j) and 0 if they 'disagree'. Then the Rand index is:

$$R(X, Y) = \binom{n}{2}^{-1} \sum_{i,j:i>j}^{n} f_2(X, Y, i, j)$$

Recall that $M = M(\Omega)$ is the similarity matrix and $M^\star$ is the true similarity matrix.

$$\begin{aligned}
D_{mat}(M, M^\star) &= \binom{n}{2}^{-1} \sum_{i,j:i>j}^{n} |M_{i,j} - M^\star_{i,j}| \\
&= \binom{n}{2}^{-1} \sum_{i,j:i>j}^{n} \left| \left(\sum_{m=1}^{z} p_m f_1(Z_m, i, j) \right) - f_1(Z^\star, i, j) \right| \\
&= \binom{n}{2}^{-1} \sum_{i,j:i>j}^{n} \left| \sum_{m=1}^{z} p_m f_1(Z_m, i, j) - \sum_{m=1}^{z} p_m f_1(Z^\star, i, j) \right| \\
&= \binom{n}{2}^{-1} \sum_{i,j:i>j}^{n} \left| \sum_{m=1}^{z} p_m \big(f_1(Z_m, i, j) - f_1(Z^\star i, j) \big) \right| \\
&= \binom{n}{2}^{-1} \sum_{i,j:i>j}^{n} \sum_{m=1}^{z} p_m \big(1 - f_2(Z_m, Z^\star, i, j) \big) \\
&= \sum_{m=1}^{z} p_m \binom{n}{2}^{-1} \sum_{i,j:i>j}^{n} \big(1 - f_2(Z_m, Z^\star, i, j) \big) \\
&= \sum_{m=1}^{z} p_m \left(1 - \binom{n}{2}^{-1} \sum_{i,j:i>j}^{n} f_2(Z_m, Z^\star, i, j) \right) \\
&= \sum_{m=1}^{z} p_m \left(1 - R(Z_m, Z^\star) \right) \\
&= \sum_{m=1}^{z} p_m \bar{R}(Z_m, Z^\star) \\
&= D(\Omega). \qquad (7)
\end{aligned}$$

5 Usage

The model is implemented in the development version v1.8.0pre of BEAST ([1], [2]), more specifically the multispecies coalescent model *BEAST [4].

5.1 Running a Dissect analysis

BEAUTi can be used to set up most of the analysis. Each species should only be assigned individuals which definitely belong together, since the program will consider merging but never splitting these minimal clusters. The word 'species', as it appears in BEAUTi and in the BEAST XML, should be interpreted as a minimal cluster.

Two changes need to be made to the XML. The `birthDeathModel` must be replaced with a `birthDeathCollapseModel`, and an operator must be added for the origin height.

The value of w can be set using the element `collapseWeight`. It can either be fixed or estimated. Even when w is fixed, the prior on the number of species k is somewhat diffuse. If you want to estimate w, you can add a hyperprior and an operator for it. The model does not permit k to have a more concentrated prior than the binomial distribution decribed above, but you can obtain a more diffuse prior on k with a suitable hyperprior for w.

The value of ϵ is set in the `birthDeathModel` using the attribute `collapseHeight`.

Optionally, `tmrcaStatistics` may be added for groups of interest. A statistic `bdcNClustersStatistic` for the number of clusters can also be added. A trace of the latter can indicate if the MCMC chain is having trouble changing the number of almost collapsed nodes.

The Yule model for speciation can be used with Dissect, by setting the initial values of `species.birthDeathCollapse.relativeDeathRate` to 0, and removing the operator wich changes it.

XML for the birth-death-collapse model

```
<birthDeathCollapseModel id="birthDeathCollapse" units="substitutions"
              collapseHeight="0.0001">
  <speciesTree>
    <speciesTree idref="sptree"/>
  </speciesTree>
  <birthMinusDeathRate>
    <parameter id="species.birthDeathCollapse.meanGrowthRate"
                  value="100" lower="0.0" upper="Infinity"/>
  </birthMinusDeathRate>
  <relativeDeathRate>
    <parameter id="species.birthDeathCollapse.relativeDeathRate"
                          value="0.5" lower="0.0" upper="1.0"/>
  </relativeDeathRate>
  <originHeight>
    <parameter id="species.birthDeathCollapse.originHeight"
                value="0.2" lower="0.0" upper="Infinity"/>
  </originHeight>
  <collapseWeight>
    <parameter id="species.birthDeathCollapse.collapseWeight"
                        value="0.5" lower="0.0" upper="1.0"/>
  </collapseWeight>
</birthDeathCollapseModel>
```

XML for the operator for origin height

```
<scaleOperator scaleFactor="0.75" weight="3">
  <parameter idref="species.birthDeathCollapse.originHeight"/>
</scaleOperator>
```

XML for the operator for w (if w is estimated)

```
<scaleOperator scaleFactor="0.75" weight="3">
  <parameter idref="species.birthDeathCollapse.collapseWeight"/>
</scaleOperator>
```

XML for a number-of-clusters statistic

```
<bdcNClustersStatistic id="nClusters" name="nClusters">
  <collapseModel>
    <birthDeathCollapseModel idref="birthDeathCollapse"/>
  </collapseModel>
  <speciesTree>
    <speciesTree idref="sptree"/>
  </speciesTree>
</bdcNClustersStatistic>
```

XML for a tmrca statistic

```
<tmrcaStatistic id="speciesTree.talpoidesHeight" name="speciesTree.talpoidesHeight">
  <speciesTree idref="sptree"/>
  <mrca>
    <taxa>
      <sp idref="Thomomys_talpoides1"/>
      <sp idref="Thomomys_talpoides2"/>
      <sp idref="Thomomys_talpoides3"/>
     </taxa>
  </mrca>
</tmrcaStatistic>
```

XML for a prior and an operator for w (if w is estimated)
This goes in the `prior` element, in the `mcmc` element:

```
<betaPrior shape="5" shapeB="2">
  <parameter idref="species.birthDeathCollapse.collapseWeight"/>
</betaPrior>
```

This goes in the `operators` element:

```
<scaleOperator scaleFactor="0.75" weight="3" >
  <parameter idref="species.birthDeathCollapse.collapseWeight" />
</scaleOperator>
```

5.2 Using SpeciesDelimitationAnalyser

You can run this tool using a command line like that for BEAST, with `dr.app.beast.BeastMain` replaced by `dr.app.tools.SpeciesDelimitationAnalyser`.

The last two arguments are always an input file name and an output filename. The input file is the MCMC samples of species trees. The output is a table which lists the clusterings which are found when the nodes with small height have been collapsed. There are three optional arguments, shown in this example:

```
SpeciesDelimitationAnalyser -burnin 10000 -collapseheight .001
  -simcutoff .95 treesamples.txt out.txt
```

- burnin is the number of samples to be considered as 'burn-in' which will be ignored.
- collapseheight is the height below which nodes get collapsed. This would normally be equal to or larger then the value of ϵ used in the BEAST analysis.
- simcutoff is the value above which two clusterings are regarded as similar enough to support one another's credibility. It may be useful to change this when summarising the results as a single clustering. The similarity between two clusterings is the Rand index, and is a value between 0 and 1. The idea is that a clustering may not be very common among the samples, but may be similar to a lot of other clusterings, and the value of simcutoff affects which clustering is regarded as the 'best' summary of the results. The summary is somewhat analogous to a maximum clade credibility tree. Smaller values of simcutoff mean taking into account more dissimilar clusterings. This option should be regarded as experimental: set it to 1.0 if you want to ignore it.

Here is an imaginary example of the output to illustrate the format.

```
count   fraction   similarity   nclusters  a  b  c  d
6000    0.6        6000.0       1          0  0  0  0
3000    0.3        3000.0       2          0  1  1  1
1000    0.1        1000.0       3          2  0  1  2
```

The count column is the number of samples containing the clustering. The fraction is this count divided by the number of samples. The similarity column is a measure of how similar the clustering is to others.

The remaining columns describe a clustering. The nclusters column is the number of clusters. The other columns are labelled with the names of the 'species' (minimal clusters) used in the BEAST XML for the analysis. The numbers in the columns show how these are grouped; the numbers themseves are arbitrary, but if two columns have the same value, the minimal clusters are grouped together in the clustering. So the table above shows clusterings **abcd**, **a+bcd** and **ad+b+c**.

The Rand indices between these clusterings are $R(abcd, a + bcd) = 3/6$, $R(abcd, ad + b + c) = 2/6$, $R(a + bcd, ad + b + c) = 2/6$. If simcutoff is above 1/2, the similarity column will be unaffected. If simcutoff is between 1/3 and 1/2, the first two clusterings will support one another.

The set of clusterings can be used to produce into a similarity matrix, which contains the posterior probabilities of each pair of minimal clusters belong to the same cluster. See table 3 for the similarity matrix corresponding to the imaginary example, and the main paper for a larger example.

	a	b	c	d
a	1	.6	.6	.7
b	.6	1	.9	.9
c	.6	.9	1	.9
d	.7	.9	1	.9

Table 3: Similarity matrix

6 Obtaining a development version of BEAST

You will need:

- Java and a Java development kit (JDK) installed.
- Subversion installed. `http://subversion.apache.org/`
- Ant installed. `http://ant.apache.org/`

6.1 Linux and Mac OS X instructions

6.1.1 Check out the code

Type

```
svn checkout http://beast-mcmc.googlecode.com/svn/trunk/ beast16
```

`beast16` names a directory; you can use another name. You should see a long listing of files ending something like

```
A    beast16/examples/incorrect/testOTFPCLikelihood.xml
 U   beast16
Checked out revision 4366.
```

6.1.2 Build BEAST

`cd` to `beast16` and type `ant`. You should see something like this:

```
[gjones@pc158250 beast16]$ ant
Buildfile: build.xml

clean:

init:
     [echo] BEAST: /home/gjones/beast16/build.xml

compile-all:
    [mkdir] Created dir: /home/gjones/beast16/build
    [javac] Compiling 1836 source files to /home/gjones/beast16/build
    [javac] Note: Some input files use or override a deprecated API.
    [javac] Note: Recompile with -Xlint:deprecation for details.
    [javac] Note: Some input files use unchecked or unsafe operations.
    [javac] Note: Recompile with -Xlint:unchecked for details.
     [echo] Successfully complied.

dist-all:
    [mkdir] Created dir: /home/gjones/beast16/build/dist
      [jar] Building jar: /home/gjones/beast16/build/dist/beast.jar
      [jar] Building jar: /home/gjones/beast16/build/dist/beauti.jar

build:

BUILD SUCCESSFUL
Total time: 37 seconds
[gjones@pc158250 beast16]$
```

6.1.3 Copy some 'properties' files from source directory to build directory

Type

```
cp beast16/src/dr/app/beast/*.properties  beast16/build/dr/app/beast/
```

6.2 Windows instructions

You have to do the same steps: download the code, compile it, and run it via a java command. I use TortoiseSVN which is a GUI for Subversion and Eclipse which is an IDE for Java. This is overkill if you just want to run a development version of BEAST, but I have not tried any other way.

6.3 Test BEAST is working

In order to run BEAST you need to execute a java command and supply a 'classpath' which is in this case a list of three places (`.` meaning 'here', `/home/gjones/beast16/build/`, and `/home/gjones/beast16/lib/*`) to tell java where to look for code. Then there is the java class `BeastMain` to run and the XML file `YuleSingleLocus.xml` which is input to BEAST. Under Linux, all as one line:

```
java -classpath '.:/home/gjones/beast16/build/:/home/gjones/beast16/lib/*'
   dr.app.beast.BeastMain beast16/examples/release/starBEAST/YuleSingleLocus.xml
```

Under Windows, the classpath is separated by `;` not `:` and the code is a different place (`bin` not `build`) and file paths use `\` not `/` and the single quotes don't seem to be needed. It might look like this:

```
java -classpath .;C:\workspace\beast16\bin;C:\workspace\beast16\lib/*
   dr.app.beast.BeastMain beast16\examples\release\starBEAST\YuleSingleLocus.xml
```

You should see a normal BEAST run. If you get an error like this

```
Exception in thread "main" java.lang.NoClassDefFoundError:
 jebl/evolution/treemetrics/RootedTreeMetric
....
```

it is probably a problem with the classpath and/or the directory you are in when you issue the command.

7 R code

7.1 R code for density of origin

```
origindensity <- function(x, n, a, b, w) {
E <- exp(-a*x)
B <- (1 - E) / (1-b*E)
z <- 1
z <- z * a
z <- z * (1-b)
z <- z * E
z <- z / (1-b*E)^2
z <- z * (w + (1-w)*B)^(n-2)
z <- z * (w + n*(1-w)*B)
z
}


log.origindensity <- function(x, n, a, b, w) {
E <- exp(-a*x)
B <- (1 - E) / (1-b*E)
z <- 0
z <- z + log(a)
z <- z + log(1-b)
z <- z - a*x
z <- z - 2 * log(1-b*E)
z <- z + (n-2) * log(w + (1-w)*B)
z <- z + log(w + n*(1-w)*B)
z
}

n <- 50
a <- 1
b <- 0.9
w <- 0.5

cat("mean nof species in prior is ", 1 + (n-1)*(1-w), "\n")
x <- seq(from=0, to=10, length.out=1001)
matplot(x, cbind(log(origindensity(x, n, a, b, w)), log.origindensity(x, n, a, b, w)), type="l")
integrate(origindensity, lower=0, upper=Inf, n=n, a=a, b=b, w=w)
```

7.2 R code for prior probabilities of number of species

If w has a beta distribution for its hyperprior, with parameters a and b (`shape` and `shapeB` in the XML) and there are n individuals, then the prior probabilities of number of species is given by

$$\Pr(k = x|n, a, b) = \frac{\Gamma(n)}{\Gamma(x)\Gamma(n+1-x)} \frac{\Gamma(x-1+b)\Gamma(n-x+a)}{\Gamma(n-1+a+b)} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}$$

and the function `px` below calculates this, for $1 \leq x \leq n$.

```
library(VGAM)
px <- function(x, n, a, b) {
dbetabinom.ab(x-1, size=n-1, shape1=b, shape2=a)
}
# Alternative, without VGAM:
px2 <- function(x, n, a, b) {
p <- 1
p <- p * gamma(n)
p <- p / gamma(x)
p <- p / gamma(n+1-x)
p <- p * gamma(x-1+b)
p <- p * gamma(n-x+a)
p <- p / gamma(n-1+a+b)
p <- p * gamma(a+b)
p <- p / gamma(a)
p <- p / gamma(b)
p
}

z <- px(1:25, 25, 8,2)
z2 <- px2(1:25, 25, 8,2)
plot(z)
z
z2
```

7.3 R code for displaying the similarity matrix

```
# Read in the table of clusterings
workdir <- "C:/Users/Graham/AAA/Programming/tmp/birthdeathcollapse/w70e0001/"
x <- read.table(paste(workdir, "SDA_OUT.txt", sep=""), header=TRUE)

# Abbreviations for display
renames <- matrix(c(
"Orthogeomys_heterodus1",  "O.het",
"Thomomys_bottae1",  "T.bot awa-a",
"Thomomys_bottae10",  "T.bot mew",
"Thomomys_bottae11",  "T.bot sax",
"Thomomys_bottae12",  "T.bot lat",
"Thomomys_bottae2",  "T.bot awa-b",
"Thomomys_bottae3",  "T.bot xer",
"Thomomys_bottae4",  "T.bot cac",
"Thomomys_bottae5",  "T.bot alb",
"Thomomys_bottae6",  "T.bot rui",
"Thomomys_bottae7",  "T.bot bot",
"Thomomys_bottae8",  "T.bot alp",
"Thomomys_bottae9",  "T.bot rip",
"Thomomys_idahoensis1",  "T.ida pyg-a",
"Thomomys_idahoensis2",  "T.ida pyg-b",
"Thomomys_mazama1",  "T.maz maz",
"Thomomys_mazama2",  "T.maz nas",
"Thomomys_monticola1",  "T.mon-a",
"Thomomys_monticola2",  "T.mon-b",
"Thomomys_talpoides1",  "T.tal oci",
"Thomomys_talpoides2",  "T.tal yak",
"Thomomys_talpoides3",  "T.tal bri",
"Thomomys_townsendii1",  "T.tow tow",
"Thomomys_townsendii2",  "T.tow rel",
"Thomomys_umbrinus1",  "T.umb chi",
"Thomomys_umbrinus2",  "T.umb atr"),
nrow=26, ncol=2, byrow=TRUE)

# Minimal cluster names are column names omitting first 4
mincl.names <- colnames(x)[-(1:4)]
# Check for typos, etc
for (i in 1:length(mincl.names)) {
  stopifnot(mincl.names[i] == renames[i,1])
  }
```

```
# Make the similarity matrix
displaynames <- renames[,2]
nmincls <- length(displaynames)
sim <- matrix(0, ncol=nmincls, nrow=nmincls, dimnames=list(displaynames, displaynames))
for (i in 1:nmincls) {
  for (j in 1:nmincls) {
    coli <- x[,mincl.names[i]]
    colj <- x[,mincl.names[j]]
    w <- coli == colj
    sim[i,j] <- sum(x[w,"fraction"])
  }
}
# ensure rounding errors don't make probabilities sum to more than 1.
sim <- pmin(sim,1)

# change the order of minimal clusters
# This depends on which patterns you want to emphasise
neworder <- c(11,7,8,9,10,13,3,2,6,4,5,12,  24,23,  26,25,  16,17, 20,21,22, 1, 14,15, 18,19)
# Currently recognised groups
dividers <- c(0,12,14,16,18,21,22,24,26)

plot.rectangle <- function(v1,v2,...)
{
polygon(c(v1[1],v2[1],v2[1],v1[1]), c(v1[2],v1[2],v2[2],v2[2]), ...)
}

# Main plotting routine
plot.simmatrix <- function() {
  par(mar= c(0,5,5,0)+.1)
  plot(NULL, xlim=c(0,nmincls), ylim=c(nmincls,0), axes=FALSE, ylab="", xlab="")
  axis(3, at=(1:nmincls)-.5, displaynames[neworder], tick=FALSE, las=2, line=-1)
  axis(2, at=(1:nmincls)-.5, displaynames[neworder], tick=FALSE, las=2, line=-1)
  for (i in 1:nmincls) {
    for (j in 1:nmincls) {
      d <- 1 - sim[neworder[i],neworder[j]]
      plot.rectangle(c(i-1,j-1), c(i,j), col=rgb(d,d,d), border="white")
    }
  }
  for (b in dividers) {
    lines(x=c(-.5,nmincls), y=c(b,b))
    lines(x=c(b,b), y=c(-.5,nmincls))
  }
}

# Display as text, on screen, and to PDF file
print(sim[neworder,neworder], digits=2)
plot.simmatrix()
pdf(file=paste(workdir, "simmatrix.pdf", sep=""))
plot.simmatrix()
dev.off()
}
```

8 Distribution on partitions from prior

Probability distributions on the partitions of a finite set are of general interest. Some of these distributions arise as the result of a stochastic process. The *Chinese restuarant process* (see Wikipedia) is one example, and there are also a variety of *preferential attachment processes* (see Wikipedia). One of the latter is known as 'the Yule process' [7] which models the emergence of new genera among species. (Note that this is an extension to the pure birth model for speciation, which is also called a Yule process or Yule model.) None of these seems to be the same as the distribution associated with the model used here. The following result gives the probability of a given clustering in the model used here, conditioned on the number of clusters. The point process described in [3] shows why the merging process in the proposition produces the same distribution as sampling a tree using the method described in the main paper (sampling from f and $(1-w)f+wm$), and then slicing through the tree. The result is almost surely already published somewhere, but I can't find it. It may be useful for calculating Bayes factors for different species delimitation hypotheses.

Proposition. *Assume a merging process which starts with n objects each assigned to its own cluster. Clusters are then merged by randomly choosing pairs among those available at each step until there are r clusters. Let $b_1, b_2, \ldots, b_r$ be the sizes of the clusters in a particular clustering B produced by this process. Then*

$$\begin{aligned}\Pr(B|r) &= \frac{r!\,(n-r)!\,(r-1)!\,b_1!\;b_2!\ldots b_r!}{(n-1)!\,n!} \\ &= r!\binom{n-1}{r-1}^{-1}\binom{n}{b_1, b_2, \ldots, b_r}^{-1}\end{aligned}$$

Proof. First we count the total number T of ordered mergings that make r clusters from $\{1, 2, \ldots, n\}$. Then we count the number S of ordered mergings that make B and obtain $\Pr(B|r) = S/T$. It may help to think of a particular example of B such as

$$B = \{\ \{1, 2, \ldots, b_1\},\ \{b_1{+}1,\ b_1{+}2, \ldots, b_1{+}b_2\}, \ldots, \{n{-}b_r{+}1,\ n{-}b_r{+}2, \ldots, n\}\ \}.$$

We have

$$\begin{aligned}T &= \binom{n}{2}\binom{n-1}{2}\cdots\binom{n-r}{2} \\ &= \frac{n!}{r!}\frac{(n-1)!}{(r-1)!}2^{n-r}\end{aligned}$$

For S, we multiply the numbers of ways in which each the clusters of sizes $b_1, b_2, \ldots, b_r$ can be formed by the number of ways that the mergings of different clusters can be interleaved. There are $b-1$ mergings for a cluster of size b, so there are

$$\binom{n-r}{b_1-1, b_2-1, \ldots, b_r-1}$$

interleavings, and

$$\begin{aligned}S &= \binom{n-r}{b_1-1, b_2-1, \ldots, b_r-1}\prod_{i=1}^{r}\frac{b_i!(b_i-1)!}{2^{b_i-1}} \\ &= (n-r)!2^{n-r}\prod_{i=1}^{r}b_i!\end{aligned}$$

and the result follows. □

Suppose that there are s_i clusters of size i $(1 \le i \le n)$. Note that $\sum_{i=1}^{n} s_i = r$. Then there are

$$\binom{n}{b_1, b_2, \ldots, b_r}\left(\prod_{i=1}^{n} s_i!\right)^{-1}$$

partitions of $\{1, 2, \ldots, n\}$ which have the same shape (the same partition of the integer n). So the probability that the r clusters have a given shape is

$$r!\binom{n-1}{r-1}^{-1}\left(\prod_{i=1}^{n} s_i!\right)^{-1}$$

Example. Suppose $n = 8, r = 3$. Then

$$\Pr(4+2+2) = 3!\binom{7}{2}^{-1}(0!\,2!\,0!\,1!\,0!\,0!\,0!\,0!)^{-1} = 1/7$$

and there are

$$\binom{8}{4,2,2}(0!\,2!\,0!\,1!\,0!\,0!\,0!\,0!)^{-1} = 210$$

partitions of $\{1, 2, \ldots, 8\}$ which have this shape.

The Chinese restuarant process is different from the merging process above. The probabilities for $n = 6$ are shown in table 4. When conditioned on $r = 2$, the Chinese restuarant process gives probabilities in ratio 72:45:20 for 5+1,4+2,3+3, whereas the merging process gives 2:2:1. When conditioned on $r = 3$, the Chinese restuarant process gives 6:8:1 for 4+1+1,3+2+1,2+2+2, whereas the merging process gives 3:6:1.

Table 4: Chinese restuarant process

Shape	P(partition)	#ptns per shape	P(shape)
6	120/720	1	120/720
5+1	24/720	6	144/720
4+2	6/720	15	90/720
3+3	4/720	10	40/720
4+1+1	6/720	15	60/720
3+2+1	2/720	60	120/720
2+2+2	1/720	15	15/720
3+1+1+1	2/720	20	40/720
2+2+1+1	1/720	45	45/720
2+1+1+1+1	1/720	15	15/720
1+1+1+1+1+1	1/720	1	1/720

References

[1] Alexei J Drummond and Andrew Rambaut. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evolutionary Biology*, 7:214, 2007.

[2] Alexei J. Drummond, Marc A. Suchard, Dong Xie, and Andrew Rambaut. Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Molecular Biology and Evolution*, 29:1969–1973, 2012.

[3] T Gernhard. The conditioned reconstructed process. *J. Theo. Biol.*, 253:769–778, 2008.

[4] J Heled and A Drummond. Bayesian inference of species trees from multilocus data. *Mol. Biol. Evol.*, 27:570–580, 2010.

[5] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[6] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association (American Statistical Association)*, 66(336):846–850, 1971.

[7] G. U. Yule. A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis, F.R.S. *Philosophical Transactions of the Royal Society B*, 213:21–87, 1925.