A ROUGH GUIDE TO SNAPPNET

Charles-Elie Rabier, Vincent Berry, Jean-Christophe Glaszmann, Fabio Pardi, Celine Scornavacca

ce.rabier@gmail.com, vberry@lirmm.fr
University of Montpellier

SnappNet is a new Bayesian method that directly relies on DNA sequences. Our method is implemented in BEAST 2 (Bouckaert et al., 2014), an improved version of the popular version BEAST 1.x dedicated to Bayesian evolutionary analyses. Our SnappNet package is built on two BEAST packages, Snapp (Bryant et al, 2012), and SpeciesNetwork (Zhang et al., 2017). It incorporates the novel MCMC operators of SpeciesNetwork to move through the network space and also benefits from operators specific to the mathematical model behind Snapp (e.g. population sizes,mutation rates ...) and extended to handle networks.

1 The stochastic model behind SnappNet

1.1 Gene tree model

For each site, the associated gene tree is obtained according to the Network Multispecies Coalescent model. The process starts at the leaves of the network and goes backward in time, until all lineages coalesce. At the beginning, coalescence occurs only between lineages that belong to the same species. 2 given lineages coalesce at rate $2\mu/\theta$ where θ denotes the population size parameter of that species. The parameter μ is equal to $2uv/(u+v)$ where u and v are mutation rates. Assuming that k lineages belong to that species, the first coalescent time follows an exponential distribution $\mathbb{E}(k(k-1)\mu/\theta)$, since the coalescence of each combination of 2 lineages is equiprobable. When $k = 2$, the expected coalescent time is $\theta/(2\mu)$. θ is the average number of mutations, separating 2 individuals. When the lineages (that fail to coalesce) enter a node delimiting two species, the remaining lineages of the two species are merged, and are allowed to coalesce within the new branch, according to the θ value associated to that branch. When the lineages (that fail to coalesce) enter a reticulation node, each lineages chooses independently (Yu et al., 2011) between the left or the right reticulation edge (above the reticulation node), according to a Bernoulli distribution $\mathcal{B}(\gamma)$. γ refers to the probability of going to the left population. Above the root of the phylogenetic network, coalescence occurs among all lineages and the process ends when only one ancestral lineage is remaining.

Note that as its cousin Snapp, SnappNet imposes the constraint $\mu = 1$ (see the explanations below).

1.2 Mutation model

As in Snapp, we consider biallelic markers and the colors red and green represent the two alleles. Markers evolve along the gene tree branches, according to a continuous time Markov chain, where u and v denote respectively the instantaneous rates of mutating from red to green, and from green to red.Under this model, on a branch of length T, there are on average $2uvT/(u+v)$ mutations. Then, the mutation rate μ is equal to $2uv/(u+v)$. Imposing the constraint $2uv/(u+v) = 1$ (i.e. $\mu = 1$), enables to measure branch lengths in substitutions per site (i.e. genetic distance).

2 Preparing your xml file with Beauti

The easiest way to handle SnappNet is to use Beauti (Bouckaert et al., 2014). This way, you do not need to worry about the xml file required for running SnappNet on your own data.

In case you want to understand the xml file generated by Beauti, some extra informations are given at "http://charles-elie.rabier.pagesperso-orange.fr/doc/SnappNet.html" . Be careful, these informations will differ slightly from the xml file obtained from Beauti. Indeed, these xml informations are related to the xml generated by our simulator Sim-SnappNet.

Let us prepare the xml file with Beauti. First, you need a nexus file in order to upload it into Beauti. For instance, the file JDD1-all.chr.nexus is located in the folder example. Recall the BEAST commmand line for launching beauti: java -jar ./beauti . Note that our beauti template looks very similar to the Snapp's template, since it is built on it, and incorporates only changes regarding our network model.

Here are the main steps to follow :

- Step 1 (loading the nexus file, see Figure 2). File → AddAlignment → choose the file JDD1-all.chr.nexus.
- Step 2 (correspondence between taxon and species, see Figures 3 and 4). Guess → use everything → after last
- Step 3 (Model parameters, see Figure 5). Calculate mutation rates (see Snapp manual by Bouckaert and Bryant, available at https://www.beast2.org/snapp/). Choose whether or not you want to include non polymorphic sites in the analysis.
- Step 4 (Setting the priors, see Figures 6 and 7)
- Step 5 (About the MCMC outputs, see Figure 8).
- Step 6 (Show operators panel, see Figure 9).

- Step 7 (Choosing the maximum number of reticulations, see Figure 10).
- Step 8 (Saving your xml file, see Figure 11)

2.1 Setting the priors (Step 4)

As a network prior, SnappNet uses the birth hybridization process of Zhang et al. (MBE, 2017). The network prior depends on the speciation rate λ, on the hybridization rate ν and on the time of origin τ_0.

Let us describe the hyperpriors imposed onto these parameters:

- netDivRate $:= \lambda - \nu$ follows an exponential distribution. You can specify the mean of that distribution. Recall that if netDivRate $\sim \mathcal{E}(\lambda)$, then $\mathbb{E}\{\text{netDivRate}\} = 1/\lambda$
- originTime $:= \tau_0$ follows an exponential distribution.
- turnOverRate $:= \nu/\lambda$ is assigned a Beta distribution with parameters α and β.

Given the values of netDivRate and turnOverRate, we can compute the networkPrior:

- NetDiversification is the starting value for netDivRate
- Turn Over is the value for turnOverRate
- shape is the value for the parameters α and β of the beta prior on inheritance probabilities (called γ).

As its cousin Snapp, SnappNet considers a Gamma distribution as a prior on population sizes θ. This Gamma prior induces a prior on the colaescence rate. So, the parameter α and β in snapprior refers to the parameters of the Gamma distribution. SnappNet will evaluate this Gamma prior at the CoalescenceRate given as a starting point. So, feel free to give a value to CoalescenceRate.
Last, as in Snapp, the user can specify fixed values for the u and v rates, or impose a prior for these rates, and let them be sampled within the MCMC.

2.2 About the MCMC outputs (Step 5)

You can specify the chain length, the pre burnin length. “Store every” refers to the frequency (i.e. number of iterations) for which the sampled networks are printed out in a file.

After having saved the .xml file (step 8), with name JDD1.xml, we can run SnappNet with the following command line :

```
java -Xmx15g -jar SnappNetProjectToRun.jar -seed 123 JDD1.xml > stdout
```

It will generate two files, named

- OutPut.xml.trace.log (see tracelog)
- OutPut.xml.Myspecies.net (see specieslog)

OutPut.xml.trace.log gives the log posterior, the log likelihood, the log prior, and a few parameter values at some MCMC steps (depending on the logEvery parameter). OutPut.xml.trace.log can be analyzed with the software Tracer for some MCMC convergence diagnostics (ESS ...), see Figure 1. Tracer can be dowloaded from http://beast.community/tracer.

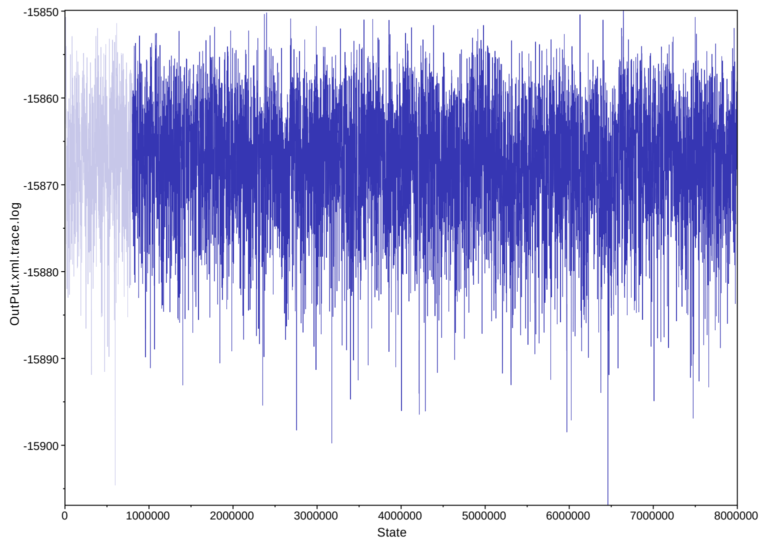


Figure 1: Analysis of the estimated posterior distribution with Tracer

An example of OutPut.xml.trace.log is the following

```
Sample posterior likelihood prior u v netDivRate:species turnOverRate:species
originTime:species

0 -88233.4055772867 -44420.84842400747 -43812.55715327924 0.56085686465 4333
4.608 2.0 0.5 0.1

1000 -37047.359125439965 -36856.64421858852 -190.7149068514516 0.56085686465 4333
4.608 1.4777196769681868 0.6150861627675254 0.1

2000 -37047.33860364854 -36856.64421858852 -190.69438506002274 0.56085686465 4333
4.608 1.272501762679686 0.3334571470715876 0.1
```

```
3000 -35989.436450990834 -35780.41889099498 -209.0175599958533 0.56085686465 4333
4.608 0.47387451366805233 0.4843068185997752 0.1
```

On the other hand, the file OutPut.xml.Myspecies.net gives the networks sampled by the MCMC algorithm. The file OutPut.xml.Myspecies.net will look like the following:

```
Begin trees;

tree STATE_0 = ((Or1[&Theta=200.0]:5.0,(Aro[&Theta=200.0]:4.0,(Ind[&Theta=200.0]:3.0,
(Aus[&Theta=200.0]:2.0,(Jap[&Theta=200.0]:1.0,Or3[&Theta=200.0]:1.0)S1[&Theta=200.0]:1.0
[&Theta=19
0.75839038575592]:1.0)S3[&Theta=200.0]:1.0)S4[&Theta=200.0]:1.0)S5[&Theta=200.0]:1.0);

tree STATE_1000 = ((Aro[&Theta=0.13335546542395332]:2.004442488495597,((Aus[&Theta=
0.20420989376448476]:1.2341992011362617,((Or3[&Theta=0.43407716819704545]:0.086438899960
Jap[&Theta=0.2563645451330878]:0.08643889996091127)S5[&Theta=0.21963856281664615]
:0.34419503331369866,Or1[&Theta=0.5594987376939713]:0.4306339332746099)S1
[&Theta=0.16921057561572125]:0.8035652678616518)S4[&Theta=0.10522247828347867]:0.3744459
[&Theta=0.319642248951935]:1.6086451896139435)S3[&Theta=0.08743670356590173]:0.395797298
]:3.995557511504403);

...
```

2.3 About the 16 MCMC operators (Step 6)

Here are the different operators handled by SnappNet.

Topological operators:

- *addReticulation:* to add a reticulation node
- *deleteReticulation:* to delete a reticulation node
- *flipReticulation:* to flip a reticulation edge
- *relocateBranch:* to relocate a branch
- *relocateBranchNarrow:*

Other operators:

- *changeUAndV:* to change the mutation rate values u (from red to green) and v (from green to red), under the constraint $2uv/(u+v) = 1$.

- *changeGamma:* to change the coalescence rate γ associated to a network branch. By definition, in the model, $\gamma = 2/\theta$, where θ denotes the population size
- *changeAllGamma:* to change coalescence rates γ of all network branches.
- *turnOverScale:* to change the value of the parameter ν/λ linked the birth-hybridization process (ν: hybridization rate, λ: speciation rate)
- *divrRateScale:* to change the value of the parameter $\lambda - \nu$ linked to the birth-hybridization process
- *inheritanceProbUniform:* to change the hybridization probability at a reticulation node chosen at random (among all reticulation nodes)
- *inheritanceProbRndWalk:* to change the value of the hybridization probability (at a random reticulation node) by applying a random walk to the logit of γ
- *originMultiplier:* to change origin height of the network
- *networkMultiplier:* to change internal node heights using a multiplier
- *nodeUniform:* to select randomly an internal network node and to move its height uniformly
- *nodeSlider:* to select randomly an internal network node and to move its height using a sliding window

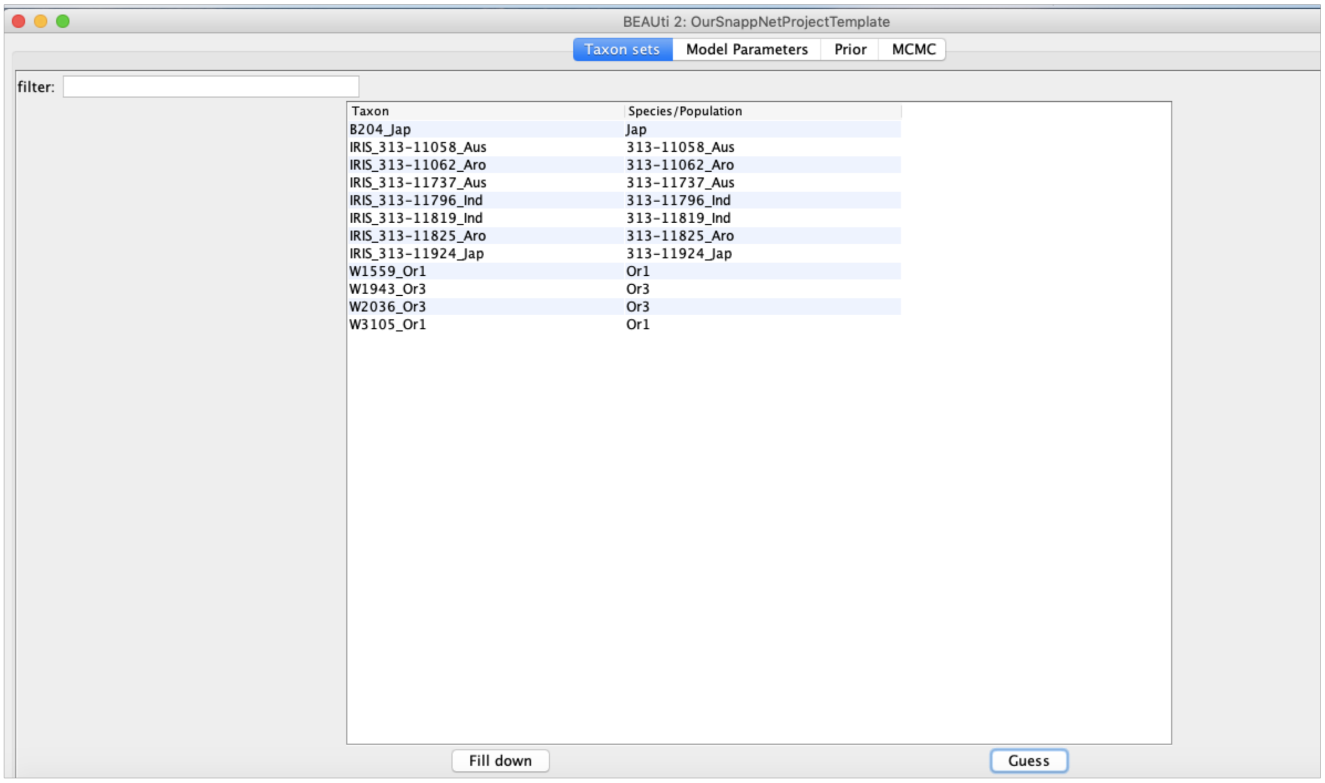
Remark : We are using here the same notation γ for the coalescence rate and the hybridization probability, because of the two BEAST packages, Snapp (Bryant et al, 2012) and SpeciesNetwork (Zhang et al., 2017). Obviously, the coalescence rates and the hybridization probabilities are different quantities, and the code handles them appropriately.

3 How to evaluate a network by Maximum Likelihood

Let us prepare the xml file with Beauti. First, you need a nexus file in order to upload it into Beauti. For instance, the file JDD1-all.chr.nexus is located in the folder example. Recall the BEAST commmand line for launching beauti: java -jar ./beauti . Our beauti template for computing Maximum Likelihood looks similar to the previous MCMC template, but there are slight changes.

Here are the main steps to follow :

- Steps 1, 2 and 3 are the same as before



Taxon	Species/Population
B204_Jap	Jap
IRIS_313-11058_Aus	313-11058_Aus
IRIS_313-11062_Aro	313-11062_Aro
IRIS_313-11737_Aus	313-11737_Aus
IRIS_313-11796_Ind	313-11796_Ind
IRIS_313-11819_Ind	313-11819_Ind
IRIS_313-11825_Aro	313-11825_Aro
IRIS_313-11924_Jap	313-11924_Jap
W1559_Or1	Or1
W1943_Or3	Or3
W2036_Or3	Or3
W3105_Or1	Or1

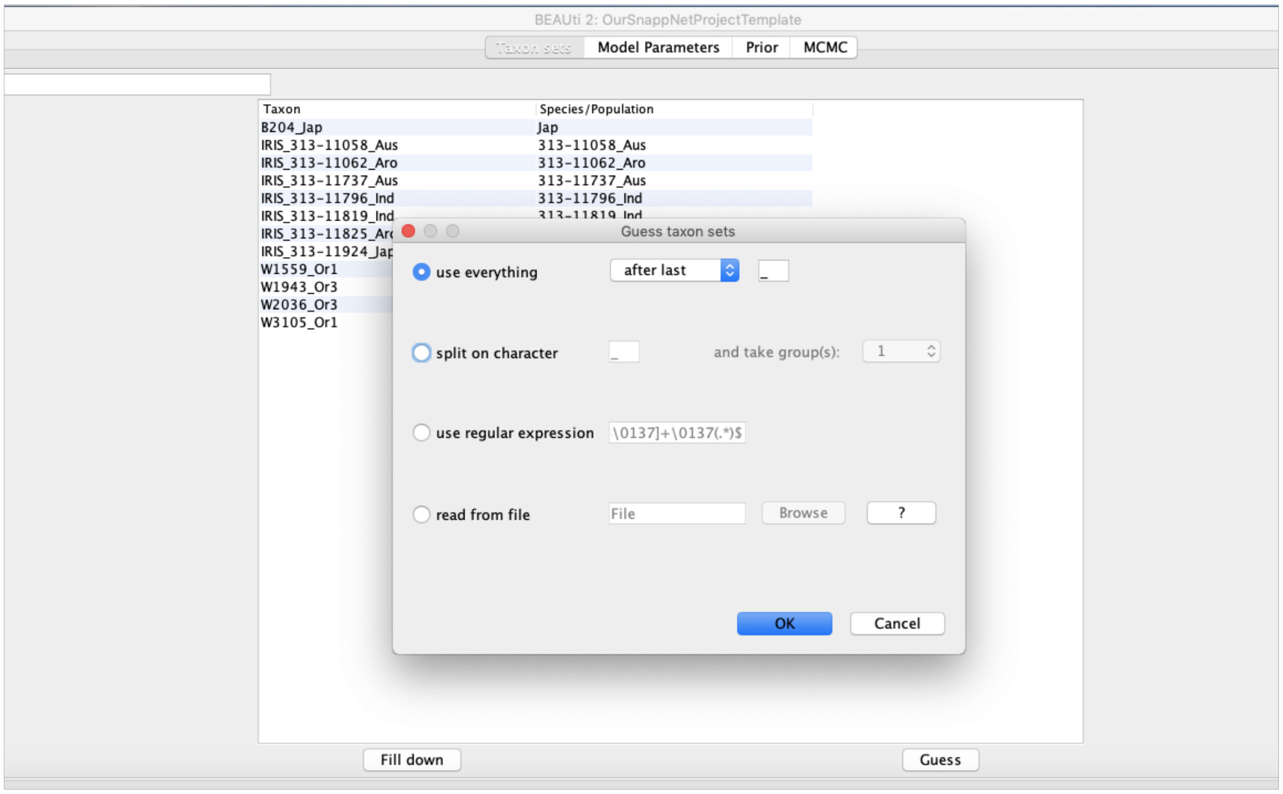
Figure 2: Step 1 (loading the nexus file). File → AddAlignment → choose the file JDD1-all.chr.nexus.

- Step 4 (Setting the number of iterations, see Figures 12). Note that we kept the name "Chain Length" from the MCMC template, although it refers now to the number of iterations. Obviously, there is no Markov Chain during our likelihood optimization.

- Step 5 (Choosing the weights of the different operators, see Figure 13)

After having saved the .xml file, we can run SnappNet with the following command line :

```
java -Xmx15g -jar SnappNetProjectToRun.jar -seed 123 JDD1ML.xml > stdout
```



Taxon	Species/Population
B204_Jap	Jap
IRIS_313-11058_Aus	313-11058_Aus
IRIS_313-11062_Aro	313-11062_Aro
IRIS_313-11737_Aus	313-11737_Aus
IRIS_313-11796_Ind	313-11796_Ind
IRIS_313-11819_Ind	313-11819_Ind
IRIS_313-11825_Arc	
IRIS_313-11924_Jap	
W1559_Or1	
W1943_Or3	
W2036_Or3	
W3105_Or1	

[✓ use everything] after last _

split on character _ and take group(s): 1

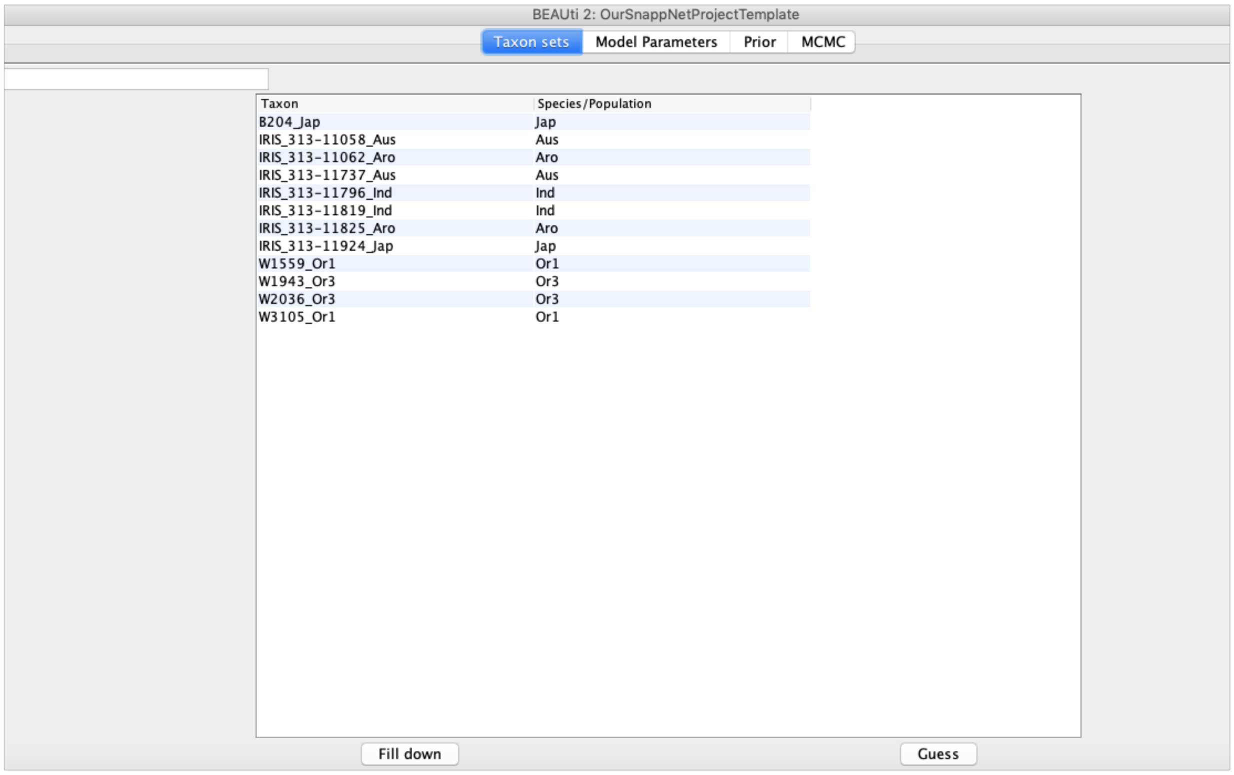
use regular expression \0137]+\0137(.*)$

read from file

Figure 3: Step 2 (correspondence between taxon and species). Guess → use everything → after last

It will generate a JDD1ML.xml.state file. Then, you must kill the process and open the JDD1ML.state file. Have a look at the following lines:

```
<statenode id='coalescenceRate'>coalescenceRate[11 1] (0.0,10.0): 16.22323694314152
7.619797004038854 10.52631471387076 8.37344687257438 13.225306620872894 9.49179576194465
33.250477619843 16.299413705267618 17.55211530348839 14.043375775142048 14.9127087285602
</statenode>
<statenode id='network:species'>((Or1[&amp;Theta=200.0]:5.076557457465168,
(Aro[&amp;Theta=200.0]:4.059680593610514,(Ind[&amp;Theta=200.0]:3.626223670387242,
(Aus[&amp;Theta=200.0]:2.7430529391333542,(Jap[&amp;Theta=200.0]:1.0129203279778487,
Or3[&amp;Theta=200.0]:1.0129203279778487)S1[&amp;Theta=200.0]:1.7301326111555055)
S2[&amp;Theta=200.0]:0.8831707312538879)S3[&amp;Theta=200.0]:0.4334569232232717)
```



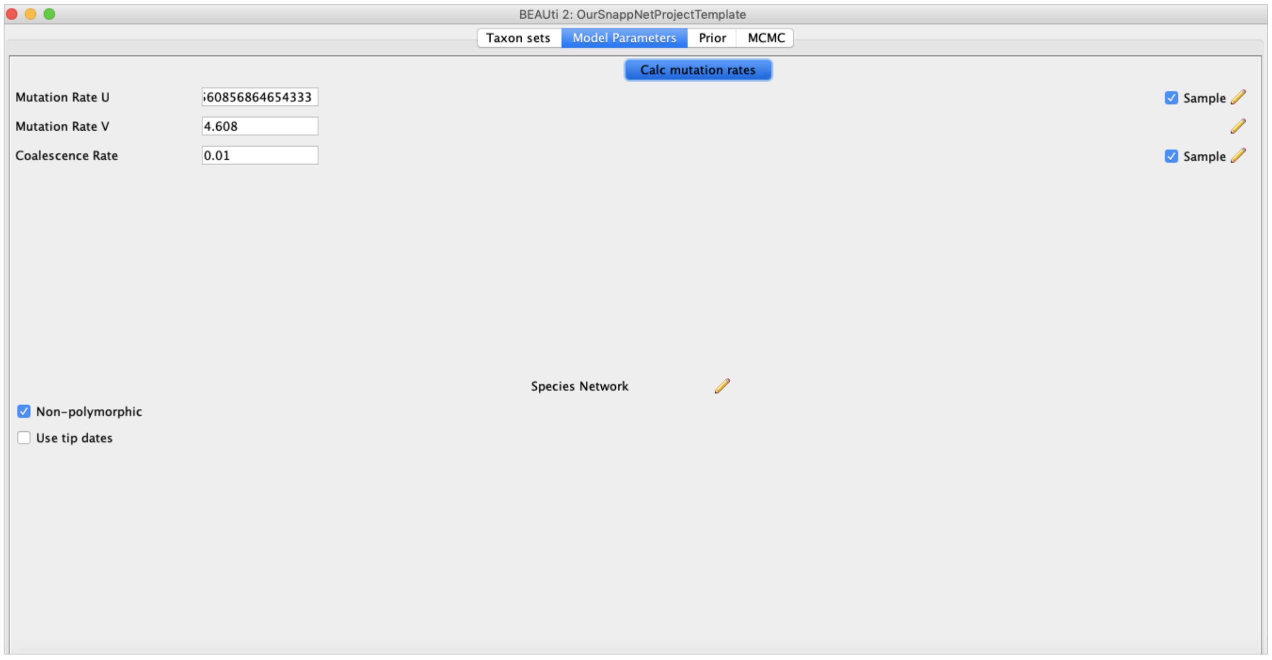
Taxon	Species/Population
B204_Jap	Jap
IRIS_313-11058_Aus	Aus
IRIS_313-11062_Aro	Aro
IRIS_313-11737_Aus	Aus
IRIS_313-11796_Ind	Ind
IRIS_313-11819_Ind	Ind
IRIS_313-11825_Aro	Aro
IRIS_313-11924_Jap	Jap
W1559_Or1	Or1
W1943_Or3	Or3
W2036_Or3	Or3
W3105_Or1	Or1

Figure 4: Result of step 2 (correspondence between taxon and species).

```
S4[&amp;Theta=200.0]:1.0168768638546544)S5[&amp;Theta=200.0]:4.4609318842475725)
</statenode>
```

In order to calculate the Maximum Likelihood of your data given a network, you have to replace those lines by your network of interest. For instance:

```
<statenode id='network:species'>(((((Aro:0.05289753496658764,Or1:0.05289753496658764)
S5:0.6992970811131721,(((Ind:0.5006293758289595,(Jap:0.07220279089382851,
Or3:0.07220279089382851)S7:0.42842658493513097)S3:0.22732316291965016)
#H1:0.005853558052244079)#H2:0.018388519278906057)S1:0.12599735718379312,
#H2[&amp;gamma=0.6653727420442962]:0.14438587646269918)S2:0.11726575859072452,
(#H1[&amp;gamma=0.7166040715675324]:0.13483785766078227
,Aus:0.8627903964093919)S4:0.13266733544488551)S6:0.05643728674372617)</statenode>
```


BEAUti 2: OurSnappNetProjectTemplate

Taxon sets | Model Parameters | Prior | MCMC

Calc mutation rates

Mutation Rate U 60856864654333 [✓ Sample]

Mutation Rate V 4.608

Coalescence Rate 0.01 [✓ Sample]

Species Network

[✓ Non-polymorphic]

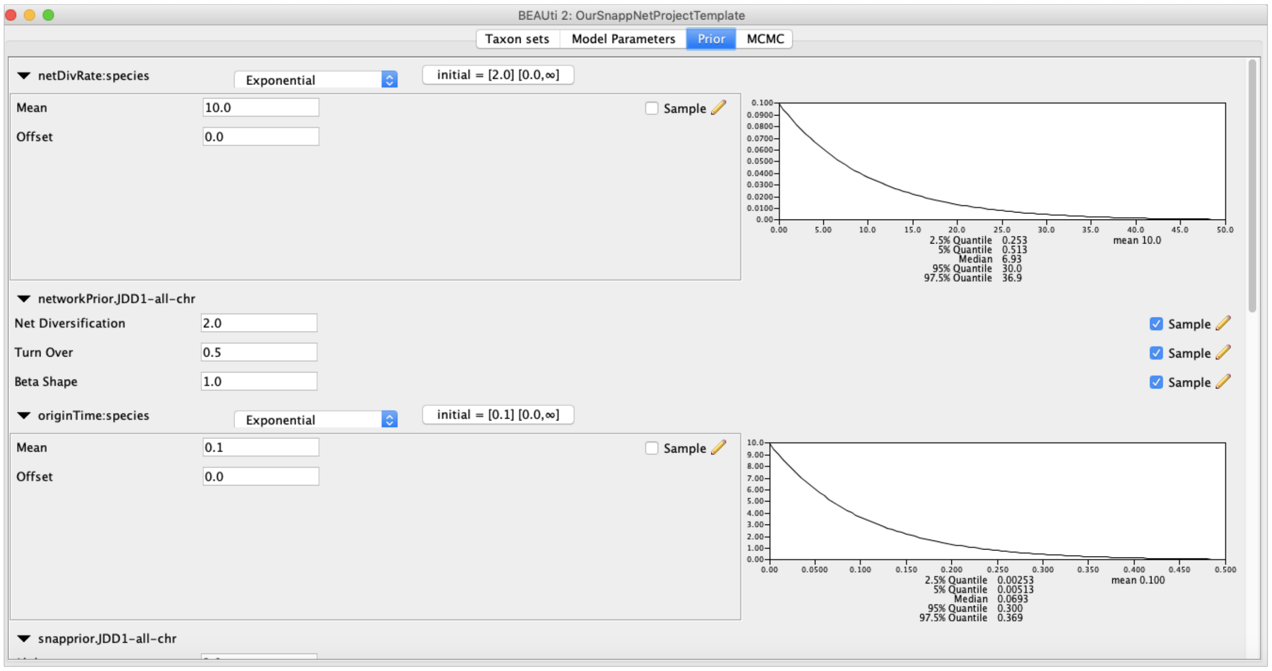
[ ] Use tip dates

Figure 5: Step 3 (Model parameters). Calculate mutation rates (see Snapp manual). Choose whether or not you want to include non polymorphic sites in the analysis.

```
<statenode id='coalescenceRate'>coalescenceRate[17 1] (0.0,10.0): 2.3352137153404153
13.836704427296574 3.8847934948861824 4.611251290455904 8.562867488025857
0.5766439098924774 10.676293910058076 10.076093636726295 6.517054392071151
5.882293514618984 2.4625726658729596 9.148744378166157 13.836704427296574
2.2493498647160353 2.3429797465985645 4.006378851420125 3.9782968863891173 </statenode>
```

Be careful, coalescenceRate has now 17 entries since the new network contains 17 edges. In case the generated .state file contains the tag 'network:sparser', you have to replace 'network:sparser' by 'network:species'.
Next, run SnappNet with the following command line:

```
java -Xmx15g -jar SnappNetProjectToRunMaxLikelihood.jar -resume  JDD1ML.xml
> stdout
```


BEAUti 2: OurSnappNetProjectTemplate

Taxon sets | Model Parameters | Prior | MCMC

netDivRate:species — Exponential — initial = [2.0] [0.0,∞]

```
Mean     10.0     [ ] Sample
Offset   0.0
```

```
2.5% Quantile   0.253        mean 10.0
  5% Quantile   0.513
       Median   6.93
 95% Quantile   30.0
97.5% Quantile  36.9
```

networkPrior.JDD1-all-chr

```
Net Diversification   2.0    [✓] Sample
Turn Over             0.5    [✓] Sample
Beta Shape            1.0    [✓] Sample
```

originTime:species — Exponential — initial = [0.1] [0.0,∞]

```
Mean     0.1     [ ] Sample
Offset   0.0
```

```
2.5% Quantile   0.00253      mean 0.100
  5% Quantile   0.00513
       Median   0.0693
 95% Quantile   0.300
97.5% Quantile  0.369
```

snapprior.JDD1-all-chr

Figure 6: Step 4 (Setting the priors 1).

4 Another optimization for evaluating a network by Maximum Likelihood

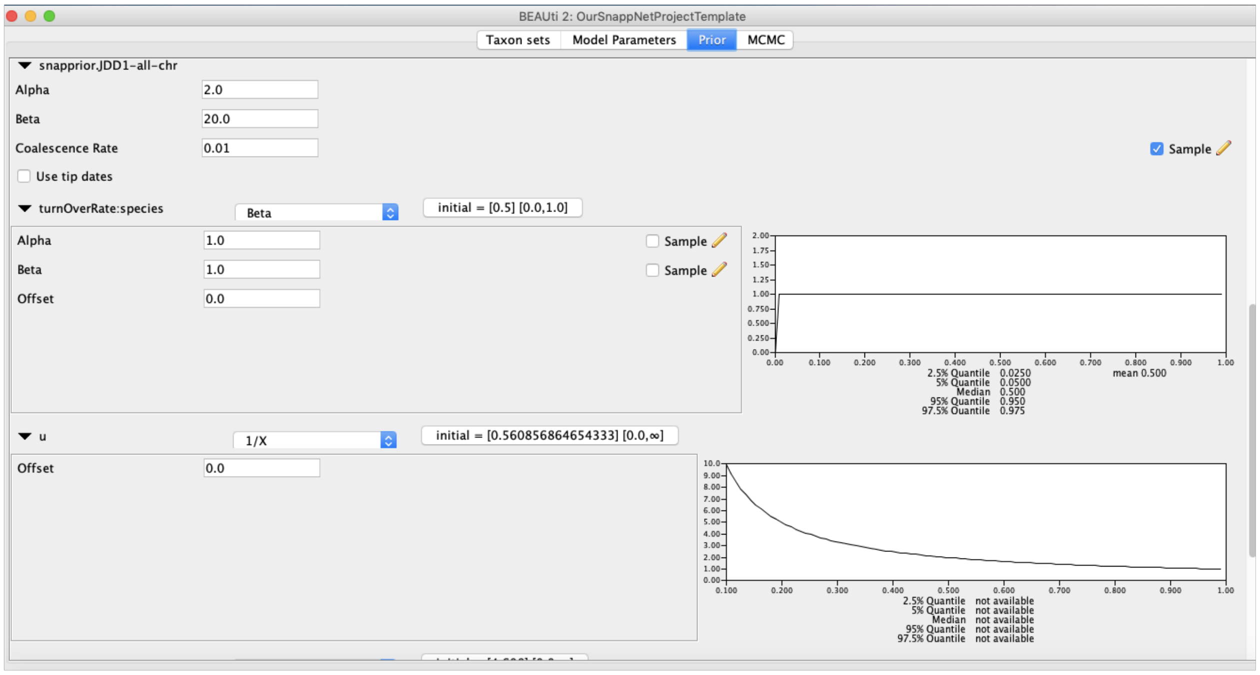
You just have to use the template dedicated to this optimization, and save the xml generated file. Next, you can run SnappNet with the following command line :

```
java -Xmx15g -jar SnappNetProjectToRun.jar -seed 123 JDD1ExtraOptim.xml > stdout
```

Then, you must kill the process and change the state file with your network of interest (as mentioned above). Finally, run SnappNet with the following command line:

```
java -Xmx15g -jar SnappNetProjectToRunResume.jar -resume  JDD1ExtraOptim.xml
> stdout
```

In the generated log file, you just have to consider the column "likelihood" and take the maximum value over all the rows.



snapprior.JDD1-all-chr
Alpha 2.0
Beta 20.0
Coalescence Rate 0.01
☐ Use tip dates
☑ Sample

turnOverRate:species Beta initial = [0.5] [0.0,1.0]
Alpha 1.0 ☐ Sample
Beta 1.0 ☐ Sample
Offset 0.0

```
 2.5% Quantile  0.0250
   5% Quantile  0.0500
        Median  0.500
  95% Quantile  0.950
97.5% Quantile  0.975
mean 0.500
```

u 1/X initial = [0.560856864654333] [0.0,∞]
Offset 0.0

```
 2.5% Quantile  not available
   5% Quantile  not available
        Median  not available
  95% Quantile  not available
97.5% Quantile  not available
```

Figure 7: Step 4 (Setting the priors 2).

References

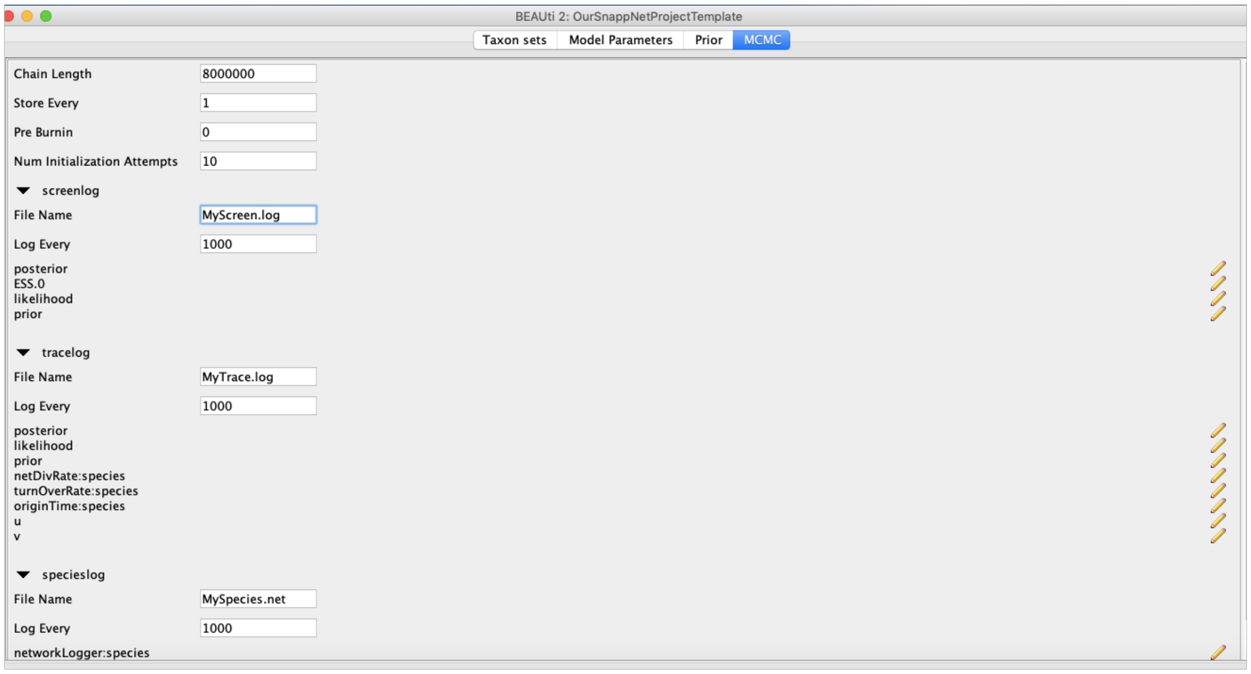
[1] Bouckaert, R., Heled, J., Kühnert, D., Vaughan, T., Wu, C. H., Xie, D., ... Drummond, A. J. (2014). BEAST 2: a software platform for Bayesian evolutionary analysis. *PLoS computational biology*, **10(4)**, e1003537.

[2] Bryant, D., Bouckaert, R., Felsenstein, J., Rosenberg, N. A., RoyChoudhury, A. (2012). Inferring species trees directly from biallelic genetic markers: bypassing gene trees in a full coalescent analysis. *Molecular biology and evolution*, **29(8)**, 1917-1932.

[3] Rambaut A, Drummond AJ, Xie D, Baele G and Suchard MA (2018) Posterior summarisation in Bayesian phylogenetics using Tracer 1.7. *Systematic Biology*. syy032. doi:10.1093/sysbio/syy032

[4] Yu, Y., Than, C., Degnan, J. H., Nakhleh, L. (2011). Coalescent histories on phylogenetic networks and detection of hybridization despite incomplete lineage sorting. *Systematic Biology*, **60(2)**, 138-149.

[5] Zhang, C., Ogilvie, H. A., Drummond, A. J., Stadler, T. (2017). Bayesian inference of



Chain Length	8000000
Store Every	1
Pre Burnin	0
Num Initialization Attempts	10

screenlog

File Name	MyScreen.log
Log Every	1000

posterior
ESS.0
likelihood
prior

tracelog

File Name	MyTrace.log
Log Every	1000

posterior
likelihood
prior
netDivRate:species
turnOverRate:species
originTime:species
u
v

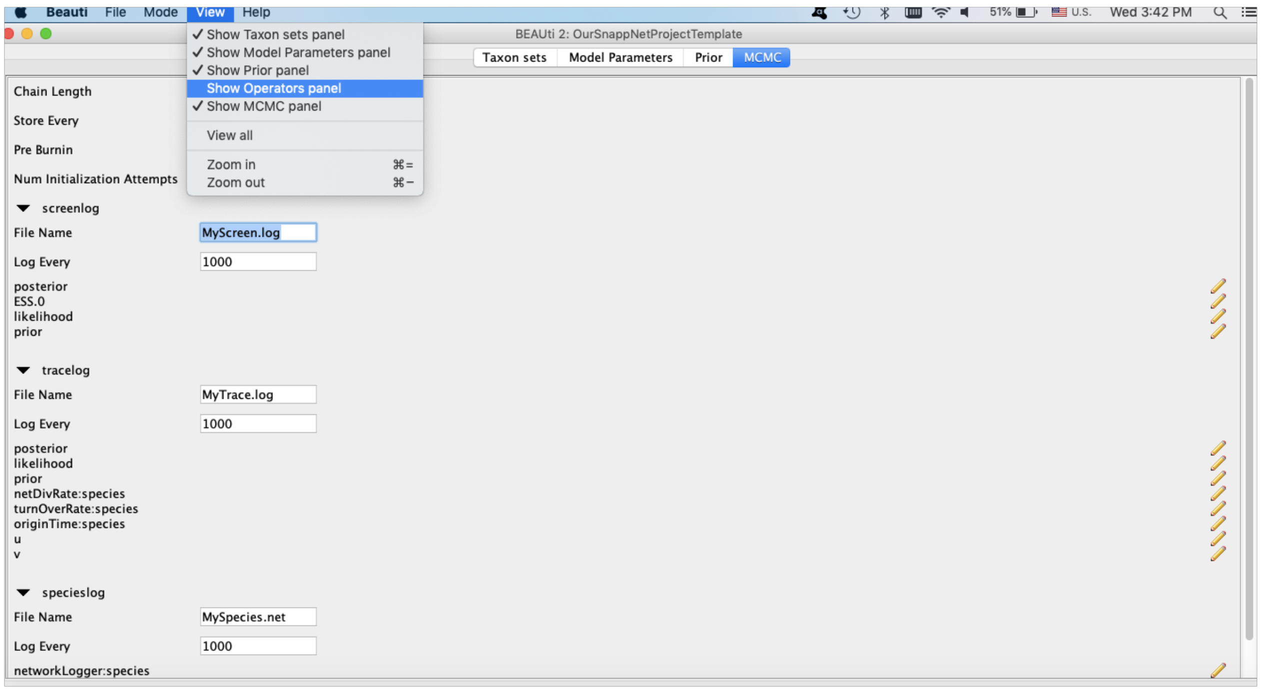
specieslog

File Name	MySpecies.net
Log Every	1000

networkLogger:species

Figure 8: Step 5 (About the MCMC outputs).

species networks from multilocus sequence data. *Molecular biology and evolution*, **35(2)**, 504-517.



Chain Length

Store Every

Pre Burnin

Num Initialization Attempts

screenlog

File Name MyScreen.log

Log Every 1000

posterior
ESS.0
likelihood
prior

tracelog

File Name MyTrace.log

Log Every 1000

posterior
likelihood
prior
netDivRate:species
turnOverRate:species
originTime:species
u
v

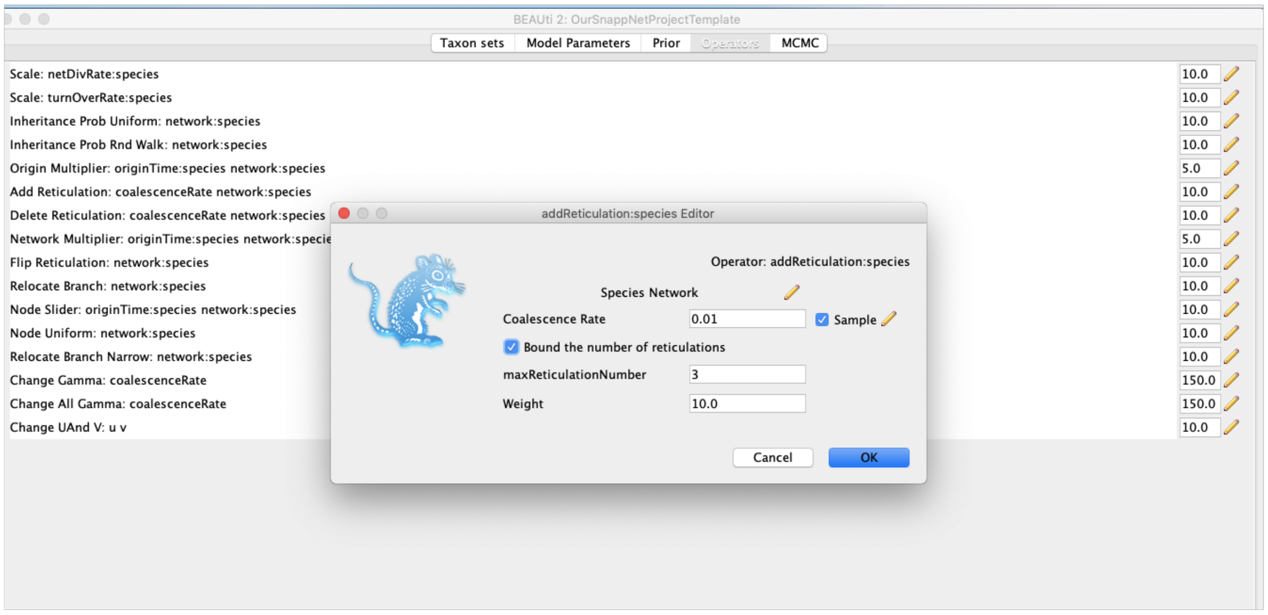
specieslog

File Name MySpecies.net

Log Every 1000

networkLogger:species

Figure 9: Step 6 (Show operators panel).


BEAUti 2: OurSnappNetProjectTemplate
Taxon sets
Model Parameters
Prior
Operators
MCMC
addReticulation:species Editor
Cancel
OK

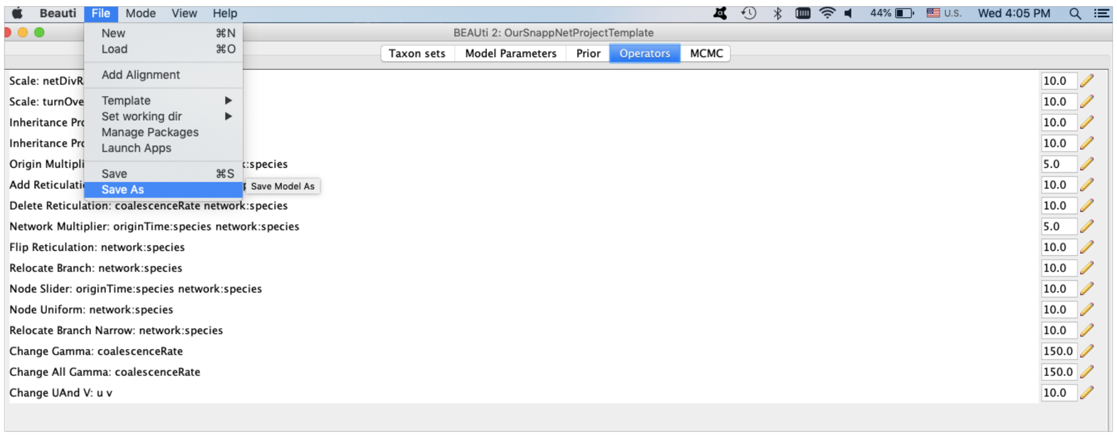
Scale: netDivRate:species	10.0
Scale: turnOverRate:species	10.0
Inheritance Prob Uniform: network:species	10.0
Inheritance Prob Rnd Walk: network:species	10.0
Origin Multiplier: originTime:species network:species	5.0
Add Reticulation: coalescenceRate network:species	10.0
Delete Reticulation: coalescenceRate network:species	10.0
Network Multiplier: originTime:species network:specie	5.0
Flip Reticulation: network:species	10.0
Relocate Branch: network:species	10.0
Node Slider: originTime:species network:species	10.0
Node Uniform: network:species	10.0
Relocate Branch Narrow: network:species	10.0
Change Gamma: coalescenceRate	150.0
Change All Gamma: coalescenceRate	150.0
Change UAnd V: u v	10.0

Operator: addReticulation:species

Species Network

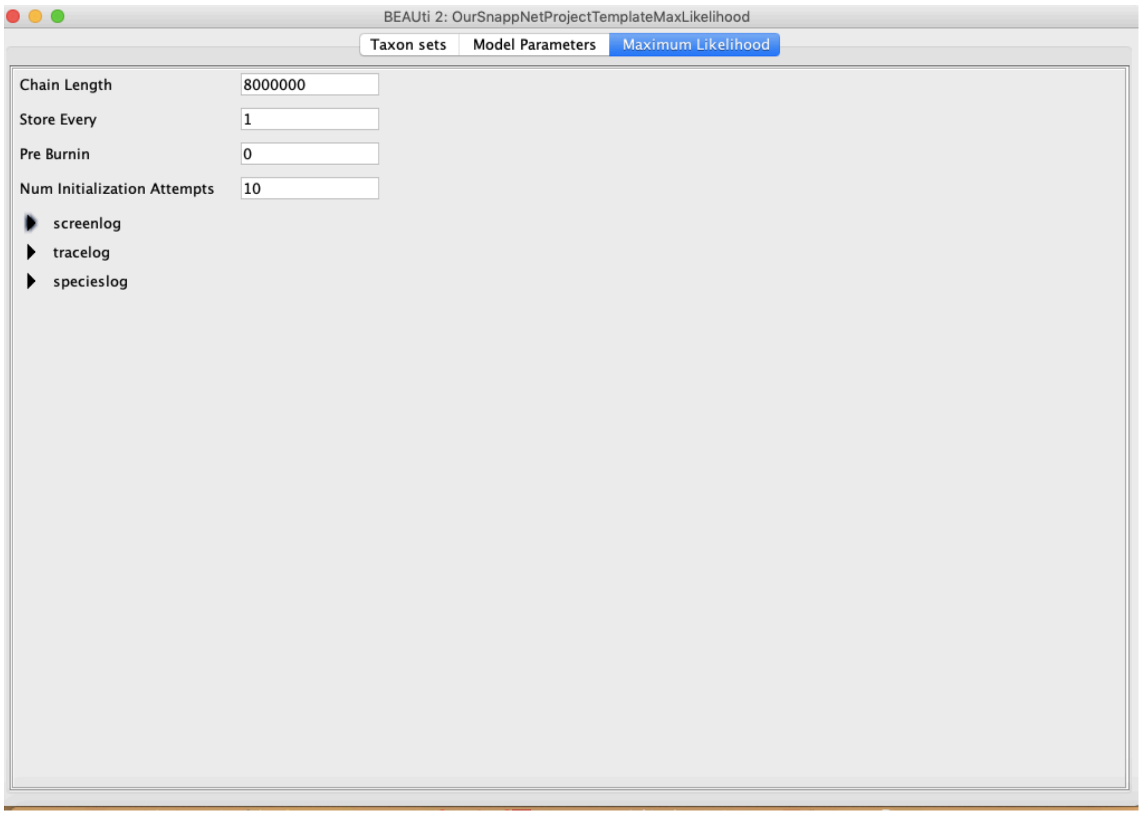
Coalescence Rate	0.01 ✓ Sample
✓ Bound the number of reticulations	
maxReticulationNumber	3
Weight	10.0

Figure 10: Step 7 (Choosing the maximum number of reticulations).



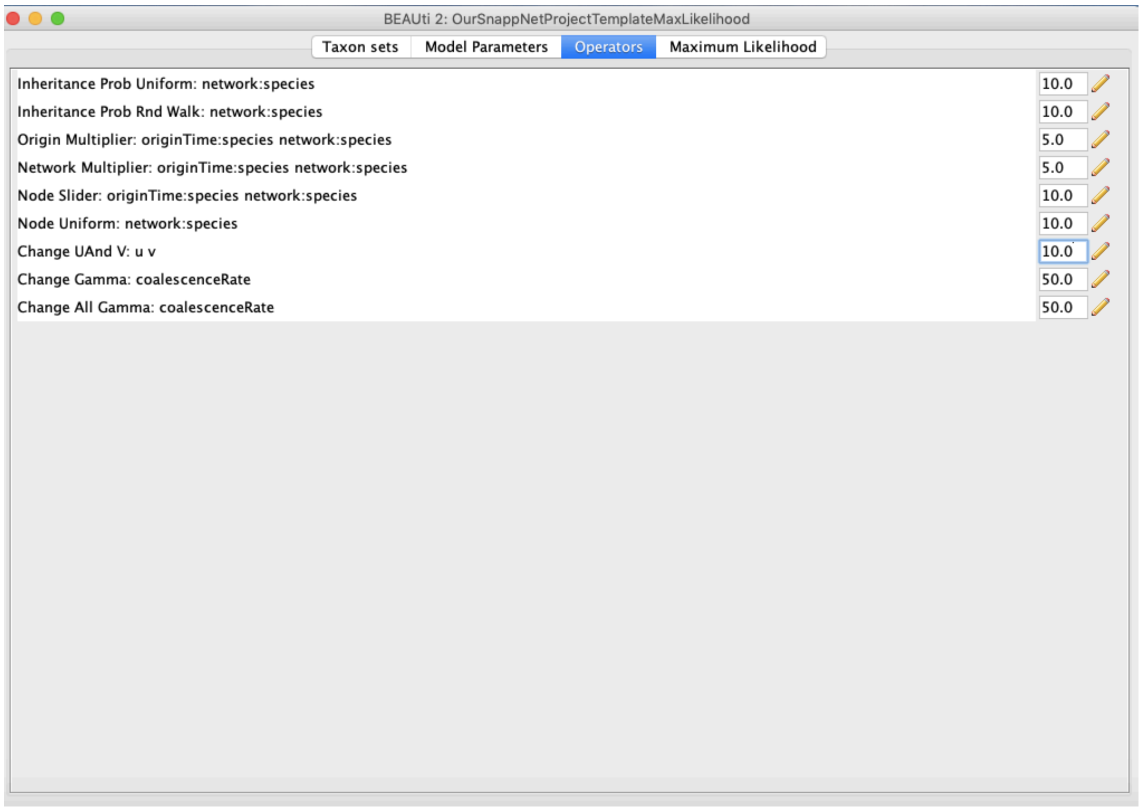
Scale: netDivR	10.0
Scale: turnOve	10.0
Inheritance Pro	10.0
Inheritance Pro	10.0
Origin Multipli …:species	5.0
Add Reticulati	10.0
Delete Reticulation: coalescenceRate network:species	10.0
Network Multiplier: originTime:species network:species	5.0
Flip Reticulation: network:species	10.0
Relocate Branch: network:species	10.0
Node Slider: originTime:species network:species	10.0
Node Uniform: network:species	10.0
Relocate Branch Narrow: network:species	10.0
Change Gamma: coalescenceRate	150.0
Change All Gamma: coalescenceRate	150.0
Change UAnd V: u v	10.0

Figure 11: Step 8 (Saving your xml file).



Chain Length 8000000
Store Every 1
Pre Burnin 0
Num Initialization Attempts 10

Figure 12: Step 4 for ML (Setting the number of iterations).



Inheritance Prob Uniform: network:species	10.0
Inheritance Prob Rnd Walk: network:species	10.0
Origin Multiplier: originTime:species network:species	5.0
Network Multiplier: originTime:species network:species	5.0
Node Slider: originTime:species network:species	10.0
Node Uniform: network:species	10.0
Change UAnd V: u v	10.0
Change Gamma: coalescenceRate	50.0
Change All Gamma: coalescenceRate	50.0

Figure 13: Step 5 for ML (Choosing the weights of the different operators).