

Discrete morphology - Models and Tree Inference

April M. Wright

Introduction to phylogenetic models of morphological evolution

Morphological data is commonly used for estimating phylogenetic trees from fossils. This tutorial will focus on estimating phylogenetic trees from *discrete* characters, those characters which can be broken into non-overlapping character states. This type of data has been used for estimation of phylogenetic trees for many years. In the past twenty years, Bayesian methods for estimating phylogeny from this type of data have become increasingly common.

This tutorial will give an overview of common models and assumptions when estimating a tree from discrete morphological data. We will use a dataset from Zamora, Rahman, and Smith (2013). This dataset contains 23 extinct echinoderm taxa and 60 binary and multistate characters.

When using RevBayes in a notebook, please put any variables you want to echo to the screen last in the markdown cell. See the below cell for an example:

```
example <- 1.0
example
## 1
```

If you will not be printing output, please leave one blank line at the end of your cell. See below for an example:

```
trash <- 2
```

It is important to be aware that some PDF viewers may render some characters given as differently. Thus, if you copy and paste text from this PDF, you may introduce some incorrect characters. Because of this, we recommend that you type the instructions in this tutorial or copy them from the scripts provided.

Data and Files

On your own computer, there should be a directory called **PaleoSoc_phylo_short_course_2019**. The data for this tutorial are in the subdirectory **data**. This directory should contain: **Cinctans_for_RevBayes.nex**.

Getting Started

Create a new directory (in **PaleoSoc_phylo_short_course_2019**) called **scripts**.

When you execute RevBayes in this exercise, you will do so within the main directory (**PaleoSoc_phylo_short_course_2019**), thus, if you are using a Unix-based operating system, we recommend that you add the RevBayes binary to your path. Alternatively make sure that you set the working directory to, for example, **PaleoSoc_phylo_short_course_2019** if this is the directory you stored the scripts and data in.

Creating Rev Files

In this exercise, you will work primarily in this R text editor and create a set of files that will be easily managed and interchanged.

In this section you will begin the file and write the Rev commands for loading in the taxon list and managing the data matrices. Then, starting in section Mk Model, you will move on to specifying each of the model components. Once the model specifications are complete, you will complete the script with the instructions given in section

Load Data Matrices

RevBayes uses the function `readDiscreteCharacterData()` to load a data matrix to the workspace from a formatted file. This function can be used for both molecular sequences and discrete morphological characters. Import the morphological character matrix and assign it the variable `morpho`.

```
morpho <- readDiscreteCharacterData("data/Cinctans.nex")
morpho

## [1] Successfully read one character matrix from file 'data/Cinctans.nex'
## [2] Standard character matrix with 27 taxa and 60 characters
## [3] =====
## [4] Origination:                Cinctans.nex
## [5] Number of taxa:              27
## [6] Number of included taxa:    27
## [7] Number of characters:       60
## [8] Number of included characters: 60
## [9] Datatype:                  Standard

morpho

## [1] Standard character matrix with 27 taxa and 60 characters
## [2] =====
## [3] Origination:                Cinctans.nex
## [4] Number of taxa:              27
## [5] Number of included taxa:    27
## [6] Number of characters:       60
## [7] Number of included characters: 60
## [8] Datatype:                  Standard
```

Create Helper Variables

We will dig into the model momentarily. But first, we will create some variables that are used in our analysis, but are not parameters. We will assign these variables with the constant node assignment operator, `<-`. Even though these values are used in our scripts, they are not parameters of the model.

We will first create a constant node called `num_taxa` that is equal to the number of species in our analysis (23). We will also create a constant node called `num_branches` representing the number of branches in the tree, and one of the taxon names. This list will be used to initialize the tree.

```
taxa <- morpho.names()
num_taxa <- morpho.size()
num_branches <- 2 * num_taxa - 2
num_branches

## 52
```

Next, create two workspace variables called `mvi` and `mni`. These variables are iterators that will build a vector containing all of the MCMC moves used to propose new states for every stochastic node in the model graph. Each time a new move is added to the vector, `mvi` will be incremented by a value of 1.

```
moves = VectorMoves()
monitors = VectorMonitors()
```

One important distinction here is that `mvi` is part of the RevBayes workspace and not the hierarchical model. Thus, we use the workspace assignment operator `=` instead of the constant node assignment `<-`.

The Mk Model

First, we will create a joint prior on the branch lengths.

```

br_len_lambda ~ dnExp(0.2)
moves.append(mvScale(br_len_lambda, weight=2))
moves
## [1] Move[] vector with 1 value
## [2] =====
## [3] [1]
## [4] Scale(br_len_lambda)

```

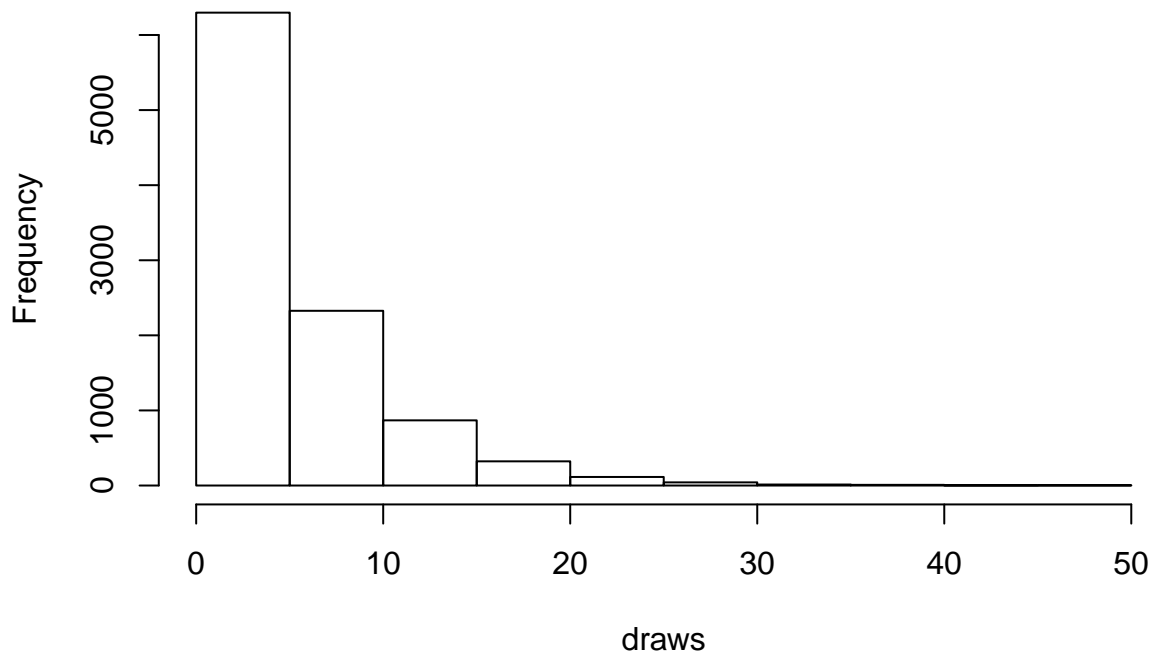
This prior specifies that branch lengths will be drawn from an exponential distribution with parameter 0.2. If you're not familiar with what an exponential distribution, try setting the below code to `eval = FALSE` to run the R code and visualize the distribution. Set `eval = FALSE` when you are done.

```

library(ggplot2)
draws <- rexp(10000, .2)
hist(draws)

```

Histogram of draws



Now, we combine the branch lengths with a uniform prior on topology to make a tree. The uniform prior simply means no tree is more likely *a priori* than any other. This can be easily changed, for example, to use a starting tree. We then specify MCMC moves on the topology, NNI and SPR. These moves propose new topologies. In this way, we propose and evaluate new sets of relationships. We perform these moves frequently because these parameters are really important. We will also move each of the branch lengths each iteration. The scale move scales the current branch length. Finally, we monitor the tree length. This is a quantity many biologists are interested in.

```

phylogeny ~ dnUniformTopologyBranchLength(taxa, branchLengthDistribution=dnExponential(br_len_lambda))
moves.append(mvNNI(phylogeny, weight=num_branches/2.0))
moves.append(mvSPR(phylogeny, weight=num_branches/10.0))
moves.append(mvBranchLengthScale(phylogeny, weight=num_branches))
tree_length := phylogeny.treeLength()

```

We will add Gamma-distributed rate variation and specify moves on the parameter of the Gamma distribution.

```

alpha_morpho ~ dnUniform( 0, 1E6 )
rates_morpho := fnDiscretizeGamma( alpha_morpho, alpha_morpho, 4 )
#Moves on the parameters of the Gamma distribution.
moves.append(mvScale(alpha_morpho, lambda=1, weight=2.0))

```

If you are unfamiliar with the gamma distribution, feel free to run the below code to visualize the distribution.

```

library(ggplot2)
alpha_morpho <- runif(1, 0, 1E6 )

draws <- rgamma(1000, shape = alpha_morpho, rate = alpha_morpho)
hist(draws)

```

Next, we will create a Q -matrix. Recall that the Mk model is simply a generalization of the JC model. Therefore, we will create a Q -matrix using `fnJC`, which initializes Q -matrices with equal transition probabilities between all states. Since we have multistate data, we need to specify different Q -matrices for the different number of character states. For example, it would not make sense to model a 5-state character using a model saying there are only two character states.

To do this, we have written a loop in which we break up the data set into partitions according to the number of character states that character has. Then, we specify a Q -matrix in the correct dimensions. We do not retain any partitions that do not have any characters. For example, if we tried to partition the characters with 4 states, and there were none, we would not create a Q -matrix.

Then, we combine each partition, Gamma-distributed rate heterogeneity, and the tree together into what is called the `phyloCTMC`. This is the joint set of model parameters that will be used the model these data. Each partition is then clamped to its model.

```

n_max_states <- 7
idx = 1
morpho_bystate[1] <- morpho
for (i in 2:n_max_states) {
  # make local tmp copy of data
  # only keep character blocks with state space equal to size i
  morpho_bystate[i] <- morpho
  morpho_bystate[i].setNumStatesPartition(i)
  # get number of characters per character size with i-sized states
  nc = morpho_bystate[i].nchar()
  # for non-empty character blocks
  if (nc > 0) {
    # make i-by-i rate matrix
    q[idx] <- fnJC(i)
  }
  # create model of evolution for the character block
  m_morph[idx] ~ dnPhyloCTMC( tree=phylogeny,
                              Q=q[idx],
                              nSites=nc,
                              siteRates=rates_morpho,
                              type="Standard")

  # attach the data
  m_morph[idx].clamp(morpho_bystate[i])

  # increment counter
  idx = idx + 1
  idx
}
}

```

```
## [1] 2
## [2] 3
## [3] 4
```

We see some familiar pieces: `tree`, `Q-matrix` and `rates_morpho`. We also have two new keywords: `data type` and `coding`. The `data type` argument specifies the type of data - in our case, “Standard”, the specification for morphology. All of the components of the model are now specified.

Complete MCMC Analysis

Create Model Object

We can now create our workspace model variable with our fully specified model DAG. We will do this with the `model()` function and provide a single node in the graph (`phylogeny`).

```
mymodel = model(phylogeny)

## Missing Variable: Variable phylogeny does not exist
## Error: Problem processing line 1 in file "rb2e6e1a5e3835.Rev"
```

The object `mymodel` is a wrapper around the entire model graph and allows us to pass the model to various functions that are specific to our MCMC analysis.

Specify Monitors and Output Filenames

The next important step for our Rev-script is to specify the monitors and output file names. For this, we create a vector called `monitors` that will each sample and record or output our MCMC.

The first monitor we will create will monitor every named random variable in our model graph. This will include every stochastic and deterministic node using the `mnModel` monitor. The only parameter that is not included in the `mnModel` is the tree topology. Therefore, the parameters in the file written by this monitor are all numerical parameters written to a tab-separated text file that can be opened by accessory programs for evaluating such parameters. We will also name the output file for this monitor and indicate that we wish to sample our MCMC every 10 cycles.

```
monitors.append( mnModel(filename="output/mk_gamma.log", printgen=10))

## Missing Variable: Variable monitors does not exist
## Error: Problem processing line 1 in file "rb2e6e2681cd20.Rev"
```

The `mnFile` monitor writes any parameter we specify to file. Thus, if we only cared about the branch lengths and nothing else (this is not a typical or recommended attitude for an analysis this complex) we wouldn't use the `mnModel` monitor above and just use the `mnFile` monitor to write a smaller and simpler output file. Since the tree topology is not included in the `mnModel` monitor (because it is not numerical), we will use `mnFile` to write the tree to file by specifying our `phylogeny` variable in the arguments.

```
monitors.append( mnFile(filename="output/mk_gamma.trees", printgen=10, phylogeny))
```

The third monitor we will add to our analysis will print information to the screen. Like with `mnFile` we must tell `mnScreen` which parameters we'd like to see updated on the screen.

```
monitors.append(mnScreen(printgen=100))
```

Set-Up the MCMC

Once we have set up our model, moves, and monitors, we can now create the workspace variable that defines our MCMC run. We do this using the `mcmc()` function that simply takes the three main analysis components as arguments.

```
mymcmc = mcmc(mymodel, monitors, moves, nruns=2, combine="mixed")
```

```
## [1] Could not compute lnProb for node m_morph[1].
## [2] Standard character matrix with 27 taxa and 60 charact
## [3] Drawing new initial states ...
## [4] Could not compute lnProb for node m_morph[1].
## [5] Standard character matrix with 27 taxa and 60 charact
## [6] Drawing new initial states ...
## [7] Could not compute lnProb for node m_morph[1].
## [8] Standard character matrix with 27 taxa and 60 charact
## [9] Drawing new initial states ...
## [10] Could not compute lnProb for node m_morph[1].
## [11] Standard character matrix with 27 taxa and 60 charact
## [12] Drawing new initial states ...
## [13] Could not compute lnProb for node m_morph[1].
## [14] Standard character matrix with 27 taxa and 60 charact
## [15] Drawing new initial states ...
## [16] Could not compute lnProb for node m_morph[1].
## [17] Standard character matrix with 27 taxa and 60 charact
## [18] Drawing new initial states ...
```

The MCMC object that we named `mymcmc` has a member method called `.run()`. This will execute our analysis and we will set the chain length to 10000 cycles using the `generations` option.

```
mymcmc.run(generations=10000, tuningInterval=200)
```

```
## [1] Running MCMC simulation
## [2] This simulation runs 2 independent replicates.
## [3] The simulator uses 5 different moves in a random move schedule with 87.2 moves per iteration
## [4] Iter          |      Posterior      |      Likelihood      |      Prior      |      elapsed      |      ETA
## [5] -----|-----|-----|-----|-----|-----|-----
## [6] 0          |      -860.776      |      -876.401      |      15.6255      |      00:00:00      |      --:--:--
## [7] 100         |      3285.83       |      3302.11       |      -16.2842     |      00:00:02      |      --:--:--
## [8] 200         |      3827.08       |      3844.83       |      -17.7444     |      00:00:04      |      00:03:16
## [9] 300         |      4144.7        |      4174.41       |      -29.702      |      00:00:06      |      00:03:14
## [10] 400        |      4863.56       |      4882.31       |      -18.7551     |      00:00:08      |      00:03:11
## [11] 500        |      4867.34       |      4888.84       |      -21.5004     |      00:00:10      |      00:03:10
## [12] 600        |      4875.59       |      4899.16       |      -23.5787     |      00:00:12      |      00:03:08
## [13] 700        |      4871.93       |      4884.4        |      -12.4683     |      00:00:14      |      00:03:05
## [14] 800        |      4869.76       |      4897.65       |      -27.8867     |      00:00:16      |      00:03:04
## [15] 900        |      4903.33       |      4921.95       |      -18.6238     |      00:00:18      |      00:03:02
## [16] 1000       |      4897.27       |      4921.08       |      -23.8065     |      00:00:20      |      00:03:00
## [17] 1100       |      4892.12       |      4910.86       |      -18.7494     |      00:00:23      |      00:03:00
## [18] 1200       |      4899.59       |      4923.29       |      -23.7007     |      00:00:27      |      00:03:10
## [19] 1300       |      4914.63       |      4935.61       |      -20.9816     |      00:00:32      |      00:03:34
## [20] 1400       |      4936.11       |      4952.71       |      -16.6037     |      00:00:39      |      00:03:55
## [21] 1500       |      4931.94       |      4954.4        |      -22.4554     |      00:00:45      |      00:04:11
## [22] 1600       |      4931.49       |      4954.27       |      -22.7782     |      00:00:48      |      00:04:11
## [23] 1700       |      4935.77       |      4954.35       |      -18.5752     |      00:00:51      |      00:04:09
## [24] 1800       |      11784.5        |      11816         |      -31.4914     |      00:00:53      |      00:04:01
## [25] 1900       |      11806.6        |      11844.8        |      -38.1869     |      00:00:55      |      00:03:54
## [26] Iter          |      Posterior      |      Likelihood      |      Prior      |      elapsed      |      ETA
## [27] -----|-----|-----|-----|-----|-----
## [28] 2000          |      11813.2       |      11857         |      -43.8192     |      00:00:58      |      00:03:52
## [29] 2100          |      11809.8       |      11847         |      -37.1801     |      00:00:59      |      00:03:41
## [30] 2200          |      11816         |      11842.9        |      -26.9148     |      00:01:01      |      00:03:30
## [31] 2300          |      11805         |      11851.3        |      -46.2466     |      00:01:03      |      00:03:30
```

##	[32]	2400		11807.3		11840.3		-32.9709		00:01:05		00:03:2
##	[33]	2500		11824.2		11861.2		-36.949		00:01:06		00:03:1
##	[34]	2600		11809.8		11855.3		-45.4256		00:01:08		00:03:1
##	[35]	2700		11803.4		11851.3		-47.9014		00:01:10		00:03:0
##	[36]	2800		11811.5		11857		-45.5826		00:01:12		00:03:0
##	[37]	2900		11825.4		11865.7		-40.3871		00:01:16		00:03:0
##	[38]	3000		11834.3		11862.5		-28.1658		00:01:20		00:03:0
##	[39]	3100		11828.4		11873.2		-44.7403		00:01:22		00:03:0
##	[40]	3200		11843.1		11873.9		-30.7753		00:01:24		00:02:5
##	[41]	3300		11834.5		11887		-52.4643		00:01:26		00:02:5
##	[42]	3400		11851.1		11881.8		-30.7019		00:01:28		00:02:5
##	[43]	3500		11850.1		11886.8		-36.6845		00:01:31		00:02:4
##	[44]	3600		11848.3		11885.6		-37.342		00:01:33		00:02:4
##	[45]	3700		11835.2		11876.7		-41.4984		00:01:35		00:02:4
##	[46]	3800		11834.4		11874.6		-40.1648		00:01:37		00:02:3
##	[47]	3900		11869.1		11908.6		-39.4943		00:01:40		00:02:3
##	[48]	Iter		Posterior		Likelihood		Prior		elapsed		ET
##	[49]	-----										
##	[50]	4000		11876.1		11916.5		-40.4577		00:01:42		00:02:3
##	[51]	4100		11864.9		11909.7		-44.7598		00:01:43		00:02:2
##	[52]	4200		11866.7		11916.6		-49.8667		00:01:45		00:02:2
##	[53]	4300		11860.7		11912.4		-51.6618		00:01:47		00:02:2
##	[54]	4400		11874.8		11921.7		-46.8948		00:01:49		00:02:1
##	[55]	4500		11881.2		11919.7		-38.5065		00:01:51		00:02:1
##	[56]	4600		11860.7		11904.9		-44.1578		00:01:53		00:02:1
##	[57]	4700		11866.4		11912.3		-45.867		00:01:55		00:02:0
##	[58]	4800		11901.1		11923		-21.8683		00:01:57		00:02:0
##	[59]	4900		11889.9		11915.4		-25.5212		00:01:59		00:02:0
##	[60]	5000		11889.6		11907.3		-17.6155		00:02:01		00:02:0
##	[61]	5100		11888.7		11918.6		-29.8822		00:02:04		00:01:5
##	[62]	5200		11885.5		11917.9		-32.4249		00:02:06		00:01:5
##	[63]	5300		11890		11915.9		-25.8435		00:02:08		00:01:5
##	[64]	5400		11895.8		11917.2		-21.3966		00:02:10		00:01:5
##	[65]	5500		11871.5		11922.1		-50.5415		00:02:12		00:01:4
##	[66]	5600		11900.6		11934		-33.3698		00:02:14		00:01:4
##	[67]	5700		11886.1		11922.1		-36.0043		00:02:16		00:01:4
##	[68]	5800		11904.6		11925.5		-20.9009		00:02:18		00:01:3
##	[69]	5900		11897.2		11924.7		-27.5142		00:02:20		00:01:3
##	[70]	Iter		Posterior		Likelihood		Prior		elapsed		ET
##	[71]	-----										
##	[72]	6000		11905.3		11928.6		-23.3099		00:02:22		00:01:3
##	[73]	6100		11894.8		11913.8		-19.0356		00:02:24		00:01:3
##	[74]	6200		11900.1		11922.8		-22.7074		00:02:26		00:01:2
##	[75]	6300		11892.7		11924.7		-32.0543		00:02:29		00:01:2
##	[76]	6400		11896.1		11920.7		-24.5423		00:02:31		00:01:2
##	[77]	6500		11906.2		11928.5		-22.2948		00:02:33		00:01:2
##	[78]	6600		11920		11936		-15.9478		00:02:35		00:01:1
##	[79]	6700		11904.1		11941.5		-37.4459		00:02:38		00:01:1
##	[80]	6800		11906.2		11930.4		-24.1496		00:02:40		00:01:1
##	[81]	6900		11918.8		11938.7		-19.8934		00:02:42		00:01:1
##	[82]	7000		11908.8		11938.6		-29.725		00:02:44		00:01:1
##	[83]	7100		11910.8		11940.9		-30.1373		00:02:46		00:01:0
##	[84]	7200		11914		11943.9		-29.862		00:02:48		00:01:0
##	[85]	7300		11916		11938.2		-22.2206		00:02:50		00:01:0

## [86]	7400	11908.8	11938.2	-29.3972	00:02:52	00:01:00
## [87]	7500	11908.5	11939.7	-31.2378	00:02:54	00:00:58
## [88]	7600	11927.6	11952.5	-24.8992	00:02:56	00:00:56
## [89]	7700	11919.2	11952.6	-33.3954	00:02:59	00:00:53
## [90]	7800	11916.9	11944.9	-28.03	00:03:01	00:00:51
## [91]	7900	11933.1	11959.5	-26.4738	00:03:03	00:00:49
## [92]	Iter	Posterior	Likelihood	Prior	elapsed	ET
## [93]	-----					
## [94]	8000	11915	11942.1	-27.0981	00:03:05	00:00:47
## [95]	8100	11928.4	11956.4	-28.0658	00:03:07	00:00:45
## [96]	8200	11921.7	11959.6	-37.8464	00:03:08	00:00:44
## [97]	8300	11916.6	11952.1	-35.4643	00:03:10	00:00:38
## [98]	8400	11913	11942.5	-29.5377	00:03:12	00:00:36
## [99]	8500	11916.7	11944.5	-27.8434	00:03:14	00:00:34
## [100]	8600	11933.6	11954.5	-20.8311	00:03:16	00:00:32
## [101]	8700	11920.2	11948.6	-28.4595	00:03:17	00:00:31
## [102]	8800	11925.2	11947.3	-22.069	00:03:19	00:00:29
## [103]	8900	11912.9	11949.3	-36.3197	00:03:21	00:00:27
## [104]	9000	11910.3	11943	-32.6793	00:03:23	00:00:25
## [105]	9100	11936.8	11950.5	-13.6902	00:03:25	00:00:23
## [106]	9200	11938.3	11959.1	-20.811	00:03:27	00:00:21
## [107]	9300	11932.3	11950.9	-18.5895	00:03:28	00:00:20
## [108]	9400	11929.9	11951.2	-21.3612	00:03:30	00:00:18
## [109]	9500	11935.5	11966.8	-31.2959	00:03:32	00:00:16
## [110]	9600	11943.4	11968.6	-25.1669	00:03:34	00:00:14
## [111]	9700	11938.7	11966	-27.2945	00:03:36	00:00:12
## [112]	9800	11914.3	11951.6	-37.2817	00:03:38	00:00:10
## [113]	9900	11928.6	11965.6	-36.9328	00:03:40	00:00:08
## [114]	Iter	Posterior	Likelihood	Prior	elapsed	ET
## [115]	-----					
## [116]	10000	11933.8	11960.9	-27.1858	00:03:41	00:00:06

When the analysis is complete, RevBayes will quit and you will have a new directory called **output** that will contain all of the files you specified with the monitors.

We can look at the log files in the software **Tracer**. We can also calculate several different types of summary trees:

```
# Read in the tree trace and construct the maximum clade credibility (MCC) tree #
trace = readTreeTrace("output/mk_gamma.trees")

# Summarize tree trace and save MCC tree to file
mccTree(trace, file="output/mk_gamma.mcc.tre" )

## [1] *****
## [2] Compiling maximum clade credibility tree from 2002 trees in tree trace, using a burnin of 500 trees
## [3] Summarizing clades ...
## [4] Progress:
## [5] 0-----25-----50-----75-----100
## [6] *****
## [7] Annotating tree ...
## [8] ((((((((((Trochocystites_bohemicus[&index=20,age_95%_HPD={0,1.21818}]:0.038547,Trochocystoides_p...
```

RevBayes can calculate MCC trees, MAP trees, and consensus trees. Have each person at your table try one, and see how they differ.

We can also use R Packages, such as **RWTY** to look at convergence in our sample:


```

library(rwty)

## Loading required package: ape

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2

my.trees <- load.trees("output/mk_gamma.trees", format="revbayes", logfile = "output/mk_gamma.log", skip=1)

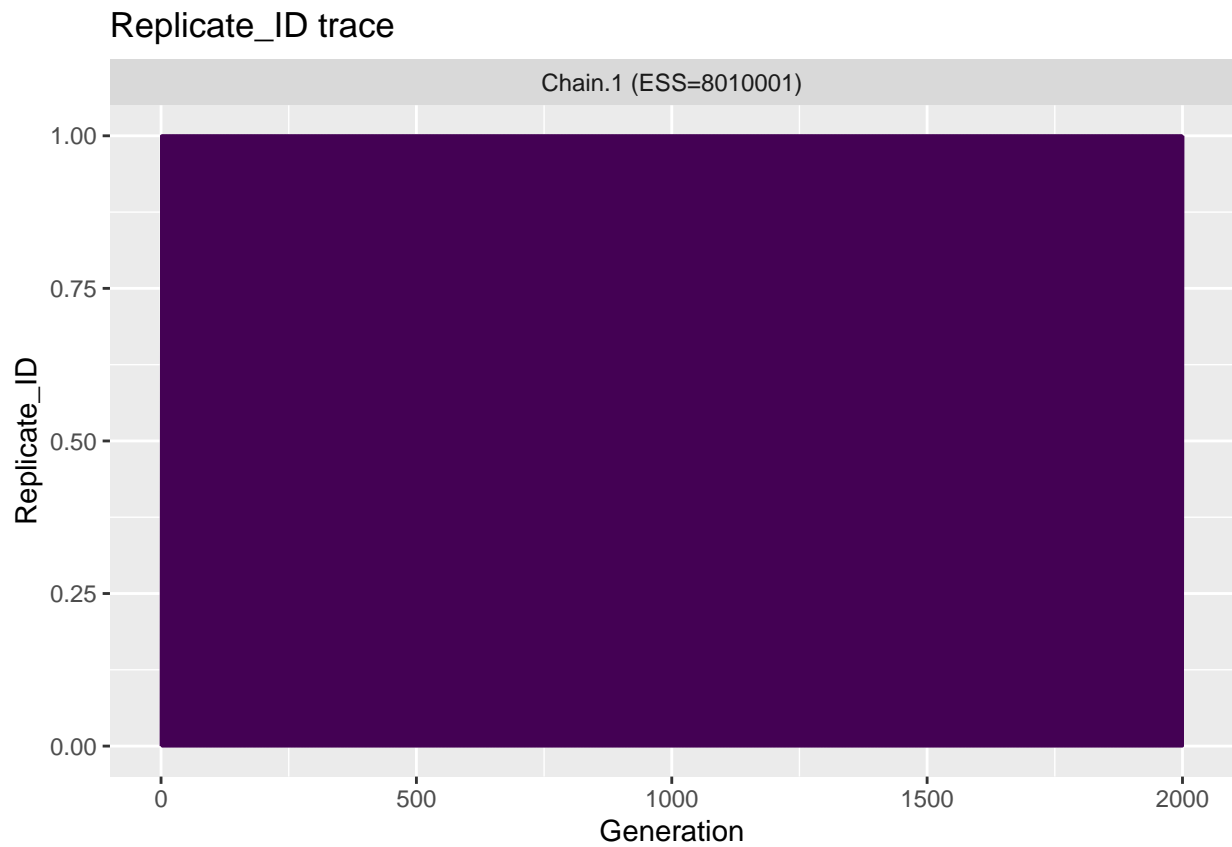
## [1] "Reading trees..."
## [1] "1 generations per tree..."
## [1] "Unrooting, this may take a while..."
## [1] "Reading parameter values from mk_gamma.log"

makeplot.all.params(my.trees, burnin=0)

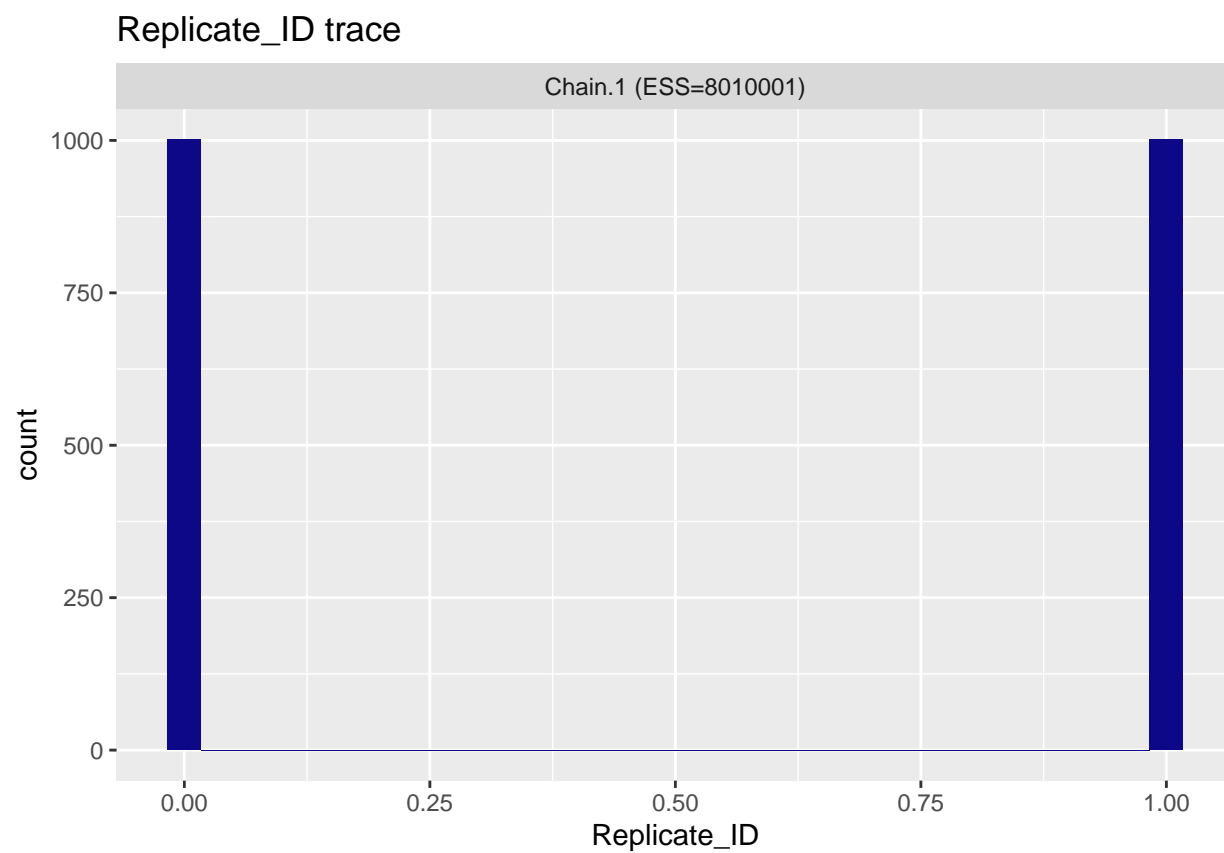
## [1] "Creating trace for Replicate_ID"
## [1] "Creating trace for Posterior"
## [1] "Creating trace for Likelihood"
## [1] "Creating trace for Prior"
## [1] "Creating trace for alpha_morpho"
## [1] "Creating trace for br_len_lambda"
## [1] "Creating trace for rates_morpho.1."
## [1] "Creating trace for rates_morpho.2."
## [1] "Creating trace for rates_morpho.3."
## [1] "Creating trace for rates_morpho.4."
## [1] "Creating trace for tree_length"
## [1] "Creating trace for tree topologies"
## [1] "Calculating approximate ESS with sampling intervals from 1 to 100"

## $Replicate_ID.trace
## $Replicate_ID.trace$trace.plot

```

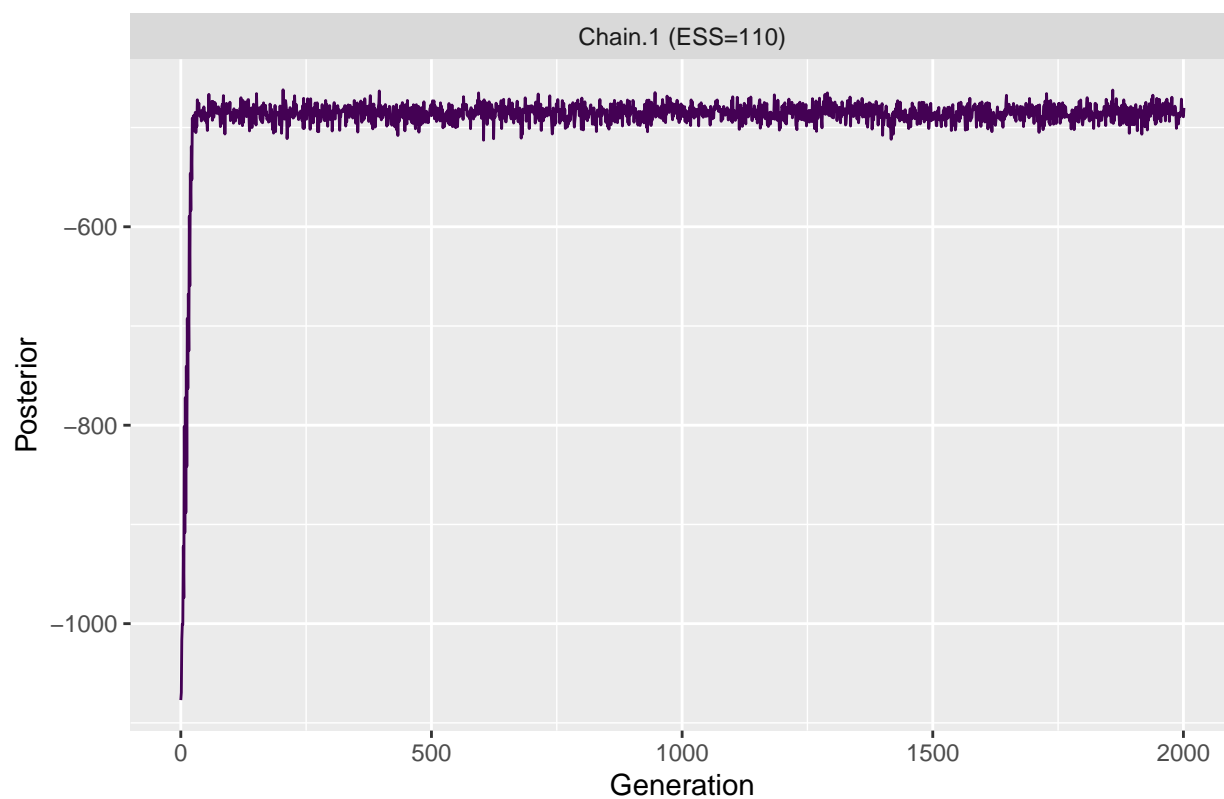


```
##  
## $Replicate_ID.trace$density.plot  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

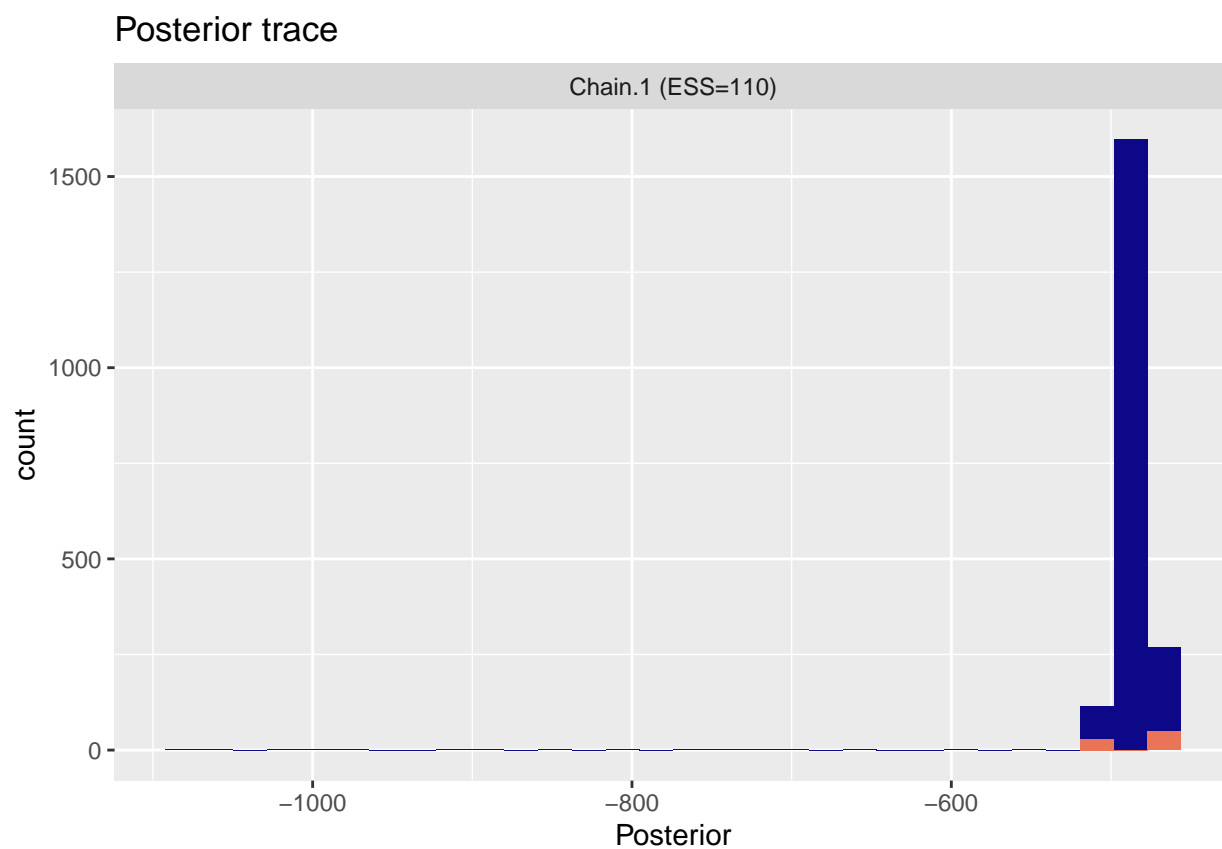


```
##  
##  
## $Posterior.trace  
## $Posterior.trace$trace.plot
```

Posterior trace

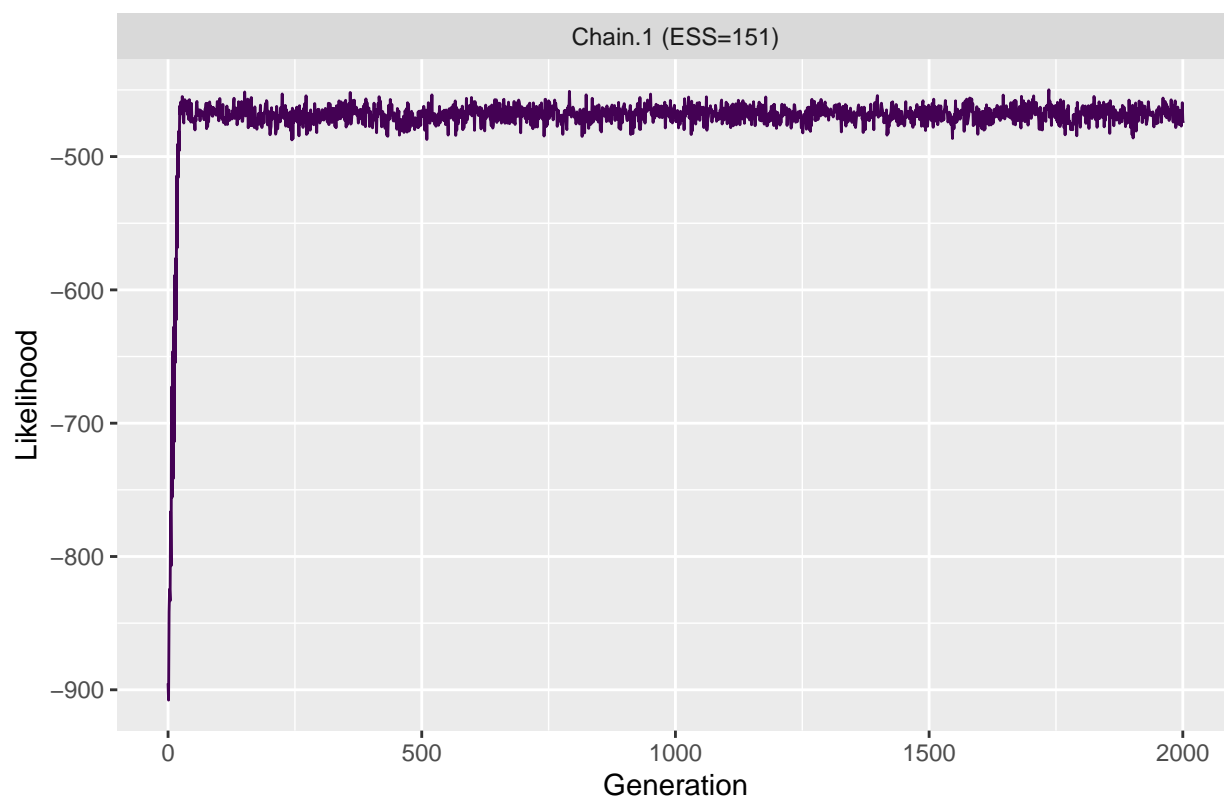


```
##  
## $Posterior.trace$density.plot  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

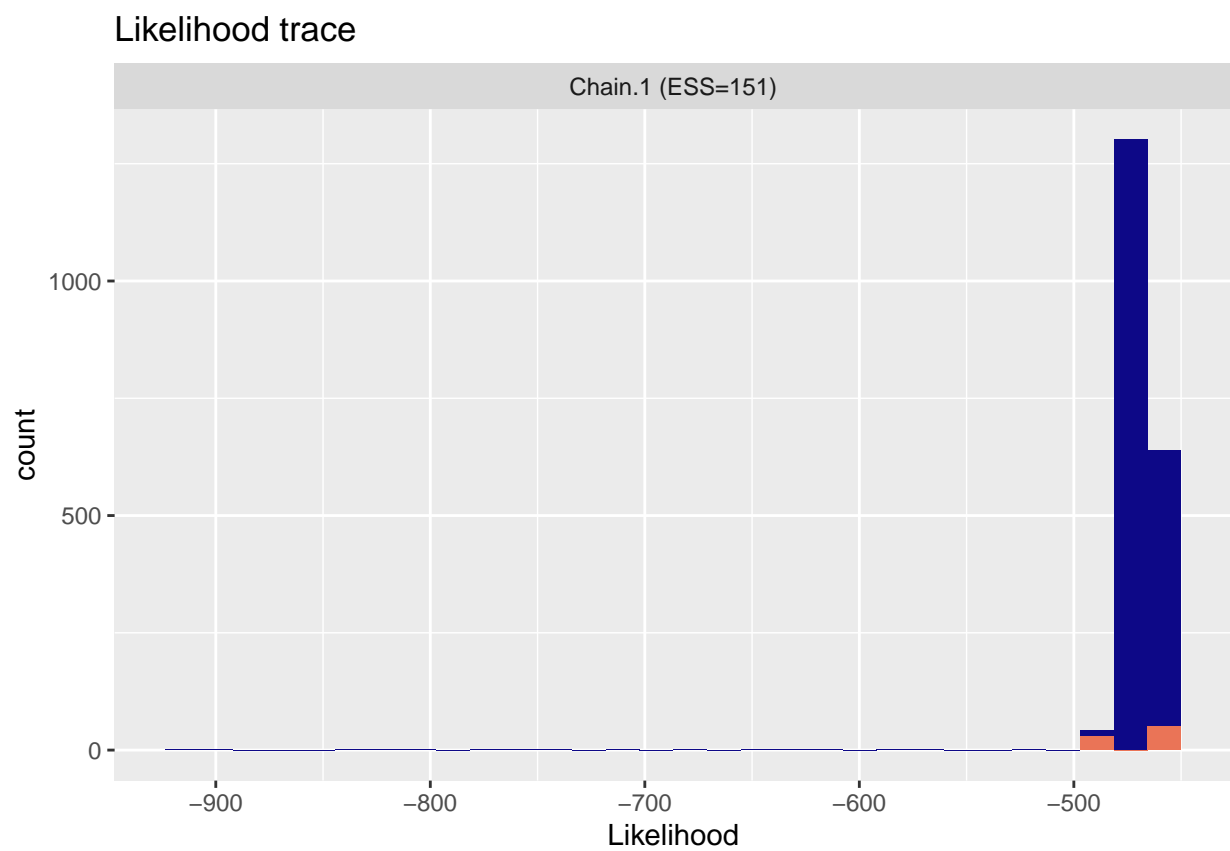


```
##  
##  
## $Likelihood.trace  
## $Likelihood.trace$trace.plot
```

Likelihood trace

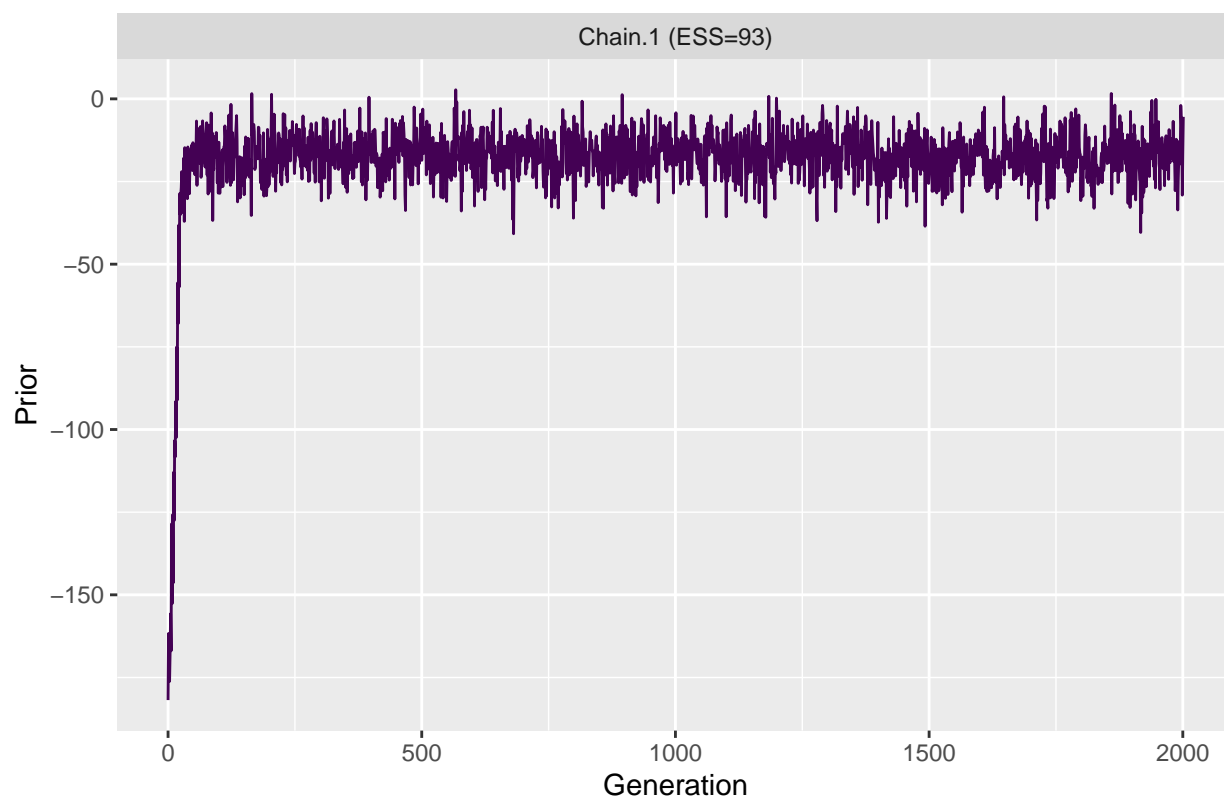


```
##  
## $Likelihood.trace$density.plot  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

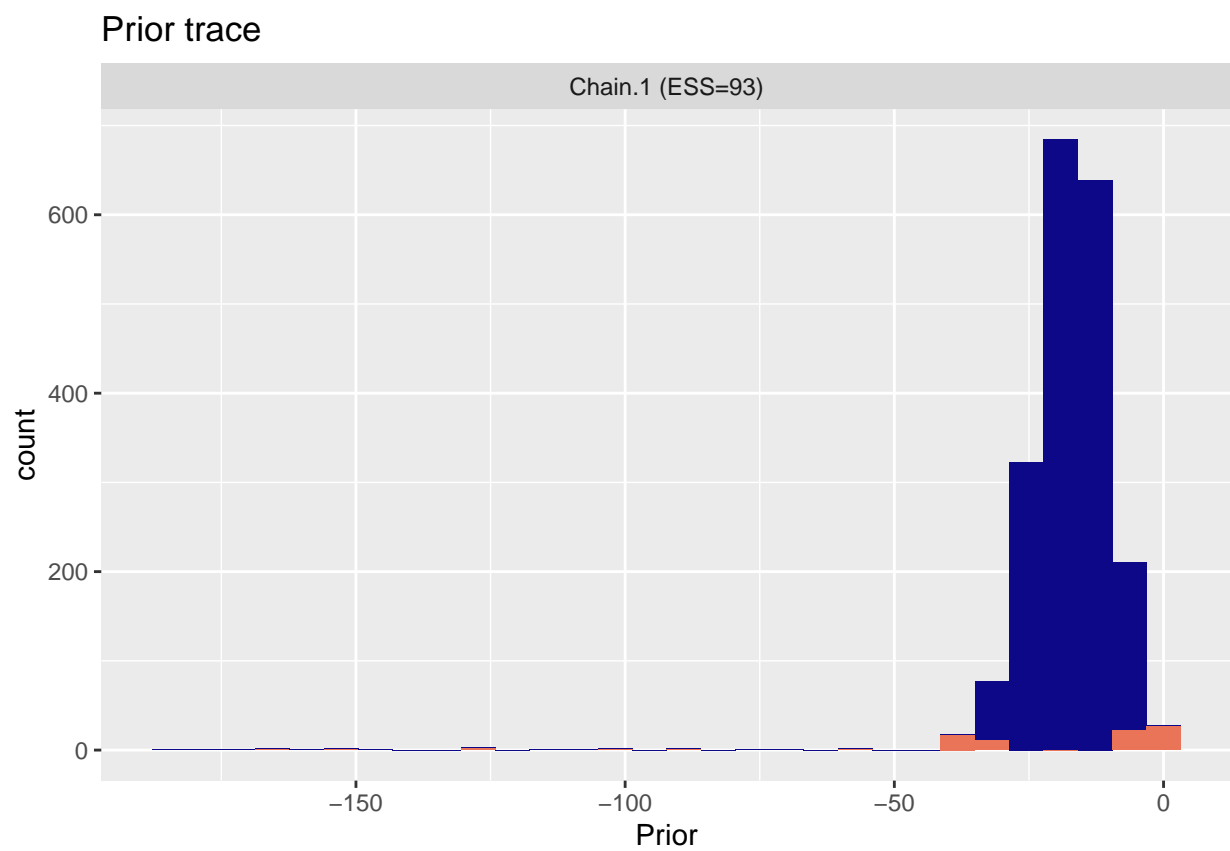


```
##  
##  
## $Prior.trace  
## $Prior.trace$trace.plot
```

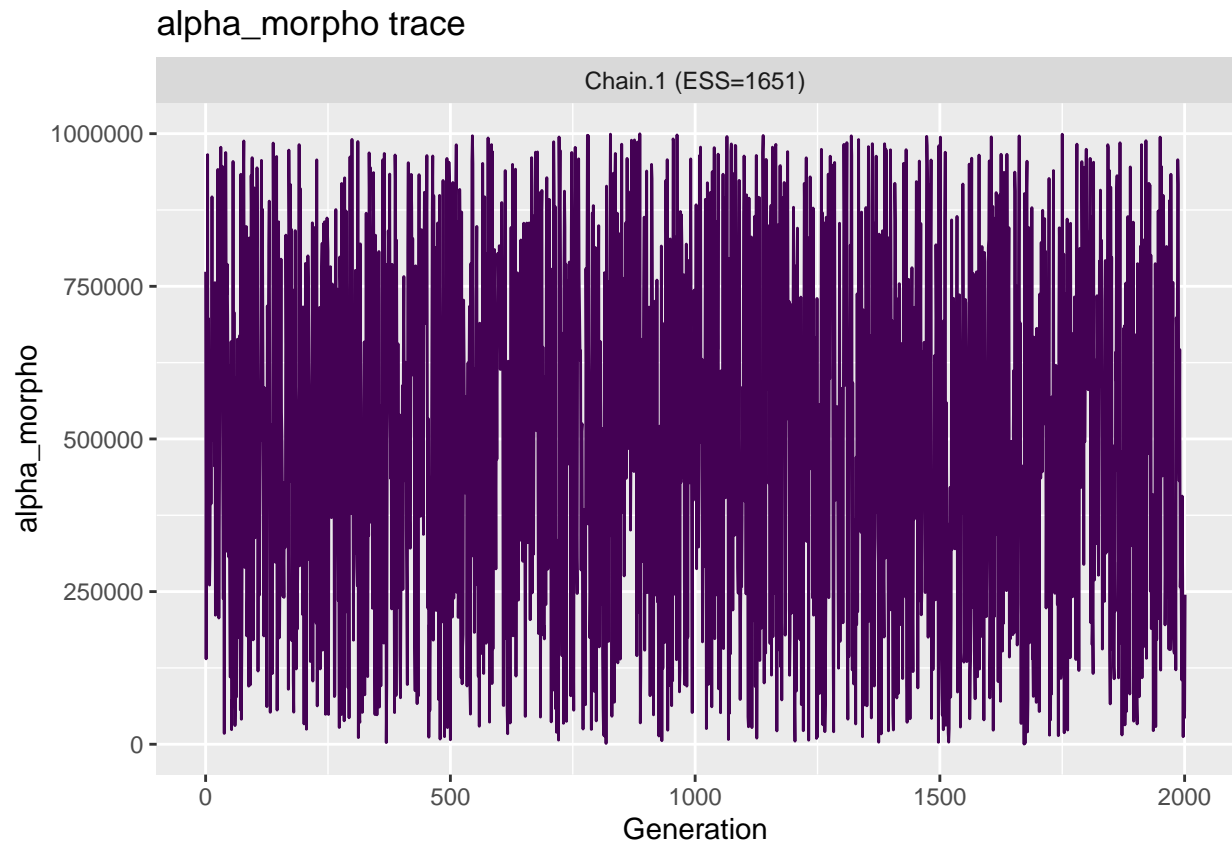
Prior trace



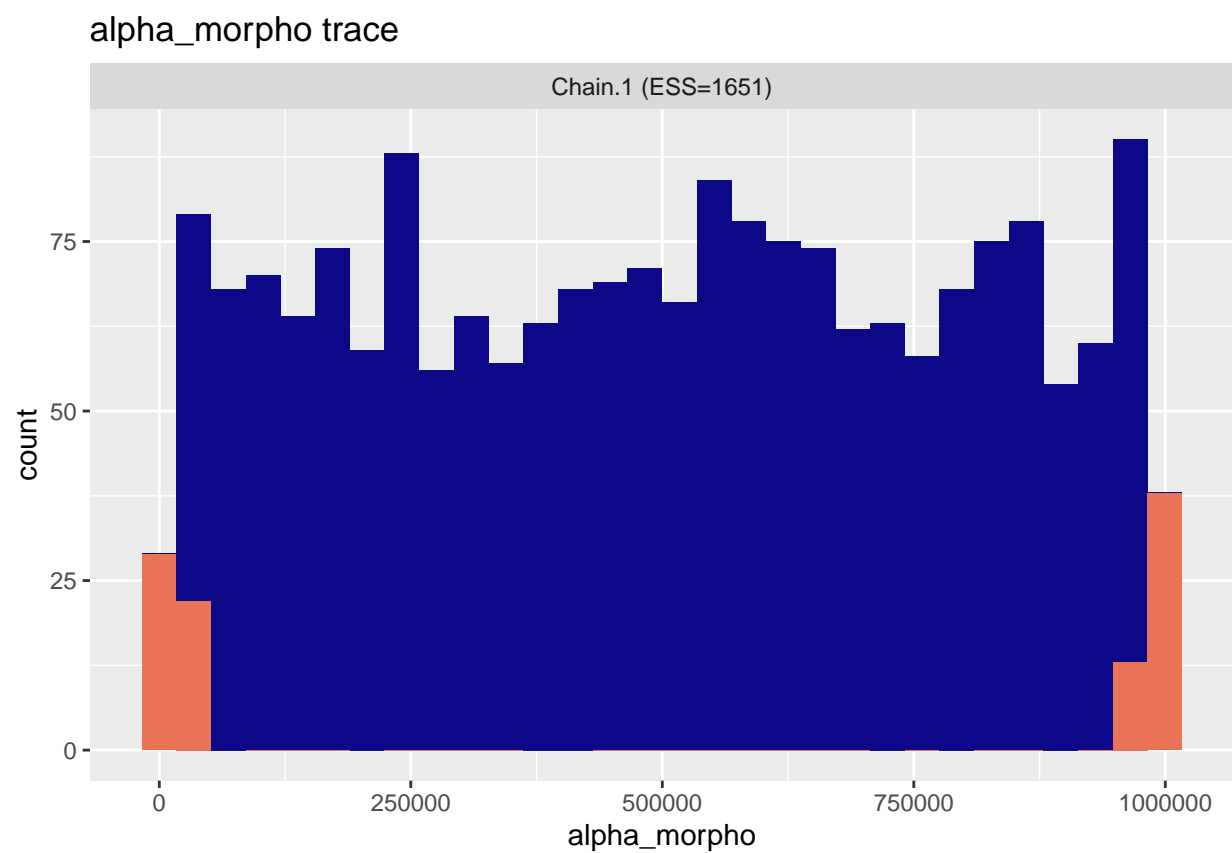
```
##  
## $Prior.trace$density.plot  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
##  
##  
## $alpha_morpho.trace  
## $alpha_morpho.trace$trace.plot
```

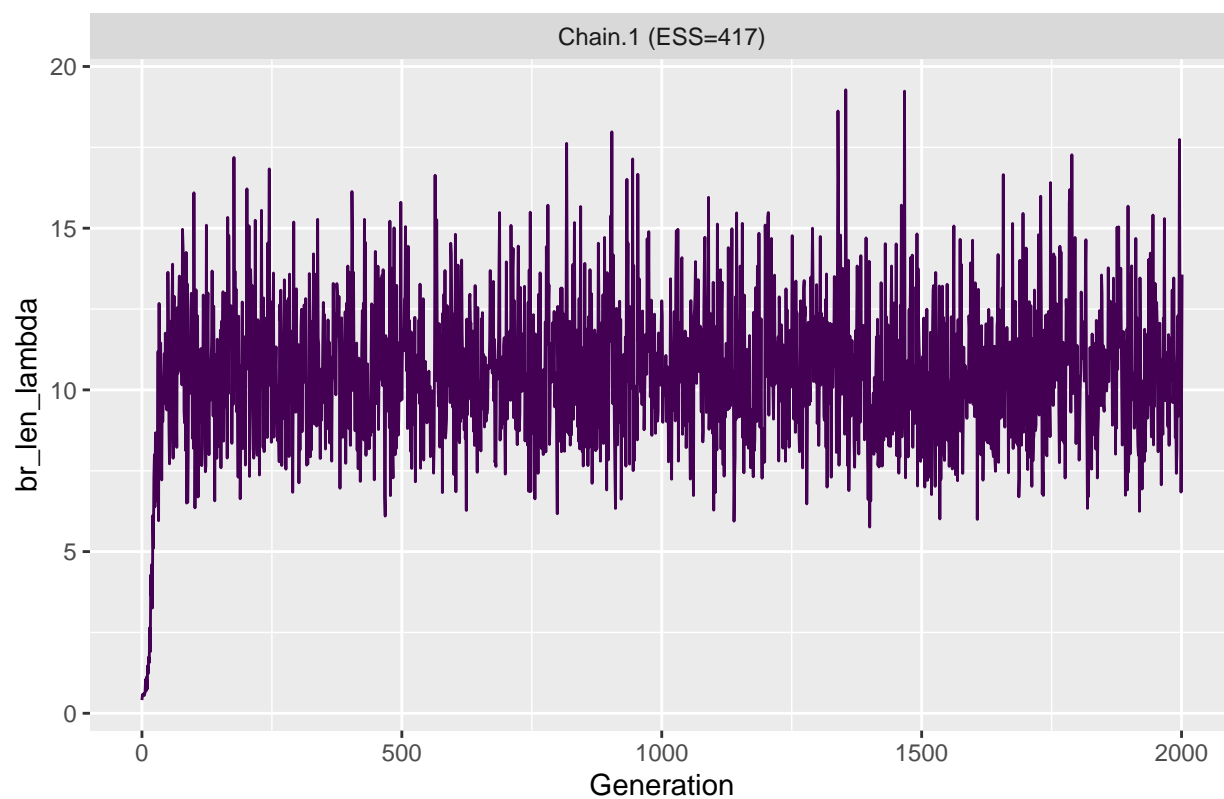


```
##  
## $alpha_morpho.trace$density.plot  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

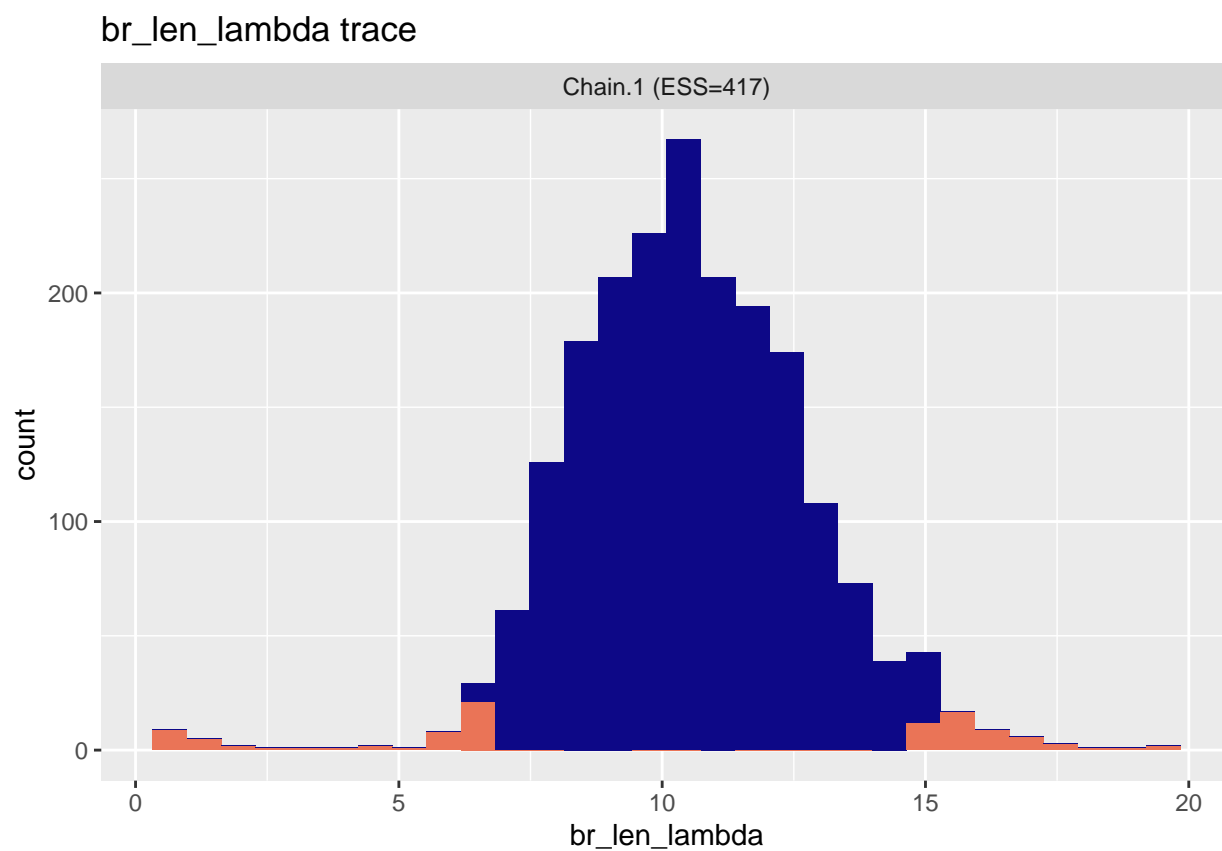


```
##  
##  
## $br_len_lambda.trace  
## $br_len_lambda.trace$trace.plot
```

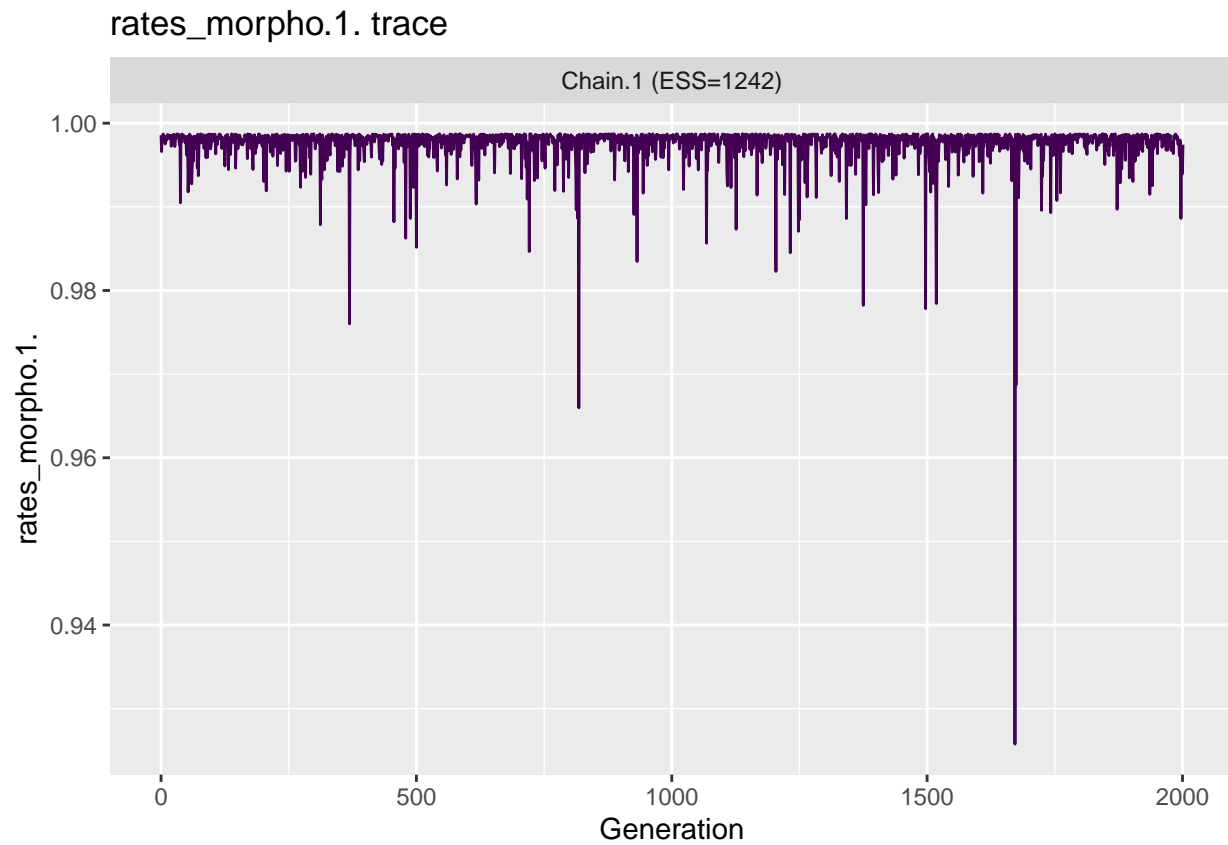
br_len_lambda trace



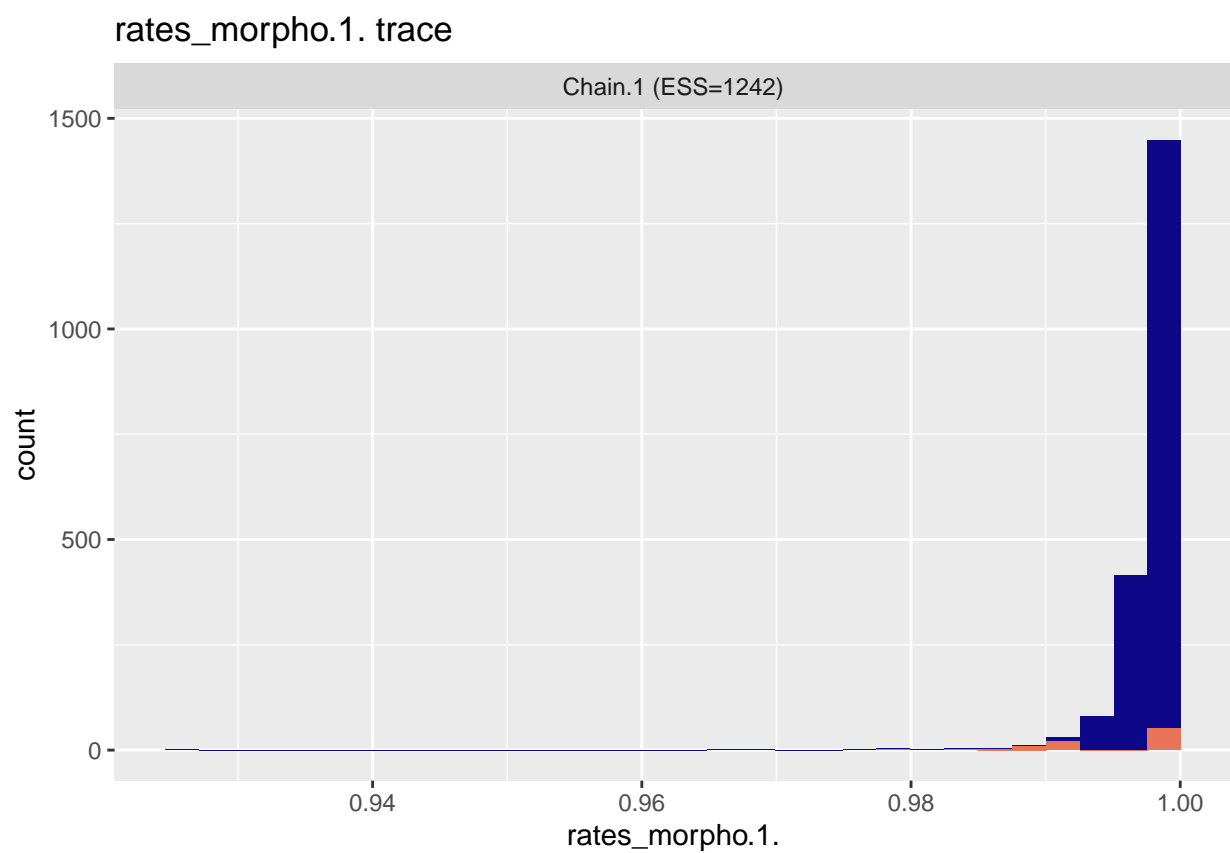
```
##  
## $br_len_lambda.trace$density.plot  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



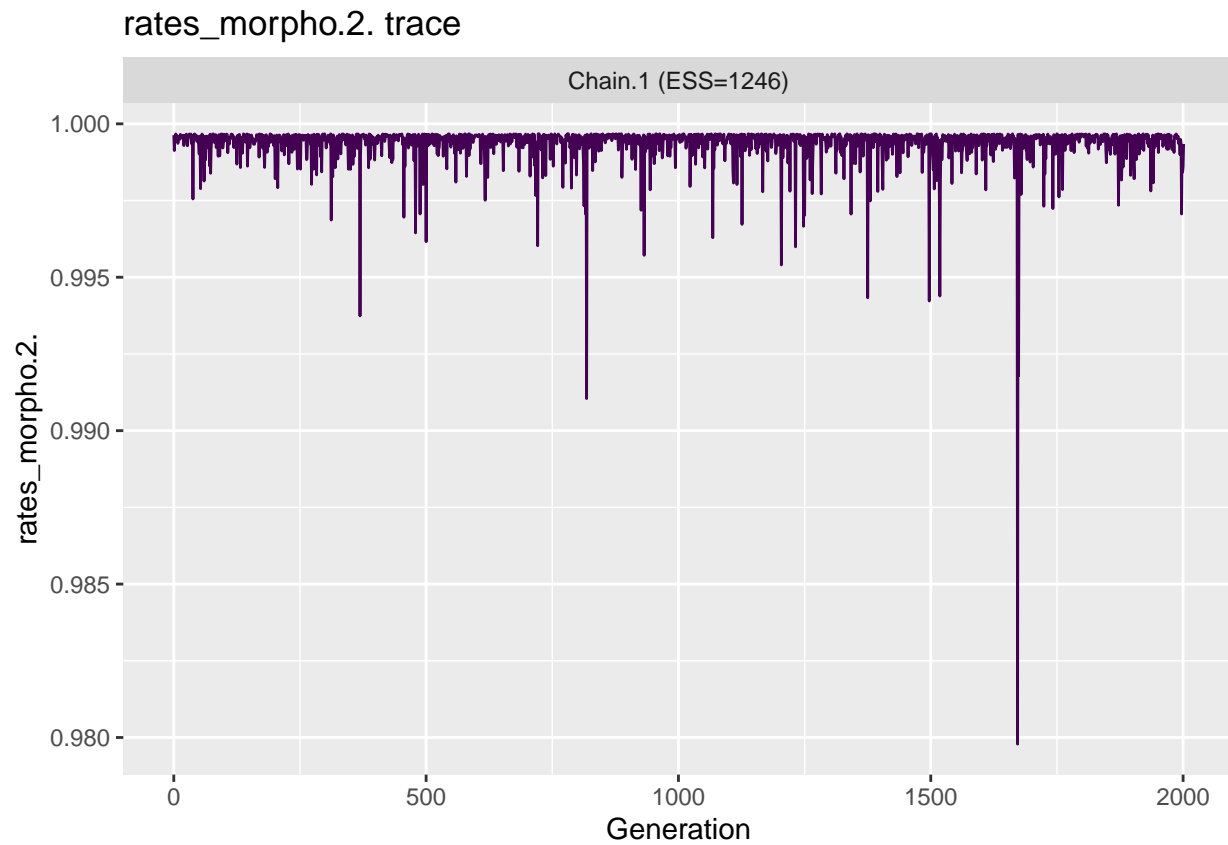
```
##  
##  
## $rates_morpho.1..trace  
## $rates_morpho.1..trace$trace.plot
```



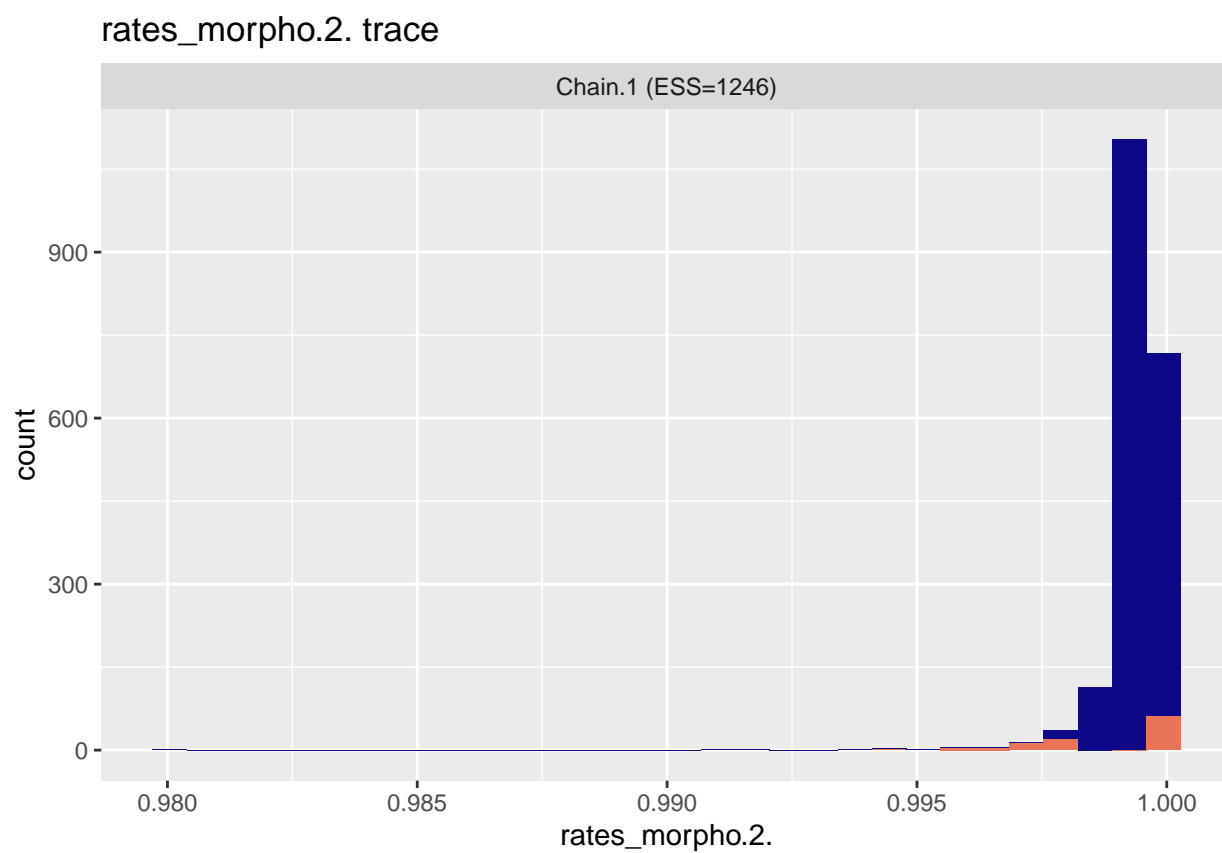
```
##  
## $rates_morpho.1..trace$density.plot  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



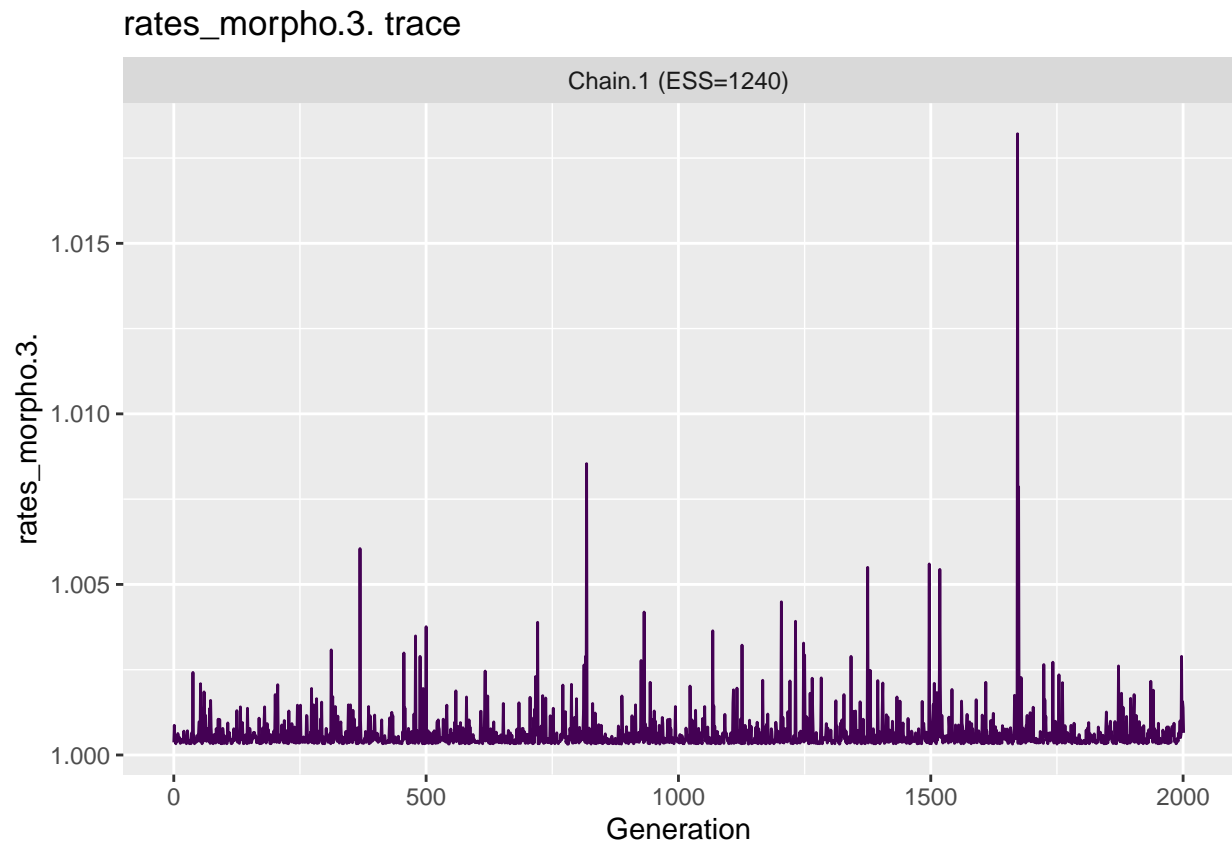
```
##  
##  
## $rates_morpho.2..trace  
## $rates_morpho.2..trace$trace.plot
```



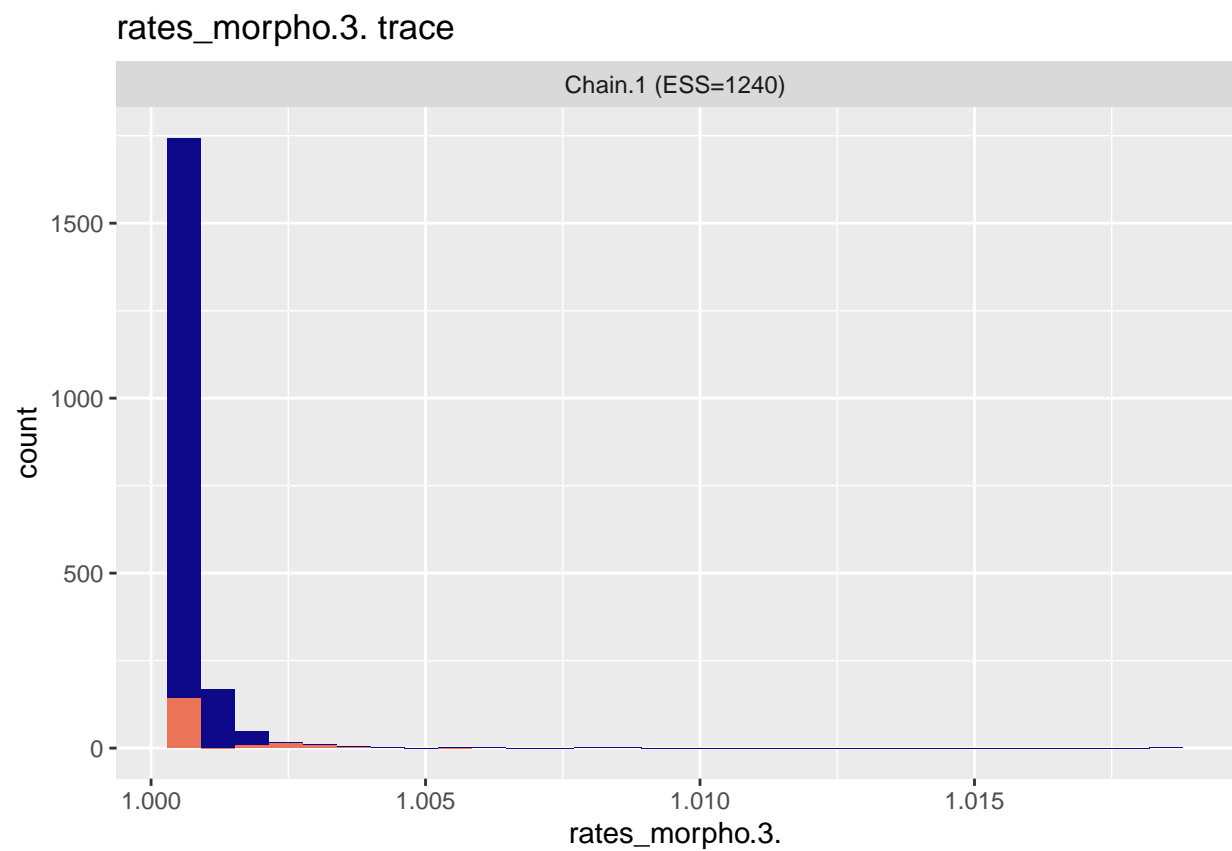
```
##  
## $rates_morpho.2..trace$density.plot  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

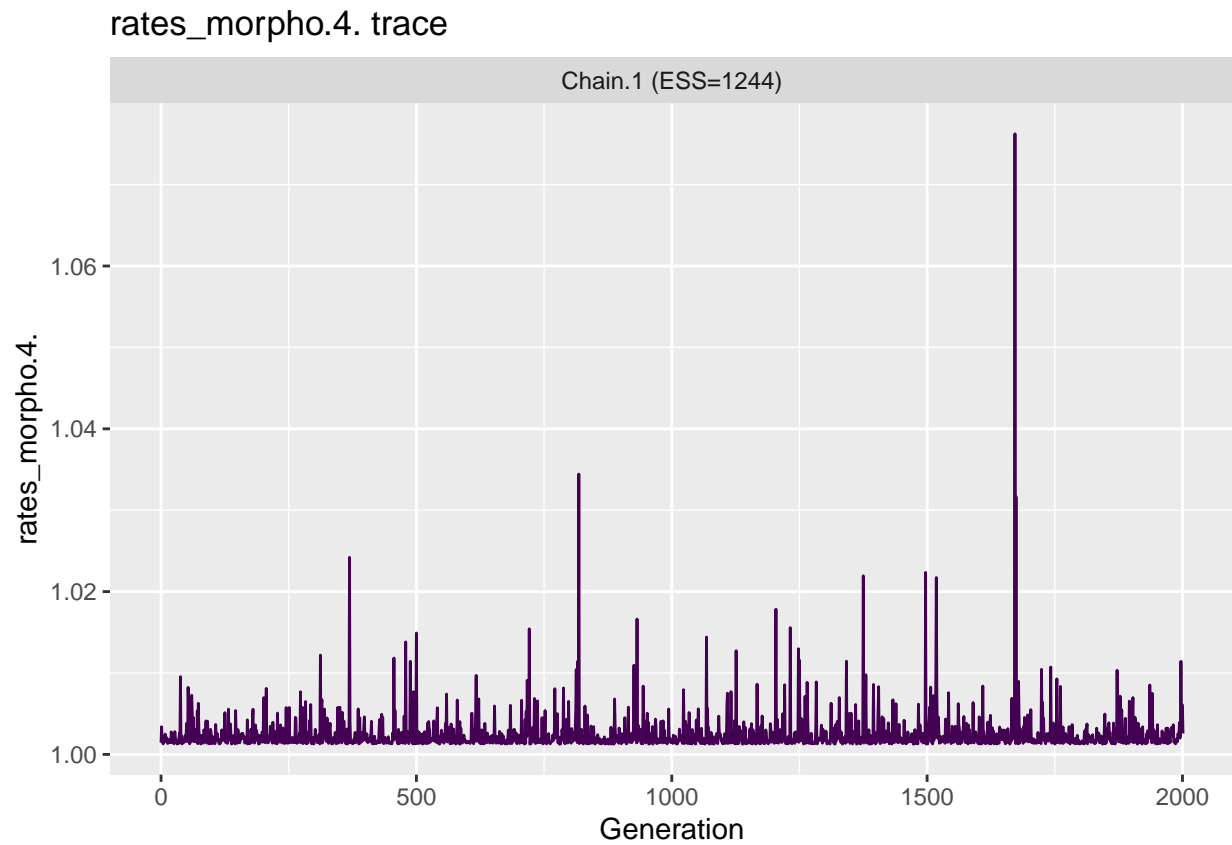
```
##  
##  
## $rates_morpho.3..trace  
## $rates_morpho.3..trace$trace.plot
```



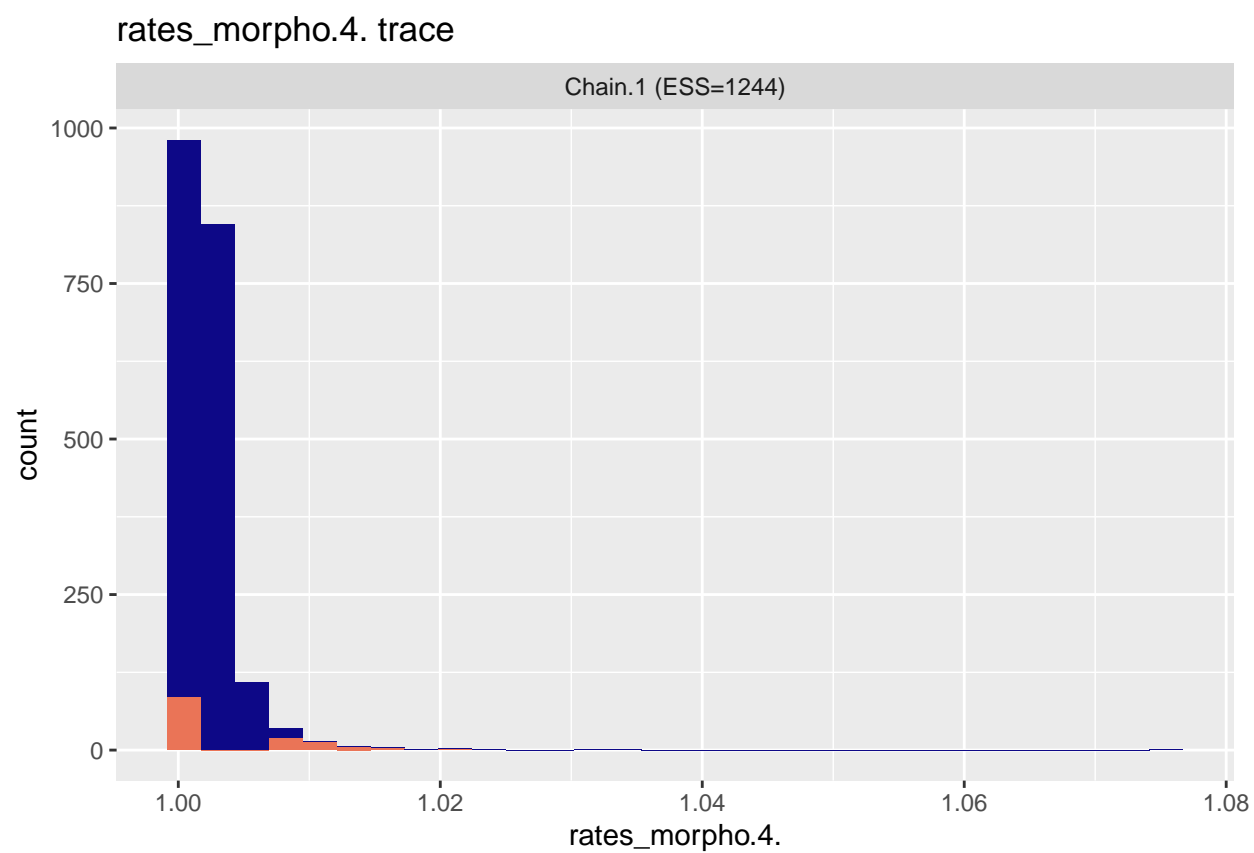
```
##  
## $rates_morpho.3..trace$density.plot  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



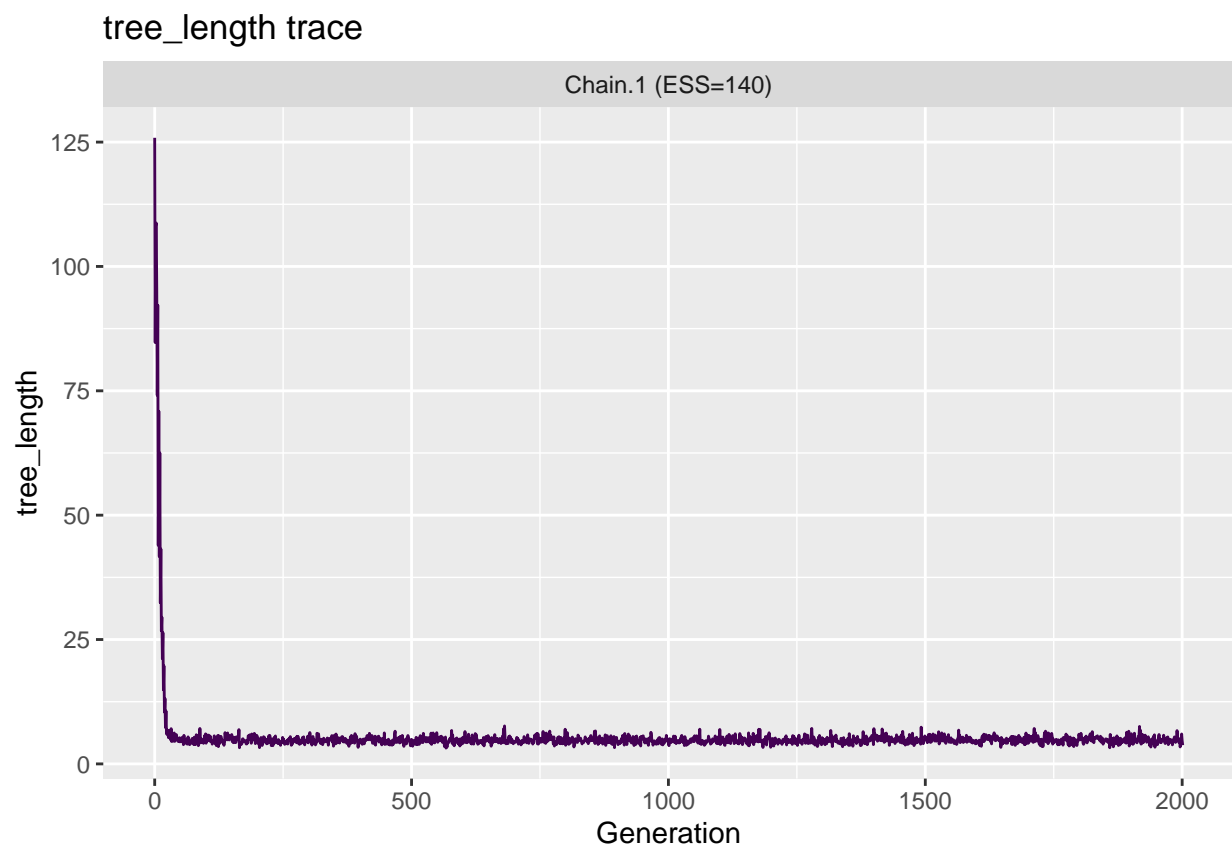
```
##  
##  
## $rates_morpho.4..trace  
## $rates_morpho.4..trace$trace.plot
```



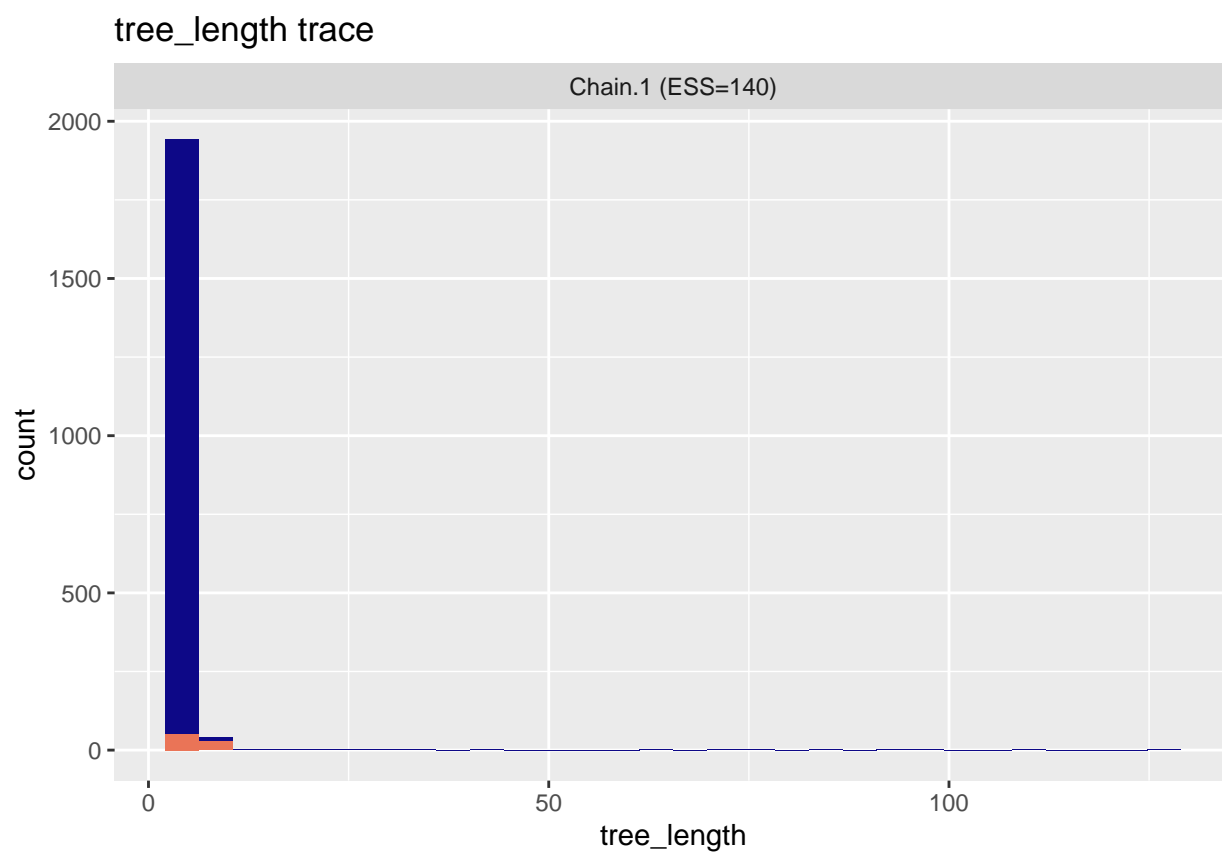
```
##  
## $rates_morpho.4..trace$density.plot  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
##  
##  
## $tree_length.trace  
## $tree_length.trace$trace.plot
```

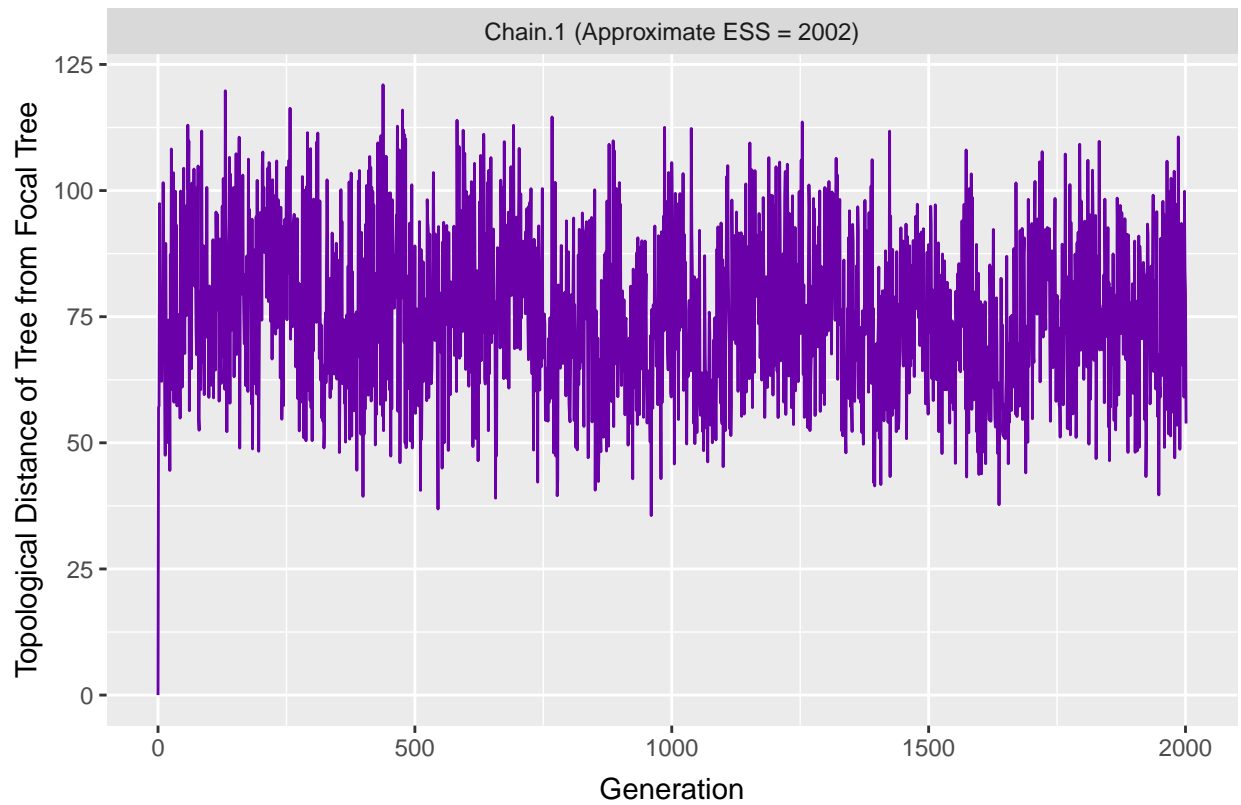


```
##  
## $tree_length.trace$density.plot  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

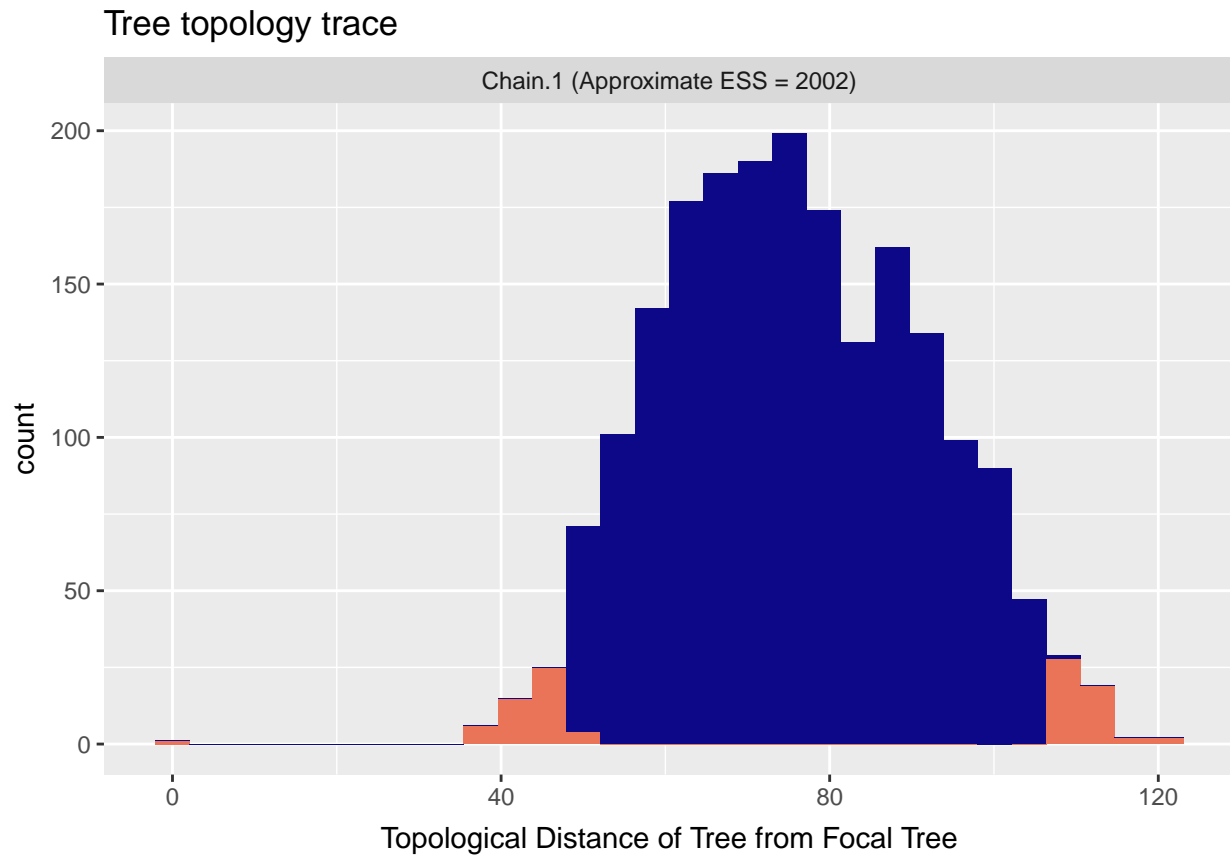


```
##  
##  
## $topology.trace.plot  
## $topology.trace.plot$trace.plot
```

Tree topology trace



```
##  
## $topology.trace.plot$density.plot  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

References

Zamora, Samuel, Imran A Rahman, and Andrew B Smith. 2013. "The Ontogeny of Cinctans (Stem-Group Echinodermata) as Revealed by a New Genus, Graciacystis, from the Middle Cambrian of Spain." *Palaeontology* 56 (2): 399–410.