Assessing MCMC convergence in R

Smithsonian Peer-led Bioinformatics Series

Oct. 25, 2016

Dietrich Gotzek

I’ll assume…

- you are familiar with MCMC
- you are familiar with your favorite software’s built-in convergence diagnostics
- you already use *Tracer* to:
  - visualize likelihood traces
  - estimate ESS
  - autocorrelation?
  - parameter correlations?

MCMC Approximation of the Joint Posterior Probability Density

MCMC in theory and practice

MCMC in theory...

an <u>appropriately constructed</u> and <u>adequately run</u> chain is guaranteed to provide an arbitrarily precise description of the joint stationary density

MCMC in practice...

although a given sampler may work well in most cases, all samplers will fail in some cases, and is not guaranteed to work for any given case

Q. When do we know that the MCMC provides an accurate approximation for a given empirical analysis?

A.

MCMC Approximation of the Joint Posterior Probability Density

MCMC in theory and practice

MCMC in theory...

an <u>appropriately constructed</u> and <u>adequately run</u> chain is guaranteed to provide an arbitrarily precise description of the joint stationary density

MCMC in practice...

although a given sampler may work well in most cases, all samplers will fail in some cases, and is not guaranteed to work for any given case

Q. When do we know that the MCMC provides an accurate approximation for a given empirical analysis?

A.

NEVER!

Brian Moore (treethinkers.org)

MCMC Approximation of the Joint Posterior Probability Density

MCMC performance and OCD

It is not sufficient to merely be deeply concerned about MCMC performance…you need to be completely obsessed about it!

- for any Bayesian inference based on MCMC
- particularly for complex models/inference problems

I will not talk about *CODA* or *BOA*…

- Both more or less do the same thing
- Easy to use (menu driven)
- Implement useful descriptive statistics, convergence diagnostics, and plots for continuous parameters:

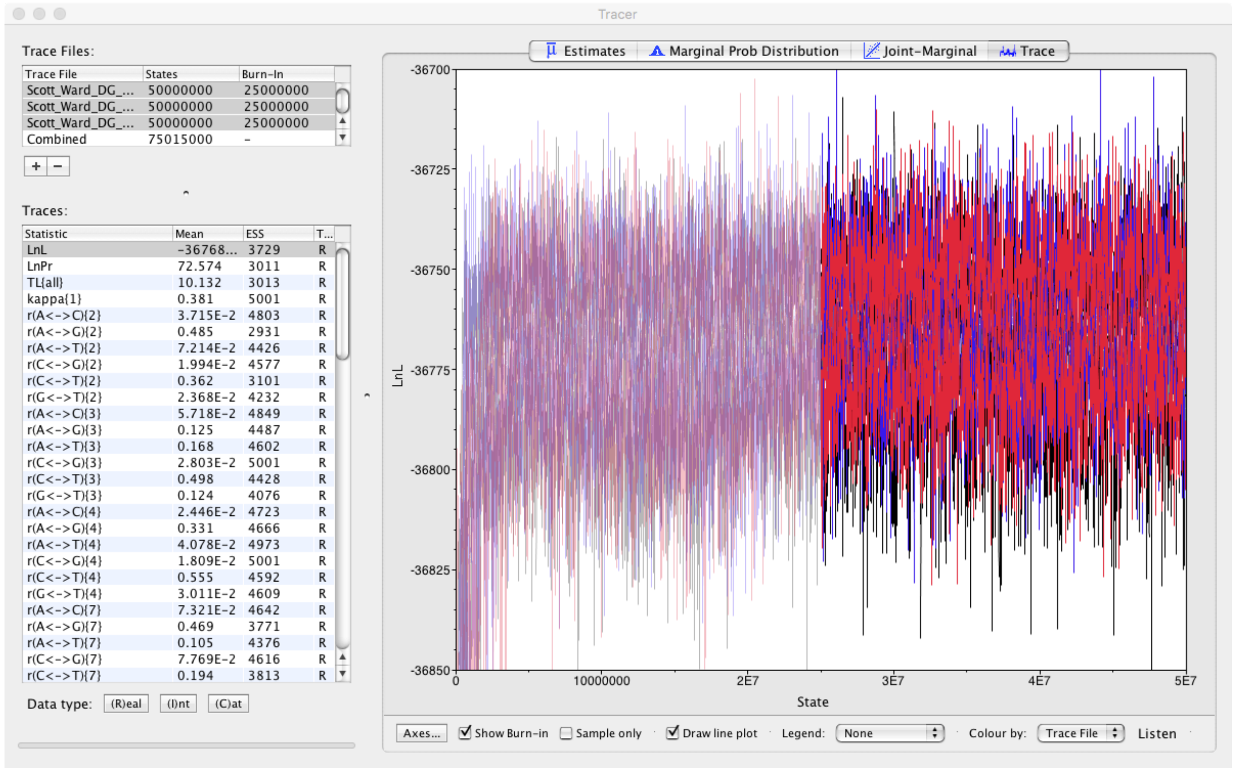
  Multiple Chain Convergence Diagnostics
  - Gelman & Rubin (1992) [PSRF in MrBayes]
  - Brooks & Gelman (1998) [*corrected scale reduction factor*]

  Individual Chain Convergence Diagnostics
  - Geweke Convergence Diagnostic (1992)
  - Heidelberger & Welch (1983)
  - Raftery & Lewis (1992)

Potential problems with ’standard’ MCMC convergence approaches

- Generally, people look at log-likelihood (LnL) to check convergence.
- LnL is calculated over the entire model
    - Some parameters can be particularly well estimated (i.e., tree length)
    - Others not so much (i.e., topology)

Average standard deviation of split frequencies = 0.023163
Maximum standard deviation of split frequencies = 0.252263
Average PSRF for parameter values = 1.000
Maximum PSRF for parameter values = 1.061



Trace Files:

Trace File	States	Burn-In
Scott_Ward_DG_...	50000000	25000000
Scott_Ward_DG_...	50000000	25000000
Scott_Ward_DG_...	50000000	25000000
Combined	75015000	-

Traces:

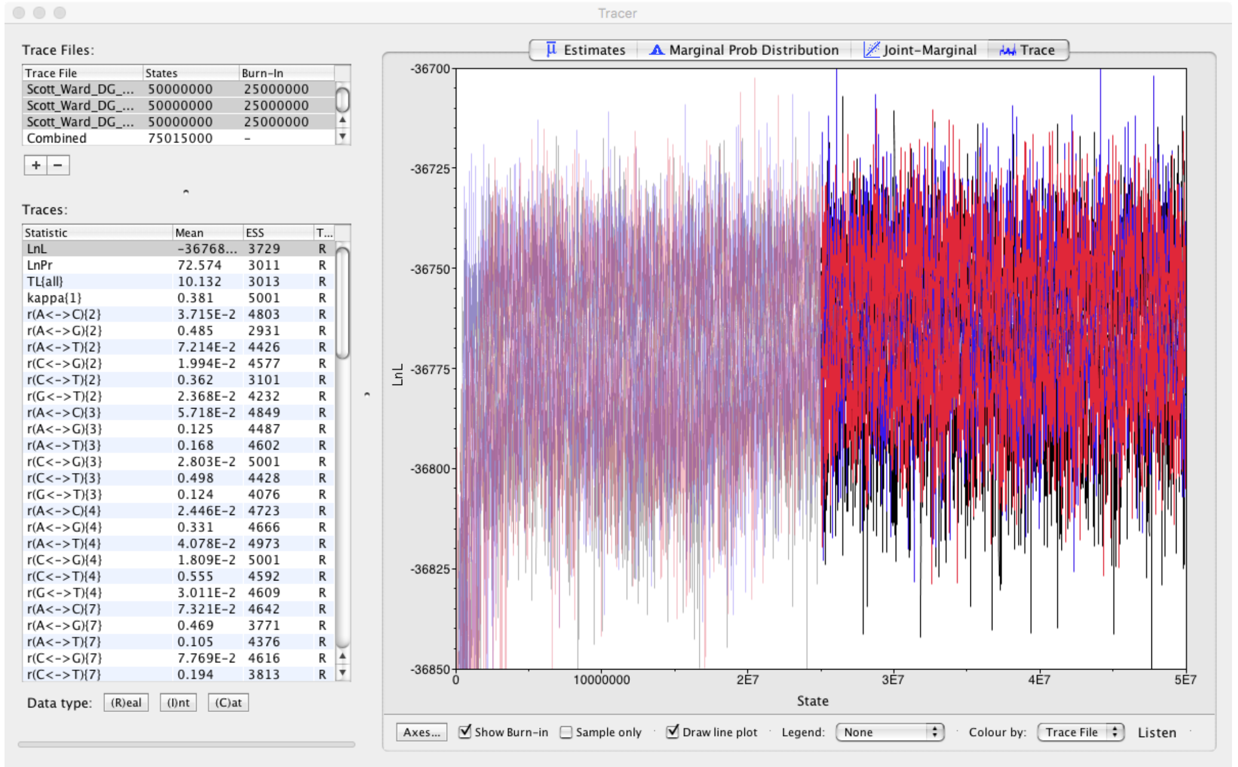
Statistic	Mean	ESS	T...
LnL	-36768...	3729	R
LnPr	72.574	3011	R
TL{all}	10.132	3013	R
kappa{1}	0.381	5001	R
r(A<->C){2}	3.715E-2	4803	R
r(A<->G){2}	0.485	2931	R
r(A<->T){2}	7.214E-2	4426	R
r(C<->G){2}	1.994E-2	4577	R
r(C<->T){2}	0.362	3101	R
r(G<->T){2}	2.368E-2	4232	R
r(A<->C){3}	5.718E-2	4849	R
r(A<->G){3}	0.125	4487	R
r(A<->T){3}	0.168	4602	R
r(C<->G){3}	2.803E-2	5001	R
r(C<->T){3}	0.498	4428	R
r(G<->T){3}	0.124	4076	R
r(A<->C){4}	2.446E-2	4723	R
r(A<->G){4}	0.331	4666	R
r(A<->T){4}	4.078E-2	4973	R
r(C<->G){4}	1.809E-2	5001	R
r(C<->T){4}	0.555	4592	R
r(G<->T){4}	3.011E-2	4609	R
r(A<->C){7}	7.321E-2	4642	R
r(A<->G){7}	0.469	3771	R
r(A<->T){7}	0.105	4376	R
r(C<->G){7}	7.769E-2	4616	R
r(C<->T){7}	0.194	3813	R

Potential problems with ’standard’ MCMC convergence approaches

- Generally, people look at log-likelihood (LnL) to check convergence.
- LnL is calculated over the entire model
  - Some parameters can be particularly well estimated (i.e., tree length)
  - Others not so much (i.e., topology)

Average standard deviation of split frequencies = 0.023163
Maximum standard deviation of split frequencies = 0.252263
Average PSRF for parameter values = 1.000
Maximum PSRF for parameter values = 1.061

an ASDSF < 0.01 is very good indication of convergence;
values between 0.01 and 0.05 may be adequate.
[MrBayes Manual]



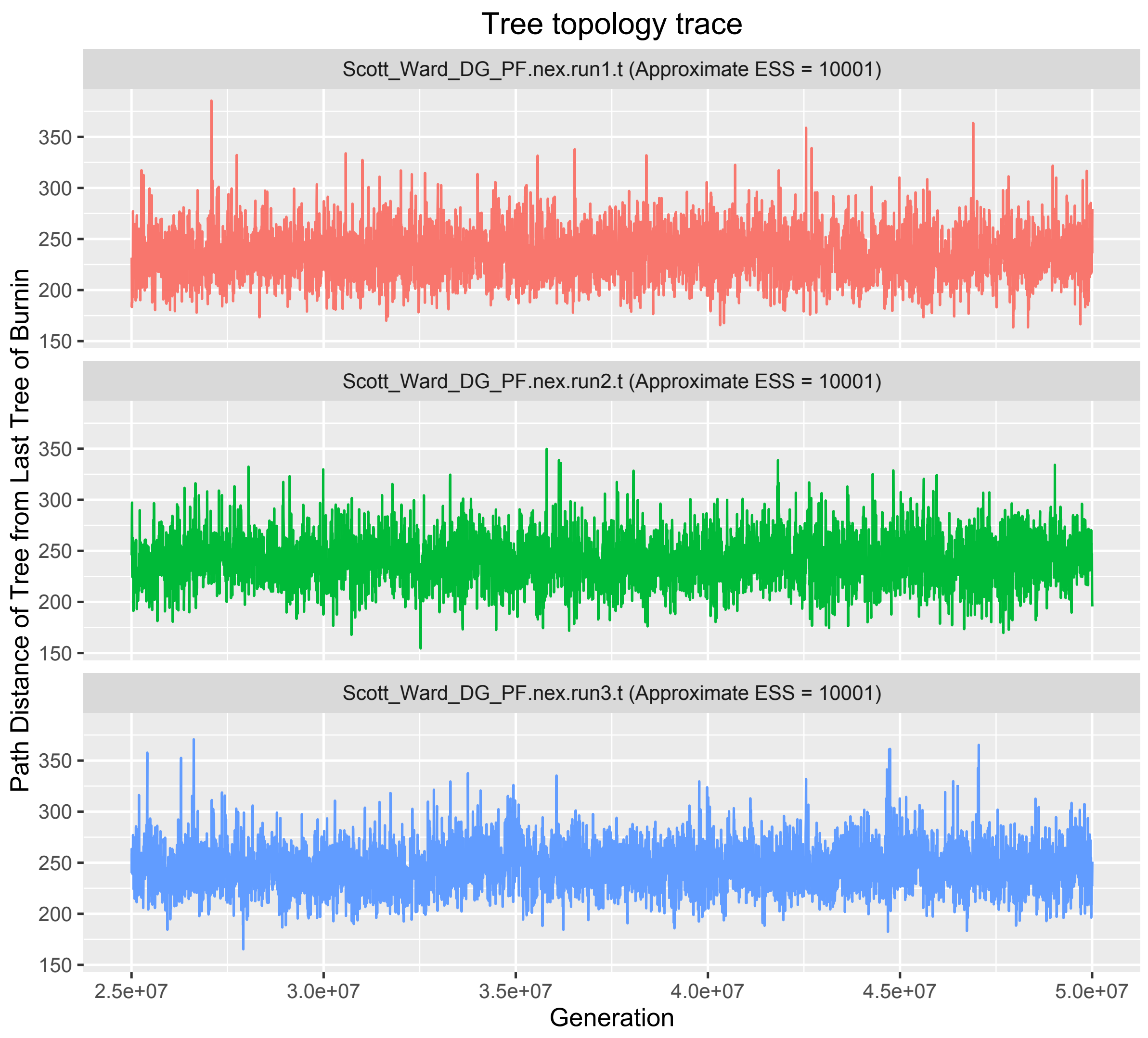
Trace Files:

Trace File	States	Burn-In
Scott_Ward_DG_...	50000000	25000000
Scott_Ward_DG_...	50000000	25000000
Scott_Ward_DG_...	50000000	25000000
Combined	75015000	-

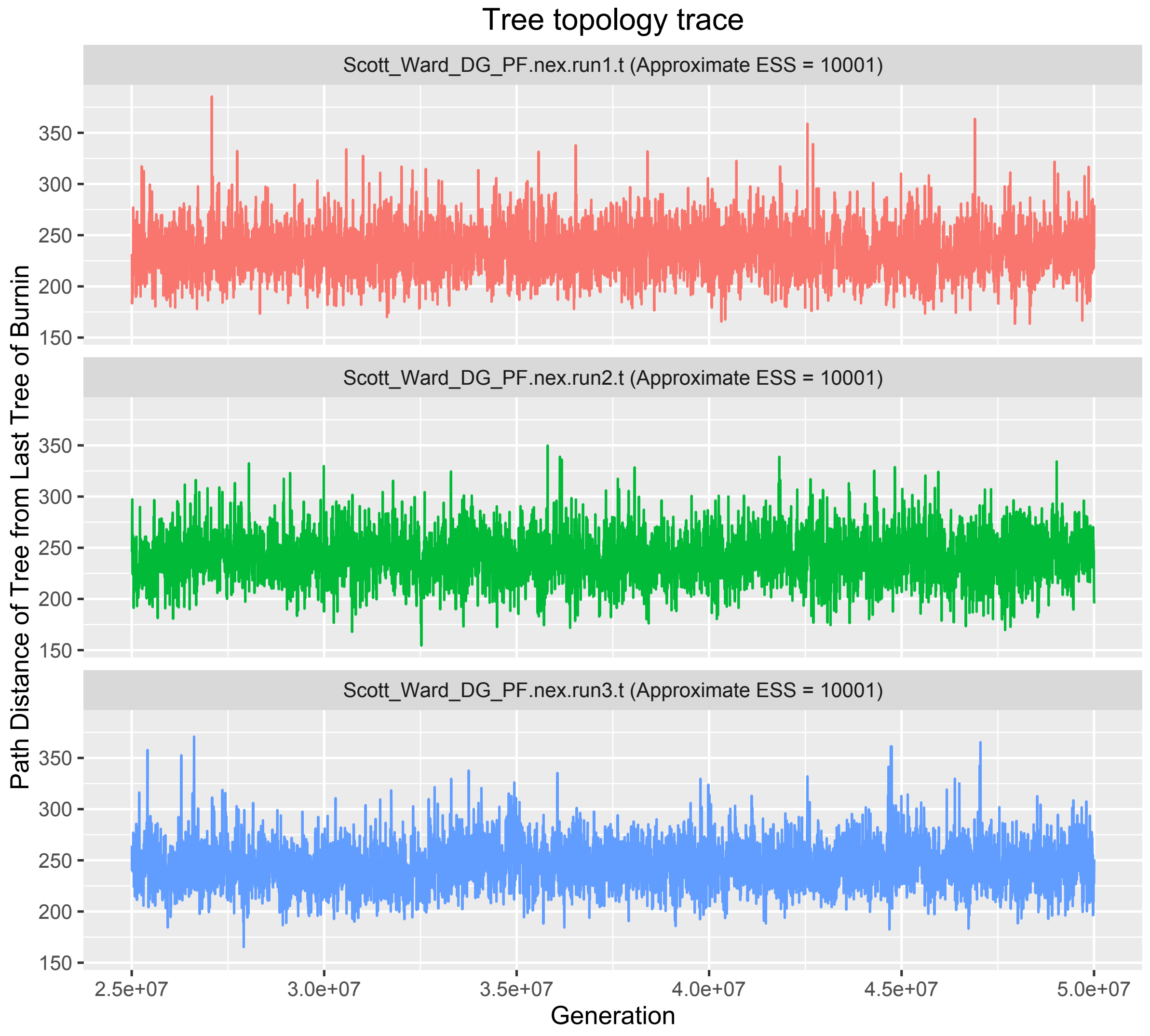
Traces:

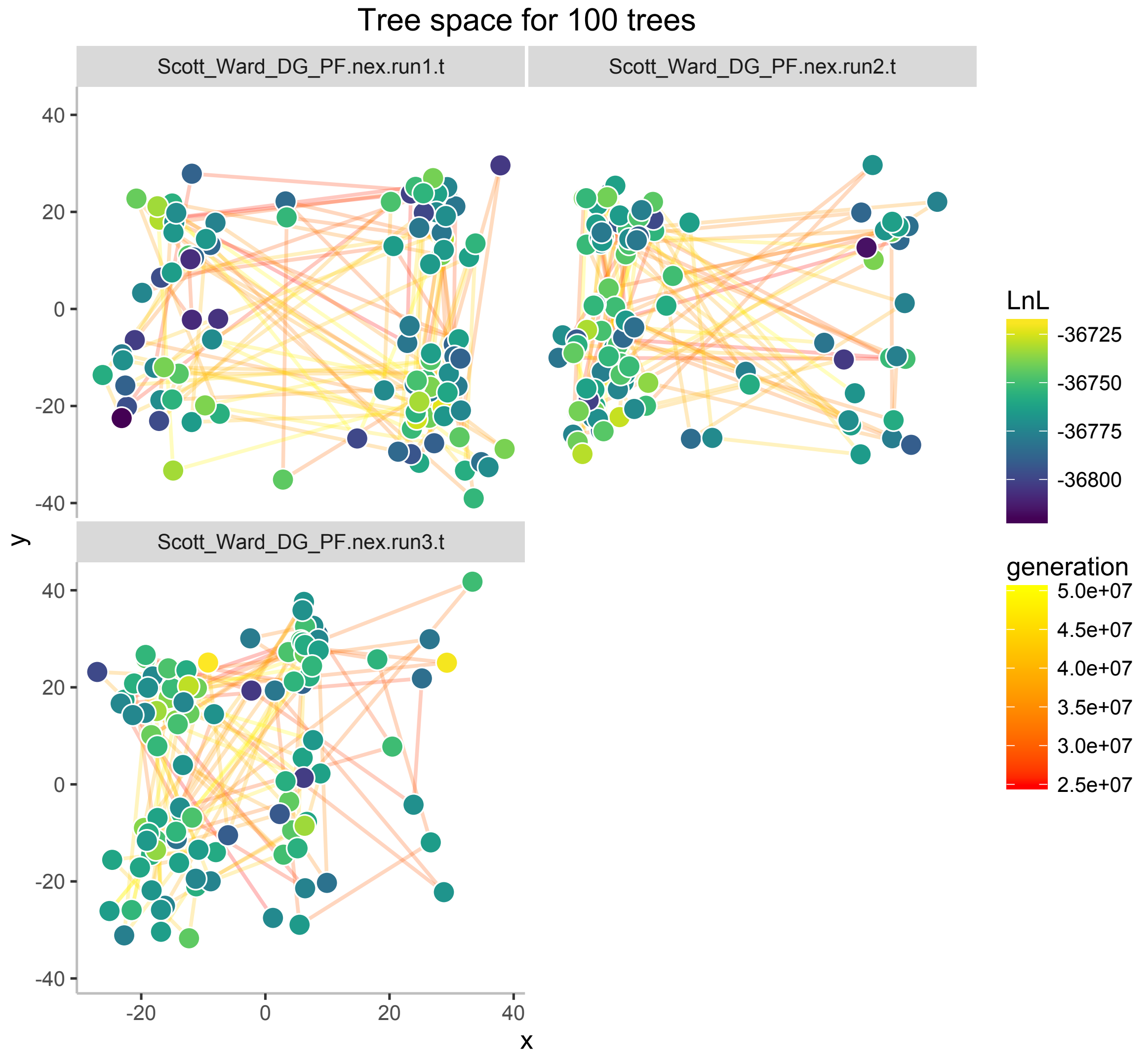
Statistic	Mean	ESS	T...
LnL	-36768...	3729	R
LnPr	72.574	3011	R
TL{all}	10.132	3013	R
kappa{1}	0.381	5001	R
r(A<->C){2}	3.715E-2	4803	R
r(A<->G){2}	0.485	2931	R
r(A<->T){2}	7.214E-2	4426	R
r(C<->G){2}	1.994E-2	4577	R
r(C<->T){2}	0.362	3101	R
r(G<->T){2}	2.368E-2	4232	R
r(A<->C){3}	5.718E-2	4849	R
r(A<->G){3}	0.125	4487	R
r(A<->T){3}	0.168	4602	R
r(C<->G){3}	2.803E-2	5001	R
r(C<->T){3}	0.498	4428	R
r(G<->T){3}	0.124	4076	R
r(A<->C){4}	2.446E-2	4723	R
r(A<->G){4}	0.331	4666	R
r(A<->T){4}	4.078E-2	4973	R
r(C<->G){4}	1.809E-2	5001	R
r(C<->T){4}	0.555	4592	R
r(G<->T){4}	3.011E-2	4609	R
r(A<->C){7}	7.321E-2	4642	R
r(A<->G){7}	0.469	3771	R
r(A<->T){7}	0.105	4376	R
r(C<->G){7}	7.769E-2	4616	R
r(C<->T){7}	0.194	3813	R

Potential problems with 'standard' MCMC convergence approaches



Potential problems with 'standard' MCMC convergence approaches





RWTY [https://github.com/danlwarren/RWTY]

RWTY

- Implements various tests, visualizations, and metrics for diagnosing convergence and mixing of MCMC chains in phylogenetics. It implements and automates many of the functions of the AWTY package in the R environment. It also adds a number of new functions not available in AWTY.
- In particular:
  - Topological Approximate ESS (topological.approx.ess(mb.rwty, burnin = 50)
  - Tree Topology Trace Plots (mb.rwty$topology.trace)
  - Topological Autocorrelation (mb.rwty$autocorr.plot)
  - Treespace Plots (mb.rwty$treespace.heatmap)
  - Some other cool stuff…

RWTY

- Requires *ape* & *ggplot2*
- Install phylogenetics packages:

```
install.packages('ctv')
library('ctv')
install.views('Phylogenetics')
```

- Install *RWTY:*

```
install.packages("rwty")
    or
install.packages("devtools")
library(devtools)
install_github("danlwarren/RWTY")
```

- Load *RWTY:*

```
library(rwty)
rwty.processors <<- 4          (N-1 is default)
```

Getting data into *RWTY*:
load.trees for single or *load.multi* for multiple chains

- Can load MrBayes, BEAST, *BEAST, RevBayes output (and force generic):

```
> mb.trees <- load.multi("path_to_FOLDER", format = "mb", gens.per.tree=1000,
trim = 10)

>  beast.trees <- load.trees("path_to_FOLDER", format = "beast",
gens.per.tree=5000, trim = 5, skip = 1000)

> other.trees <- load.trees("path_to_TREE_file", type = "nexus", logfile =
"path_to_LOG_file", skip = 1, gens.per.tree = 1000)

> mb.trees <- load.multi("/Users/dgotzek/Desktop/Smithsonian 2016/2016 R
workshop/mb_ants", format = "mb", gens.per.tree = 5000, trim = 50)
```

Doing stuff with *RWTY*:

- analyze.rwty()

  Does all analyses in one fell swoop.

  Bug in that only last plot is automatically output to screen.

  Always direct output to an object, so you can call up all plots!

- topological.approx.ess()

  Calculates the approximate ESS of tree topologies from all of the chains.

  The approximate ESS is similar to the ESS used for continuous parameters, but in certain cases an upper bound will be provided rather than a point estimate.

- makeplot.x()

  Only outputs desired plots.

Doing stuff with *RWTY*:

- analyze.rwty()

```
> mb.rwty <- analyze.rwty(mb.trees, burnin = 0, fill.color = 'LnL', filename =
"/Users/dgotzek/Desktop/Smithsonian 2016/2016 R
workshop/mb_ants/RWTYoutput_PD.pdf", treedist = "PD”)

> names(mb.rwty)

> mb.rwty
```

Doing stuff with *RWTY*:

- topological.approx.ess()

```
> approx.ess <- topological.approx.ess(mb.trees, burnin = 500)
> approx.ess

operator approx.ess                     chain
1    =       501 Scott_Ward_DG_PF.nex.run1.t
2    =       501 Scott_Ward_DG_PF.nex.run2.t
```

RWTY [https://github.com/danlwarren/RWTY]

Output:

- Plot of autocorrelation of tree topologies at different sampling intervals along a chain
- Plot of split frequencies calculated in sliding windows for the most variable clades
- Plot of change in split frequencies between sliding windows for all clades
- Plot of cumulative split frequencies as the MCMC progresses
- Plot of change in cumulative split frequencies as the MCMC progresses
- Heatmap and point depictions of chains in treespace.
- Plot of the Average Standard Deviation of Split Frequencies (ASDSF) between chains as the MCMC progresses
- Plot of pairwise correlations between split frequencies among chains
- Plot of chains clustered by their pairwise ASDSF values

Treescape [http://shiny.imperial-stats-experimental.co.uk/users/mlkendal/treescape/]

treescape

Treescape (a bare bones tour)

- Implements statistical tools for the exploration of sets of phylogenetic trees describing the evolutionary relationships between the same taxa.
- In particular:
  - 2- or 3-D plots to visualize trees using Principal Coordinates Analysis
  - Superior tree distance metrics (let’s get away from RF)

    e.g., Kendall Colijn (2016)

    Billera, Holmes, Vogtmann’s geodesic distance (2001)
  - Identifies clusters of trees and helps identify differences between them

Treescape [http://shiny.imperial-stats-experimental.co.uk/users/mlkendal/treescape/]

treescape

Treescape (a bare bones tour)

- Requires *ape*, *ade4*, further *adegenet*, *adegraphics*, *rgl* and XQuartz for 3D
- Install *treescape:*

```
install.packages(”treescape")
      or
install.packages("devtools")
library(devtools)
install_github("thibautjombart/treescape")
```

- Load *treescape:*

```
library(“treescape”)
library(“adegenet”)
library(“adegraphics”)
library(“rgl”)
```

Treescape [http://shiny.imperial-stats-experimental.co.uk/users/mlkendal/treescape/]

treescape

Getting data into *Treescape*

- Requires rooted trees!!
- Load trees using ape’s functions:

```
> trees <- read.nexus(file = "/Users/dgotzek/Desktop/Smithsonian 2016/2016 R
workshop/beast_ants/tree_model2.trees", tree.names = NULL)

# to save time and such, we will randomly subsample from the
> some.trees <- sample(trees,size=300)

# sanity check that trees are really rooted
> all(lapply(some.trees, is.rooted))
```

Treescape [http://shiny.imperial-stats-experimental.co.uk/users/mlkendal/treescape/]

treescape

Tree Analysis using *Treescape*

- treescape()
- The function treescape defines typologies of phylogenetic trees using a two-step approach, returning both the matrix of pairwise tree comparisons ($D), and the PCoA ($pco). :
  - performs pairwise comparisons of trees using various (Euclidean) metrics: Kendall-Colijn, Billera-Holmes-Vogtmann, Kuhner-Felsenstein, Robinson-Foulds, weighted Robinson-Foulds, Steel & Penny tip-tip path, Steel & Penny patristic tip-tip path difference metric, Abouheif.
  - uses Principal Coordinates Analysis (PCoA aka MDS) to project pairwise distances between trees into a few dimensions.
- > KC <- treescape(some.trees, method = "treeVec", nf=4, lambda=0)

treescape

Plotting treespace in *Treescape*

- plotGroves()
- Uses scatter() function of *adegraphics*
  - Can be used to gussy up plots but *ggplot2* is probably more versatile.
  - Can be given commands from function treescape() to directly plot treespace.

```
> plotGroves(KC$pco, method = "treeVec", nf=4, lambda=0)
```

```
> plotGroves(KC$pco)
> plotGrovesD3(KC$pco)
> plotGrovesD3(KC$pco, treeNames=1:300)
```

Treescape [http://shiny.imperial-stats-experimental.co.uk/users/mlkendal/treescape/]

treescape

Identifying clusters of trees in *Treescape*

- findGroves()

```
> groves <- findGroves(KC, nclust = NULL)
> groves.4 <- findGroves(KC, nclust = 4)
```

- Plot 3D:

```
> colours <- fac2col(groves.4$groups, col.pal=funky)
> plot3d(groves.4$treescape$pco$li[,1],
groves.4$treescape$pco$li[,2],
groves.4$treescape$pco$li[,3],
col=colours, type="s", size=1.5, xlab="", ylab="", zlab="")
```

Treescape [http://shiny.imperial-stats-experimental.co.uk/users/mlkendal/treescape/]

treescape

Finding median trees in *Treescape*

- medTree()

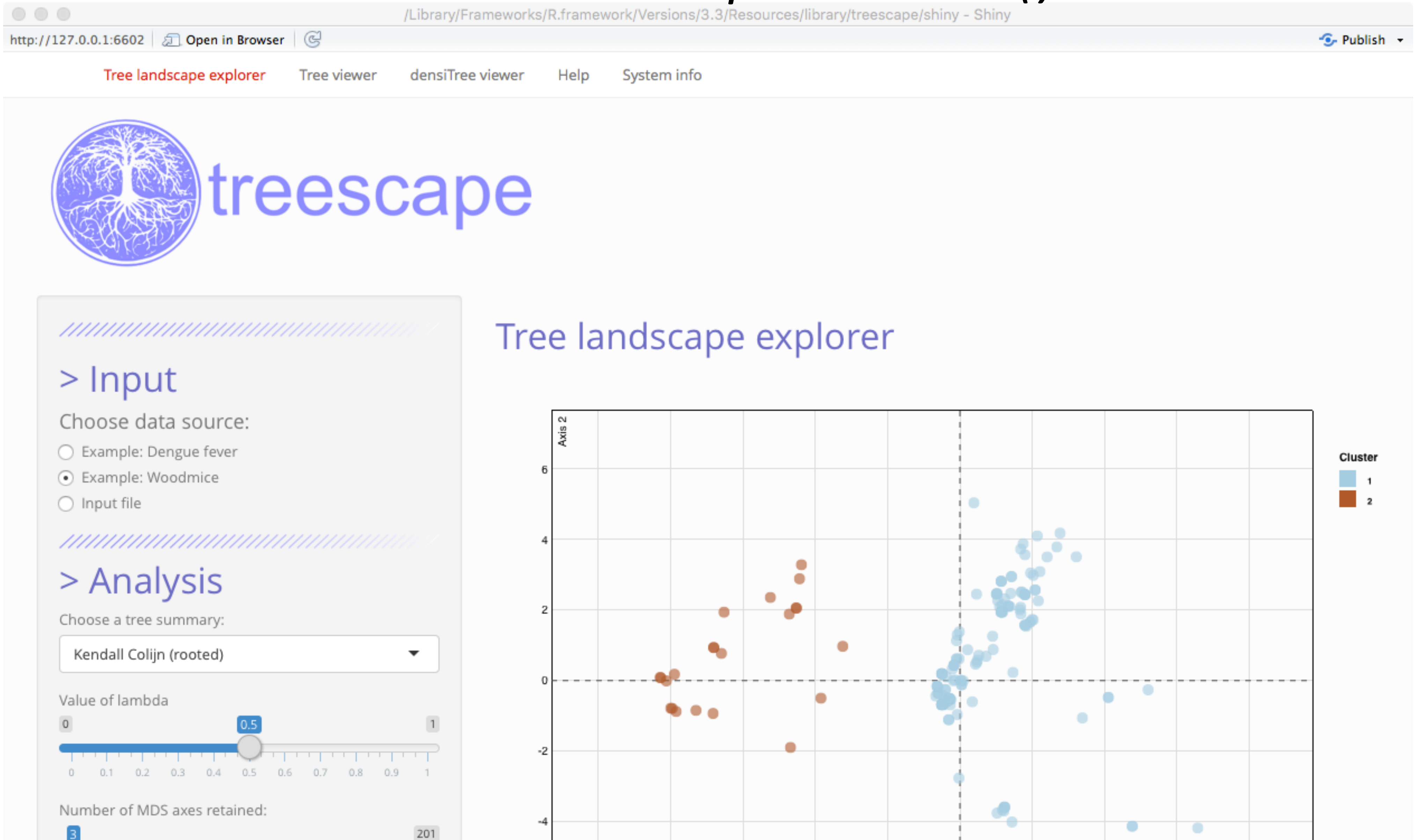
- finds the *median tree* for a set of trees according to the Kendall and Colijn metric (i.e., the tree which is closest to the center of the set of trees in the tree landscape defined in treescape).

```
> median.tre <- medTree(some.trees, groves.4$groups)
# sanity check:
> names(median.tre)
# get the first median tree of each cluster
> med.trees <- lapply(median.tre, function(e) ladderize(e$trees[[1]]))
# plot trees
> par(mfrow=c(2,2))
> for(i in 1:length(med.trees)) plot(med.trees[[i]], main=paste("cluster",i),cex=1.5)
# Compare median trees from clusters 1 and 2:
> plotTreeDiff(med.trees[[1]],med.trees[[2]], use.edge.length=FALSE)
```

Treescape [http://shiny.imperial-stats-experimental.co.uk/users/mlkendal/treescape/]

treescape

How to cheat: *treescapeServer()*


/Library/Frameworks/R.framework/Versions/3.3/Resources/library/treescape/shiny - Shiny

http://127.0.0.1:6602 Open in Browser Publish

Tree landscape explorer | Tree viewer | densiTree viewer | Help | System info

treescape

> Input

Choose data source:

- ○ Example: Dengue fever
- ● Example: Woodmice
- ○ Input file

> Analysis

Choose a tree summary: Kendall Colijn (rooted)

Value of lambda: 0.5

Number of MDS axes retained: 3

Tree landscape explorer

Some worthwhile reading…

- Whidden & Matsen (2015) Quantifying MCMC Exploration of Phylogenetic Tree Space. Syst Biol 64(3):472-491.
- Lanfear et al. (2016) Estimating the Effective sample Size of Tree Topologies from Bayesian Phylogenetic Analyses. GBE
- Kendall & Colijn (2015) A Tree Metric using Structure and Length to Capture distinct Phylogenetic Signals. arXIV 1507.05211v3
- Kendall & Colijn (2016) Mapping Phylogenetic Trees to Reveal Distinct Patterns of Evolution. MBE 33(10):2735-2743.