

TUTORIAL: RUNNING MARKOV CHAIN MONTE CARLO (MCMC) SIMULATIONS & ASSESSING OUTPUT

Tuesday 18th November 2014, University of Adelaide

Simon Tierney and Kate Sanders

AIM

For this tutorial we will set up an analysis to run in RevBayes and examine the performance of the MCMC using:

- Visual inspection the output of RevBayes using the operator summary and "Tracer"
 - (a) ESS - Effective Sample Size
 - (b) Diagnosis of trace plots
 - (c) Acceptance rates
- Visual inspection of the trees using "FigTree"
- Two post-hoc convergence diagnostics
 - (a) ASDSF - Average Standard Deviations of Spilt Frequencies
 - (b) PSRF - Potential Scale Reduction Factor

Keep an eye out for:

- Badly mixed chains
- Poor MCMC performance
- Correlated parameter estimates

DATA

13 species of beech tree (genus *Fagus*)

2576 nucleotide sites

3 gene regions:

- 1 nuclear gene region (ITS)
- 2 chloroplast gene regions (rbcL&matK)

MODEL - GTR+I+G

Model type:

GTR (General Time Reversible Model)

Across-site rate variation:

- I (Proportion of invariant sites)
- G (Gamma distribution)

Partitioning scheme:

Uniform scheme, shared evenly across all sites of the matrix

PART 1: CONSTRUCTION OF REVBAYES SCRIPT

Refer to instructions in the script file “simon_mcmc.txt” to set up the analysis to run in RevBayes

This should result in three output files:

```
simon_mcmc_posterior_run1.log
simon_mcmc_posterior_run1.trees
simon_mcmc_run1.tree
simon_r1_Consensus.tre
```

PART 2: ASSESS & OPTIMISE THE PERFORMANCE OF THE MCMC SET UP IN PART 1

After running the analysis you set up in Part 1, your final output to the screen will be an overview of the moves that you have used for this MCMC run using the ‘mymcmc.operatorSummary()’

```
mymcmc.operatorSummary()
```

This operator summary tells you what the weight for each move was, the variables it has been working on, the number of times the move was used, and how often the move was accepted. Good acceptance rates for continuous parameters are between 20% and 60%. For discrete characters, such as the phylogeny, there is no rule of thumb what good acceptance is. Instead, the acceptance rate of tree topology proposal strongly depends on the current data.

[SCREEN OUTPUT EXAMPLE*]

Name	Param	Weight	Tried	Accepted	Acc. Ratio	Parameters
Sliding	pinvar	1.0000	347	12	0.0346	delta = 10.0000
NNI	topology	1.0000	367	73	0.1989	-

*NB Analyses should start from independent points in parameter space, so you should expect to get slightly different results from the example above

2.1: SAVE RESULTS AS AN EXCEL FILE

Copy the operator summary from screen to **TextWrangler** or **MS-Word** and then paste into or create a tab-delimited **MS-Excel** file. You might see that acceptance rates of several parameters are very low – they are not updated often enough.

2.2: EXAMINE RESULTS IN TRACER

Now open the generated output file in **Tracer**.

NB***You will need to open your RevBayes log output in Excel first, SAVE AS a tab-delimited file in order for Tracer to read the file (temporary problem).

The file should be called 'simon_mcmc_run1.log'

If you are unfamiliar with Tracer, a tutorial is available at:

https://web.archive.org/web/20130125173544/http://beast.bio.ed.ac.uk/Analysing_BEAST_output

What you may notice is that the effective sample sizes (ESS) are very low and there are large scale fluctuations in the trace plots suggesting that the MCMC is mixing poorly.

2.3: SAVE YOUR SCRIPT AS A BATCH FILE

Now that your script is working you can make a short cut and run this analysis as a batch file. To do this replace the file extension '.txt' with '.Rev' and execute the source function, after you have launched RevBayes:

```
./rb  
source("simon_mcmc.Rev")
```

This will save you from re-entering your script line-by-line, every time you wish to tweak the model. Make sure any commentary in your script is preceded by a hash-tag. For example you do not need to re-launch RevBayes from within your script, block it from being read in batch mode by editing your script as such:

```
# Launch RevBayes  
# ./rb
```

2.4: CHANGE & ADD MOVES

Use the str() function to check the RevType of parameters with low acceptance rates. Refer to the help file "[RevBayes_moves.pdf]" to check which other types of moves are appropriate for these variables. Edit the source file to try different types of moves for the parameters that aren't mixing well. Remember you can apply more than one type of move to a parameter simultaneously (as in the examples below).

Examples:

- The variable pi (base frequencies) is of RevType 'Simplex', which scales from 0 to 1, so the appropriate moves are "mvSimplex" and "mvSimplexElementScale":

```
pi_prior<- v(1,1,1,1)  
pi ~ dnDirichlet(pi_prior)  
moves[mi++] = mvSimplexElementScale(pi, alpha=0.1, tune=false, weight=1.0)  
moves[mi++] = mvSimplex (pi, alpha=0.1, tune=false, weight=1.0)
```

- Pinvar (probability of invariant sites) is of RevType 'Probability', so both "mvSlide" and "mvScale" moves can be used:

```
pinvar ~ beta(1,1)  
moves[mi++] = mvScale(pinvar, lambda=10.0, tune=false, weight=1.0)
```

```
moves[mi++] = mvSlide(pinvar, delta=10.0, tune=false, weight=1.0)
```

- The current MCMC only has one tree topology move. Add the “mvSPR” tree move (Subtree Pruning & Regrafting) and see if this improves the outcome:

```
topology ~ dnUniformTopology(n_species, names)
moves[mi++] = mvNNI(topology, weight=4.0)
moves[mi++] = mvSPR(topology, weight=4.0)
```

Save your changes to a new script file (e.g. “kate_mcmc_run2.Rev”) and change the output file names in monitors [1] & [2], e.g:

```
monitor[1] = mnModel(filename="kate_mcmc_posterior_run2.log", printgen=10,
separator = " ")
monitor[2] = mnFile(filename=" kate_mcmc_posterior_run2.trees", printgen=10,
separator = " ", phylogeny)
```

Now re-run the analysis in RevBayes as a batch file and view the operator summary using a handle that identifies this as the second run:

```
>source("kate_mcmc_run2.Rev")
```

Running in batch mode will automatically print-to-screen the operator summary for you. Look how often the different types of moves are accepted. Also look at the output in Tracer to see the effect on ESS values and mixing of the chain.

2.5: ADJUST TUNING PARAMETERS ON MOVES

If acceptance rates are still low it is likely that the moves you used proposed too many bad new parameters - the MCMC proposals might have been too timid or too bold. So try changing the tuning parameters on the proposal distributions for the variables that aren’t mixing well.

Examples:

- A sliding window proposal is used for Pinvar:

```
pinvar ~ dnBeta(1,1)
moves[mi++] = mvSlide(pinvar, delta=0.1, tune=false, weight=1.0)
```

The tuning parameter for this distribution is delta – you can make larger proposals by increasing delta, and more modest proposals by decreasing delta: try changing delta from 0.1 to 10 and check the operator summary to see if acceptance rates have improved.

- A simplex scale move is used for the parameter pi:

```
pi_prior<- v(1,1,1,1)
pi ~ dnDirichlet(pi_prior)
moves[mi++] = mvSimplexElementScale(pi, alpha=0.1, tune=false, weight=1.0)
```

The tuning parameter for this distribution is alpha – you can make bolder proposals by increasing alpha.

- A scale move is used for the branch length vector:

```
br_lens[i] ~ dnExponential(10.0)
moves[mi++] = mvScale(br_lens[i], lambda=1, tune=true, weight=1)
```

The tuning parameter for this distribution is lambda – you can make bolder proposals by increasing lambda.

Again, save your changes to a new file remembering to change the output file names in the monitors so these aren't over-written. Then re-run the analysis in RevBayes and check acceptance rates in the operator summary. Also open the outputs in Tracer to check ESS values and mixing of the MCMC.

2.6: AUTO-TUNING

It's important to understand mcmc moves, but it can be time consuming to adjust tuning parameters for individual proposal distributions. RevBayes allows **auto-tuning** of the moves. Just change "tune=false" to "tune=true" for all the moves in the .Rev file.

Example:

```
moves[mi++] = mvSlide(pinvar, delta=0.1, tune=true, weight=1.0)
```

This updates the tuning parameters so that you achieve a good acceptance rate. That means if previously a move accepted too few proposals it will make smaller proposals. Look at the output given by the operator summary. Notice how the tuning parameters have been updated. Also look at the output in Tracer. You should see that the ESS values are much better because the chain is mixing better.

2.7: ADJUST PROPOSAL WEIGHTS ON MOVES

Another way to improve ESS values for parameters that are not mixing well is to increase the proposal weights for those parameters so that moves are used more often. By default, a single generation in RevBayes updates all stochastic nodes in proportion to the weight argument.

Example:

By setting **weight=4** for the topology parameter we can ensure that new moves are made on the tree four times per iteration:

```
# moves on the tree
```

```
moves[mi++] = mvNNI(topology, weight=4.0)
```

Try changing the proposal weights for “difficult” parameters and check MCMC performance as before.

The ‘moveschedule’ is currently set to single in ‘mymcmc’. Change the moveschedule to random:

```
mymcmc = mcmc(mymodel, monitor, moves, moveschedule="random")
```

...so the MCMC simulator will execute moves for all parameters at random. Alternatively, you can try setting the moveschedule to ‘sequential’ so that moves are made in the order that parameters are given in the source file:

```
mymcmc = mcmc(mymodel, monitor, moves, moveschedule="sequential")
```

Think about how you might order moves to improve MCMC performance.

2.8: RUN MCMC FOR EXTENDED GENERATION TIME & EXAMINE BETWEEN-RUN DIAGNOSTICS

By now you should have an MCMC analysis that is mixing fairly well, with auto-tuning for all parameters and multiple moves that are executed at random each iteration. However it is essential that independent runs are performed for every analysis to properly check convergence.

So now run two independent runs for 500,000 generations and we will then examine between run convergence diagnostics.

Recommend running four independent runs and view together in Tracer. Compare various model parameters means & 95% Higher Posterior Distributions (HPD’s).

Potential Scale Reduction Factor (PSRF) & Average Standard Deviation of Split Frequencies are not yet live on RevBayes.

Mike Lee to give spontaneously heroic tutorial on AWTY!

http://king2.scs.fsu.edu/CEBProjects/awty/awty_start.php

Use FigTree to view trees – already in your list of Applications.

Enjoy.