Testing DENIM

Graham Jones

2017-06-18, August 17, 2017

Overall plan

Most of this is about testing DENIM on the simulated data of Leaché et al. (2014), supplied by Adam Leaché as MCcoal control files. The basic plan is

1. Re-organise the MCcoal control files. They were named and arranged in directories with different conventions.
2. Run MCcoal on all the control files, to make sequence files. The control files have different file names inside the control files, so rename the output to standard names.
3. For each 'basic pattern' (same species, same alleles, there are 1+4+1+6=12 such) make ten NEX files (one per locus) from a sequence file.
4. Load the sets of NEX files into Beauti, and set models and priors, to make a BEAST XML per basic pattern.
5. Manually divide these into pre-data, data, post-data sections, then make BEAST XMLs for all replicates
6. Run BEAST on the XML files.
7. Analyse the results, make tables and plots.

In order to change the prior on the migration rate, or make other small changes to the BEAST XMLs I usually edit the post-data XML sections (up to 7 parallel edits). Then rerun code to make ones for all replicates.

Code organisation

The main R scripts are in `DENIMSimulations`.

`Prior`. `make-xml-dirs.R` just makes some directories. `lookat-prior-logfiles.R` is for looking at results.

`Leache-FourSpp`. Re-organising the MCcoal control files, running them with `MyMCcoal.exe` to make files for sequences, migration events, etc, using `fourspp-run-MCcoal-do-time-zero-swaps.R`. Making BEAST XML files using `fourspp-mccoal-data-tobeastxml.R`.

`Leache-TenSpp`. Similar to above (using `tenspp-run-MCcoal-do-time-zero-swaps.R` and `tenspp-mccoal-data-tobeastxml.R`), but extra step using `make-0p01-ctrlfiles.R` to make more MCcoal control files first.

`Analysis`. `lookat-coverage-cred-trees.R` for 95% credibility of topologies. `lookat-ESSs.R` for ESS. `lookat-migrations.R` runs `MigrationAnalyser` to make a summary of migrations. `lookat-migs-yes-no.R` looks

at estimates vs truth for loci with/without migrations. `lookat-tree-distances.R` for branch scores. `runTreeAnnotator.R` runs `TreeAnnotator` to make maximum clade credibility trees.

`Utils`. These are R scripts `source`'d in by others especially ones in `Analysis` directory. `make-nex-empty-seqs.R` makes empty NEX files for prior tests. `mccoal-data-to-nex.R` converts MCcoal sequences to NEX for Beauti. `mccoal-seqs-to-xml.R` converts MCcoal sequences to BEAST XML fragment for making whole BEAST XML files. `mcoal-tree-to-Newick.R` parses a MCcoal control file and converts the species tree to Newick. `scenario-dir-names.R` has lists of various directories for different subsets of scenarios.

`Utils\MCcoal-src`. This is the C source for `MCcoal`, with my minor edits. The edits are in `bpp.c` and can be found by searching on 'GRJ'. They make `MCcoal` produce a files `migs.txt` containing information about individual migration events.

1 Testing on prior only

To make XML files for testing prior. Details depend on DENIM template.

1. `make-nex-empty-seqs.R` makes nex files with empty "?" sequences. Eg s6i3locus5.nex has 6 species, 3 individuals per species, and is the fifth locus like this.
2. Run Beauti2. Choose DENIMtemplate.
3. Load (eg) 6i3locus1.nex, ... 6i3locus9.nex
4. Link all clocks (or put a prior on all but first.)
5. Coalescence+Migration: Don't estimate Pop Prior Scale. Set to (eg) 0.0005, 0.005, or 0.05
6. Coalescence+Migration: Set GammaComponent1 as needed
7. Priors: Don't estimate Birth rate, leave at 100
8. Priors: Don't estimate Relative Death Rate, set to 0.0. Don't estimate Collapse Weight, set to 0.0
9. Priors: Set priors on Relative Death Rate and Collapse Weight: offset Beta to -0.5 for both.

Need to use Beauti2 for each basic configuration (number of species, individuals, loci). So (2,6) spp x (3,9) loci x (1,3,9) individuals is 12 times through Beauti.

Then copy and edit XML (in text editor or Beauti) to make different ones to run for different Pop Prior Scale.

2 Running BEAST on data made by MCcoal

Notes on 4-species data

Adam Leaché sent MCcoal control files in 4 directories for different migration rates.

```
M0.001
M0.01
M0.1
M1
```

Inside there are 400 control files, named

```
AB0.ctl - AB99.ctl   (I renamed as A-B.)
Anc0.ctl - Anc99.ctl  (I renamed as AB-C.)
BC0.ctl - BC0.ctl     (I renamed as B-C.)
coal99.ctl - coal99.ctl  (I renamed as NoMig.)
```

The `coal` ones are no migration, so these are the same settings repeated 4x, though they are different replicates.

I used `fourspp-run-MCcoal-do-time-zero-swaps.R` to copy the ctrl files to a removable disc, putting each replicate in its own directory, and renaming so that for example `M0.001/AB7.ctl` becoomes `A-B-0p001/rep008/ctrl.txt`. (My reps run 1 to 100, not 0 to 99.) Then the control files were run using a version of `MCcoal.exe` which I edited to output information about each migration, not just a summary. The `MCcoal` output files were renamed. The result is directories containing

```
ctrl.txt   Imap.txt   migs.txt   seq.txt   trees.txt
```

for each replicate for the 13 scenarios above.

There are then 4 time-zero swaps which start with the NoMig (no migration) scenario and rename appropriate sequences.

From here on, describe 4 and 10 spp cases together (below).

Notes on 10-species data

Adam Leaché sent `May-2013.tar.gz`, unpacked as `10_species\May-2013` and later `10_species.zip`, unpacked as `10_species\10_species`

In `May-2013` these six directories with 100 control files each:

```
migC3_0.1ctl   migC3_1.0ctl      (I renamed as ABCD-EFGH.)
migefgh0.1ctl  migefgh1.0ctl     (I renamed as E-F_E-G_E-H_F-G_F-H_G-H. This is "n-island")
migHI0.1ctl    migHI1.0ctl       (I renamed as H-I.)
nomigctl
```

In `May-2013\re10speciessimulations` these directories with 100 control files each:

```
migA2_1.0ctl   migA20.1ctl    (I renamed as E-F. Note typo in filename.)
migB2_0.1ctl   migB2_1.0ctl   (I renamed as F-G.)
migC2_0.1ctl   migC2_1.0ctl   (I renamed as G-EF.)
```

I copied the control files into new directories with my naming, putting the migrating branch names into the file names.

I think there are no new ctrl files in `10_species`, but there are `XML` examples.

```
introgressionFG is all XML files, presumably for migA2_1.0ctl or migA20.1ctl.
introgressionDE_run3.xml
nomig_run3.xml
singmigDE_run3.xml
```

I then used this R array to do re-organising and renaming.

```
      leache.to.repnum                                    leaf.in.ctrl      my.dir
 [1,] "ABCD-EFGH_0p1/run"                                 "run"             "ABCD-EFGH_0p1"
 [2,] "ABCD-EFGH_1p0/run"                                 "run"             "ABCD-EFGH_1p0"
 [3,] "E-F_0p1/run"                                       "run"             "E-F_0p1"
 [4,] "E-F_1p0/run"                                       "run"             "E-F_1p0"
 [5,] "E-F_E-G_E-H_F-G_F-H_G-H_0p1/migEFGH_0.1_run" "migEFGH_0.1_run" "E-F_E-G_E-H_F-G_F-H_G-H_0p1"
 [6,] "E-F_E-G_E-H_F-G_F-H_G-H_1p0/migEFGH_1_run"   "migEFGH_1_run"   "E-F_E-G_E-H_F-G_F-H_G-H_1p0"
 [7,] "EF-G_0p1/run"                                      "run"             "EF-G_0p1"
 [8,] "EF-G_1p0/run"                                      "run"             "EF-G_1p0"
 [9,] "F-G_0p1/run"                                       "run"             "F-G_0p1"
[10,] "F-G_1p0/run"                                       "run"             "F-G_1p0"
[11,] "H-I_0p1/migHI_0.1_run"                             "migHI_0.1_run"   "H-I_0p1"
[12,] "H-I_1p0/migHI_1_run"                               "migHI_1_run"     "H-I_1p0"
[13,] "COAL/nomig_run"                                    "nomig_run"       "COAL"
```

`leache.to.repnum` means the directory and leaf name as far as the replication number 0-99. `leaf.in.ctrl` is the name used inside the control files for sequence and tree files output by MCcoal, which use different conventions. A further wrinkle is in `migefgh0.1ctl`, the tree file was `migEFGH_11_run0.tre` not `migEFGH_1_run0.tre`. I edited the control files in this case.

Then I augmented the control files by adding ones with a migration rate of 0.01 but otherwise identical to the ones with a migration rate of 0.1. These were made by `make-0p01-ctrlfiles.R` by copying, and making substitutions in, the '0.1' control files. This adds 6 more scenarios to get:

```
      leache.to.repnum                                    leaf.in.ctrl      my.dir
 [1,] "ABCD-EFGH_0p01/run"                                "run"             "ABCD-EFGH_0p01"
 [2,] "ABCD-EFGH_0p1/run"                                 "run"             "ABCD-EFGH_0p1"
 [3,] "ABCD-EFGH_1p0/run"                                 "run"             "ABCD-EFGH_1p0"
 [4,] "E-F_0p01/run"                                      "run"             "E-F_0p01"
 [5,] "E-F_0p1/run"                                       "run"             "E-F_0p1"
 [6,] "E-F_1p0/run"                                       "run"             "E-F_1p0"
 [7,] "E-F_E-G_E-H_F-G_F-H_G-H_0p01/run"                  "migEFGH_0.1_run" "E-F_E-G_E-H_F-G_F-H_G-H_0p01"
 [8,] "E-F_E-G_E-H_F-G_F-H_G-H_0p1/migEFGH_0.1_run" "migEFGH_0.1_run" "E-F_E-G_E-H_F-G_F-H_G-H_0p1"
 [9,] "E-F_E-G_E-H_F-G_F-H_G-H_1p0/migEFGH_1_run"   "migEFGH_1_run"   "E-F_E-G_E-H_F-G_F-H_G-H_1p0"
[10,] "EF-G_0p01/run"                                     "run"             "EF-G_0p01"
[11,] "EF-G_0p1/run"                                      "run"             "EF-G_0p1"
[12,] "EF-G_1p0/run"                                      "run"             "EF-G_1p0"
[13,] "F-G_0p01/run"                                      "run"             "F-G_0p01"
[14,] "F-G_0p1/run"                                       "run"             "F-G_0p1"
[15,] "F-G_1p0/run"                                       "run"             "F-G_1p0"
[16,] "H-I_0p01/run"                                      "migHI_0.1_run"   "H-I_0p01"
[17,] "H-I_0p1/migHI_0.1_run"                             "migHI_0.1_run"   "H-I_0p1"
[18,] "H-I_1p0/migHI_1_run"                               "migHI_1_run"     "H-I_1p0"
[19,] "COAL/nomig_run"                                    "nomig_run"       "COAL"
```

There are two cases where file names inside the control files contain 0.1 and disagree with the 0.01 migration rate. This discrepancy disappears when MCcoal output is renamed later.

I then used `fourspp-run-MCcoal-do-time-zero-swaps.R` or `tenspp-run-MCcoal-do-time-zero-swaps.R` to copy the ctrl files to a removable disc, putting each replicate in its own directory, and renaming so that for example

ABCD-EFGH_0p1/run0.ctl becomes ABCD-EFGH_0p1/rep001/ctrl.txt. Then the control files were run using a version of MCcoal.exe which I edited to output information about each migration, not just a summary. The MCcoal output files were renamed. There are then 6 time-zero swaps which start with the COAL (no migration) scenario and rename appropriate sequences.

The end result is 25 directories

```
 [1] "ABCD-EFGH_0p01"                [2] "ABCD-EFGH_0p1"                 [3] "ABCD-EFGH_1p0"
 [4] "E-F_0p01"                      [5] "E-F_0p1"                       [6] "E-F_1p0"
 [7] "E-F_E-G_E-H_F-G_F-H_G-H_0p01" [8] "E-F_E-G_E-H_F-G_F-H_G-H_0p1"  [9] "E-F_E-G_E-H_F-G_F-H_G-H_1p0"
[10] "EF-G_0p01"                    [11] "EF-G_0p1"                    [12] "EF-G_1p0"
[13] "F-G_0p01"                     [14] "F-G_0p1"                     [15] "F-G_1p0"
[16] "H-I_0p01"                     [17] "H-I_0p1"                     [18] "H-I_1p0"
[19] "COAL"
[20] "D-E-1-allele"                 [21] "D-E-1-mig"
[22] "F-E-1-allele"                 [23] "F-E-1-mig"
[24] "F-G-1-allele"                 [25] "F-G-1-mig"
```

each containing

```
ctrl.txt   Imap.txt   migs.txt   seq.txt   trees.txt
```

where ctrl.txt is the control file, which contains the true species tree, Imap.txt is species names (not used), migs.txt contains the individual migrations that occured, seq.txt has the sequences for conversion into BEAST XMLs, and trees.txt contains the gene trees.

Example of a migrations file (E-F_E-G_E-H_F-G_F-H_G-H_0p01\rep001\migs.txt).

```
locus 1 F to H at 0.000457   E to G at 0.003920   F to E at 0.004002
locus 2 H to F at 0.000694   E to H at 0.007054   G to E at 0.018058
locus 3 F to H at 0.003945   G to E at 0.008476   H to E at 0.010433
locus 4
locus 5 F to H at 0.003343   F to G at 0.017159
locus 6 G to H at 0.003306   G to H at 0.003580   H to E at 0.015789
locus 7
locus 8 G to E at 0.005770   F to H at 0.012362
locus 9 F to H at 0.010221
locus 10 G to H at 0.017963
```

Making BEAST XML files

This is very similar for the 4 and 10 species data. There are a total of 1+4 (for 4spp) and 1+6 (for 10 spp) patterns of taxa and loci. So need to do the following 12 times.

For each taxa-loci pattern, a seq.txt was converted to NEX using mccoal-data-to-nex.R. This makes 10 NEX, files, one per locus, which can be loaded into Beauti.

1. Load DENIM template
2. Add the 10 alignments
3. link clock models, link site models

4. Use Guess, Everything after `^`. Species are A,B,C,D or A-J.
5. Site model is GTR, frequencies fixed at 0.25 0.25 0.25 0.25
6. Clock model stays as Strict
7. Ploidy's stay at 2
8. Coalescence+Migration: GammaComponent.1, RelatednessFactor, Migration Delay Scale edited. These are for experimentation.
9. Priors: In Tree.t:Species, set Collapse Weight to zero, don't estimate. Put uniform [-0.5,0.5] prior on it.
10. Priors: In Tree.t:Species, set Death Rate to zero, don't estimate. Put uniform [-0.5,0.5] prior on it (Yule).
11. Priors: lognorm(5,2) for bdcGrowthRate.Species (Leache has 1OnX)
12. Priors: lognorm(-5,2) for popPriorScale. Leache used InvGamma(3, 0.03) for *BEAST's species.popMean. *BEAST then uses a gamma distribution with mean 2 at the rootward end of branches and 4 at the tipward ends. DENIM uses a InvGamma(3, 2) per branch. It would be possible to match the model used by Leache for the simulation in MCcoal, but not to match what they did with *BEAST.
13. Priors: GTR: ac ag at cg gt have gamma(0.5,10) priors

The `BEAST XML` file is then split manually into a before-data part, a (discarded) data part, and an after-data part. Then `tenspp-mccoal-data-tobeastxml.R` (or similar for 4spp) can make a new XML data section for each replicate and join them together. The number of replicates is from an R `tenspp-scenario-dir-names.R` (or similar for 4spp).

Running the BEAST XML files

This is simple. I used batch files to run BEAST 3x in parallel. I used another computer for this.

3 Analysing the results

Prior only

`lookat-prior-logfiles.R` is for looking at results. Species tree height, number of migrations, ESSs, for different number of species, individuals, amounts of ILS.

Leaché data

The same code is used for 4-species and 10-species scenarios. The scripts have lines like this near the start.

```
source("C:/Users/GRJ/AAA/Programming/Biology/DENIMSimulations/Utils/scenario-dir-names.R")
number.of.species <- 10
just.coal.paraphyly <- TRUE
just.leache <- FALSE
```

These variables determine which subset of scenarios to use. As of 2017-06-24, these are: all 4-species, all 10-species including ones I added, all 10-species in Leaché data set, seven scenarios from 10-species which are six deep cases, and no migration.

`lookat-coverage-cred-trees.R`. Uses `beast.jar dr.app.tools.TreeLogAnalyser` to make a list of trees grouped by topology, with posterior probabilities per topology. The list is sorted on the posterior probabilities and is a NEX files. My R code reads this and finds the 95% credible set. It just prints the results (TRUE, FALSE) to console for each replicate.

`lookat-ESSs.R`. Uses CODA to look at he ESSs.

`lookat-migrations.R` Uses `MigrationAnalyser` to make a summary of migrations.

`lookat-migs-yes-no.R`. For each replicate, reads the true migrations from the `migs.txt` file made by `MCcoal` as edited by me. Reads the estimated ones from the MCMC log file. Make graphs for each migration pattern of the presence/absence of migration on a locus by locus basis.

`lookat-tree-distances.R`. For each replicate, reads the true species tree from the MCMC control file. Reads the MCMC samples and evaluates the whole posterior by finding average branch score between true tree and the samples. Make graphs for each migration pattern.

`runTreeAnnotator.R`. Makes a maximum clade credibility tree for each replicate.

References

A D Leaché, R B Harris, B Rannala, and Z Yang. The influence of gene flow on species tree estimation: A simulation study. *Systematic Biology*, 63(1):17–30, 2014.