

Appendix B. Marginal likelihood calculation using BFdriver 4.0

Ziheng Yang, 1 December 2016

The C program BFdriver generates control files and job subscription scripts for running MCMC (using BPP or MCMCtree) to calculate the marginal likelihood (or the Bayes factor), as described in Rannala and Yang (2016, Syst Biol).

The program takes a control file you provide (such as bpp.ct1) and generates $K = 16$ control files with different beta values, which are used to run bpp to sample from the different power posterior distributions. The program also generates job submission scripts and submit the jobs using qsub. All generated control files and output files are in the same current directory. The frogs dataset in the bpp release (Yang 2015 Curr Zool) is used as the example.

You need the following: a linux system with SUN grid engine managing job submission (including commands such as qsub, qstat, qdel, etc.), and a C compiler. If you don't have this job submission system, you can use BFdriver to generate the control files and run the MCMC jobs from the command line.

(A) Compiling and running BFdriver

```
cc -o BFdriver -O3 BFdriver.c tools.c -lm
BFdriver <controlfilename> <npoints> <scriptname.sh>
BFdriver A00.ct1 16 tmp.sh
```

You may need to edit the following two lines inside BFdriver.c, and if you do, remember to compile the program after editing. Here bpp is assumed to be on your search path. You can use a full path for the executable program, such as /home/gooduser/bin/bpp3.3. Also the second line is for submitting the jobs using qsub. Here the limits are set to 4G of RAM and 360 hours of running time. Check those values if necessary (and recompile).

```
fprintf(fcommand, "      echo \"bpp %s.b$I.ct1 > log.b$I.txt\" > %s\n", ctrlf, scriptf);
fprintf(fcommand, "      qsub -S /bin/bash -l h_vmem=4G -l tmem=4G -l h_rt=360:0:0 -cwd %s\n", scriptf);
```

(B) Running the program

Create a folder inside frogs/, say bfl:

```
mkdir frogs/bfl
cd frogs/bfl
```

Prepare a control file (A00.ct1, say) for the A00 analysis in the folder. Check that it works. This should have a fixed species tree, which is ((K, C), (L, H)). Species delimitation and species tree estimation should be turned off. The control file specifies the priors for theta, tau, and also specifies burnin, nsample, sampfreq etc. Run bpp to confirm that the control file works. Then run BFdriver as follows:

```
BFdriver A00.ct1 16 tree1.sh
```

Here A00.ct1 is the control file we have prepared. $K = 16$ is the number of points in the Gauss-Legendre quadrature algorithm for numerical integration. You can use 8 for testing, and 16 or 32 for real calculation. tree1.sh is the temporary script file for job submission using qsub. The BFdriver command does a few things. First it reads the control file specified (A00.ct1) and creates 16 control files with names like A00.b01.ct1, ..., A00.b16.ct1.

Each of those control files has the same content as A00.ctl except that one extra line is inserted at the beginning, like the following

```
BayesFactorBeta = 0.122298 * w=0.124629.ctl
```

This specifies the beta value when the control file is used to run bpp.

Second BFdriver creates a file named betaweights.txt, which lists the beta values and Gauss-Legendre weights. I have copied those values into an excel file in frogs/BFdriver.frogs.xls.

Third BFdriver creates a file named commands, which has the bash shell scripts for submitting the 16 jobs using qsub. You use the following to submit the jobs.

```
source commands
```

You can look at the content of tree1.sh to see the script for the last job:

```
more tree1.sh
```

which should have the content like the following:

```
bpp bpp.b16.ctl > log.b16.txt
```

You can use qstat to check the status of the 16 jobs you have submitted. When the jobs are running, they generate output files in the current folder, such as mcmc.b01.txt, out.b01.txt, and log.b01.txt (which logs the screen output). After all jobs are finished, you can use grep to extract the line with "BFbeta" from screen log (this command is at the bottom of the file commands).

```
grep BFbeta log.b*.txt
```

Then copy the ElnfX values into the excel file, and estimate the logarithm of the marginal likelihood by summing (weights * ElnfX / 2) over the 16 points.

This gives the log marginal likelihood to be [-3185.93](#).

(C) Exercise & results

Duplicate the calculation for the alternative species tree (tree2) in the folder /bf2. Change the species tree topology to (((L, H), C), K), and use tree2.sh as the temporary job script file. Everything else should be the same as described above.

My runs gave the log marginal likelihood for tree 2 to be [-3186.14](#). The ratio of the posterior probabilities for the two trees is then estimated to be $P_1/P_2 = \exp\{-3185.93 - (-3186.14)\} = \exp\{0.21\} = 1.24$. With BPP4.0 and the inverse gamma priors on θ s and τ s, the MCMC runs (A01) gave the posterior probabilities for trees 1 and 2 as [0.167](#) and [0.137](#), with the ratio [1.22](#). The MCMC runs and the marginal likelihood calculations seem to agree with each other. (Note that with BPP3 and the gamma priors on θ s and τ s, the posteriors are 0.16 and 0.13 with the ratio 1.2 (Yang 2015 fig. 4).

(D) Common errors and problems

Check the bpp control file by running bpp at the command line before submitting the jobs. Make sure that the bpp program is on your path or use a full path. You may have to edit the source file BFdriver.c and recompile.