

We here describe an algorithm to perform the ABC estimation of the recurrent hitchhiking (RHH) parameters  $s$ ,  $\lambda$  and  $\theta$ , as described in Jensen, Thornton & Andolfatto (2008). This description is general, and requires the following:

A) programs needed:

a) RHH simulator (rsweep\_stochCG), in order to simulate priors with fixed parameters. See documentation in rsweep\_stochCG.cc

b) RHH simulator (rsweep\_stochCGdist), in order to simulate priors with distributed parameters. See documentation in rsweep\_stochCGdist.cc

c) ABC estimator (ABCreg), in order to perform estimation. See file README\_ABCreg

---the above, and necessary supporting files, may be found at:

<http://www.molpopgen.org/software/index.html>

d) ms (Hudson 2002), in order to examine estimates under demographic models relevant for a given population of interest. For example, point estimates obtained by fitting a demographic model to the data may be of particular interest (*e.g.*, Ometto *et al.* 2005; Thornton and Andolfatto 2006). The program may be found at:

<http://home.uchicago.edu/~rHUDSON1/source/mksamples.html>

e) R, in order to calculate MAP estimates and quantiles from posterior distributions. Also extremely useful for plotting posteriors.

<http://www.r-project.org/>

B) Calculations needed:

The estimation procedure relies upon the observed empirical values of the means and standard deviations of  $S$  (the number of segregating sites),  $\pi$ ,  $\theta_H$ , and  $ZnS$ . These values may be obtained using your favorite analysis program.

C) With these, the algorithm to perform estimation is as follows (more details are also available in the documentation of ABCreg):

1) Calculate the observed empirical values of the above statistics, and create a 'data file', looking like:

summary1 summary2 summary3 ...

2) Choose prior distributions for the parameters. In our implementation,  $s \sim \text{Uniform}(1.0\text{E-}06, 1.0)$ ,  $2N\lambda \sim \text{Uniform}(1.0\text{E-}07, 1.0\text{E-}01)$ , and  $\theta \sim \text{Uniform}(0.0001, 0.1)$ , and the tolerance used is  $\delta = 0.001$ .

3) Draw random parameters from the prior distributions, and simulate data using those parameters, using the RHH simulation programs. Each replicate should be simulated to match the empirical data configuration. For example, if the dataset consists of 50 1kb regions, and 25 500 bp regions, the replicate datasets must be simulated in this configuration as well. The command line for the simulator is as follows:

```
rsweep_stochCG nsam nreps N s  $\Lambda$   $\rho$  nsites  $\theta$  seed
```

where,

nsam = sample size

nreps = number of replicates

N = effective population size

s = selection coefficient

$\Lambda$  = rate of sweeps per base pair \* 4N generations

$\rho$  = 4Nr per locus

nsites = length in base pairs

$\theta$  = 4N $\mu$  per locus

seed = random number

4) Calculate the 8 summary statistics for the data simulated in step 3. In a file, record the parameters and their corresponding summary statistics.

5) Return to step 3 a large number of times (*e.g.*, 1 million). This will result in a 'prior file' looking like:

```
param1 param2 ... summary1 summary2 ...  
param1 param2 ... summary1 summary2 ...  
etc...
```

6) Perform estimation. Usage of ABCreg, is as follows:

```
~/Projects/ABCreg/reg -P 3 -S 8 -p prior -d infile -b infile -t 0.001 -T -m 1000000
```

-P = number of parameters

-S = number of summaries

-p = prior file

-d = data file

-b = base name of output files. If you say -b file, the output will be file.1.log.post, file.2.log.post, etc.

-t = tolerance

-T = transform the parameters. always do this.

-m = max # of lines to read from prior file. if you don't use this, all lines are read.

7) Using the output posterior file, the MAP estimates and quantiles for the three parameters may be obtained in R, using the following command lines:

```
#read in data file
x=read.table("posterior.txt",colClasses=c(rep("numeric",3)))
source("~/Rcode/getmap.R")
y=getmap(x$V1)
quantile(x$V1,prob=c(0.025,0.975))
z=getmap(x$V2)
quantile(x$V2,prob=c(0.025,0.975))
w=getmap(x$V3)
quantile(x$V3,prob=c(0.025,0.975))
```

Plotting is also relatively straightforward in R. For example, for plotting the three parameters for our empirical dataset (corresponding to Figure 6), for both distributed (black lines) and fixed (gray lines) priors:

```
#read in data file
x=read.table("fig6_R.txt",colClasses=c(rep("numeric",3)))
source("~/Rcode/getmap.R")
s_empirical_fixed_all_log = log10(x$V1)
L_empirical_fixed_all_log = log10(x$V2)
t_empirical_fixed_all_log = log10(x$V3)
s_empirical_dist_all_log = log10(x$V4)
L_empirical_dist_all_log = log10(x$V5)
t_empirical_dist_all_log = log10(x$V6)
s_empirical_fixed_all<-process2plot(s_empirical_fixed_all_log)
L_empirical_fixed_all<-process2plot(L_empirical_fixed_all_log)
t_empirical_fixed_all<-process2plot(t_empirical_fixed_all_log)
s_empirical_dist_all<-process2plot(s_empirical_dist_all_log)
L_empirical_dist_all<-process2plot(L_empirical_dist_all_log)
t_empirical_dist_all<-process2plot(t_empirical_dist_all_log)
require(gplots)
library("geneflower") ## from BioConductor
require("RColorBrewer") ## from CRAN
#name output pdf file and give dimensions
pdf("plotsFig6.pdf", width=12, height=6, fontsize=24)
#figure with one row and three panels
par( mfcol= c(1, 3))
#plot a line
plot(s_empirical_dist_all[,1], s_empirical_dist_all[,2], col="black", xlim=c(-6,0),ylim=c(0,1),type="l",
#label the axes
xlab="log10(s)", ylab="density", lwd=4)
#add a second line
lines(s_empirical_fixed_all[,1], s_empirical_fixed_all[,2], col="gray", lwd=4)
#title the panel
title("empirical(strength)", cex.main=1)
#add a legend
legend("topleft", c("distributed", "fixed"), col = c("black", "gray"), lty =c("solid", "solid"), bty = "n",
text.col = "black", lwd = c(4, 4), cex = 0.75, merge = TRUE, bg = 'white')
#repeat for other panels
plot(L_empirical_dist_all[,1], L_empirical_dist_all[,2], col="black", xlim=c(-6,0),ylim=c(0,1),type="l",
```

```
xlab=expression(paste(log10,"(2N",lambda,")")), ylab="density", lwd=4)
lines(L_empirical_fixed_all[,1], L_empirical_fixed_all[,2], col="gray", lwd=4)
title("empirical(rate)", cex.main=1)
plot(t_empirical_dist_all[,1], t_empirical_dist_all[,2], col="black", xlim=c(-6,0),ylim=c(0,1),type="l",
xlab=expression(paste(log10,"(",theta,")")), ylab="density", lwd=4)
lines(t_empirical_fixed_all[,1], t_empirical_fixed_all[,2], col="gray", lwd=4)
title(expression(paste("empirical (neutral ",theta,")")), cex.main=1)
```