10_speciation

June 5, 2024

Practical: Speciation time using ABC

Preparation For this practical you need some R packages, namely `coda`, `abc`, `maps`, `spam`, `fields`. For plotting purposes you may also want to use `ggplot2`.

You will also need the software `ms` to be installed. You can find the executable for linux in `bin/ms`. If it doesn't work, you can compile the source `Software/ms.tar.gz` by `tar -xzvf ms.tar.gz; cd msdir; gcc -o ms ms.c streec.c rand1.c -lm`. Finally you need some data and R functions provided in `Data`.

I suggest to copy `functions.R` and `polar.brown.sfs*` in the workspace where you will run this practical without overwriting the repository.

You can work in teams for this practical.

Project In this practical you are going to estimate the divergence (or speciation) time between polar bears and brown bears using genomics data. You will be using Approximate Bayesian Computation (ABC) methods to inference such time.

[ ]:
```
# Open R and load all R functions and data needed:
source("Data/functions.R")
load("Data/polar.brown.sfs.Rdata")

# Inspect the objects:
ls()
```

The file `polar.brown.sfs` includes the joint (2 dimensions) site frequency spectrum (SFS) between polar bears (on the rows) and brown bears (on the columns). This is based on real genomic data from 18 polar bears and 7 brown bears. The site frequency spectrum is a matrix $N \times M$ where cell (i, j) reports the number of sites with allele frequency $(i-1)$ in polar bears and $(j-1)$ in brown bears. If you want to see this file type `cat polar.brown.sfs` in your terminal.

[ ]:
```
# You can plot this spectrum:
plot2DSFS(polar.brown.sfs, xlab="Polar", ylab="Brown", main="2D-SFS")
```

Each population has $2n+1$ entries in its spectrum, with n being the number of individuals. The number of chromosomes for each species (bears are diploids, like humans) can be retrieved as:

[ ]:
```
nChroms.polar <- nrow(polar.brown.sfs)-1
nChroms.polar
nChroms.brown <- ncol(polar.brown.sfs)-1
nChroms.brown
```

The only thing you need to remember about the site frequency spectrum is that we can easily calculate several summary statistics from it. These summary statistics can be used for inferences in an Approximate Bayesian Computation (ABC) framework.

For instance, from the site frequency spectrum, we can easily calculate the number of analysed sites (in this example all sites are polymorphic, and thus variable in our sample), which is simply the sum of all entries in the SFS.

[ ]:
```
nrSites <- sum(polar.brown.sfs, na.rm=T)
nrSites
```

This value is important as we will condition the simulations to generate this number of sites for each repetition. In other words, when simulating data we will simulate exactly this number of polymoprhic sites to calculate the site frequency spectrum and all corresponding summary statistics afterwards.

I provide a function to easily calculate several summary statistics from a site frequency spectrum.

[ ]:
```
obsSummaryStats <- calcSummaryStats(polar.brown.sfs)
obsSummaryStats
# These are the OBSERVED summary statistics! Keep them.
```

These are the summary statistics available in this practical and their meaning is the following: * fst: population genetic differentiation; it measures how much species are genetically different; it goes from 0 (identical) to 1 (completely different); * pivar1: genetic diversity of species 1 (polar bears); * pivar2: genetic diversity of species 1 (brown bears); * sing1: number of singletons (sites with frequency equal to 1) for species 1 (polar bears); * sing2: number of singletons (sites with frequency equal to 1) for species 2 (brown bears); * doub1: number of doubletons (sites with frequency equal to 2) for species 1 (polar bears); * doub2: number of doubletons (sites with frequency equal to 2) for species 2 (brown bears); * pef: proportion of sites with equal frequency between polar bears and brown bears; * puf: proportion of sites with unequal frequency between polar bears and brown bears (note that puf=1-pef).

It is not important that you understand the significance (if any) of all these summary statistics in an evolutionary context. If interested, a nice review is "Molecular Signatures of Natural Selection" by Rasmus Nielsen (you can find a pdf copy in `Readings/Papers/Nielsen_2005.pdf`). However, some of these summary statistics might be more informative than others. It is your first goal to understand which summary statistics to keep.

The parameter you want to estimate is the divergence time between polar and brown bears (T). You first aim is to performs N simulations of data by drawing from a prior distribution of T and record (separately) the drawn values and the corresponding summary statistics generated by that value of T.

You can define how many simulations you want to perform (ideally a lot).

[ ]:

```
nrSimul <- 1e4 # but change this accordingly
```

Then you should define the prior distribution of our parameter to be estimated, the divergence time T. You can use any distribution you find suitable. However, you may want to consider that a reasonable range of values for T is between 200k and 700k years ago.

The function to simulate data (specifically the site frequency spectrum) given values of T (and M, the migration rate) is `simulate`:

[ ]:

```
simulate
```

This function takes as parameters: T (divergence time), M (migration rate), how many sites to simulate, the directory for `ms` program and the text file in output. This function simulates a joint evolutionary history for both polar and brown bears according to what we know in terms of their respective changes in size. However, you can set when they speciated (T in years ago) and the migration rate (M). (Note that the migration rate is scaled by the reference population size so a reasonable range of M is between 0 and 2.)

As an example, assuming T=200k and M=0 the command to simulate data and calculate summary statistics is the following:

[ ]:

```
# first, set the path to the "ms" software you installed
msDir <- "bin/ms" # this is my specific case, yours could be different
msDir <- "/home/matteo/Documents/Teaching/Workshops/EMBO-POPGEN/Day3/Notebooks/
 ↪Software/msdir/ms"

# second, set the name for the output text file
fout <- "ms.txt" # leave it like here

# then we can simulate data:
```

```
simulate(T=2e5, M=0, nrSites, msDir, fout)

# and finally calculate the summary statistics for this simulation
#(note that you need to specify the number of chromosomes for the two species)
simulatedSFS <- fromMStoSFS(fout, nrSites, nChroms.polar, nChroms.brown)
calcSummaryStats(simulatedSFS)

# you can even plot the simulated site frequency spectrum
plot2DSFS(simulatedSFS, xlab="Polar", ylab="Brown", main="simulated 2D-SFS")
```

Based on the observed summary statistics 'fst', which measures how different polar and brown bears, in relation to the one calculated simulating T=2e5, can you make some initial (very rough) considerations on the most likely values of T (higher or lower than 200k years ago)?

You can use the abc package and the abc function to calculate the posterior distribution (as well as to compute the distance between observed and expected summary statistics).

[ ]:
```
library(abc)
```

[ ]:
```
#?abc
```

As you can see, to perform an ABC analysis you need 3 objects: * target: a vector of the observed summary statistics; * param: a vector, matrix or data frame of the simulated parameter values; * sumstat: a vector, matrix or data frame of the simulated summary statistics.

You already have 'target' as it is the vector of observed summary statistics called 'obsSummaryStatistics'.

You now have everything to estimate the divergence time. For simplicity assume that $M = 0$. Also, you are free to choose a rejection or local-regression method, as specified in the 'abc' function. This is not strictly required, but if you want to explore the estimation of two parameters simultaneously, you can estimate M by defining a prior for it, draw random samples jointly of T and M, calculate summary statistics, and so on.

Hints Please consider these points carefully when completing the project. * Assess which summary statistics are more or less informative for the parameter estimation (e.g. after a first run of simulations with all parameters, look for correlations between the simulated parameter value and summary statistics). * You can also look for correlations between summary statistics and eventually use only one of the pair if two summary statistics are highly correlated. If you are a pro, you can also perform a principal component or multidimensional scaling analysis (e.g. with package 'pls') and by using each statistic's loadings, you can create novel uncorrelated summary statistics which are linear combinations of the previous ones (this part is purely suggestive). * Remember to scale your simulated (jointly with the observed) summary statistics separately, so that the mean is zero and standard deviation is one. * Generate a plot with the posterior distribution of the parameter of interest. You can also show the chosen prior distribution on the same plot. * Calculate the posterior mean, mode, median and other notable quantities (e.g. 95% HPD interval) to summarise the posterior distribution. * I suggest you to use the 'abc' package in R instead of implementing methods (e.g. regression) yourself. * A useful diagnostic plot to show is the distribution of sampled values from the prior: do they cover the whole range of the prior (and are they distributed as expected)?

[ ]:

```
# ...
```