Exercise 3

Matteo Fumagalli

February 15, 2017

Preparation

For this exercise you need some R packages, namely: 'coda', 'abc', 'grid', 'maps', 'spam', 'fields'. For plotting purposes you may also want to use: 'ggplot2'.

You will also need the software 'ms' to be installed. This can be obtained here:
https://uchicago.app.box.com/s/l3e5uf13tikfjm7e1il1eujitlsjdx13
where you should download 'ms.tar.gz'. Then unzipped it and compile it with
`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 here:
https://github.com/mfumagalli/BayesianMethods.
Assuming that you will be working in a folder called 'Bayesian', this what you should do:

```
git clone https://github.com/mfumagalli/BayesianMethods.git
cd BayesianMethods
git pull
cd ..
mkdir Bayesian
cd Bayesian
cp ../BayesianMethods/functions.R .
cp ../BayesianMethods/polar.brown.sfs* .
```

If you encounter difficulties please do let me know.

Instructions

A document in pdf format must be submitted before the final exam on 28^{th} March 2017. Any time before the start of the exam is fine. The R code used must be clearly visible and properly formatted in the document. The use of latex to generate the pdf is preferable but not mandatory. A simple list of R lines used is not sufficient. The reasoning for each step must be clearly indicated, either using brief comments in the code or in a separate text. Likewise adding plots and/or tables for diagnostic and/or final analyses are encouraged. I am not interested in the correct answer *per se* (if any) but rather in seeing your reasoning behind the different analyses you performed. For instance, if you start

with a certain prior and then choose another way you can present both analyses and explain why you moved from a first prior to a second one (e.g. because the estimate was close to the border with the firstly used bounded prior?). In general, you are free to choose the approach you want and there is no right or wrong way of doing it, but I am interested to see that you took informed decisions in your project.

The completion of this exercise will count up to 30% of the question from the Bayesian Statistics class. This means that if you don't submit this exercise, then your question at the exam will count no more than 70%.

This project can be completed individually or in small groups but with no more than 4 students per group. I expect that each student in the working group has actually participated in the project completion. I won't assign a different weight depending on whether the project was completed by one or four students.

Once completed, the document can be sent via email to m.fumagalli@imperial.ac.uk. As the exam is anonymous, please do not put your name(s) in the document but rather indicate the CID numbers for all students that have participated to the submitted project. Of course I won't keep track of the email address that has been used for sending a project. However, if you prefer, you can send the document to Samraat who will then forward it to me.

Please do not hesitate to contact me if you have additional questions.

Project

In this

```
# Open R and load all R functions and data needed:
source("functions.R")
load("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. This site frequency spectrum is a matrix N x M where at cell i,j 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
```

It is not important that you understand the concept of site frequency spectrum. The only thing to remember is that we can easily calculate several summary statistics from it. These summary statistics can be used for inferences based on Approximate Bayesian Computation techniques. Also consider that it is easy to calculate a site frequency spectrum from simulations.

For instance, from the site frequency spectrum, we can easily calculate the number of analysed sites (in this example all sites are polymorphic), 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 sites to calculate the site frequency spectrum and the corresponding summary statistics afterwards.

Notably, as said before, from the site frequency spectrum we can calculate several summary statistics. I provide a function to easily do that from a site frequency spectrum.

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

These are the summary statistics available and their meaning is the following:

1. fst: population genetic differentiation; it measures how much species are genetically different; it goes from 0 (identical) to 1 (completely different);
2. pivar1: genetic diversity of species 1 (polar bears);
3. pivar2: genetic diversity of species 1 (brown bears);
4. sing1: number of singletons (sites with frequency equal to 1) for species 1 (polar bears);
5. sing2: number of singletons (sites with frequency equal to 1) for species 2 (brown bears);
6. doub1: number of doubletons (sites with frequency equal to 2) for species 1 (polar bears);

7. doub2: number of doubletons (sites with frequency equal to 1) for species 2 (brown bears);

8. pef: proportion of sites with equal frequency between polar bears and brown bears;

9. 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 these summary statistics in an evolutionary context. If interested, a nice review is "Molecular Signatures of Natural Selection" by Rasmus Nielsen, Annual Review of Genetics, Vol.39:197-218, 2005, DOI: 10.1146/annurev.genet.39.073003.112420.

However some of these summary statistics might be more informative than others. It is your first goal to understand which summary statistics to keep (although you may decide to keep them all).

The parameter we want to estimate is the divergence time between polar and brown bears (T). Additionally (but this is not required to get the full score) one can estimate the migration rate (M) too.

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.

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

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

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. (For the brave ones, the migration rate is scaled by the reference population size so you can consider a reasonable range of M being between 0 (included) and 3.)

The function to simulate data (specifically the site frequency spectrum) given values of T (and M) 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).

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 <- "~/Software/msdir/ms" # this is my specific case, yours
    could be different
# 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(polar.brown.sfs, xlab="Polar", ylab="Brown",
    main="simulated 2D-SFS")
```

Quick question: based on the observed summary statistics 'fst' which measure how different polar and brown bears are compared to the one calculated simulating T=2e5, can you make some 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 required to get the full score, 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. A diagnostic plot is the joint posterior probability of T and M.

Good luck!

Hints

Please consider these points carefully when completing the project.

- Assess which summary statistics are more or less informative for the parameter's estimation (e.g. after a first run of simulation 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 statistics' loadings, you can create novel uncorrelated summary statistics which are linear combinations of the previous ones (this part is purely suggestive and it is not required to obtain the full score).
- 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) you consider of interest to summarise the posterior distribution.
- I suggest you to use the 'abc' package in R as it implements the local-linear regression method too. However you can also implement a rejection sampling method yourself, as seen in class.
- 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)?