Making Complex Radiocarbon Calibration Software More Accessible: a New Approach?

C. E. Buck, J. Andrés Christen

Abstract

The present authors have spent many years developing a Bayesian framework for the calibration of radiocarbon determinations in the light of a range of archaeological information. In the process we have written thousands of lines of robust C code which is, in general, very well tested. The code is complex and the statistical theory behind what it does is far beyond the comprehension of most members of the archaeological community. As a result only a handful of researchers world-wide have access to and can utilise our software. Surely there must be a way to give other researchers access while at the same time not devoting all our research time to rewriting the code and developing `pretty' front ends for a range of different hardware types! In this paper we report on our efforts to develop an Internet-based Bayesian radiocarbon calibration facility (BCal). This facility will allow archaeologists to issue instructions in terms of the archaeological and radiocarbon concepts that they already understand rather than using the C programming language directly. We hope that BCal will be a user-friendly graphical interface (perhaps in HTML or JAVA) to allow the user to submit archaeological information and radiocarbon determinations for calibration without any need for programming knowledge at all. The calibrations would be undertaken remotely on our UNIX workstations regardless of the operating system and processor type available on their local machines and the results made available for use locally in a range of graphical and text formats.

1 Introduction

For several decades many researchers interested in the application of statistics to archaeological data interpretation have decried the use of 'off the peg' statistical methodologies developed primarily for use in other disciplines. Nonetheless those same researchers (including the present authors) have for various reasons made only a small proportion of their purpose built software available to the wider archaeological community. This means that, even if the feasibility of the statistical approaches is demonstrated and the results are well received, very few other researchers ever adopt the methodologies.

The present authors have spent many years (along with other workers) developing a Bayesian framework for the calibration of radiocarbon determinations in the light of a range of archaeological information. In the process we have written thousands of lines of robust C code which is, in general, very well tested. The code is complex and the statistical theory behind what it does is far beyond the comprehension of most members of the archaeological community. As a result only a handful of researchers world-wide have access to and can utilise our software. This paper reports on attempts to remedy the situation.

2 Bayesian radiocarbon calibration

Since the amount of atmospheric ^{14}C has not been constant over time, radiocarbon determinations as supplied by the professional laboratories must be calibrated onto the calendar scale before they are a really useful aid in dating past human activity. The need for such calibration has been known for some time, but it is only recently that statisticians have developed Bayesian radiocarbon calibration techniques that take into account the very special nature of archaeological dating evidence.

Bayesian radiocarbon calibration allows us to combine information from a range of different sources to arrive at a coherent interpretation of all the available dating evidence. In formal terms we combine: *a priori* chronological information with radiocarbon data and the calibration curve via a statistical model to arrive at posterior probability distributions for the calendar dates of interest.

There are now published Bayesian methodologies for tackling a wide range of radiocarbon dating problems. These include:

1. Calibrating single determinations with or without *termini*.
2. Calibrating groups of related determinations with simple or complex stratigraphic relationships.
3. Probabilistic assessment of the chronological location of samples of unknown stratigraphic origin within a radiocarbon sequence that is well understood.
4. Inferring the dates of chronological events (such as phase boundaries) whose relative chronological location is known, but for which we have no direct dating evidence.
5. Identification of outliers in a collection of radiocarbon determinations.
6. Selection of samples from a currently available pool of organic materials in order to optimise our chances of answering the chronological questions posed.

With the appropriate prior information and relative chronological sequence modelled into the calibration process, users of this framework obtain posterior probability distributions for each calendar date in which they are interested (including those for which they do not have direct dating evidence). These can be summarised using either graphical methods or summary statistics and a range of these are now well documented and demonstrated elsewhere.

3 What are the problems?

The archaeological information can be modelled and there is a Bayesian framework in place for tackling a wide range of chronological issues, so what's stopping the majority of the archaeological community from adopting these techniques? There are two main problems.

1. Each archaeological project has unique prior information, unique data and requires different statistical models from other projects we've tackled. This means that each project requires its own computer code via which the posteriors are calculated.
2. The techniques used to calculate the posteriors are based on Markov chain Monte Carlo simulation and are commonly extremely CPU intensive.

There are a number of approaches we could take to solving these problems:

1. Simplify the procedures using 'reasonable' assumptions so that the same models and similar priors can be used for a large number of projects and calculations can be done on standard desktop PCs.
2. Assume that statisticians will always have to help in the process and set up consultancies to provide archaeologists with the advice they need.
3. Find a way to standardise the code we use without compromising on the quality and range of statistical models available.
4. Make our code available to others for use on their own machines if they have sufficiently powerful ones.
5. Make CPU on our machines available to others who do not have the necessary power on their own desktop.

In deciding which approach to take, there were a number of achievements upon which we could build:

1. A huge library of robust and well tested C routines for tackling a large number of radiocarbon calibration problems within the carefully developed Bayesian framework.
2. A large number of case studies which make use of the library and from which we have gained the experience to begin to standardise the software without compromising on the quality of the statistical modelling available.

There were, however, a number of things that we still needed to develop:

1. An interpreter that sits between the C code and the user, preventing them from having to learn a programming language, but at the same time allowing them to provide explicit and coherent information from which posterior probabilities will be calculated.
2. A graphical user interface to the interpreter with built in checking mechanisms for internal consistency.
3. The CPU time needed to undertake the calculations.
4. Access to the results in a range of formats suitable for presentation both on-line and for publications.

We have already developed much of what is needed and we have a clear structure in which our new calibration environment will operate (Figure 1). It is time now to decide what form the graphical user interface should take and how best to make use of available CPU.

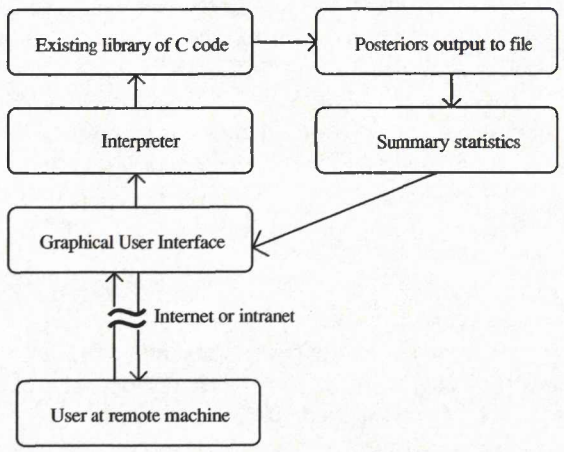

Figure 1: The structure of the proposed Internet-based Bayesian radiocarbon calibration facility (BCal).

4 The interpreter - *mexcal*

This part of the software is written in C++. It takes lines from a text-based input file (in which all the objects needed for the current calibration are declared), interprets them and makes calls to the necessary calibration routines. A *mexcal* input file is text-based and contains one line for each piece of modelling information and data, given in a standard form. Since the standard format relates directly to the archaeological problem and the data available it is a good deal easier to program in *mexcal* than in C++. For those of us who have worked in this area for sometime, the saving achieved by using the interpreter rather writing new C code is several days work. Typically one can define a new problem in a matter of minutes or hours rather than hours or days.

5 The graphical user interface - *BCal*

This part of the software is still at the design stage and our ideas about it are evolving all the time as access to the Internet becomes more wide spread and the bottle necks get worse. Nonetheless, the interface will need to have the capability

1. for user-friendly submission of information ready for calibration, and
2. presentation of results to the user when they are complete.

We already have the necessary code to provide all of the input and output. What we still require is the mechanism for interfacing with the user. We are planning:

1. a WWW server on which the software will sit,
2. HTML (or JAVA) interface screens with associated CGI scripts to check the consistency of the input information.
3. submission of the checked information to *mexcal*.

The mexcal program will then be submitted to run on an appropriate server.Upon completion of the calibration the user will be e-mailed with information about the success or failure of the job and a URL at which the output can be viewed.

6 Questions on which we are still working and would appreciate comments

1. Are HTML and JAVA the best way to provide the platform independent interface or are there better ways?
2. Are there any useful tricks for providing server security in situations where one is providing users with access to CPU in this way?
3. How much post processing of the output data should our software do and how much should we leave to the user?
4. How should we undertake the consistency checking of user input?

7 Obtaining further information

Since this paper was presented, the authors have obtained funding for the Bcal project from the University of Wales, Cardiff. Readers interested in the latest progress should visit http://bcal.cf.ac.uk.

Contact details

Caitlin E. Buck
School of History and Archaeology
University of Wales
Cardiff
WALES

J. Andrés Christen
Instituto de Matemáticas
Unidad Morelia
UNAM
MÉXICO