Bacon manual – v2.2

Maarten Blaauw[1] and **J. Andrés Christen**[2]

1 School of Geography, Archaeology and Palaeoecology, Queen's University Belfast, Belfast, UK
2 Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Mexico

If a man will begin with certainties, he shall end in doubts,
but if he will content to begin with doubts, he shall end in certainties
Francis Bacon (AD 1561 – 1626)

Contents **page**

1 Introduction 2
2 Prior information 2
3 Dates 3
4 Installation 5
5 Getting started 6
6 User-relevant Bacon functions 7
6.1 Running age-depth models 7
6.2 Producing age-depth graphs 8
6.3 Post-run analyses 9
6.4 Reloading existing runs 10
7 Version history and future development 10
8 References 11

Disclaimer

Bacon is open-source software; you are free to use, copy, distribute and modify it, but please do read this manual and accompanying paper before using the program. This software is distributed under the terms of the GNU General Public Licence. Bacon does not come with any warranty and the authors do not assume any responsibility for the usefulness of any portion of this program. Do cite this program when modifying or using it, including its version, applied settings and calibration curves.

Parts of this software remain under construction (particularly the Bacon.R interface). Details such as the default settings and behaviour could change between versions, so please check this manual for the latest information. You are most welcome to report bugs, ideas and missing features to Maarten (maarten.blaauw@qub.ac.uk) or Andrés (jac@cimat.mx).

1 Introduction

Bacon (Blaauw and Christen, 2011) is an approach to age-depth modelling that uses Bayesian statistics to reconstruct Bayesian accumulation histories for deposits, through combining radiocarbon and other dates with prior information. Prior assumptions about accumulation rate and its variability over time are taken into account explicitly, much like Bacon's predecessor Bpeat (Blaauw and Christen, 2005). In many cases Bacon isn't bothered by outlying dates, since the dates are modelled using a student-t distribution with wide tails (Christen and Pérez, 2009). The software runs on Windows, Mac and Linux, using R (R Development Core Team, 2013) as an interface.

Any age-depth model produces estimates of accumulation rates, either implicitly or explicitly. For example, the most popular method of connecting the mid-points of dated levels using linear sections (Blaauw, 2010), assumes that a deposit accumulated linearly between each dated level, and that changes in accumulation rate took place abruptly and exactly at the dated depths.

Bacon divides a core into many thin vertical sections (by default of *res*=5 cm thickness), and through millions of Markov Chain Monte Carlo (MCMC) iterations estimates the accumulation rate (in years/cm; so more correctly, sedimentation times) for each of these sections. Combined with an estimated starting date for the first section, these accumulation rates then form the age-depth model. The accumulation rates are constrained by prior information as described below.

2 Prior information

The accumulation rate prior consists of a gamma distribution (Fig. 1, left panel), which is much like a normal/Gaussian distribution but is often asymmetric and always positive (since we can often safely assume that deposits did not accumulate backwards in time). It has two parameters: *acc.shape* (default 1.5, higher values result in more peaked shapes) and *acc.mean* (default 20, which can be changed to, e.g., 5, 10 or 50 for different kinds of deposits). The default prior values are currently based on Goring et al. (2012), allowing for more variability than the defaults of former Bacon versions (2 and 10 respectively; Blaauw and Christen 2011). In case initial rough age estimates indicate an accumulation rate much different from the default, Bacon will suggest an alternative acc.*mean* based on round values such as 1, 2, 5, 10, 20, 50, …, 500 yr/cm. Ensure that you agree with the prior distributions applied to your core.

The section thickness (default *res=5*) will dictate to some degree the flexibility of the age-depth model. With few and thick sections, the model will be rather rigid yet with abrupt changes in accumulation rate, whereas with many sections the model will often appear more smooth. For cores longer than a few metres it could make sense to increase the section thickness for an initial quick run (e.g., setting *res=50*), followed by decreasing res until the model appears sufficiently smooth.

The memory defines how much the accumulation rate of a particular depth in a core depends on the depth above it (Fig. 1, right panel). If this memory, or autocorrelation, is assumed to be low, accumulation rate is thought to have changed much over time (i.e. the environmental conditions or internal dynamics that influence accumulation were very variable), while a high memory will imply a smooth, more constant accumulation history. The prior for the memory is a beta distribution, which looks much like the gamma distribution but its values are always between 0 (no assumed memory) and 1 (100% memory). Its default settings of *mem.strength=4* (higher values result in more peaked shapes) and *mem.mean=0.7* allow for a large range of posterior memory values.

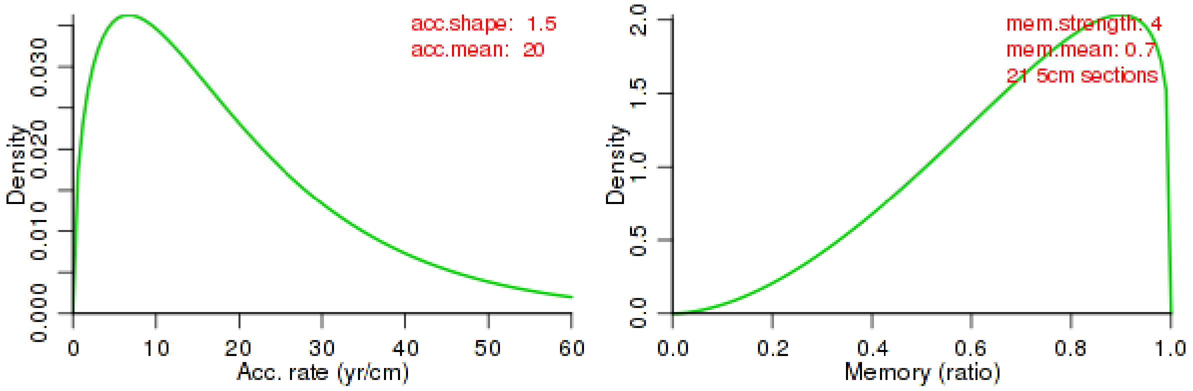

Fig. 1. Prior information for accumulation rate (left) and its memory or variability (right). Defaults are shown.

At certain core depths, hiatuses can be inferred. The prior distribution for the time gap of such hiatuses is modelled by a gamma distribution (Fig. 2). We suggest that unless there are clear indications otherwise, you keep the default value of *hiatus.shape=1*, which causes short hiatuses to be the most likely (*hiatus.shape* <1 will never reach 0 yr gaps, and as such is not an advisable value). The mean time gap for this prior is set by *hiatus.mean* (default 1000 yr). If you expect hiatuses to be very small to non-existent, simply put, e.g., *hiatus.mean=10*. Accumulation rates are assumed not to be auto-correlated between the depths just before and after a hiatus, so, memory is reset to 0 while crossing the hiatus.

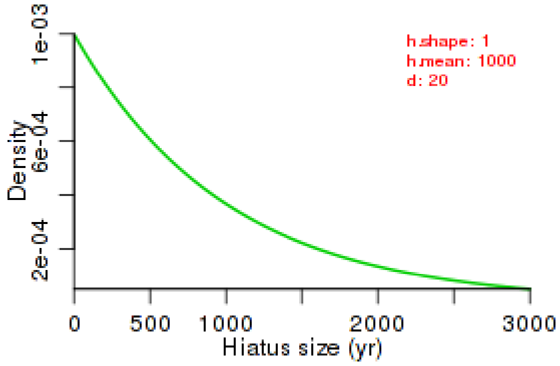

Fig. 2. Example of a prior distribution for hiatus size, here set at 20 cm depth.

The assumed depths for any hiatus should be provided as, e.g., *hiatus.depths=20* for one at 20cm depth, and *hiatus.depths=c(20,40)* for two hiatuses at 20 and 40 cm depth. By default, similar priors for accumulation rate are assumed for all sections above and below hiatuses. If you want to define distinct accumulation rate priors for sections below and above a hiatus, set these as follows, e.g.: *acc.mean=c(20,100)*, *acc.shape=c(1.5, 2)* if the prior for accumulation rate should be mean 20, shape 1.5 above the hiatus, and mean 100, shape 2 below. Similarly for cores with two or more hiatuses, e.g. *hiatus.depths=c(20,40)*, *acc.mean=c(20, 100, 500)*. The prior for memory stays the same throughout the core, however memory is set at 0 crossing a hiatus.

3 Dates

Although Bacon works with any kind of absolute dates (e.g., ^{210}Pb, OSL, tephra or other dates on a calendar scale), it is often used to age-model ^{14}C-dated sequences. Radiocarbon dates should be calibrated using either IntCal13 (for terrestrial northern hemisphere material; Reimer et al., 2013), Marine13 (for marine dates; Reimer et al., 2013), SHCal13 (for southern hemisphere dates; Hogg et al., 2013) or any other calibration curve (see below), while modern ^{14}C dates are calibrated using one of the post-bomb calibration curves (NH1, NH2 or NH3 for the northern hemisphere, SH1-2 or SH3 for the southern hemisphere; Hua et al., 2013). See http://intcal.qub.ac.uk/CALIBomb if you are unsure which postbomb curve you need. If Bacon finds postbomb dates (negative ^{14}C ages) and you haven't specified a postbomb curve, you will be prompted. Provide postbomb curves as, e.g., *postbomb=1* for the NH1 postbomb curve (*2* for NH2, *3* for NH3, *4* for SH1-2, *5* for SH3).

All ^{14}C dates should be provided in ^{14}C BP, not in cal BP, F^{14}C or pMC (% modern carbon). Any pMC dates can be translated to ^{14}C BP using Bacon's *pMC.age()* function (see below).

Bacon's calendar scale is cal BP (calendar years before AD 1950). Adapt AD/BC ages accordingly (positive BC/AD values represent AD ages, negative ones BC ages. If we neglect that 0 BC/AD never existed, then cal BP ages can be calculated as 1950 - BC/AD). The calendar scale of graphs and age output-files can be changed to BC/AD providing the option *BCAD=TRUE.*

The default ^{14}C calibration curve is IntCal13 (*cc=1*). You can change this to *cc=2* (Marine13), *cc=3* (SHCal13), or *cc=4* (an alternative curve). For calendar dates, i.e. dates that are already on the calendar scale and thus should not be calibrated, set *cc=0* (see below if your core contains a mixture of ^{14}C and calendar dates). Provide values for cc either as option within the *Bacon()* command (for the entire core only), or as a fifth column within the core's .csv file (see below).

The file containing the dates should have at least four columns separated by commas (default, alternative separators can be provided as, e.g. *sep*=”;”). The first column should contain the lab IDs (e.g., UBA-18000 or tephra1), the second and third should contain the ages and reported errors (at 1 standard deviation; errors should always be >0 since we must firmly suspect 0yr errors in anything related to age-depth modelling), and the fourth one should list the depths of those dates. The file should start with a line containing headers such as “labID”, “age”, “error”, and “depth”. Avoid using spaces within the individual names. For an example, open Bacon/Cores/MSB2K/MSB2K.csv (taken from Blaauw et al., 2003) in Excel or a plain text editor. Store the file in its own folder within the Cores folder, e.g. Bacon/Cores/CoreA/CoreA.csv if your core is called CoreA.

Previous Bacon versions used plain text files with a .dat extension and columns separated by tabs or spaces. Since users reported problems with producing files containing .dat extensions in Mac and Windows, from now on Bacon works with .csv files. If no such file is found, Bacon will try to read the .dat file if present, and convert it to a .csv file if no unexpected values or columns are found. Bacon’s .csv files are not compatible with clam’s .csv files, since both expect different information. (NB: a very easy way of tripping some versions of MS Excel is having it open a .csv file that starts with the letters *ID*. Therefore please avoid using that name for the first column!)

If your core has a mix of dates (e.g. ^{14}C and calendar ones), add a fifth column called *cc* with the corresponding number 0-4 for each date. NB this differs from previous Bacon versions! As explained above, calendar dates receive a 0, northern hemisphere terrestrial ^{14}C dates a 1, marine ^{14}C dates a 2, southern hemisphere terrestrial ^{14}C dates a 3, and ^{14}C dates that need an alternative calibration curve a 4. See Bacon/Cores/RLGH3/RLGH3.csv for an example (where the surface has been given a calendar date based on when this core was sampled). The number of any required postbomb curve should be provided within the Bacon command.

Any alternative calibration curves should be put in the folder Bacon/Curves. They should be plain text files with three columns (cal BP ages, ^{14}C ages, and errors) ordered to increasing cal BP ages, separated by spaces and without any headers or other text. See Bacon/Curves/3Col_intcal13.14C for an example. Provide the name of the curve within the option calcurve4.

If your material stems from a mix of two reservoirs (e.g., atmospheric and marine), you can produce 'mixed' curves. Within the function *mix.curves()*, provide a ratio (default 0.5) for either of the curves (cc1, default IntCal13, and cc2, default Marine13), and a name for the new curve (default mixed.14C). Offsets and their uncertainties can be included through, e.g., *offset*=c(100,50) (default 0 resp. 0). Once this curve has been produced tell Bacon to use it, e.g. providing 4s for the relevant dates in column 5 of the dates file, or if all dates require this custom-built calibration curve:

Bacon(, cc=4, cc4=”mixed.14C”)

3.1 Offsets

In case some or all of your radiocarbon dates are suffering from a reservoir effect, this should be corrected for. You can provide a mean offset dR and associated error dSTD within the Bacon command, for example:

Bacon(, dR=80, dSTD=40)

Alternatively, open the .csv file of the relevant core in Excel or a plain text editor, and add the dR and dSTD values as columns to the right of the existing five columns (assuming that the column specifying the calibration curves has already been added). The layout should be: first the four default ones (lab ID, age, error, depth), then the one specifying the calibration curves (cc, set at 1 for default IntCal09), and then the values for dR and dSTD. These columns should contain the values for the mean reservoir age and its associated 1 standard deviation error, respectively, for each date in the core. Distinct dR or dSTD values can be provided for individual dates. For dates without assumed offsets, enter 0 and 0 respectively. Non-radiocarbon dates are not corrected for any reservoir effect (except if you specify within the Bacon command that cc=0 and specify non-zero values for d.R and d.STD).

3.2 Error sizes

The reported 1 standard deviation errors for radiocarbon and other dates are usually assumed to be correct, accurate and precise estimates of their true errors. However, Christen and Pérez (2010) argue that these reported point estimates of the errors can underestimate – or sometimes overestimate – the true errors, and therefore they suggest to replace these point-wise error estimates by a (student’s-t) distribution that aims to encompass the unknown true error. In effect these extended tails act as an accommodator for ‘outliers’. Even if the dates show a scatter, with the student-t distribution the dates often do fit age-depth models, making the chronological modelling process much more robust against outlying dates. The student’s-t distribution has two parameters, *t.a* and *t.b*, set at 3 and 4 by default (see Christen and Pérez, 2010). If you want to assign more narrow error distributions (more closely resembling the normal distribution), set *t.a* and *t.b* at for example 33 and 34 respectively. For symmetry reasons, *t.a* must always be equal to *t.b-1*. Use normal=TRUE to use the normal/Gaussian distribution.

If you want to assign specific parameter values to specific dates (e.g., setting t.a and t.b to 33 and 34 to avoid long tails for precise/reliable dates, and t.a 3 and t.b 4 to indicate tails for other dates), add two extra columns to the core's dates file (columns 8-9, after the four default columns, the one specifying the calibration curve, and the two columns specifying the reservoir offsets, dR and dSTD). These columns should come after the cc column and the columns for age offsets (see 1.2.1), i.e. the student-t parameter values should be in columns 8 (*t.a*) and 9 (*t.b*). If all dates are thought to have the same student-t distributions, it's probably best to leave these two extra columns out.

4 Installation

Bacon can be downloaded freely from http://chrono.qub.ac.uk/blaauw/bacon.html. Make sure you select the version for your operating system: WinBacon_2.2.zip for Windows (works on 32 and 64 bit systems), MacBacon_2.2.zip for Mac OSX (64 bit only) or LinBacon_2.2.zip for Linux (this version will need to be compiled for your computer; see below).

Unzip the file to an easily accessible place, e.g., straight onto C:\ for Windows. On Windows computers, the Desktop is a less ideal place since it is often quite tedious to find its location on your hard drive (e.g., *C:\Documents and Users\%username%\Desktop* or *C:\Users\%username%\Desktop* where *%username%* is your user name). Users have reported problems when running software from external drives so we suggest to store Bacon onto *C:\* or ~ only. In case you have a previous Bacon version installed, for reasons of safety store the more recent Bacon version somewhere else and copy any of your required cores (folders and their files you've previously added to Bacon/Cores) to the new Bacon/Cores folder.

Different unzipping programs will treat folders within zip-files in different ways. Thus you might end up with a Bacon folder nested within another Bacon folder. You can drag the correct Bacon folder (the one containing the file *Bacon.R* and folders *Cores*, *Curves*, *bin* and *cpp*) to your destination of choice.

Linux users will have to compile the Bacon executables before running Bacon. Make sure you've got g++, gsl and R installed (Fedora: *sudo yum install gcc-c++ gsl gsl-devel R-base R-devel*, Ubuntu: *sudo apt-get install g++ gsl-bin libgsl0-dev R-base R-base-dev*). Then in a terminal go into Bacon/cpp and type *make –f makefileLinux.*

More adventurous Mac/Windows users can also compile Bacon themselves. Windows users will have to install a c/c++ compiler such as MinGW with msys. On Mac OSX, make sure you've installed the Xcode environment (a heavy download of >1GB). Now install the GNU Scientific Library (gsl) by downloading and compiling the source code. Then in a terminal, go to the Bacon/cpp folder and type *make -f makefileX*, replacing *X* with *Windows* or *MaxOSX* as required.

If you run into problems with the installation of Bacon, contact maarten.blaauw@qub.ac.uk or jac@cimat.mx and we will do our best to assist.

Bacon works within R, which can be downloaded freely for your operating system. Visit www.r-project.org, click on CRAN, then choose a mirror near to you, and select the link to download R for either Mac OS X or Windows. See above for Linux users. Bacon has been tested using the currently latest version of R, 3.0.1, and might not work as expected on older R versions. We haven't tested running Bacon through third-party R implementations such as Rstudio.

5 Getting started

It is vital that R is told in which folder it should be working (i.e. where to find all necessary code, calibration curves and core information). There are several ways to do this:
1) open R by clicking on its icon or through a menu, and then change to the correct working directory. It shouldn't matter whether you use the 32 or 64 bit version of R. This can be done by clicking on the *File* drop-down menu on the top-left of the R screen, then on *Change Working Directory...*, and then by selecting the location where the (unzipped) Bacon folder lives. As an alternative you can also type within the R command window, e.g., *setdir("C:\Bacon").*
2) on Mac/Linux open a command-line terminal, go to the Bacon working directory, and open R (e.g., *cd ~/Bacon; R*).
3) Construct a custom icon on your Desktop, telling it to open R in Bacon's working directory. Copy any existing R Desktop shortcut icon, right-click on that icon, select "Properties" and specify the directory under *Start in:*.

Once you've opened R and are in the Bacon directory, load the Bacon.R code by typing within the R command window:

```
source('Bacon.R')                    <press enter after each command>
```

If all is fine, Bacon will greet you. If instead R complains that it cannot open that file, ensure that you're in the right directory, and that you've typed everything correctly.

Now produce a Bacon age-model for the default core MSB2K accepting all default settings:

```
Bacon()
```

Answer *y* to the question about running MSB2K with 21 sections (you might need to click on the command window to make it active again before typing), and wait for the run to be performed and the results to appear. A graph with four panels will be plotted (Fig. 3). Also provided are the mean, minimum and maximum 95% confidence ranges (based on quantiles, here 2.5% and 97.5%). Within the core's folder, Bacon/Cores/MSB2K, a text file ending in "_ages.txt" will have been produced containing the 95% age ranges, median and weighted mean for each cm. In the same folder you will find a pdf of the graph, and text files MSB2K_settings.txt describing the applied settings, MSB2K_21.out with the parameter values that make up the age-model (each line contains an MCMC iteration), and MSB2K_21.bacon that contains the chosen commands to produce the age-model.

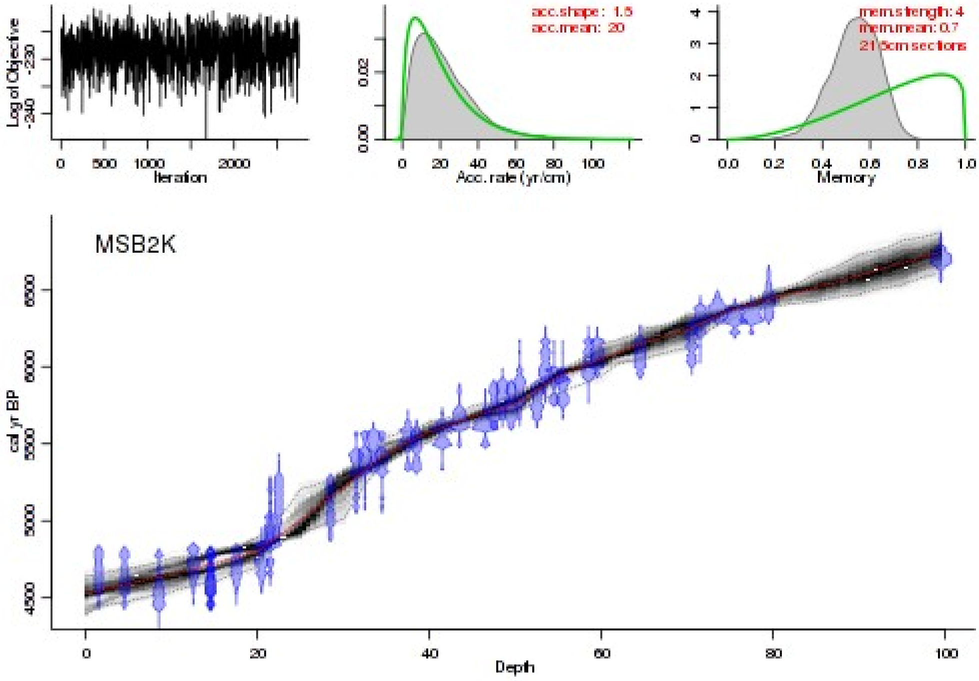

Fig. 3. Bacon output graph. Upper panels depict the MCMC iterations (left; good runs show a stationary distribution with little structure among neighbouring iterations), the prior (green curves) and posterior (grey histograms) distributions for the accumulation rate (middle panel) and memory (right panel). Bottom panel shows the calibrated ^{14}C dates (transparent blue) and the age-depth model (darker greys indicate more likely calendar ages; grey stippled lines show 95% confidence intervals; red curve shows single 'best' model based on the weighted mean age for each depth).

6 User-relevant Bacon functions

As introduced above, the *Bacon()* function will produce an age-depth model from a dated core. This section will introduce the most user-relevant Bacon functions, their uses and options. The *Bacon()* function itself has many options outlined below.

6.1 Running Bacon age-depth models

Bacon()'s first option, *core* (default “MSB2K”), specifies which core is going to be modelled. The command *Cores()* shows which cores are present within Bacon's Cores folder. New cores can be added to this Cores folder as explained in section 3. Most cores will only require four columns separated by columns (lab ID, Age, error, depth). However if your core's dates require a mixture of calibration curves, or have variable age offsets or student-t distributions, make sure you specify these as additional columns within the .csv file: calibration curve (5th column *cc*, values 0-4), age offsets (6th and 7th columns, *d.R*, *d.STD*), or any non-default settings for the student-t distribution (8th and 9th columns, *t.a*, *t.b*).

Bacon will divide the core into numerous sections of equal thickness specified by *thick* (default 5), from the uppermost to the lowermost dated depth, and model accumulation rates for each of these sections. Alternative upper (*d.min*) and lower (*d.max*) depths can be provided. If these alternative depths lie within the range of dated depths, all dated depths will be used but an age-model provided only within the specified depth ranges. If provided *d.min* or *d.max* values lie outside the dated depths, this will require extrapolation (a dangerous excursion into unknown territory). For long cores or small values of *thick*, very many parameters will need to be estimated. This will take a long time to calculate and might result in 'unresponsive' age-depth models since the model becomes over-parametrized. On the other hand, too few sections will result in age-models with unrealistically abrupt 'elbows'. By default (*suggest*=TRUE), Bacon will warn you if the number of sections lies outside the safe range (default between 10 and 200 sections; *reswarn*=c(10,200)) and suggest an alternative value. Accept this change by typing *y*, or typing *n* to continue with the old value.

The default *unit* is 'cm', and by default ages are calculated for each cm between *d.min* and *d.max,* and stored as files ending in '_ages.txt' within the core's folder. Alternative depth intervals can be provided using, e.g., *d.by*=0.5. This can be handy especially for very long cores, where calculating ages for every cm would take a long time. If more than *maxcalc* (default 500) depths will need to be calculated, Bacon will warn you and suggest an alternative value for *d.by* that will take less time to calculate. Instead of regularly spaced depths, you can also provide a file containing the depths for which you require ages. This file, containing the depths in a single column without header, should be stored within the core's folder, and the file name should start with the core's name and end with '_depths.txt' (see Bacon/Cores/MSB2K/MSB2K_depths.txt for an example). Then within the Bacon command specify that *depths.file*=TRUE (default FALSE).

As outlined in section 2, specifying prior information is essential for producing Bacon age-depth models. Prior distributions need to be set for the accumulation rate, the memory (dependence of accumulation rate between neighbouring depths), and, if hiatuses are present, for their expected sizes. The prior for accumulation rate is set by a gamma distribution with shape *acc.shape*=1.5 (default) and mean *acc.mean*=20 (default). However if initial ballpark estimates of accumulation rates differ much from the provided or default prior acc.mean, Bacon will suggest to change the prior. Accept this suggested alternative prior by typing *y*, or leave as is by typing *n*. To get rid of these suggestions, use *suggest*=FALSE. The memory prior is a beta distribution with default parameters *mem.strength*=4 and *mem.mean*=0.7. Depths of any hiatuses can be provided by, e.g., *hiatus.depths*=20. Default parameters for the gamma distribution describing hiatus length are *hiatus.shape*=1 and *hiatus.mean*=1000.

The default calibration curve for ^{14}C dates is IntCal09 (*cc*=1). Setting cc to 2 will calibrate using Marine13, and cc=3 will calibrate using SHCal13. For cores with postbomb dates (negative ^{14}C ages), you will need to specify a postbomb curve (no default; *postbomb*=0); see section 3. Core-wide age offsets can be specified within the Bacon command with *d.R* (default 0) and *d.STD* (default 0), although it often makes more sense to provide such offsets within the core's dates file (see above). By default (*normal*=FALSE) we use the student-t distribution for all dates. Default parameter values are *t.a*=3 and *t.b*=4. Again, it might make more sense to specify alternative student-t values within the core's dates file. By default, values in the dates files are separated by commas (*sep*=',') while *dec*='.' are the default decimal points .

Bacon works in cal BP and uses this time scale for graphs and files unless told to use the BC/AD scale (*BCAD*=TRUE). Default probability intervals are 95% (*prob*=.95). Ages are rounded to single years by default (*round*=0).

By default Bacon will ask you to confirm that you want to run the core with the provided settings. Disable this (*ask*=FALSE) for batch runs. Alternatively, if you want to load an existing Bacon run instead of producing a new one, use *run*=FALSE (defaults to TRUE). Bacon will try to remember which settings you've applied to your cores (default *remember*=TRUE). If you run into inconsistencies or other problems, try running your core in Bacon again with remember=FALSE, or, start cleanly by typing *Bacon.cleanup()*. Bacon cleans up temporary files by default (cleanup=TRUE), but you might want to avoid this if you want to inspect the files after a run.

Generally Bacon will perform millions of MCMC iterations for each age-model run, although only a fraction of these will be stored. In most cases the remaining MCMC iterations will be well mixed (the upper left panel of the fit of the iterations shows no strange features such as sudden systematic drops or rises). However if the iterations seem not well mixed, or if too few remain (say less than 1,000), then it would be wise to run the core again but with a larger sample size (e.g., *ssize*=8000, default 2000). Alternatively, if the run has a visible remaining burn-in, this can be removed with the command *scissors()*. To remove, e.g., the first 300 iterations, type *scissors(300).* The remaining iterations can also be 'thinned', by randomly removing a proportion. For example to remove 10% of the iterations, type *thinner(0.1)*. After running either the scissors or thinner command, plot the age-model again using the command *agedepth().*This will also update the pdf version of the graph in your core's folder (unless you specify *plot.pdf*=FALSE).

Internal Bacon options that should not normally require interaction or adaptation by users include *th0* (starting years for the MCMC iterations), *burnin* (amount of initial, likely sub-optimal MCMC iterations that will be removed), *MinYr* and *MaxYr* (age limits for Bacon runs, default at safe limits of -1000 resp. 1,000,000 cal BP), *find.round* (temporary measure to deal with rounding problems at hiatuses; default 4), *bins* (number of bins for histograms; no default), and *cutoff* (avoid plotting very low probabilities of dates).

6.2 Producing age-depth graphs

There are many options to steer the look of the main age-depth plot (either within the *Bacon()* command, or after a run through *agedepth()*). The default of plotting depth on the horizontal axis and age on the vertical one can be changed with *rotate.axes*=TRUE. The directions of the depth and age axes can be reversed using *rev.d*=TRUE resp. *rev.yr*=TRUE. Alternative limits for the calendar axis can be specified with *yr.min* and *yr.max* (within the agedepth() command alternatives for *d.min* and *d.max* can also be provided). By default the core's name is plotted in the topleft corner of the main panel. Alternatives can be provided, e.g., to leave empty provide *plot.title*=' ', and to change its location provide, eg., *title.location*=”topright”. The labels for the depth (default *d.lab*='Depth') and calendar (default *yr.lab*='cal BP' or 'BC/AD' if BCAD=TRUE) can be changed similarly. By default panels showing the MCMC iterations and the priors and posteriors for accumulation rate and memory are plotted above the main age-depth model panel. This can be avoided by supplying *model.only*=TRUE.

By default the age-model is plotted by calculating *d.res*=200 and *yr.res*=200 greyscale pixels, using *grey.res*=100 shades. The darkest grey value is *dark*=1 by default; lower values will result in lighter grey but values >1 are not allowed. Date distributions are plotted using by default *date.res*=100 points. By default, more precise dates will cover less time and will thus peak higher than less precise dates. This can be avoided by specifying *normalise.dists*=FALSE. The widths of the date distributions are set by *width* (default *width*=15).

The default colours of the calibrated ^{14}C dates (transparent dark blue borders and fills; *C14.border*=rgb(0, 0, 1, 0.5) and *C14.col*=rgb(0, 0, 1, 0.35)) and other dates (transparent light blue borders and fills; *cal.col*=rgb(0, 0.5, 0.5, 0.35), *cal.border*=rgb(0, 0.5, 0.5, 0.5)). The confidence ranges of the age-model are indicated with, by default, dark grey stippled curves (*range.col*=grey(0.5), *range.lty*='12'). Turn these off by supplying *plot.range*=FALSE. By default the weighted mean age-depth model is drawn using a red stippled line (*wm.col*="red", *wm.lty*="12"), whereas the alternative median model is left unplotted by default (*med.col*=NA, *med.lty*="12"). Finally there are options to set the amount of space for margins (default *mar*=c(3, 3, 1, 1)) and graph labels (default *mgp*=c(1.5, 0.7, 0)), and the drawing of boxes around plots (default L-shaped, *bty*='l').

Age estimates of specific depths can also be plotted, using the function *Bacon.hist()*. As first option within this command, one or more depths should be provided. If only one depth is given, a plot will be made (unless you specify *Plot*=FALSE) and an age range reported (default 95% ranges, *prob*=0.95), e.g.:

Bacon.hist (20)

Alternative values can be provided for the calendar axis limits (*xlim*) or label (*xlab*), vertical axis limits (*ylim*) or label (*ylab*). Alternative colours can be given to the histogram (defaults *hist.col*=grey(0.5) and *hist.border*=grey(0.2)), confidence ranges, weighted mean and median (defaults *range.co*l=”blue”, *wmean.col*=”red”, *med.col*=”green”, respectively). Bacon will try to automatically estimate reasonable bin sizes, however changing the histograms bin size could alter the shape of the histogram considerably. Try for example the difference between *bins*=10 and *bins*=100.

Temporary files used to produce the plot are removed automatically after the plot has been made unless *cleanup*=TRUE. Age ranges are given as quantiles (e.g., 2.5% and 97.5% for the default of 95% confidence limits).

6.3 Post-run analyses

Obtaining an age-depth model is often only a step towards a goal, e.g., plotting a core's proxies against calendar time. Bacon has several functions to help you there. One potentially useful function is *Bacon.Age.d()*, which outputs all MCMC-derived ages for a given depth. These ages will be in cal BP, except if BCAD has been set to TRUE during the Bacon run or within the *Bacon.Age.d* function. You can then capture these ages into a new variable, e.g.:

```
ages.d20 = Bacon.Age.d(20, BCAD=TRUE)
```

Proxies analysed along the depths of a core can be plotted as 'proxy-ghost' graphs against calendar time while taking into account chronological uncertainties. Here darker grey indicates more likely calendar ages for specific proxy values (Blaauw et al. 2007). For this to work, place a csv file with the values of proxies against depth within your core's folder. The values should be in columns separated by commas (default *sep*=”,”), the first column containing the depths and the first line containing the proxy names. The file name should start with the core's name and end with “_proxies.csv”. For an example see Bacon/Cores/MSB2K/MSB2K_proxies.csv.

The first option within the *proxy.ghost()* function tells which proxy to use (counting from the column number in the .csv file after the depths column). By default, greyscale pixels are calculated for *proxy.res*=200 equally spaced proxy values, and also for *yr.res*=200 equally spaced time bins. The default is to use *grey.res*=100 grey shades. Bin sizes to calculate the grey-scales are calculated automatically (*bins*). The darkest grey value is *dark*=1 by default; lower values will result in lighter grey but values >1 are not allowed. The default is to plot the calendar horizontally, however this can be rotated (*rotate.axes*=TRUE). The axes can also be reversed (*rev.proxy*, *rev.yr*) and their limits adapted (*proxy.lim*, *yr.lim*). Axis labels (*proxy.lab, yr.lab*) can be changed from their defaults as well. The ages will be given in cal BP, except if BCAD has been set to TRUE during the Bacon run or within the *Bacon.Age.d* function. You can optionally add the weighted mean ages of the proxy values (*plot.wm*=TRUE, default colour *wm.col*=”red”). By default an L-shaped box will be drawn around the graph (*draw.box*=”l”). Temporary files to produce the greyscales are removed automatically after the plot has been produced (*cleanup*=TRUE).

Considering accumulation rates is crucial for age-depth modelling, and even more so if subsequently used for calculating proxy influx values, or interpreted as proxy for environmental change such as carbon accumulation. Bacon deals explicitly with accumulation rate and its variability through defining prior distributions. Functions are provided to obtain accumulation rates (in years per cm, so actually sedimentation times) as estimated by the MCMC iterations for any depth (*accrate.depth()*) or age of a core (*accrate.age()*). Have a look at the following examples:

```
accrate.d20 = accrate.depth(20)
accrate.a5000 = accrate.age(5000)
plot(accrate.a5000, type=”.”)
hist(accrate.a5000)
```

A core’s accumulation rates can be plotted with *plot.accrate.depth()* and *plot.accrate.age()*. The darkest grey value is *dark*=1 by default; lower values will result in lighter grey but values >1 are not advised. The default is to plot the accumulation rates horizontally and the depth vertically (*rotate.axes*=FALSE). The direction of the axes can be reversed from the defaults (*rev.d*=TRUE, *rev.acc*=TRUE). Calculating accumulation rates against calendar age will take some time to calculate, and might show unexpected rates around the core’s maximum ages (only a few of all age-model iterations will reach such ages and they will tend to have modelled accumulation rates for the lower depths much lower than the other iterations). Axis limits for accumulation rates are estimated automatically, however upper limits can be very variable (and thus hard to predict) if calculated in cm/yr; therefore you might want to manually adapt the axis limits after plotting with default settings (e.g., *acc.lim*=c(0,1)). Ages will be given in cal BP, unless BCAD=TRUE has been set within the *Bacon()* or *plot.accrate.age()* functions.

Flux rates (e.g. pollen grains/cm^2/yr) for proxies can also be plotted as greyscales. For this to work, provide a plain text file with headers and the data in columns separated by commas, ending in ‘_flux.csv’ and saved in your core’s folder. The first column should contain the depths, and the next columns should contain the proxy concentration values (leaving missing values empty). See Bacon/Cores/MSB2K/MSB2K_flux.csv for an example. Then type for example *flux.age(1)* to plot the flux values for the first proxy in the .csv file. Instead of using a _flux.csv file, you can also define a flux variable within the R session (consisting of depths and their proxy concentrations in two columns) and provide the name of this variable, e.g.: *flux.age(flux=flux1).* As for plotting, *flux.age* has similar options to the *plot.accrate* functions outlined above.

Probabilities of 'events' in an age-modelled core can be plotted against time, taking into account chronological uncertainties (Blaauw et al. 2007). Such events could be for example core depths at which proxies indicate changes toward wetter local conditions. This can be expressed as values between 0 (no event) and 1 (event at 100% probability) for each depth. Blaauw et al. 2010 propose to estimate probabilities of events by finding specific proxy features such as rising curves. Probabilities are then estimated through resampling, where low to modest rises of proxy curves result in low event probabilities, and clear proxy rises in high probabilities. Probabilities are calculated for specific *time windows*, and the user should specify the window width and the step size with which the window moves. The resulting probabilities are plotted and saved within the core's folder (file names ending with the window width and "_probs.txt"). The age limits between which the window will move (*min.age* and *max.age*) are calculated from the age-model by default. To calculate the event probabilities, produce a file with two columns (depth and corresponding proxy-derived probabilities, separated by white spaces). Provide headers at the file's first line, and save with extension "_events.txt". See Bacon/Cores/MSB2K/MSB2K_events.txt for an example. Then within R type, for example:

AgesOfEvents(100, 10)

to calculate event probabilities at 100 year window widths, with the windows moving at 10 year steps from the core's bottom ages to its top age. Any alternatives to *min.age* or *max.age* should be given in cal BP, even if BCAD is set to TRUE. *AgesOfEvents* has similar plotting options to the *plot.accrate* functions outlined above.

6.4 Reloading existing runs

Bacon age-model runs are stored within the core's folder, and will remain in R's memory while the session lasts. This information is stored within the session as a a variable called *info*. For example, type

info$ranges

to see the modelled age ranges, median and weighted mean ages for each depth of the currently loaded core. However previous Bacon runs and their specific settings can be reloaded into an R session, for example in order to produce updated plots or additional analysis after you've closed an R session. One way to do so is to type, e.g.,

Bacon("CoreA", run=FALSE)

which will plot the core's existing age-model, and load the settings and results into R's memory as variable *info*. The command *Cores()* shows which cores are present within Bacon's Cores folder.

7 Version history and future developments

Major new features in Bacon 2.2

Uses the latest IntCal13 and postbomb calibration curves (Hogg et al. 2013, Hua et al. 2013, Reimer et al. 2013)

The default prior for the accumulation rate has been changed to acc.mean=20 (yr/cm) and acc.shape=1.5 (based on Goring et al. 2012; previous values were 10 and 2 respectively)

Bacon now suggests values for accumulation rate priors and section thickness if initial values differ from the defaults

Many bug fixes in the Bacon.R and underlying C/C++ codes

Planned new features

Ages for depths close to hiatuses are not currently calculated in an optimal way; we plan to enhance this

Currently Bacon works with ^{14}C dates on the ^{14}C scale, however we plan to change this to the more linear fractional scale F^{14}C (as do, e.g., clam, OxCal and Calib)

Develop a feature to include information regarding known synchronous markers within Bacon age-models (BLT)

Include tutorial and FAQ sections within this manual

8 References

Blaauw, M., Heuvelink, G.B.M., Mauquoy, D., van der Plicht, J., van Geel, B., 2003. A numerical approach to ^{14}C wiggle-match dating of organic deposits: best fits and confidence intervals. Quaternary Science Reviews 22, 1485-1500

Blaauw, M., Christen, J.A., 2005. Radiocarbon peat chronologies and environmental change. *Applied Statistics* 54, 805-816

Blaauw, M., Christen, J.A., Mauquoy, D., van der Plicht, J., Bennett, K.D., 2007. Testing the timing of radiocarbon-dated events between proxy archives. *The Holocene* 17, 283-288

Blaauw, M., 2010. Methods and code for 'classical' age-modelling of radiocarbon sequences. *Quaternary Geochronology* 5, 512-518

Blaauw, M., Wohlfarth, B., Christen, J.A., Ampel, L., Veres, D., Hughen, K.A., Preusser, F., Svensson, A., 2010. Were last glacial climate events simultaneous between Greenland and France? A quantitative comparison using non-tuned chronologies. *Journal of Quaternary Science* 25, 387-394

Blaauw, M., Christen, J.A., 2011. Flexible paleoclimate age-depth models using an autoregressive gamma process. *Bayesian Analysis* 6, 457-474

Christen, J.A., Pérez E., S., 2010. A new robust statistical model for radiocarbon data. *Radiocarbon* 51, 1047-1059

Goring, S., Williams, J.W., Blois, J.L., Jackson, S.T., Paciorek, C.J., Booth, R.K., Marlon, J.R., Blaauw, M., Christen, J.A., 2012. Deposition times in the northeastern United States during the Holocene: establishing valid priors for Bayesian age models. *Quaternary Science Reviews* 48, 54-60

Hogg, A.G., Hua, Q., Blackwell, P.G., Buck, C.E., Guilderson, T.P., Heaton, T.J., Niu, M., Palmer, J., Reimer, P.J., Reimer, R., Turney, C.S.M., Zimmerman, S.R.H., 2013. ShCal13 Southern Hemisphere calibration, 0-50,000 cal yr BP. *Radiocarbon* 55(4), doi:10.2458/azu_js_rc.55.16783

Hua, Q., Barbetti, M., Rakowski, A.Z., 2013. Atmospheric radiocarbon for the period 1950-2010. *Radiocarbon* 55(4), doi:10.2458/azu_js_rc.v55i2.16177

R Development Core Team, 2013. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.r-project.org

Reimer, P.J., Bard, E., Bayliss, A., Beck, J.W., Blackwell, P.G., Bronk Ramsey, C., Buck, C.E., Edwards, R.L., Friedrich, M., Grootes, P.M., Guilderson, T.P., Haflidason, H., Hajdas, I., Hatté, C., Heaton, T.J., Hoffmann, D.L., Hogg, A.G., Hughen, K.A., Kaiser, K.F., Kromer, B., Manning, S.W., Niu, M., Reimer, R.W., Richards, D.A., Scott, M.E., Southon, J.R., Turney, C.S.M., van der Plicht, J., 2013. IntCal13 and Marine13 radiocarbon age calibration curves 0-50,000 yr cal BP. *Radiocarbon* 55(4), 1869-1887