



HAL
open science

ChronoModel version 2.0: Bayesian Statistics Applied to Chronological Modelling

Philippe Lanos

► **To cite this version:**

Philippe Lanos. ChronoModel version 2.0: Bayesian Statistics Applied to Chronological Modelling. 2024. hal-04574542

HAL Id: hal-04574542

<https://hal.science/hal-04574542>

Preprint submitted on 14 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

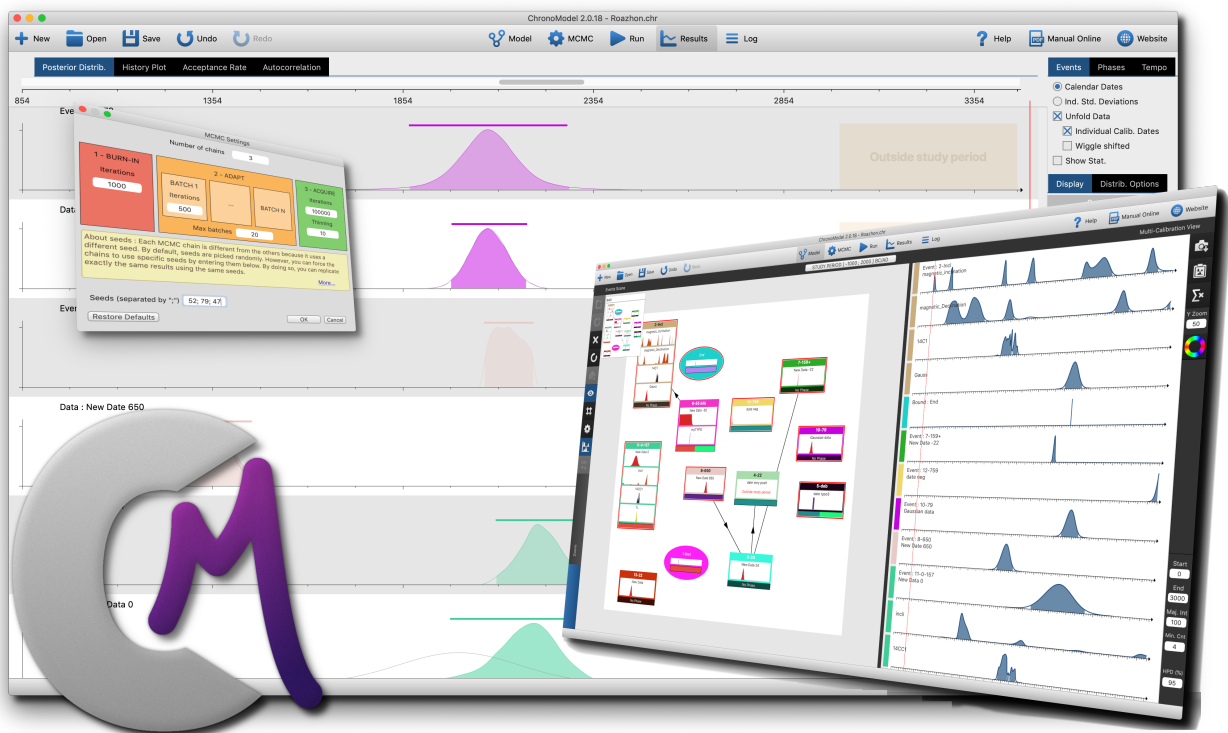


Distributed under a Creative Commons Attribution 4.0 International License

ChronoModel

version 2.0

Bayesian Statistics Applied to Chronological Modelling



Author

Philippe Lanos, CNRS, philippe.lanos@univ-rennes1.fr

Manual version : 1

May 14, 2024

Copyright © 2024 CNRS

Creative Commons License Notice: This book is licensed under **Creative Commons Attribution 4.0 International License**.

<https://creativecommons.org/licenses/by/4.0/>



You are free to

- **Share** — copy and redistribute the material in any medium or format
- **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Contents

1	INTRODUCTION TO CHRONOMODEL	4
1.1	Welcome to ChronoModel	4
1.2	Development context	5
1.3	CM citation	5
1.4	Funding	6
1.5	Contacts	6
2	TEMPORAL CONCEPTS USED IN CHRONOMODEL	6
3	CREATING PROJECTS	7
3.1	Project manager	7
3.2	Creating/Opening a project - Defining the study period	8
3.3	Creating an event	9
3.3.1	<i>Adding measurements into an event</i>	10
3.3.1.1)	<i>Data management</i>	10
3.3.1.2)	<i>Plugins</i>	10
3.3.1.3)	<i>Reference curves</i>	10
3.3.1.4)	<i>Advanced options</i>	11
3.3.2	<i>Calibration process</i>	11
3.3.2.1)	<i>individual calibration process</i>	11
3.3.2.2)	<i>Special case: calibration from multiple radiocarbon measurements</i>	12
3.4	Creating Bounds, TPQ, TAQ	13
3.5	Creating phases	13
3.6	Creating temporal constraints	14
4	MCMC CALCULATION	14
4.1	Markov chain Monte Carlo	14
4.2	Building a Markov chain in multiple dimensions: the Gibbs sampler	14
4.3	Drawing from full conditional posterior distributions	15
4.3.1	<i>Full conditional posterior distribution of the event θ</i>	15
4.3.2	<i>Full conditional posterior distribution of calibrated dates t_i</i>	15
4.3.3	<i>Full conditional posterior distribution of the variance of a calibrated date σ_t^2</i>	16
4.3.4	<i>Full conditional posterior distribution of a wiggle matching δ_i</i>	16
4.3.5	<i>Full conditional posterior distribution of a bound θ^B</i>	16
4.4	Settings	16
4.5	Methods	17
5	RESULTS PRESENTATION	18
5.1	Checking the Markov chains	18
5.1.1	<i>History plot</i>	18
5.1.2	<i>Acceptance rate</i>	19
5.1.3	<i>Correlation between successive values</i>	19

5.2	Marginal posterior densities	19
5.3	Checking the Markov chains	21
5.3.1	<i>Is the equilibrium reached ? Look at the history plots.</i>	22
5.3.2	<i>Correlation between successive values ? Look at the autocorrelation functions.</i>	23
5.3.3	<i>Look at the acceptance rates.</i>	24
5.4	Statistical results	24
5.4.1	<i>Credibility interval (CI)</i>	25
5.4.2	<i>Highest posterior density (HPD) region</i>	25
5.5	Duration, Tempo-Plot, Activity	25
5.6	LOG, saving, summary, post-treatment	25
6	TERMS AND CONDITIONS	26
6.1	Disclaimer	26
6.2	Copyright notice	26

1 Introduction to ChronoModel

1.1 Welcome to ChronoModel

ChronoModel (CM) is a user-friendly Bayesian chronological modeling tool for interpreting dating data along with expert chronological information. The software is freeware, open source and cross-platform (Mac, Windows, Linux). It presents a user-friendly interface where the data are handled using intuitive graphical elements (thumbnails). These elements explicitly show the data and all the prior information introduced in the model. We recall that the translation of the archaeologist's ideas and knowledge into a chronological model have to be clearly justified, especially when publishing the results.

Bayesian statistical models and methods implemented in CM are documented in peer-reviewed literature (see references). The user guide 1.5 provides an introduction to the software and a detailed description of how to use it.



Figure 1: CM Web site: <http://www.chronomodel.com/>

CM can be freely downloaded from the web site <https://www.chronomodel.com>. The pre-compiled binaries of the latest version of the application are :

Chronomodel 2.0 for Mac OS X (February 2019): Supported : 10.7 (Lion), 10.8 (Mountain Lion), 10.9 (Mavericks), 10.10 (Yosemite), 10.11 (El Capitan)

Chronomodel 2.0 for Windows 32 bits (February 2019): Supported : Windows 7, Windows 8, Windows 10

Chronomodel 2.0 for Linux (February 2019): Supported : Ubuntu v14 (Linux)

You can download the source code and compile everything yourself from scratch. It is built on Qt5 (Qt is a C++ toolkit for cross-platform application development) and uses the FFTW library (<https://www.fftw.org/>). The only pre-requisite to build it yourself is to have Qt5 installed on your system. The project is hosted on GitHub.com. You can clone the repository by typing : `git clone https://github.com/Chronomodel/chronomodel.git`.

1.2 Development context

In the 2000s, the “Rennes” team of archaeomagnetic dating (CNRS, University of Rennes 1 and University of Bordeaux-Montaigne) began to develop a piece of software for converting the inclination, declination or intensity determinations of the Earth’s magnetic field into calendar dates using geomagnetic calibration curves (Lanos, 2004). This software, based on a Bayesian approach and written in Pascal/Delphi language for Windows, was first called RenCMT, then RenDate (2004), and finally RenDateModel (2008), this last version including chronological modeling functions.

At the same time, the team of palaeomagnetism of the “Institut de Physique du Globe” at Paris (IPGP, France) developed a non Bayesian technique for calculating archaeomagnetic dates, based on directional statistics of McFadden (Le Goff et al., 2002). More recently, following the ideas of Lanos (2004) implemented in RenDate software, the team of the Palaeomagnetic Laboratory at the Complutense University of Madrid (Spain) developed an application on MatLab for calculating archaeomagnetic dates (Pavón-Carrasco et al., 2011).

In addition to the aspect of calculating an archaeomagnetic date, the RenDateModel software also helped to construct chronological models which take into account relative chronology (based on stratigraphy or evolution criteria). For this aim, a new statistical concept for estimating the date of a target event has been developed by introducing an individual error on each chronometric date (Sapin et al., 2008; Cassen et al., 2009; Lanos and Dufresne, 2012). This approach is quite different from the one implemented in BCal or OxCal software and now constitutes the basic idea at the core of CM software. Thus, the CM project arises from ten years of collaborative research with archaeologists and experienced a new development, from 2012, thanks to the collaboration with the Laboratory of Mathematics “Jean-Leray” of the University of Nantes.

1.3 CM citation

The CM team of version 2.0 comprises Philippe Lanos, Director of the project (CNRS, IRAMAT-CRPAA, University of Bordeaux-Montaigne and University of Rennes 1), Philippe Dufresne (CNRS, IRAMAT-CRPAA, University of Bordeaux-Montaigne and University of Rennes 1) and the contributors who have tested ChronoModel v2.0 : William BANKS, Juan Antonio BARCELÓ ÁLVAREZ, Pascal BERTRAN, Didier BINDER, Jacques-Elie BROCHIER, Sandrine CABUT, Thomas DYE, Gwenaél HERVÉ, Jean-Claude LEFÈVRE, Christine OBERLIN, Christophe VASCHALDE.

When writing up results arising from CM 2.0, please cite our work by giving the URL for the software: <https://chronomodel.com/> with the reference: Lanos Ph. and Dufresne Ph. (2019). ‘ChronoModel version 2.0: Software for Chronological Modelling of Archaeological Data using Bayesian Statistics’. Available at <https://chronomodel.com/Documentation> on ChronoModel software:

- Lanos, Ph. and Dufresne, Ph. (2019). ChronoModel version 2.0, User manual : <https://chronomodel.com/download-chronomodel-software-mac-windows>
- Vibet, M.-A., Philippe, A., Lanos, Ph. and Dufresne, P. (2016). User manual of ChronoModel V1.5.
- Vibet, M.A. , Philippe, A. , Lanos, Ph. and Dufresne Ph. (2015). ChronoModel User’s manual, version 1.1.

Papers associated with the software:

- Ph. Lanos and A. Philippe (2017) Hierarchical Bayesian modeling for combining Dates in archaeological context. *Journal de la Société Française de Statistique*, Vol. 158 (2) p. 72-88. <http://journal-sfds.fr/article/view/621>.
- Ph. Lanos and A. Philippe (2018). Event date model: a robust Bayesian tool for chronology building. *Communications for Statistical Applications and Methods (CSAM)*, Vol. 25, No. 2, 131–157. <https://doi.org/10.29220/CSAM.2018>

1.4 Funding

CM has been funded by the French National Research Agency (ANR), project ChronoModel, ANR 11 MONU 007, 2011-2015. Since this date, the software is maintained by the Rennes team : Philippe Lanos and Philippe Dufresne (CNRS, Archeosciences Bordeaux laboratory, University of Bordeaux-Montaigne and University of Rennes).

1.5 Contacts

For further information on the software, send an email to : support@chronomodel.com

2 Temporal concepts used in ChronoModel

ChronoModel (CM) is based on important temporal concepts that are at the core of the software.

Event date: CM is based on the concept of unknown Event date which is defined by using a hierarchical Bayesian statistical model. This parameter represents the unknown calendar date of a temporal event within a chronology. In CM, different types of measurement may be combined in order to estimate the date of the event of interest. Assuming that this event can reliably be associated with one or several suitable samples, out of which measurements can be made and calendar dates can be obtained, the Event model combines these calendar dates, assumed to be contemporaneous, in order to estimate the unknown target calendar date. The measurements may be ^{14}C ages in radiocarbon, ages in luminescence, inclination, declination or intensity of the geomagnetic field in archeomagnetism, typo-chronological references or any Gaussian measurements.

Group of events = Phases: In CM, a Phase is a group, a collection of archaeologically coherent events defined on the basis of archaeological, historical, geological, environmental or other criteria we want to locate in time. Unlike the event date model, the Phase in CM does not respond to a statistical model: indeed, we do not know how Events can a priori be distributed in a phase. However, we can estimate the beginning, end or duration of a phase from the Events that are included in it.

Prior temporal information: The aim is to improve our knowledge about the event date parameters using Bayesian modeling. For this, it is necessary to provide as much information as we can: Events and/or Phases may be linked by temporal order relationships. These order relationships can be defined in different ways: by the stratigraphic sequence (physical relationship observed in the field) or by criteria of stylistic, technical, architectural, etc. development which may a priori be known. All these relationships operate directly onto the target event dates. As a consequence, in a global modeling project, different phasing systems (multiphasing) can be defined on the basis of different criteria and so a phasing system can intersect an other phasing system: for example ceramic phasing can intersect lithic phasing in the sense that some target events can belong to two or several phases. Such a constraints network, when available, can significantly contribute to improving the precision of the estimates.

Events in a phase may also be constrained by a known duration or by a known hiatus between two phases (this last constraint assumes a temporal order between two groups of Events).

Moreover, a "Bound", such as a historical date or a so-called *terminus post quem* (TPQ) or *terminus ante quem* (TAQ), may be introduced in order to constrain one or several Event dates.

Unlike BCal and OxCal which have additional hierarchical parameters called boundary parameters, we do not introduce such prior information in CM modeling (see the argumentation in "Event date model...", sections 3 and 4).

Outliers: There is no “outlier” detection in CM, unlike the BCal and OxCal software. Thus, we do not consider in principle that some specific samples can be outliers because, for instance, they are suspected to be contaminated or they do not precisely date the event that they are stratigraphically associated with. In fact, we consider that all the dated Events related to the target Event can a priori be disturbed owing to unknown error sources. This disturbance is modeled by individual error parameters within the Bayesian hierarchical framework of the event date model (see article "Event date model...", section 2). However, note that the posterior distribution of the error parameter can provide us with a way to assess the outlying degree of a sample.

Terminology: The “measurement” value (chronometric data) in CM is equivalent to the determination term and that the unknown calendar date of the target event corresponds to the “determination parameter” term used in BCal software (Litton and Buck, 1995; Buck et al., 1996). Moreover, the “floating parameter” term defined in BCal can correspond to the “Gauss” dating or to the Bounds in CM, in the sense that there is no antecedent measurement, i.e. no calibration process.

3 Creating projects

The basic concept of the modeling interface in CM is to represent event dates, groups of phases and prior temporal information through graphical items which follow the Harris matrix (Harris, 1989). This intuitive approach of chronological modeling, already implemented and experienced in RenDateModel software, allows both a quick implementation of a model and an easy way to explore the data and their relationships. For practical reasons and unlike the RenDateModel software, here we separated models into two scenes: the Events scene where Event dates and stratigraphic constraints are entered, and the Phases scene where groups of Event dates and temporal constraints are entered. This separation is motivated by the fact that event dates may belong to several phases. This also turns out to be very useful for managing stratigraphic constraints between groups having numerous event dates (this allows to avoid a forest of stratigraphic arrows in the Events scene!).

3.1 Project manager

The main page of the project manager appears immediately when the software is launched and allows you to manage one project at a time. This tool provides the functionality to create and edit projects, calibrate measurements, calculate posterior densities with MCMC techniques and view the results of the current project.

The icons contained in the top of the window are:

On the left hand side: three icons to help manage projects and two icons to help manage modeling actions.

- Create a new project.
- Open a pre-existing project. ChronoModel projects are saved using the extension file *.CHR
- Save the current project.
- Undo the last actions.
- Redo the last actions.

In the middle: the icons refer to the different steps at the core of the modeling with ChronoModel.

- Design the model.
- Open the MCMC settings window.

- Run the Bayesian modeling.
- Show the results of the modeling.
- Show summaries of the MCMC methods used for this modeling and its results.

On the right hand side:

- Able or disable the context-sensitive tips seen in yellow bubbles.
- Reach the user manual on the web site.
- Open the ChronoModel website.

Options valid for all the current project can be set up in the *Settings* of the application.

- Choice of the language (English, for now. Any language can be added)
- Automatic loading/saving options
- CSV format options: CM reads a CSV file using options defined in the regional parameters of the operating system (in English: a comma is a cell separator and a dot is a decimal separator).
- Image resolution options
- Date format options: results may be seen in different date formats. We can choose:
 - BC/AD : Before Christ/Anno Domini
 - Age cal. BP = 1950 - date in BC/AD
 - Date cal. BP = date in BC/AD - 1950
 - Age cal. B2K = 2000 - date in BC/AD
 - Date cal. B2K = date in BC/AD - 2000
- Possibility to add/delete reference curves for *GAUSS*, *14C* and *AM* dating methods

3.2 Creating/Opening a project - Defining the study period

After having launched ChronoModel, the interface in Figure 2 appears

To open a pre-existing project, use the *Open* button and then choose the project. ChronoModel projects are saved with the extension “.chr”. Otherwise, in order to create a new project, click on the *New* button. This action opens a new window asking to name this new project and to save it in a chosen directory.

Then a dialogue box appears asking to define the prior study period: the period under which one assumes that the unknown calendar dates of the events are likely to be, from external information such as the historical or archaeological context. The study period should be given in the date format BC/AD as the Bayesian modelling is computed in this date format.

The project window comprises two frames (see Figure 2). The left frame is dedicated to event modeling, called the Events scene, where stratigraphic constraints can be inserted in the form of arrows placed between event boxes. The right frame is dedicated to three different functions:

- *Properties*: data management, information about events and associated dates.
- *CSV Data*: import of a CSV file containing information about dates or export of all the data used in the Events scene of the current project.
- *Phases*: Phases scene: for phase modeling, and where succession or duration constraints can be inserted.

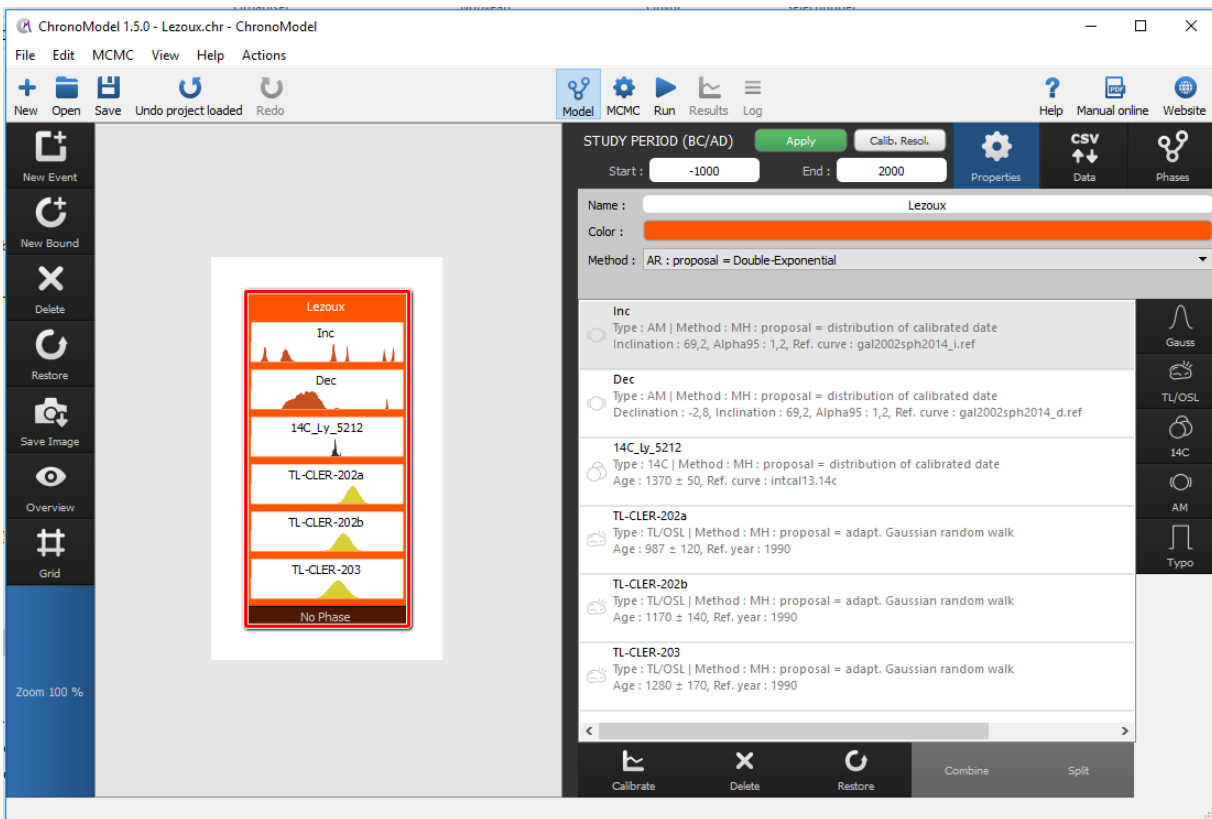


Figure 2: The interface of CM has Events created on the left scene and corresponding data information on the right hand side.

3.3 Creating an event

The parameter of interest within CM is the calendar date for a target event for which we have direct chronometric information. The user supplies the chronometric data related to each event date in the form of measurements. A measurement is for instance a radiocarbon age and associated error produced by a radiocarbon dating laboratory. In most cases an event date has only one measurement associated with it, but in some situations we can have "pooled" measurements supplied by one or several dating techniques.

To create a new event, with name and color, select the *NewEvent* button. Once validated, the event box appears in the Events scene. The properties of the selected event (surrounded in red by clicking on it) may be seen using the *Properties* button. Here, the name, the color and the MCMC method associated with this event might be changed.

By default, the Events scene is white. A grid may be added using the *Grid* button. It may help to order the events so that this order is followed for the plots in the results window.

The *overview* button and *zoom* scroll bar may be useful to better visualize event boxes when they are very numerous in the scene.

We may also export the image of the Events scene by clicking the *Save Image* button, either in PNG format or in Scalable Vector Graphics (SVG) format.

An event may be deleted from the Events scene first by selecting the Event and then by clicking on *Delete* button. An event is then deleted with all its measurements. Any element deleted may be restored by clicking on the *Restore* button. An event containing measurements is restored with all its measurements.

All measurements included in the Events of the current project may be exported in a CSV file using the *export all project data as CSV* button accessible from the *CSV Data* tab.

The event model is represented in CM by a box as in Figure 2, where the individual calibrated date distributions are displayed.

3.3.1 Adding measurements into an event

3.3.1.1 Data management. An event is associated with data information such as measurements or typochronological references, managed through “plugins”. There are two ways to insert data with ChronoModel:

- By using the *Properties* button: we select a measurement type by clicking on the corresponding plugin icon. An extension window will be opened to enter the data (see for instance Figure 3 for radiocarbon dating). Thus, data are added one by one.
- By using the *CSV Data* button: data are loaded using a CSV file organized according to each type of measurement.

For each measurement, we insert the reference name, the measurement value given by the laboratory and its associated standard error, and some additional information according to the type of measurement. By selecting an event (then surrounded in red) in the event’s scene and using the *Properties* button, it is possible to modify, delete or restore a measurement within an event.

3.3.1.2 Plugins. In CM, different types of measurement may be combined in order to estimate the date of the event of interest (see Figure 2). These types are defined according to five plugins:

- Radiocarbon dating *14C*. We insert the age value given by the laboratory and its associated standard error. If necessary we add the reservoir effect ΔR with its associated standard error.

It is sometimes possible to combine radiocarbon dating and dendrochronology, when we have several radiocarbon dates referring to an unknown calendar date θ and shifted by known quantities called δ_i . This is referred to as “wobble-matching”. δ_i may either be a deterministic or a stochastic parameter. If δ_i is stochastic, then its prior distribution function may have a uniform distribution or a Gaussian distribution. By default, the wobble matching is set to 0.

- Luminescence dating *TL/OSL*. We insert the age, its error and the reference year of measurement given by the laboratory.
- Archaeomagnetic dating *AM*. We insert the inclination, declination or intensity of the geomagnetic field given by the laboratory, with associated errors.
- Gaussian dating *GAUSS*. Certain calendar dates in CM can be given without any measurement process. Such a date represents an event within the chronology about which we have absolute prior chronological information but no chronometric measurement. In CM, it is defined as a Gaussian distribution characterized by a mean and an error. A Gaussian date is assumed to be defined in date format BC/AD.
- Typo-chronological reference *Typo Ref*. This is defined by a uniform dating interval corresponding, for instance, to the making period of an artifact. A Typo-reference is defined by its *Lower date* and its *Upper date* defined using date format BC/AD.

3.3.1.3 Reference curves. Measurements M_i from chronometric dating methods have to be calibrated into calendar dates. This is the case of radiocarbon, archaeomagnetism, luminescence and even Gaussian ages. CM converts these measurements into calendar dates using the appropriate calibration curve (see here below). When adding a radiocarbon dating, one may select one of these reference curves using the drop-down menu. In CM, several calibration curves are already implemented. Other curves may be included in CM from the *Application settings* option.

Figure 3: The radiocarbon measurement extension window with advanced options

- ^{14}C and AM reference curves: see list with references in CM user guide (Vibet et al., 2016)
- TL/OSL reference curve: the function of calibration is $g(t) = t_0 - t$, where t_0 is the reference year.
- GAUSS reference curve: it is possible to simulate a calibration process by using a quadratic function $g(t) = a * t^2 + b * t + c$. If $a = c = 0$ and $b = 1$, then $g(t) = t$. This corresponds to the *none* choice: dating without calibration. Otherwise, we may customize the calibration curve changing parameters a , b , and c . We can also load a CSV file containing a customized calibration curve (refer to *Application settings*).

3.3.1.4 Advanced options. These options allow you to define MCMC methods applied to the dates t_i associated with the measurements M_i . For this, we use the drop-down menu *method*.

- ^{14}C and AM: The default MCMC method is the Metropolis-Hastings algorithm using a distribution that mimics the individual calibration density as a proposal. This method is well adapted to multimodal densities such as those obtained from the calibration of archaeomagnetic or radiocarbon measurements. Two other proposals may also be selected, the (event date) prior distribution and a Gaussian adaptive random walk.
- TL/OSL, GAUSS and *Typo Ref*: The default MCMC method is the Metropolis-Hastings algorithm using a Gaussian adaptive random walk as a proposal. Two other proposals for TL/OSL and GAUSS may also be selected, the prior distribution and a distribution that mimics the individual calibration density. The other proposal for *Typo Ref* is solely the prior distribution.

Note that in the case of radiocarbon dating, the wiggle matching constraint is inserted via the advanced option.

3.3.2 Calibration process

3.3.2.1 individual calibration process. With the *Calibrate* button, each chronometric measurement (inclination, declination, ^{14}C age, TL/OSL age) can be calibrated against a calendar date using the appropriate calibration

curve. CM calibrates the individual data using the same Bayesian calibration framework as that used for radiocarbon dating. The calibration process is described in details in [Lanos and Philippe \(2017\)](#). An example of the individual calibration process is shown in Figure 4 in the case of radiocarbon dating. Several statistical results according to the distribution of the calibrated date are summarized, in particular the HPD region is given at a chosen confidence level.

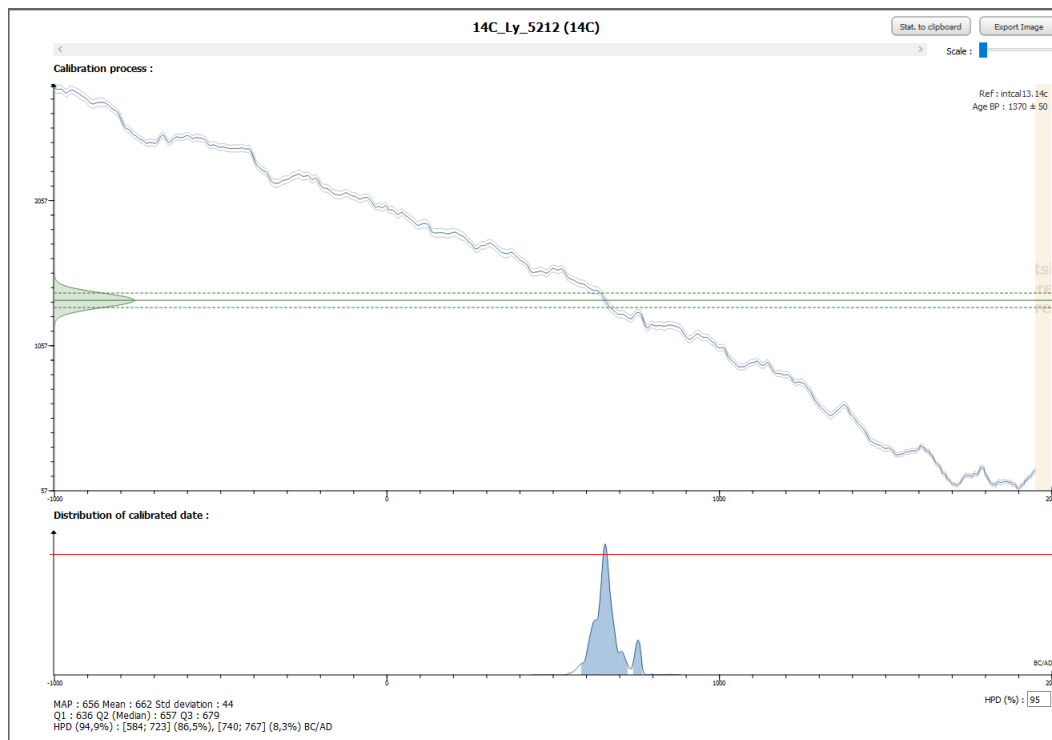


Figure 4: 14C age calibration using IntCal13 curve. No reservoir offset is taken into account.

3.3.2.2 Special case: calibration from multiple radiocarbon measurements. If several radiocarbon ages have been obtained from the same organic sample or are known for other reasons to date to exactly the same calendar year, then a “pooled mean” can be calculated as with BCal and OxCal. The radiocarbon measurements and associated errors are mathematically combined as part of the calibration process to create a single estimate for the event date. Note that these measurements should have the same reference curve for the calibration and the same MCMC method.

In CM, this situation can be treated in two ways. One way is to calculate the “pooled mean” \bar{M} as described above and introduce it as a single measurement in the target event date model. Thus, there will be only one individual error with respect to the target calendar date. Be aware that the pooled mean \bar{M} is not robust to outliers. To combine several radiocarbon measurements, select the associated event in the Events scene, then select the *Properties* tab. Now select all measurements to be combined from the list of measurements and click on the *Combine* button. A unique measurement appears. To undo, click on the *Split* button to split the measurements.

The other way is to consider the measurements as independent with different individual errors within the target event date model. This last approach leads to results similar to those obtained with “pooled means” and outlier penalization in BCal and OxCal : see examples in [Lanos and Philippe \(2017, 2018\)](#).

3.4 Creating Bounds, TPQ, TAQ

A bound, such as a historical date or a so-called *terminus post quem* (TPQ) or *terminus ante quem* (TAQ), may be introduced in order to constrain one or several event dates. The Bound does not behave like an event date because there is no individual error attached to it. It means that the bound represents exactly the date of a target event.

To create a new bound in the Events scene, select the *NewBound* button. A new box will be opened asking to name this new bound. After validation, an ellipse appears in the Event scene. When the bound is selected in the event's scene, the bound's properties may be seen with the *Properties* Button. From there, its name and its color might be changed and values may be added. A bound may either have a fixed value or have a uniform distribution within a range of values.

3.5 Creating phases

In CM, the Events scene and the Phases scene are separated into two frames (see example in Figure 5). This makes possible to model multi-phasing where an event may belong to several phases. To create a phase, click on the *Phases* button. We are then asked for the name, the color and its maximum duration. By default, the duration of the phase is unknown. If prior information about the maximum duration is available, it may be included in the model. It is of course possible to modify, delete or restore a phase.

By default, the Phases scene is white. A grid may be added. The phase boxes may also be seen with the *overview* button and the *zoom* scroll bar when they are very numerous on the scene. Finally, the image of the Phases scene may be saved in PNG or SVG format.

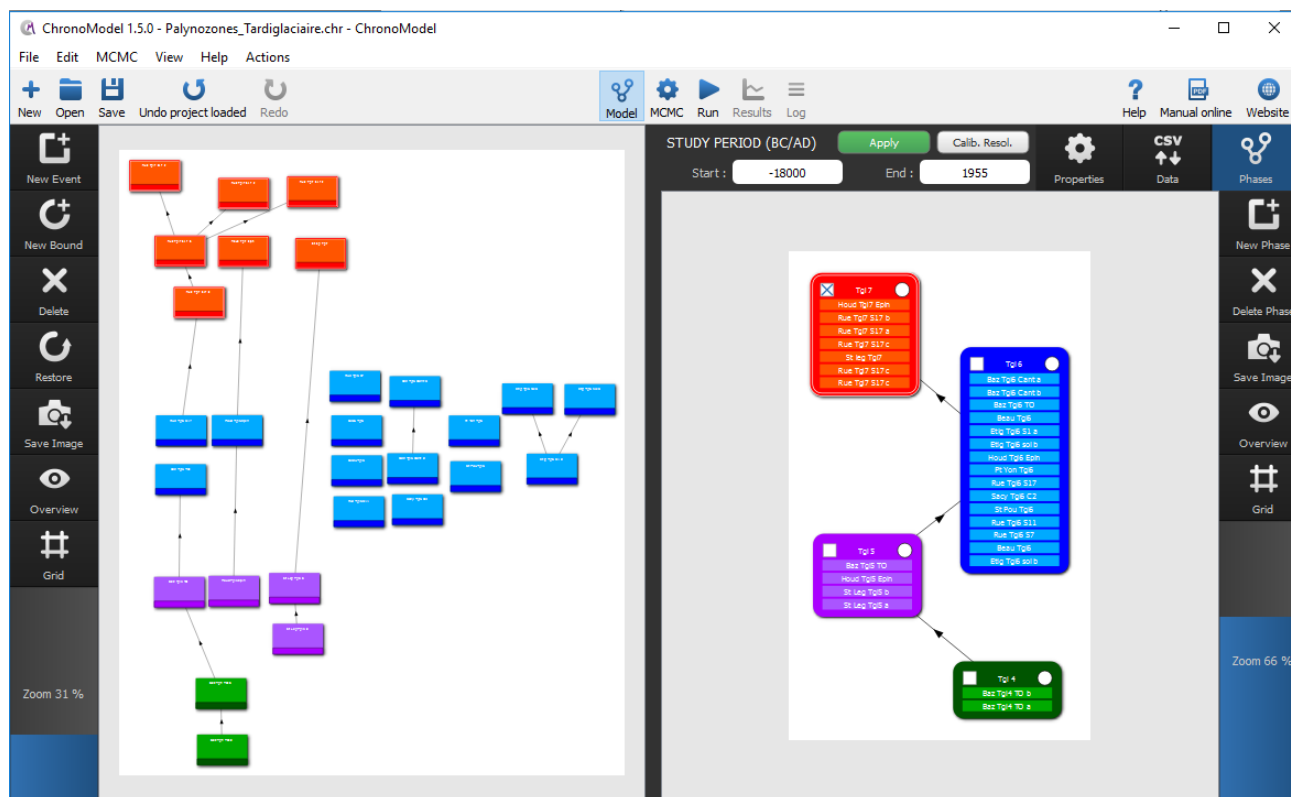


Figure 5: The interface of CM, with events created on the left scene, and phases on the right hand side.

3.6 Creating temporal constraints

In the Events scene, a stratigraphic constraint (relative dating) may be added or deleted between any two events or bounds. In the same way, in the Phases scene, a succession constraint may be added or deleted between any two phases. This constraint is represented by a black arrow going from the oldest element to the youngest.

When a set of event dates precedes an other set of event dates, thus leading to the creation of a lot of stratigraphic links (arrows) on the Events scene, then it is much easier and quicker to create two phases containing these events and between which only one stratigraphic link (one arrow) will be put.

A hiatus expresses the minimum time elapsed between two phases. By default, the hiatus between these phases is unknown. However, if prior information about the hiatus is available, it might be included in the model by clicking on the arrow.

4 MCMC calculation

In Bayesian modelling, posterior densities and their marginal densities are of interest. In general, we cannot derive posterior distributions analytically. However, posterior distributions are always known up to proportionality.

Let's define by θ the parameter of interest and by y the data sample. The posterior distribution $p(\theta|y)$, is known up to proportionality, by multiplying the prior and likelihood functions as follow:

$$p(\theta|y) \propto p(\theta)p(y|\theta)$$

We can build up a sample of θ from $p(\theta|y)$ even if we only know $p(\theta|y)$ up to proportionality using simulation methods based on Markov chain Monte Carlo (MCMC).

4.1 Markov chain Monte Carlo

Monte Carlo methods aim at estimating numerical values using repeated random sampling. By the law of large numbers, integrals described by the expected value of some random variable can be approximated by taking the empirical mean of independent samples of the variable. However, independent samples are usually difficult to compute. That is why Markov chains, more easily computed, are usually implemented.

A Markov chain is a sequence of random variables $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(I)}$, for which, for any t , the distribution of $\theta^{(t)}$ given all previous θ 's depend only on the recent value, $\theta^{(t-1)}$ (Buck et al., 1996; Robert and Casella, 2004). Under regularity conditions, when I becomes large, the Markov chain converges to its stationary (or equilibrium) distribution. When a Markov chain has reach its equilibrium (or its stationary state), any sample from the Markov chain is a sample from the stationary distribution.

Now, for any posterior distribution $p(\theta|y)$, it is possible to build a Markov chain $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(I)}$, whose stationary distribution is the posterior distribution $p(\theta|y)$. Once the equilibrium state is reached, any sample from the Markov chain may be used to estimate the posterior density, the posterior mean ...

As the sampling is done sequentially, with the distribution of the sampled draws depending on the past value drawn, correlation between successive values is high. To reduce this correlation, it is required to thin the Markov chain and use a subsample to estimate the posterior density distribution.

4.2 Building a Markov chain in multiple dimensions: the Gibbs sampler

A convenient algorithm useful in many multidimensional problems is the Gibbs sampler (Buck et al., 1996; Robert and Casella, 2004). Let's say we want to approximate the joint posterior probability density function, $p(\theta_1, \theta_2, \dots, \theta_d|y)$.

The algorithm starts with a sample of initial values $(\theta_1^{(0)}, \theta_2^{(0)}, \dots, \theta_d^{(0)})$ randomly selected. The first step of the algorithm is to update the first value by sampling a candidate value of $\theta_1^{(1)}$ knowing $\theta_2^{(0)}, \dots, \theta_d^{(0)}$ from the full conditional distribution $p(\theta_1|\theta_2^{(0)}, \dots, \theta_d^{(0)})$. The next step is to find a candidate value $\theta_2^{(1)}$ knowing $\theta_1^{(1)}, \theta_3^{(0)}, \dots, \theta_d^{(0)}$ using the full conditional distribution $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \dots, \theta_d^{(0)})$. And so on... Then the d^{st} step is to find a candidate value for $\theta_d^{(1)}$ knowing $\theta_1^{(1)}, \theta_2^{(1)}, \dots, \theta_{d-1}^{(1)}$. This process is then iteratively repeated.

When the number of iterations, I , becomes large, the distribution of the sample $(\theta_i^{(0)}, \dots, \theta_i^{(I)})$ becomes extremely close to the marginal posterior distribution of θ_i . In practical terms, this means that if we can simulate a Markov chain for θ_i until it reaches its equilibrium, then any sample from this Markov chain may be used to estimate information about the marginal posterior distribution.

4.3 Drawing from full conditional posterior distributions

Again, full conditional posterior distributions are not always easy to simulate. Algorithms are required to approximate these posterior distributions. In ChronoModel, two main algorithms are implemented, the **rejection sampling method** (also called acceptance-rejection - AR) and the **Metropolis Hastings algorithm** (MH). Both algorithms requires a proposal density function that should be easily sampled from in order to generate new candidate values. For the rejection sampling algorithm, it is common to use, if possible, the prior function or the likelihood as a proposal function. For the Metropolis-Hastings algorithm, a common choice is to use a symmetric density, such as the Gaussian density.

Depending on the type of the parameter, event θ , calibrated date t_i , variance of calibrated date σ_i^2 or bound θ^B , different methods are proposed. These methods are described here in turn.

4.3.1 Full conditional posterior distribution of the event θ

Three different methods can be chosen.

- **Rejection sampling with a Gaussian proposal** (Robert and Casella, 2004)
- **Rejection sampling with a Double exponential proposal** (Robert and Casella, 2004)
- **Metropolis-Hastings algorithm with an adaptive Gaussian random walk** (Roberts and Rosenthal, 2001)

The two first methods are exact methods. We recommend to use one of them.

4.3.2 Full conditional posterior distribution of calibrated dates t_i

In this case, three different methods can be chosen.

- **Metropolis-Hastings algorithm using a density that mimics the individual calibration density**
This method is adapted for multimodal densities, such as calibrated measures.
- **Metropolis-Hastings algorithm using the parameter prior distribution** This method is recommended when no calibration is needed, namely for TL/OSL, Gaussian measurements or typo-chronological references.
- **Metropolis-Hastings algorithm using an adaptive Gaussian random walk**
This method is adapted when the density to be approximated is unimodal. The variance of this proposal density is adapted during the process.

4.3.3 Full conditional posterior distribution of the variance of a calibrated date σ_i^2

As the full conditional density is unimodal, the Metropolis Hastings algorithm can be implemented here. The proposal density involved is an adaptive Gaussian random walk (Roberts and Rosenthal, 2001). The variance of this proposal density is adapted during the process.

4.3.4 Full conditional posterior distribution of a wiggle matching δ_i

If the δ_i is fixed, no sampling is needed.

If the δ_i has a uniform distribution on $[\delta_m, \delta_M]$ or if it has a Gaussian distribution, then the rejection sampling is used in order to sample out of the full conditional. In both cases, the proposal is a Gaussian distribution.

4.3.5 Full conditional posterior distribution of a bound θ^B

If the bound is fixed, no sampling is needed.

If the bound has a uniform distribution, the full conditional distribution is also a uniform distribution.

The aim of the calculation is to estimate the posterior marginal densities for the event date θ and for the other parameters : chronometric dates t_i and individual standard deviations σ_i . Posterior distributions are always known up to proportionality and we use Markov chain Monte Carlo (MCMC) methods to estimate numerical values using repeated random sampling.

CM software is not only open source, it also allows you to choose MCMC algorithms and parameters, and to thoroughly inspect the numerical calculations necessary for estimating the marginal posterior densities of the parameters of interest.

4.4 Settings

Before running the project calculation, we have to determine the setting parameters of the MCMC calculation. These settings are fixed by default, but we also offer the possibility to change them by selecting the *MCMC* button.

The MCMC settings are described here below and summarized in Figure 6, and the initialization values generated from the chosen "seed" can be seen with *MCMC initialization* tab in the *LOG* window.

Number of chains: One can run several chains in parallel using different starting values. This may be used to state whether the equilibrium is reached before the end of the Adaptation period. For the final report, a long run is performed, discarding values from the burn-in and adapting periods and thinning the remaining values (see below).

Starting values: Each Markov chain starts with a starting value that is randomly selected among all solutions compatible with constraints. For instance, in Figure 6, the starting value of the seed is set to equal 5.

Burn-in period: By default, we set 1 000 iterations for this period in order to "forget" the starting values.

Adapting: During the adaptation period, variances of all adaptive Metropolis-Hastings Gaussian random walks are adapted so that the corresponding acceptance rates are close to 44% (see section 5.1.2). The adaptation period continues with another batch until the acceptance rate calculated on the last batch is included between 41% and 47% unless the maximum number of batches is reached. By default, we use 500 iterations in each of the 20 batches.

Acquiring: In this period, we set a large number of iterations (typically 1 000 000), so that all Markov chains are assumed to have reached their equilibrium distribution. However, successive value of a Markov chain are not independent. In order to limit the correlation in the sample, we choose to thin the sample by only keeping equally spaced values.

Thinning: We keep a value every m value in each Markov chain. m is chosen so that the remaining values may be considered as independent. Autocorrelograms allow us to check the correlation between two successive values in the acquiring period. Hence, we start with the default thinning interval ($= 1$) and we change it if one autocorrelogram is not correct. In Figure 6, the thinning interval is set to equal 10 for final results.

Mixing level: This value (set to equal 0.95 or 0.99) is related to the proposal used to estimate the dates t_i (see section 2.4 in the article).

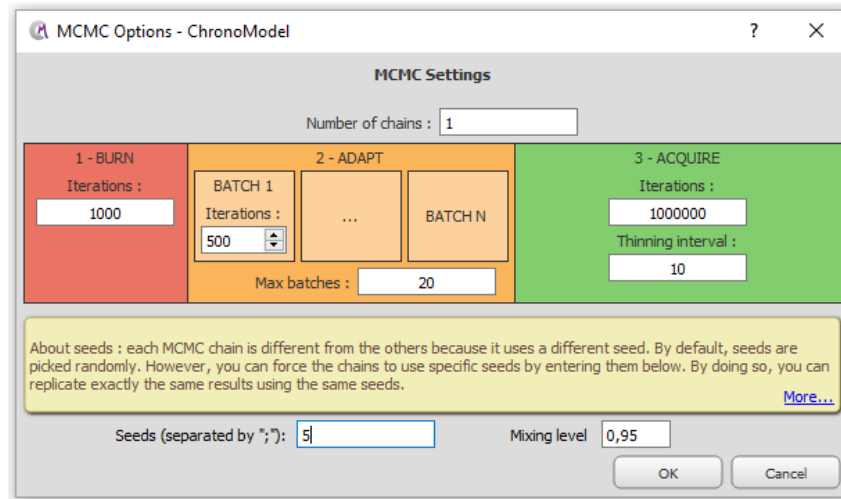


Figure 6: MCMC settings

4.5 Methods

Different MCMC methods are used in CM according to the parameters to be estimated: event date θ , chronometric dates t_i , individual errors σ_i . Two main algorithms are implemented, the rejection sampling method (also called acceptance-rejection) and the Metropolis-Hastings algorithm. Both algorithms require a proposal density function in order to generate new candidate values. For the rejection sampling algorithm, it is common to use, if possible, the prior function or the likelihood as a proposal function. For the Metropolis-Hastings algorithm, a common choice is to use a symmetric density, such as the Gaussian density.

The Gibbs sampler is used to generate a Markov chain for each parameter. These Markov chains are samples of the marginal posterior distribution of each parameter. Each Markov chain starts with a given starting value, uses a burn-in phase to forget this starting value, uses an adaptive phase to adapt all random walk variances, before the acquiring phase itself.

Depending on the type of parameter (Event θ , dates t_i , standard deviations σ_i), the following methods are proposed:

- The MCMC method used to generate new values at each step of the Gibbs sampler for the event date θ is rejection sampling using a double exponential proposal.
- In the case of chronometric dates t_i , the MCMC methods used are described in section 3.3.1.4.
- In the case of individual standard deviations σ_i , we use the Metropolis-Hastings algorithm with an adaptive Gaussian random walk. This method is well adapted here because the target density to be estimated is expected to be unimodal.

After execution of the “run” process, the Markov chains have to be checked in order to know whether they have reached their equilibrium before the acquiring period. Looking at the results, we verify two points : the convergence of the chains and the absence of correlation between successive values.

5 Results Presentation

The results presentation tool may be accessed from the project manager by selecting *Results*.

The results page provides four different tabs for viewing the results: *Posterior distrib.*, *History plots*, *Acceptance rate* and *Autocorrelation*.

We are given some control over the format of the results produced. For example, we can change the temporal ranges, the size and style of the plots, etc. (see *Display options* on the right panel).

When the MCMC calculations are finished, the marginal posterior density plots (*Posterior distrib.*) are first displayed. However, before interpreting the plots, it is necessary to ensure that the MCMC chains have converged.

5.1 Checking the Markov chains

The most important single issue is whether or not the MCMC simulation has converged. At present, there is no automatic checking facility that is fool-proof and we must take responsibility for checking convergence for ourself. This is why we put three tools in ChronoModel that allow us to check for convergence.

5.1.1 History plot

First, we can observe the history plot (or the trace) of the Markov chains and inspect it for signs of convergence. Figure 7 displays one Markov chain in the neighborhood of the starting values. We can see here that the Markov chain quickly finds its equilibrium state which is very different to the starting state. So, we have to check that the complete history plot shows good mixing properties during the acquiring period (green period) as they do not show tendencies or constant stages.

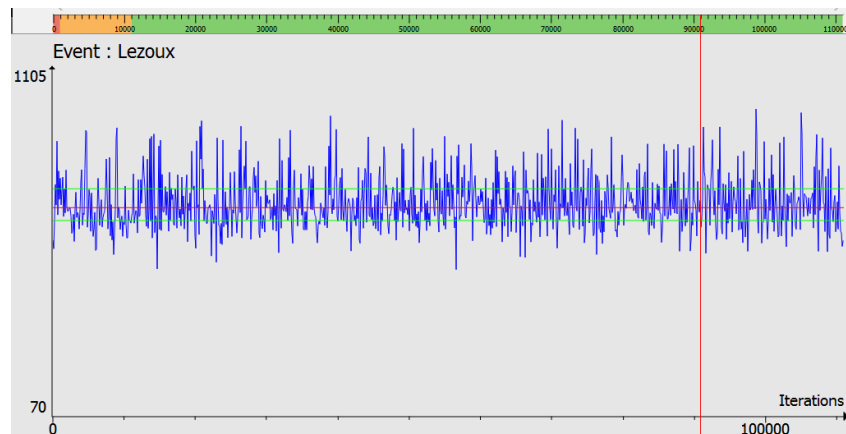


Figure 7: Event date history plot showing good mixing properties

Looking at parallel Markov chains, all computed with different starting values (therefore with different seed values), is also a way to decide whether a chain has reached its equilibrium. If the posterior distributions estimated by several different chains are very similar, this means that the equilibrium is reached. If equilibrium is not reached, the number of iterations of the burn-in period (red period) should be augmented. A higher number of iterations per batch may also be used. It might also be of help to change the algorithm used to draw values from full conditional posterior distributions (see the MCMC methods that can be selected for each parameter in section 3.3.1.4).

Because CM uses Markov chain Monte Carlo methods to obtain the results, no two runs of the code will produce the same answers (unless we fix the seed for the random number generator). However, we should find that multiple runs with different random seeds produce results which are very similar. Indeed, it is easy to run the project several times.

5.1.2 Acceptance rate.

The Metropolis-Hastings algorithm generates a candidate value from a proposal density and this candidate value is accepted with a probability. In the case of the Event date parameter θ , with the MCMC algorithm implemented being the rejection sampling method (called acceptance-rejection) from the full conditional distribution of θ , the acceptance rate is always equal to 1. This is an exact simulation.

In the case of the Metropolis-Hastings algorithm using an adaptive Gaussian random walk, the theoretical optimal rate is 44% (Roberts and Rosenthal, 2001). For example, in Figure 8, we can see that the acceptance rate for the TL date for which the Metropolis-Hastings algorithm using an adaptive Gaussian random walk is used, is close to 44% all throughout the acquiring period. If all batches are used in the adapting period, this might tell us that the adaptive Gaussian random walk did not reach the optimal interval. We then recommend asking for a longer number of iterations per batch in MCMC settings. Acceptance rate examples for Inc, Dec or 14C, displayed in Figure 8, are very different from 44%. This is due to the specific proposal used in these cases. In the case of date Inc, the acceptance rate is low: this comes from the fact that the target distribution is multimodal with peaked and asymmetrical modes. We could change the proposal (see *Advanced Method* drop down menu, as in Figure 3) by using the “prior distribution” or the “adaptive Gaussian random walk”. But it will not be ensured that all the states will be visited during sampling.

5.1.3 Correlation between successive values

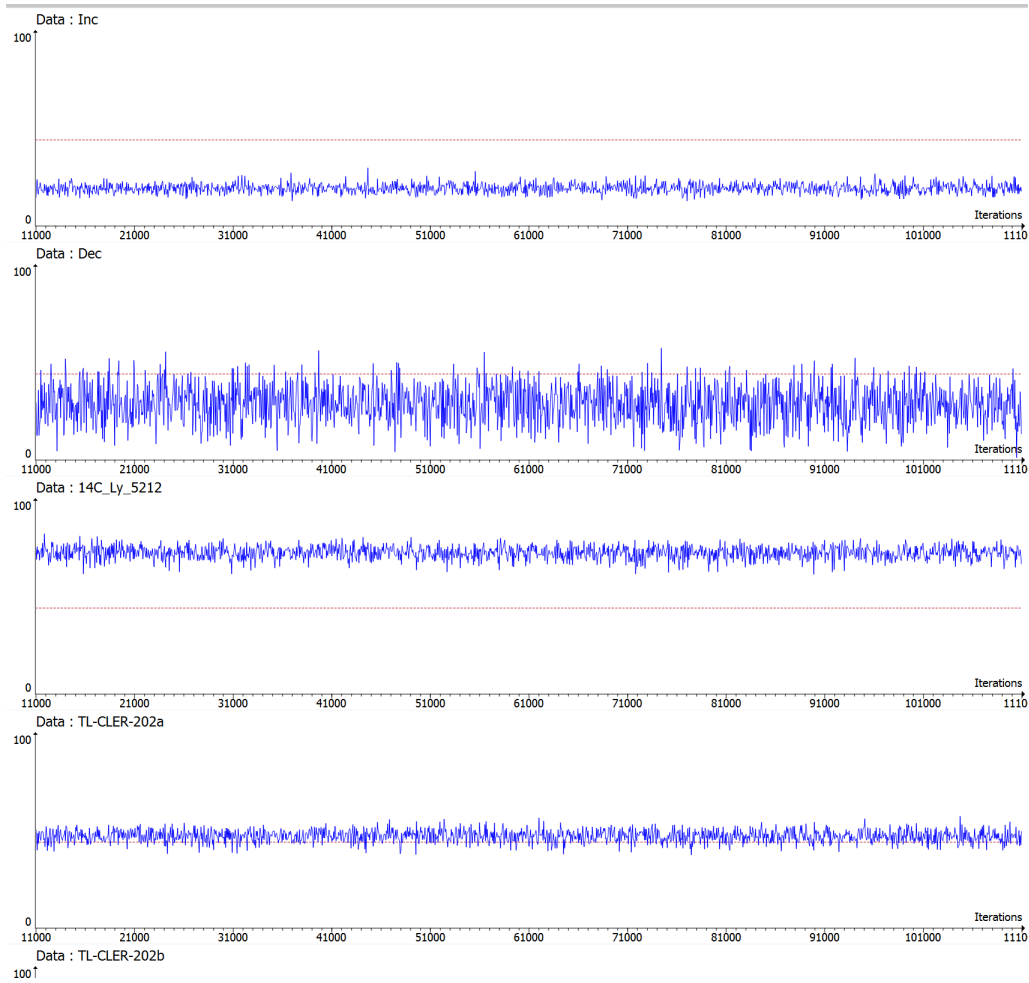
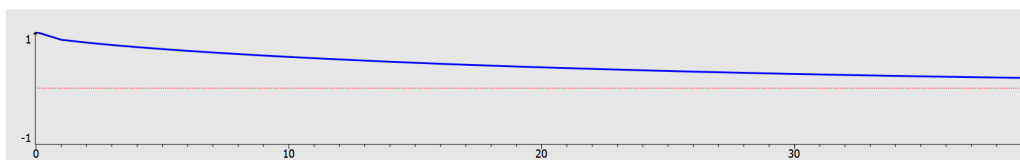
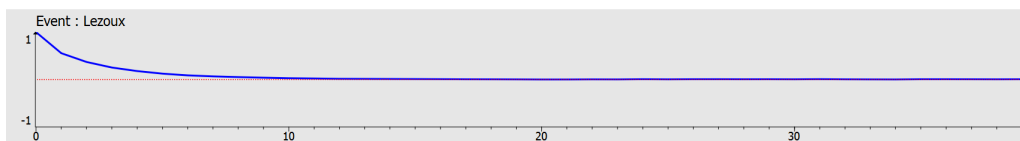
As the sampling is done sequentially (the current sample depending on the previous value drawn), correlation between successive values is high and will bias the estimations. In order to reduce this correlation, we use a subsample (from the thinning process) to estimate the posterior density distribution. Figure 9 displays an example of the autocorrelation function (autocorrelogram) obtained for event date θ when the thinning interval is equal to 1. We can see that the autocorrelation function decreases exponentially and falls under the 95% confidence interval after a lag of 40. This autocorrelation can be reduced by increasing the thinning interval to 10. For instance, using 1,000,000 iterations during the acquiring period, means that we keep 100,000 observations for final analysis. The obtained autocorrelogram decreases exponentially and falls under the 95% confidence interval after a lag of about 10 (Figure 10). We can also check that all other results remain similar when increasing this thinning interval.

Sometimes, there are circumstances, depending on the modeling project, under which the software will fail. We have done our best to eliminate these, but if you construct statistical or archaeological scenarios that we have not thought of before you may have problems. We cannot guarantee that everything you try to do will work without problems. If you do have problems, do not hesitate to contact us.

5.2 Marginal posterior densities

Considering that the Markov chains have reached their equilibrium and are not autocorrelated, the statistical results may be interpreted. The *Results* presentation tool provides us with the means to view and save the results of the MCMC calculations. Results are always expressed as marginal posterior probability distributions.

By default, when there are no phases in the modeling project, event (and bounds) plots are first displayed (on a grey background) and it is possible, using the *Unfold* button, to drop down the associated dates (on a white background) along with the individual calibrated dates (outlined in grey black line). An example of posterior

Figure 8: Acceptance rates for the dates t_i Figure 9: Autocorrelogram for event date θ with thinning interval equal to 1.Figure 10: Autocorrelogram for event date θ with thinning interval equal to 10. It quickly falls under the 95% confidence interval

densities for an event date with its associated dates is shown in Figure 11. To see marginal posterior distributions of individual standard deviations (example in Figure 12), click on *individual std deviations* in the *Results options* menu.

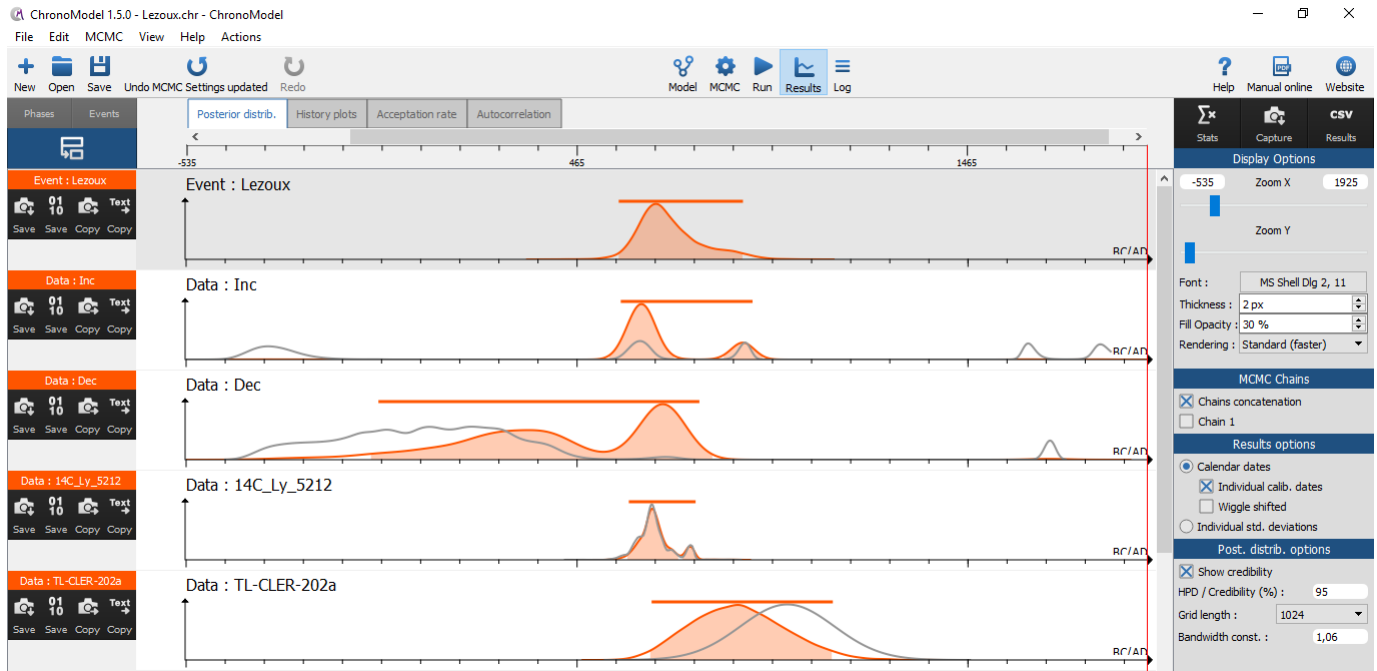


Figure 11: Posterior densities for event θ [grey background] and chronometric dates t_i [white background]. Individual calibrated densities are superimposed in dark grey lines to show how the dates are corrected by the event date model. Credibility intervals are shown above the posterior densities by a thick line

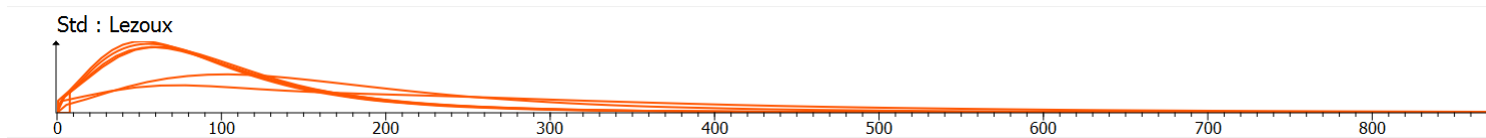


Figure 12: Posterior densities obtained for individual standard deviations σ_i .

When there are phases, “beginning” and “end” posterior probability distributions are first displayed on the same plot (see Figure 13). Then, using the *Unfold* button, it is possible to drop down event dates with their associated dates and also with individual calibrated dates. In this case it is also possible to plot, by using the *Show Duration* button, an estimate of the time elapsed between the beginning and end parameters.

All the density curves are obtained using a Gaussian kernel with a bandwidth defined by the Silverman criterion. For Gaussian data, the bandwidth is equal to 1.06 (Härdle et al., 2012). This value can be changed in the *Post. distrib. options* menu, as well as the grid length (1024 points by default) used by the FFTW routine (Frigo and Johnson, 2005) for calculating the densities.

5.3 Checking the Markov chains

When Markov chains are generated, two points have to be verified : the convergence of the chains and the absence of correlation between successive values. If the Markov chain has not reach its equilibrium, values extracted from the chains will give inappropriate estimates of the posterior distribution. If high correlation remains between successive values of the chain, then variance of the posterior distribution will be biased. Here are some tools to detect whether

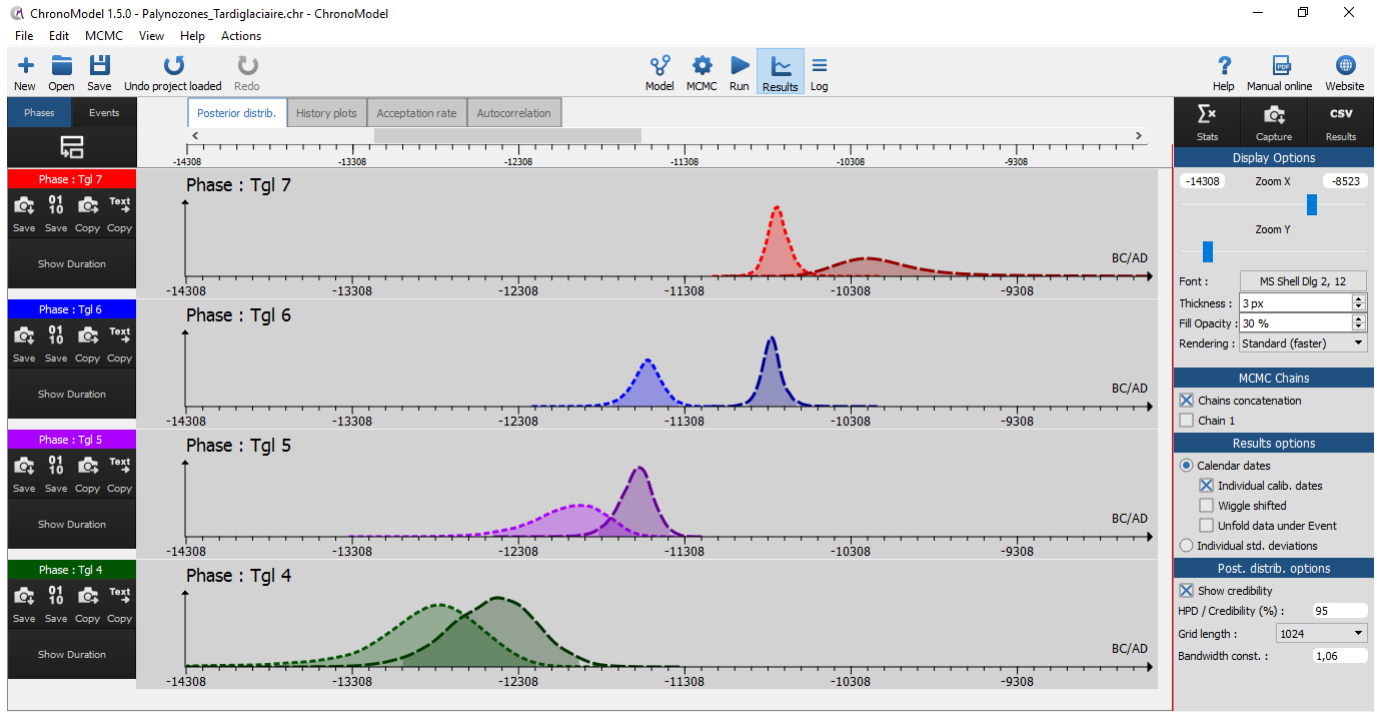


Figure 13: Posterior densities for the beginning and the end of different phases

a chain has reached its equilibrium and whether successive values are correlated. We also give indications about what can be done in these unfortunate situations.

By default, results shown correspond to the date density functions. To see results corresponding to individual standard deviations (σ_i), select "Individual std deviation" under the "Results options" section.

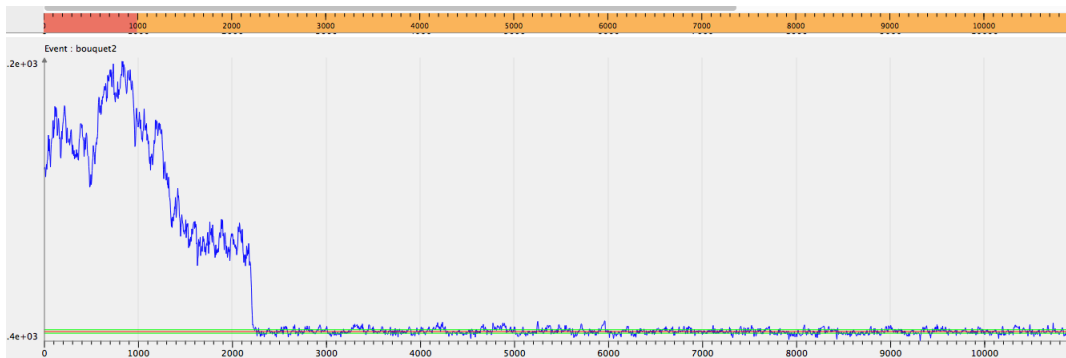


Figure 14: History plot near starting state

5.3.1 Is the equilibrium reached ? Look at the history plots.

Unfortunately, there is no theoretical way to determine how long a burn-in period needs to be. The first thing to do is to observe the history plot of the chains (dates and variances) and to inspect it for signs of convergence. Figure 14 displays an example where the Markov chain explored a first state in the neighbourhood of the starting value. Then during the **ADAPT period** (orange period), the Markov chain finally found its equilibrium state that was very different from the starting state.

History plots should have good mixing properties during the **ACQUIRE period** (green period). They should not

show tendencies or constant stages as shown in the example of Figure 15.

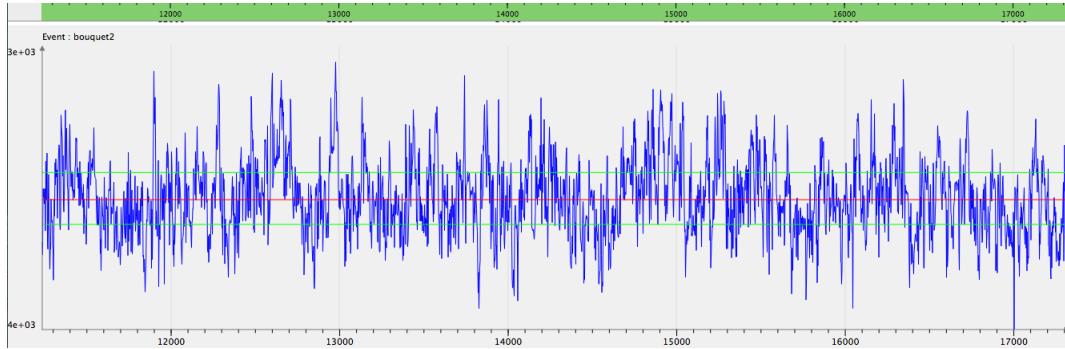


Figure 15: History plot with good mixing properties

Producing parallel Markov chains, all with different starting values, can help deciding whether a chain has reached its equilibrium. When the equilibrium is reached, the posterior distributions estimated by each chain should be similar.

What should be done if equilibrium is not reached ? First, the number of iterations of the **BURN period** (red period) should be augmented. A higher number of iterations per batch may also be used.

It might also be of help to change the algorithm used to draw values from full conditional posterior distributions (the MCMC method selected for each parameter).

5.3.2 Correlation between successive values ? Look at the autocorrelation functions.

A Markov chain is a sequence of random variables $\theta^{(1)}, \theta^{(2)}, \dots$, for which, for any t , the distribution of $\theta^{(t)}$ given all previous θ 's depend only on the recent value, $\theta^{(t-1)}$ (Buck et al., 1996; Robert and Casella, 2004). Hence, a high correlation between two consecutive values is expected. However, correlation will bias the estimations. That is why thinning the Markov chains is required.

To check whether the chain is correlated, observe the autocorrelation plot. The autocorrelation function should have an exponential decrease. Only the first autocorrelation should be high (at lag 0), the remaining correlations should be negligible. Figure 16 displays several autocorrelation functions having a good behavior.

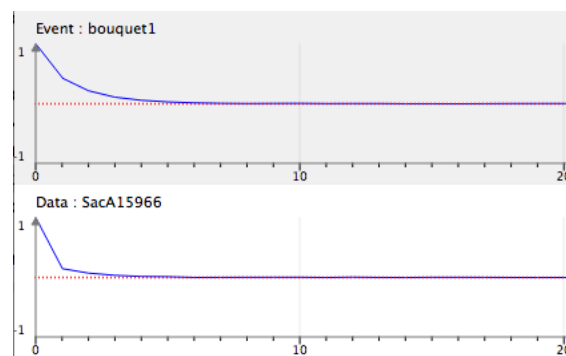


Figure 16: Examples of autocorrelation functions that fall quickly enough under the 95% confidence interval

What should be done if correlation is high ? By default, the thinning interval from the MCMC settings is 10, that is only one value out of 10 is kept from the Markov chains. This thinning interval should be long enough to reduce the correlation between successive values. If not, it should be increased.

5.3.3 Look at the acceptance rates.

The Metropolis Hastings algorithm generates a candidate value from a proposal density. This candidate value is accepted with a probability. An interesting point is the acceptance rate of these candidate values. This rate gives an idea of the suitability of the MCMC method used. This rate should not be too low or too high. However, there are no theoretical criterion except for adaptive Metropolis Hastings Gaussian random walk.

The theoretical optimal rate for the adaptive Gaussian random walk is 44% (Roberts and Rosenthal, 2001).

In ChronoModel, this algorithm is used to draw values from the full conditional posterior density of individual variances. But this method may be also chosen for other parameters.

Figure 17 displays an example of acceptance rates that are close to 44% (within the interval of 40% and 46%). Acceptance rates of each adaptive Gaussian random walk should be checked.

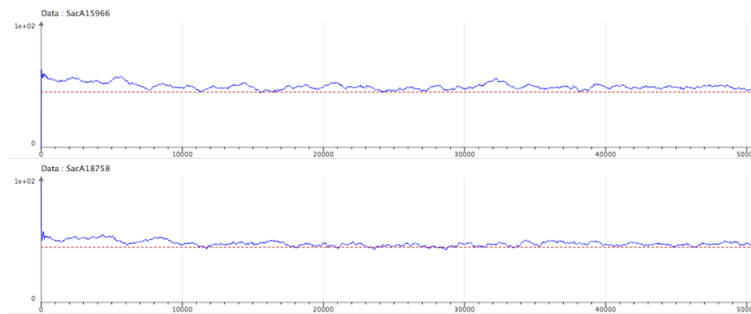


Figure 17: Examples of acceptance rates close to 44%

What should be done if the acceptance rates are not close to 44% ? From the MCMC settings, ask for a longer number of iterations per batch.

5.4 Statistical results

For each marginal posterior density of a parameter, the statistical results are summarized as the following:

- MAP : Maximum a posteriori, the highest mode of the marginal posterior density.
- Mean : the mean of the posterior density function. The mean is usually preferred to MAP or Median as its statistical properties are better than the others.
- Std deviation : the standard deviation of the chain
- Q1 : the numerical value separating the lower 25% of the data from the higher 75%
- Q2 (MEDIAN): the numerical value separating the lower 50% of the data from the higher 50%.
- Q3 : the numerical value separating the lower 75% of the data from the higher 25%
- Credibility interval (CI): the shortest credible interval expressed at a given confidence level.
- HPD region : indicates the most likely calendar date range(s) for the parameter in question. It is a set of intervals having the highest posterior density at a given confidence level. In other words, any year included in a reported HPD region has a higher posterior probability density than any year outside the region. It might be an interval (if the density is unimodal) or a discontinuous set (if the density is multimodal). When the density is unimodal, HPD region and CI should give similar results. However, due to numerical problems, HPD regions may be less precise than CI. Hence we recommend the use of CI when the density is unimodal. By default, HPD regions and credibility intervals are given at 95%.

- Acceptance rate : global acceptance rate. In the case of event date θ , this rate is always 100%

5.4.1 Credibility interval (CI)

Once the Markov chain of the parameter of interest is obtained, the credibility interval is constructed using the empirical quantile function. In ChronoModel, a $100(1 - \alpha)\%$ credibility interval is the shortest interval, defined by two empirical quantile function values, out of which $\alpha\%$ of the Markov chain remains.

5.4.2 Highest posterior density (HPD) region

In ChronoModel, highest posterior density regions are estimated by

$$J = \{x \in Grid | \hat{f}_n(x) \geq k\}$$

The value of k is chosen so that

$$\delta * \sum_{x \in J} \hat{f}_n(x)$$

where δ is the grid stepsize.

However, a better estimation would be the one developed by [Samworth and Wand \(2010\)](#).

5.5 Duration, Tempo-Plot, Activity

Description which will be done for next version CM3 which will be released in 2023 [Philippe and Vibet \(2020\)](#)

5.6 LOG, saving, summary, post-treatment

Using the *LOG* button, an html list of summaries is displayed:

- *Model description* is a summary of the model structure along with the type of measurement, age, standard deviation, reference curve for the calibration process, MCMC method, stratigraphic constraints, and so on.
- *MCMC initialization* is the list of starting values used in the MCMC process and for each chain.
- *Posterior distrib. results* gathers all statistical results according to all posterior distributions.

Each graph may be saved either individually, using the *Save Image* button, or globally using the *Capture* button. Different formats may be chosen : PNG, JPG, SVG or even BMP for global graphs.

Individual graphs may also be copied and pasted using the clipboard *Copy image* button. Statistics related to each parameter are saved thanks to the *Copy text* button. It is also possible to extract the results from the Markov chains using the *Save 0110* button.

The *CSV Results* button allows you to save a table (*Stats-table.CSV*) summarizing the statistical results and to save several CSV files (in BC/AD format): a file called "events.csv" containing the MCMC samples of all events, a file called "phases.csv" containing all parameters (beginning and end) of phases if at least one phase is modeled, and a file per phase (if any) containing the MCMC samples of the parameters of the phase and all events included in it. HTML files from the LOG tab are also saved.

Thus, values of MCMC chains (saved in CSV files) may be subject to additional post-treatments.

6 Terms and conditions

6.1 Disclaimer

ChronoModel is released under the CeCILL Licence V2.1: http://www.cecill.info/licences/Licence_CeCILL_V2.1-en.html.

6.2 Copyright notice

ChronoModel

<https://www.chronomodel.com>

Copyright © CNRS, 2024. All rights reserved.

Project director : Philippe LANOS (philippe.lanos@univ-rennes1.fr)

Authors :

Helori LANOS (helori.lanos@gmail.com)

Philippe DUFRESNE (philippe.dufresne@CNRS.fr)

Contributors :

William BANKS, Juan Antonio BARCELÓ ÁLVAREZ, Pascal BERTRAN, Didier BINDER, Jacques-Elie BROCHIER, Sandrine CABUT, Thomas DYE, Gwenaël HERVÉ, Jean-Claude LEFÈVRE, Christine OBERLIN, Christophe VASCHALDE.

The FFTW software component ([Frigo and Johnson, 2005](#)) embedded within CM is owned by third parties: "A Fast Fourier Transform Compiler," by Matteo Frigo, in the Proceedings of the 1999 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '99), Atlanta, Georgia, May 1999.

<https://www.fftw.org/>

<https://github.com/LionelAuroux/hell-lang/blob/master/thesis/synthese/fftw.bib>

References

- Buck, C. E., Litton, C. D., and Cavanagh, W. G. (1996). *The Bayesian Approach to Interpreting Archaeological Data*. Chichester, J.Wiley and Son, England.
- Cassen, S., Lanos, P., Dufresne, P., Oberlin, C., Delqué-Kolic, E., and Le Goffic, M. (2009). Datations sur site (tables des marchands, alignement du grand menhir, er grah) et modélisation chronologique du néolithique morbihannais. In Cassen, S., editor, *Autour de la Table, explorations archéologiques et discours savants sur une architecture restaurée à Locmariaquer, Morbihan (Table des Marchands et Grand Menhir), Colloque international de Vannes 2007*, pages 737–768. LARA, CNRS et Université de Nantes.
- Frigo, M. and Johnson, S. G. (2005). The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231. Special issue on “Program Generation, Optimization, and Platform Adaptation”.
- Härdle, W. K., Müller, M., Sperlich, S., and Werwatz, A. (2012). *Nonparametric and Semiparametric Models*.
- Harris, E. (1989). *Principles of archaeological stratigraphy*. Interdisciplinary Statistics, XIV, 2nd edition. second ed. Academic Press, London.

- Lanos, P. (2004). Bayesian inference of calibration curves: Application to archaeomagnetism. In Buck, C. and Millard, A., editors, *Tools for Constructing Chronologies*, volume 177, pages 43–82. Lecture Notes in Statistics, Springer-Verlag.
- Lanos, P. and Dufresne, P. (2012). Modélisation statistique bayésienne des données chronologiques. In de Beaune, S. A. and Francfort, H.-P., editors, *L'Archéologie à découvert*, pages 238–248. CNRS Editions, Paris.
- Lanos, P. and Philippe, A. (2017). Hierarchical Bayesian modeling for combining dates in archaeological context. *Journal de la Société Française de Statistique*.
- Lanos, P. and Philippe, A. (2018). Event date model: a robust bayesian tool for chronology building. *Communications for Statistical Applications and Methods (CSAM)*, 25:131–157.
- Le Goff, M., Gallet, Y., Genevey, A., and Warmé, N. (2002). On archeomagnetic secular variation curves and archeomagnetic dating. *Physics of the Earth and Planetary Interiors*, 134:203–211.
- Litton, C. D. and Buck, C. E. (1995). The Bayesian approach to the interpretation of archaeological data. *Archaeometry*, 37(1).
- Pavón-Carrasco, F., Rodríguez-González, J., Osete, M., and Torta, J. (2011). A matlab tool for archaeomagnetic dating. *Journal of Archaeological Science*, 38:408–419.
- Philippe, A. and Vibet, M.-A. (2020). Analysis of archaeological phases using the r package archaeophases. *Journal of Statistical Software, Code Snippets*, 93(1):1–25.
- Robert, C. and Casella, G. (2004). *Monte Carlo Statistical Methods*. 2nd ed. Springer.
- Roberts, G. and Rosenthal, J. (2001). Optimal scaling for various metropolis-hastings algorithms. *Statistical Science*, 16(4):351–367.
- Samworth, R. J. and Wand, M. P. (2010). Asymptotics and optimal bandwidth selection for highest density region estimation. *The Annals of Statistics*, 38(3):1767–1792.
- Sapin, C., Bayle, M., Buttner, S., Guibert, P., Blain, S., Lanos, P., Chauvin, A., Dufresne, P., and Oberlin, C. (2008). Archeologie du bati et archeometrie au Mont-Saint-Michel, nouvelles approches de Notre-Dame-Sous-Terre. *Archeologie Medievale*, 38:71–122.
- Vibet, M.-A., Philippe, A., Lanos, P., and Dufresne, P. (2016). Chronomodel v1.5 user's manual. www.chronomodel.fr.