

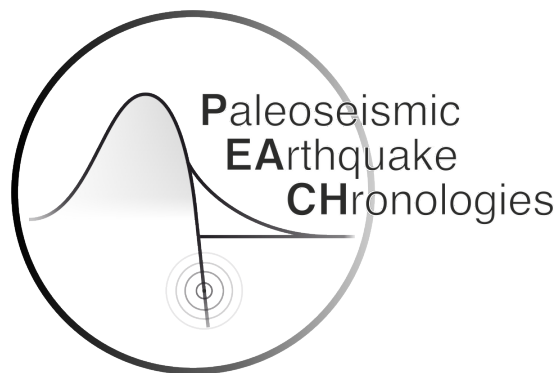
# PEACH

---

## Paleoseismic **E**arthquake **C**hronologies

### User Manual

Version 1 (January 2023)



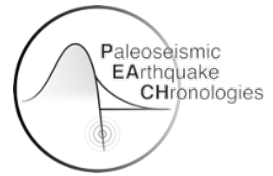
Octavi Gómez Novell, Bruno Pace, Francesco Visini

Update: January 2023

*Disclaimer: this approach has been developed during the postdoctoral stay of O. Gómez at the University of Chieti-Pescara (Italy) and in the framework of the confection of the second release of the Fault2SHA Central Apennines Database. The research has been supported by INGEO Department funds and postdoctoral fellowship "Margarita Salas" of O. Gómez, funded by the University of Barcelona with EU's Next Generation funds. The approach and derivatives are published under the Creative Commons license CC-BY-NC 4.0. For more info: <https://creativecommons.org/licenses/by-nc/4.0/deed.es>*







### Table of contents

<b><u>1. WHY PEACH?</u></b>	<b><u>1</u></b>
<b><u>2. WHAT ARE THE FUNDAMENTS OF PEACH?</u></b>	<b><u>1</u></b>
<b><u>3. FUNCTIONING OF THE APPROACH</u></b>	<b><u>2</u></b>
PART 1 – BUILDING INDIVIDUAL EVENT PDFs	3
PART 2 – ALL SITE PDF INTEGRATION	3
PART 3 – EVENT PEAK IDENTIFICATION	4
PART 4 – EXTRACTION OF THE PALEOEARTHQUAKE CHRONOLOGY	5
	<b><u>8</u></b>
<b><u>4. INPUT FILE PREPARATION</u></b>	<b><u>9</u></b>
<b><u>5. OUTPUT FILES</u></b>	<b><u>10</u></b>
<b><u>6. RESULT INTERPRETATION</u></b>	<b><u>11</u></b>
<b><u>7. REQUIREMENTS FOR THE CODE</u></b>	<b><u>14</u></b>
<b><u>8. STEP BY STEP GUIDE TO RUN THE CODE</u></b>	<b><u>14</u></b>





## 1. Why PEACH?

One of the main challenges in paleoseismic research is to constrain the timing of paleoearthquakes that are identified in the trenches. Limitations in numerical dating techniques, insufficient sedimentary resolution or slip variability along strike, are common causes of large event date uncertainties. This further difficulties the correlation of paleoseismic datasets between sites and the confection of a reliable paleoearthquake chronology.

PEACH is an approach intended to assist paleoseismologists in constraining paleoearthquake chronologies on faults, especially in the cases where the previous issues arise. PEACH is particularly useful when trench data is complex (e.g., multiple sites available), presents large uncertainties and is difficult to correlate manually. The approach follows an automatic workflow that treats event times probabilistically, allowing for a proper treatment of the uncertainties relating to dates and number of events. The inputs and outputs of the approach are simple, easy to prepare and user friendly.

## 2. What are the fundamentals of PEACH?

The approach relies on three **basic assumptions** that need to be considered.

### A. Paleoeearthquakes can be recorded in several sites along fault

Therefore, paleoearthquakes are potentially correlatable between sites if their time ranges of occurrence overlap. This assumption is strongly related to the extent of fault ruptures that are viewed as feasible for a tectonic structure, i.e., which structures are capable of rupturing simultaneously given a morphogenetic earthquake (earthquakes whose rupture reaches the surface).

### B. Correlations are restricted to the fault scale

The **fault** level from the Fault2SHA Central Apennines Database (CAD; Faure Walker et al., 2021) is adopted. According to the authors: *Faults display how the traces are connected at the surface and/or at depth. How the traces connect to form faults is based on fault geometry continuation (two traces can start and finish within a few metres from each other or may be kilometres apart), continuation of Late Pleistocene-Holocene offset and total offset across the fault and known earthquake ruptures. In showing how traces are connected, the fault map is of use to those wanting to understand the structures of the region.*

Essentially, this means that a fault is a singular structure that is estimated to have the potential of rupturing the entirety of its length. A fault can be formed by smaller sections, namely traces according to the CAD, that show different levels of activity evidence (e.g., geomorphic footprint, etc.) or location certainty. However, these traces are not expected to prevent fault rupture propagation as they are part of the same structure.

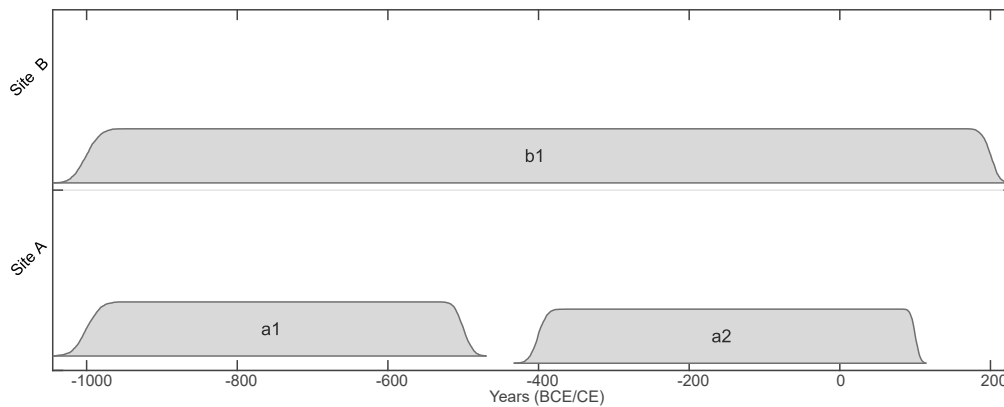
**Important:** The extent to which paleoseismic data might be correlated is up to the user. However, please note that we do not recommend using the approach to correlate data beyond the fault boundaries (i.e., multi-fault ruptures). The code



is not yet adapted to accommodate such instances and therefore the validity of the results might be compromised.

### C. Correlation is not restricted to single events

The record of events observed in paleoseismic trenches is always a minimum due to the issues mentioned at the beginning of this document. This means that, the wider the event timing uncertainties, the higher the chance that the paleoearthquake record is underestimated in that time period. Hereby, event correlations between sites should not be restricted to accommodate this uncertainty. For instance, considering that two events (a1 and a2) in a site A overlap with one event (b1) in another site B (Fig. 1), this latter should participate in both correlations with the former because it cannot be discarded that b1 represents the occurrence of two earthquakes.



**Figure 1.** Synthetic example of a paleoearthquake record of two paleoseismic sites (A and B), susceptible to be correlated with the PEACH approach.

## 3. Functioning of the approach

PEACH is an approach based on an algorithm written in MATLAB language (v. 2022b) that probabilistically models earthquake chronologies on faults using sets of basic input data from trenches (i.e., numerical date distributions limiting each event horizon; Table 1). These data are used exclusively as inputs to ensure objectivity in the whole process, because the numerical dates come from independent laboratory analyses with quantified uncertainties. The algorithm is divided into four steps that are detailed below:

Event ID	Site ID	Old	Error ( $1\sigma$ )	Younger date	Error ( $1\sigma$ )
a1	Site A	-500	30	1200	20
a2	Site A	-3000	50	-500	30
a3	Site A	-3000	50	-500	30
b1	Site B	-1000	40	-500	20
b2	Site B	-4300	50	-1000	40

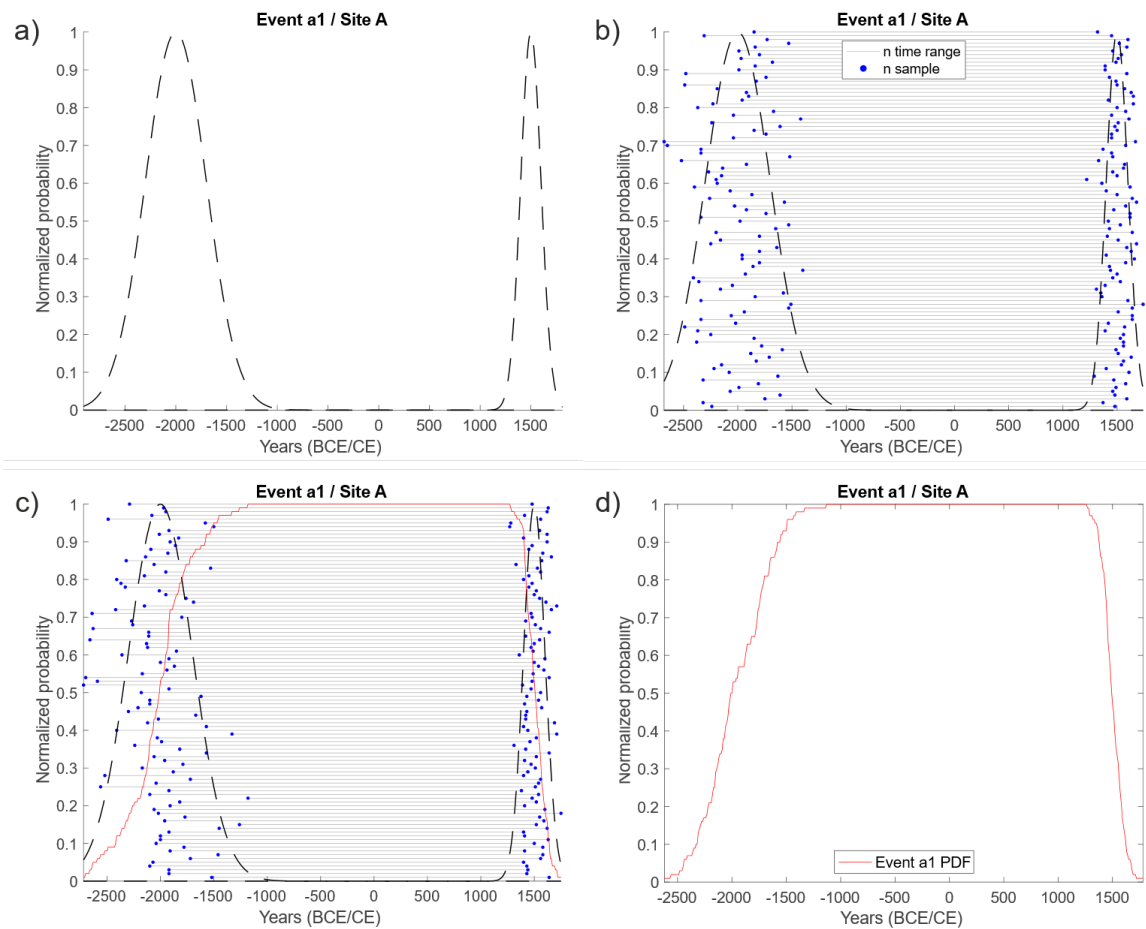
**Table 1.** Example of the input file required for the computation. Note. Dates are negative values for Before Common Era (BCE) and positive for Common Era (CE).



## Part 1 – Building individual event PDFs

In this first step, the code builds all the event PDFs from the numerical dates provided in the input (Fig. 2).

Initially, the numerical dates that constrain a specific event are transformed into normally distributed PDFs (Fig. 2a). A random sampling of  $n$  Monte-Carlo simulations is then performed within each PDF to establish  $n$  time ranges that define possible occurrence spans of that event. Each time range is defined by a random age sample  $n$  in the older PDF and a random age sample  $n$  in the younger PDF (Fig. 2b). Lastly, the time ranges are summed and normalized to obtain a PDF that represents the event age distribution (Figs. 2c and 2d). This process is done for all the events in all sites included in the input.



**Figure 2.** Visualization steps of the first part of the workflow for synthetic event a1 in Table 1: **a)** PDFs of the numerical dates provided in the input table; **b)** random sampling of the PDFs of both numerical dates and establishment of time ranges that represent the time span for the paleoearthquake a1 occurrence. In this case, the number of simulations is  $n=100$ ; **c)** Normalized summation of the time ranges to obtain a PDF (red solid line) representing the age of a1; **d)** resulting PDF of a1, which is used in the further steps of the workflow.

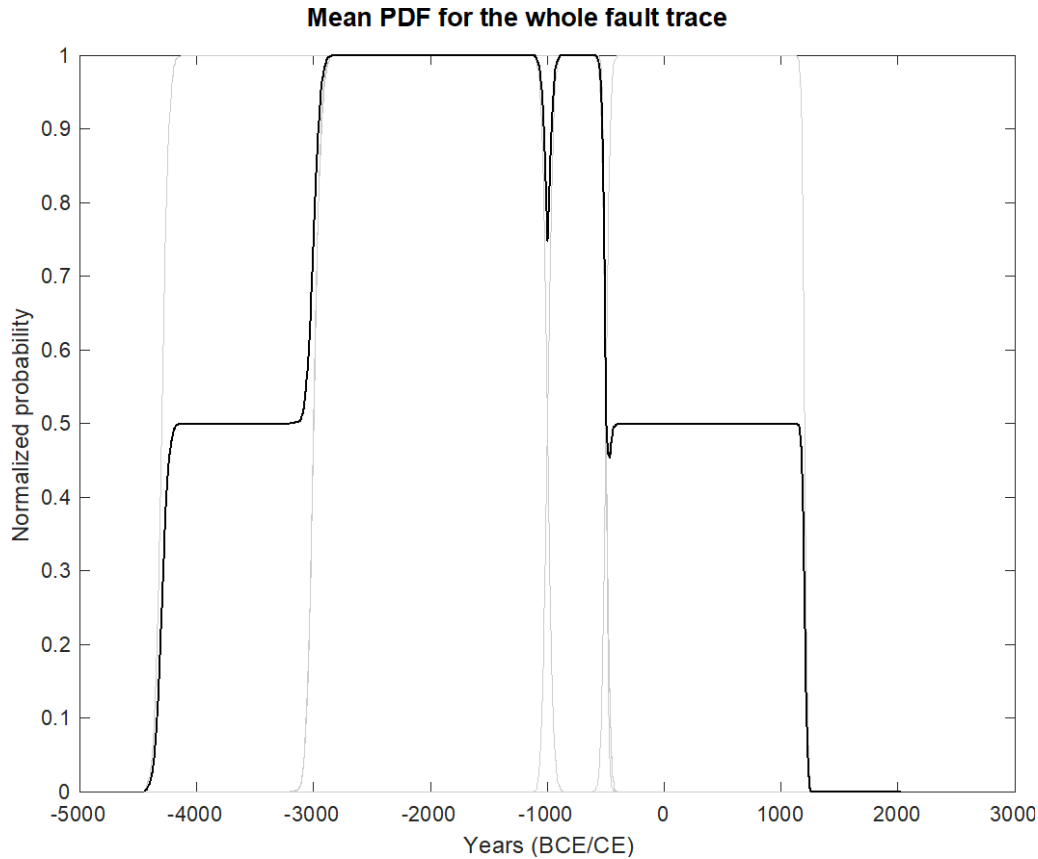
## Part 2 – All site PDF integration

Following the event PDF computation, the code integrates the distributions of all sites by computing a mean PDF of the maximum probability values for each



year bin (Fig. 3a). This distribution represents the overall earthquake occurrence probabilities in the studied fault for the time span recorded in the trenches.

The resulting shape of the mean PDF generally displays a series of peaks that represent time spans with higher event PDF coincidence, thus higher event occurrence probability. In this respect, the peaks can be considered as indicative of paleoearthquake occurrences in the fault analyzed. The next step is therefore to automatically identify the peaks.



**Figure 3.** Normalized mean PDF obtained from averaging all the event PDFs from all sites within the studied fault trace (data from the example in Table 1).

### Part 3 – Event peak identification

The peak identification process is based on the in-built MATLAB function *findpeaks*, that allows to identify peaks in a distribution. Because this function can lead to noisy peak detections, we adjust its accuracy by setting a threshold on the prominence of the peaks. The prominence measures how much the peak stands out based on its height and with respect to other peaks in the distribution.

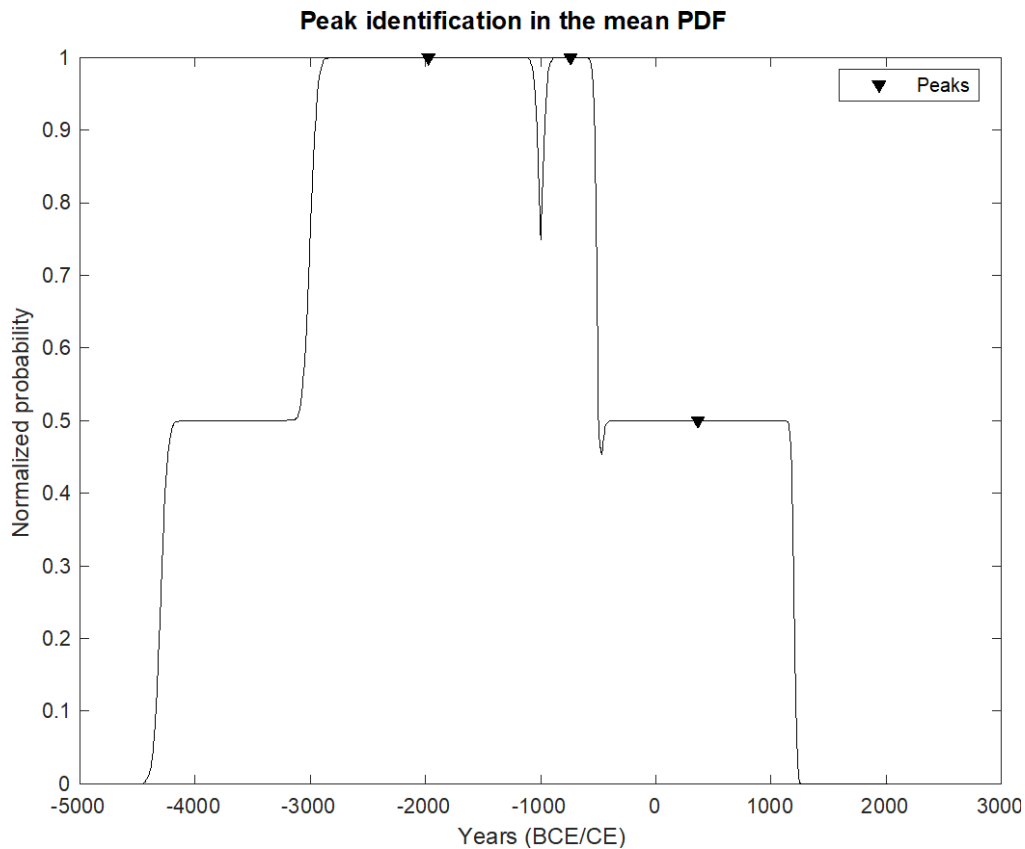
The default prominence threshold is set at 0.03 or 3% of the normalized probability, which has demonstrated the best performance in peak identification for several tests carried out in different real-case and artificial datasets.

The *findpeaks* function identifies by default the peaks from left to right (older to younger ages). In plateau-shaped peaks this can be problematic as it systematically places the peak position to the left end of the plateau. To avoid





this, the detection function is run a second time in the reverse direction (right to left) to find the peak in the right end of the plateau. Then, the middle point between both ends is computed as the true peak, which yields a more realistic position.



**Figure 4.** Peaks identified by the code in the mean PDF distribution. The time location of these peaks is used for the following steps of the approach.

#### Part 4 – Extraction of the paleoearthquake chronology

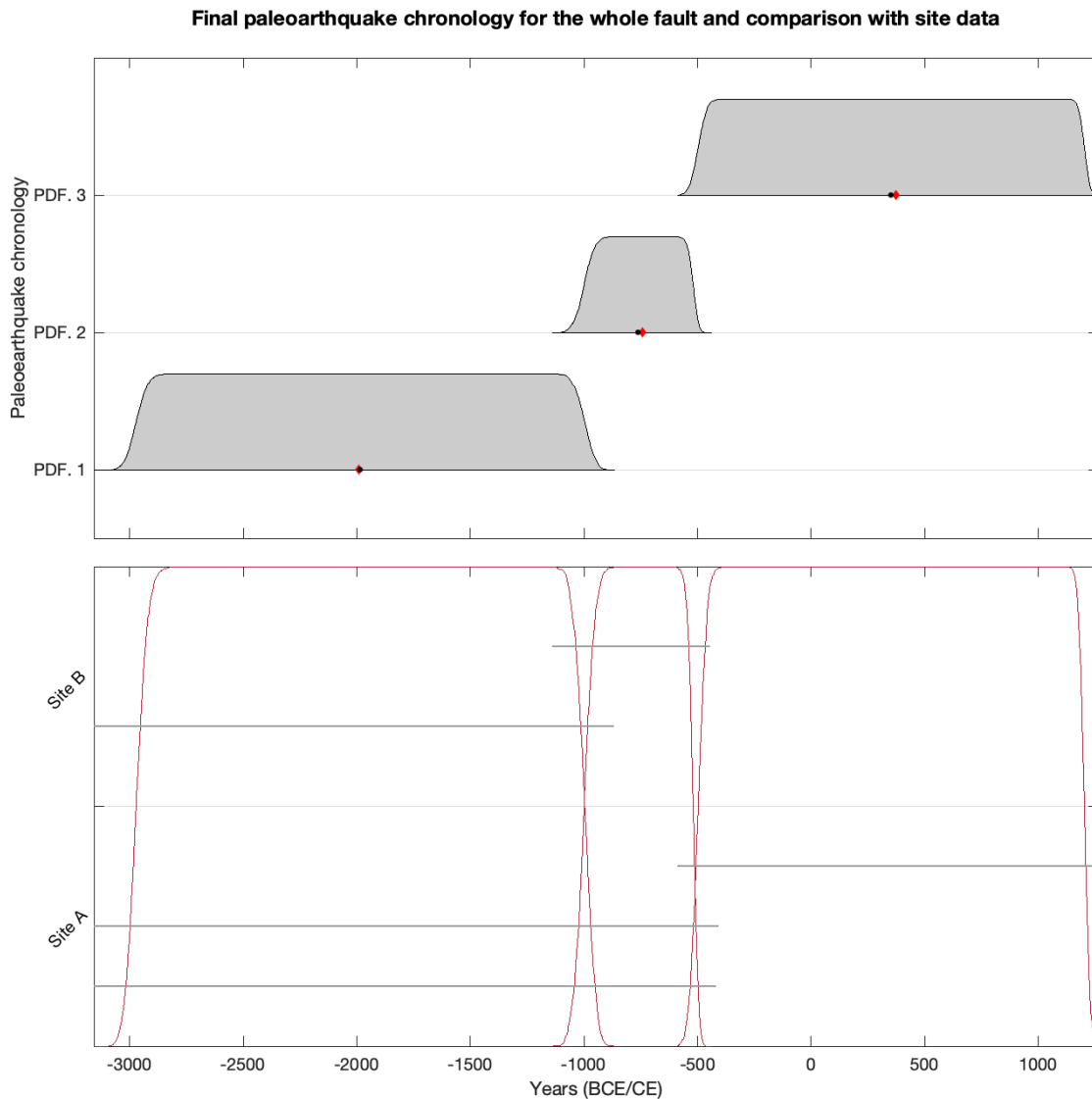
The peaks identified in the previous step are used to derive PDFs that are representative of the event in that position. To do so, the code extracts all the individual site event PDFs that intersect each peak position and multiplies them to obtain a final PDF. The full set of final PDFs derived from each peak represent the paleoearthquake chronology of the studied fault (Fig. 4).

In some instances, in a same site two or more event PDFs can have large overlaps or even the same age constraints. This can happen when there is clear stratigraphical evidence of  $>1$  event, but not enough dates are available to individually constrain each one of them. In these cases, the code cannot distinguish peaks for each event and ends up returning a single final PDF. To solve this, the code identifies when a final PDF is formed by two or more event PDFs from a same site that overlap significantly in time. This overlap is defined by when both the mean and -1 sigma end of one event PDF fall within the 1 sigma range of the other event PDF. Otherwise, is not considered an overlap.

For each final PDF, the site with the maximum number of effective overlapping events determines the number of events that form such final PDF. If there is more than one final PDF in the time span of those overlaps, the total



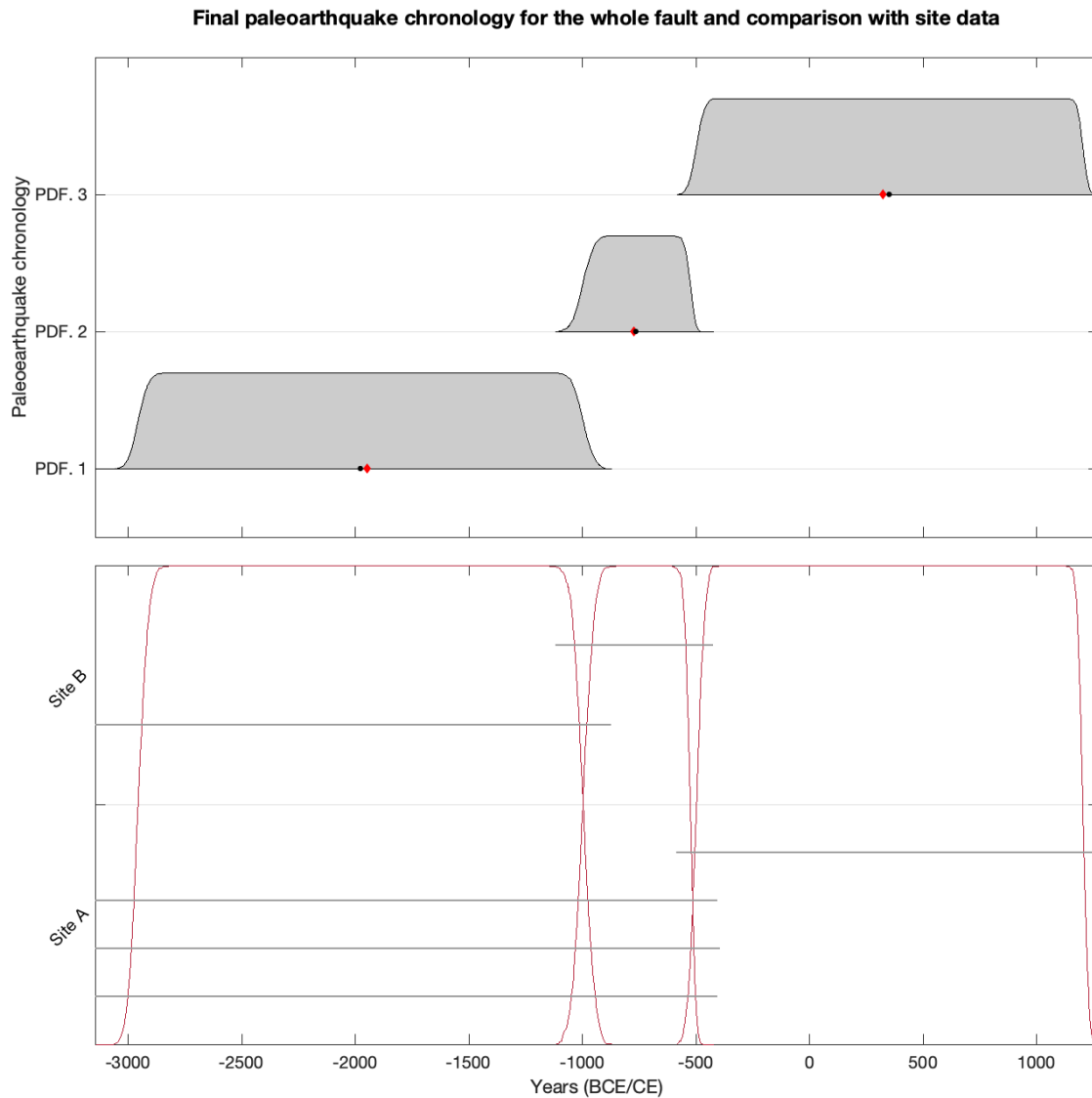
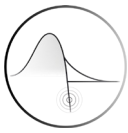
event count (e.g., number of overlaps) is distributed (divided) between the final PDFs. For instance, if in a site there are 2 overlapping events and 2 final PDFs for that time span, each final PDF would represent 1 event (Fig. 5). If this division does not lead to whole numbers, all combinations that add to the number of overlaps are computed. For instance, if instead of 2 overlapping events there are 3, because these have to be distributed into 2 final PDFs, 2 combinations of events are possible for each PDF (Table 2 and Fig. 6). It is the user the one that ultimately selects which combination is more feasible for the study area.



**Figure 5.** Final paleoearthquake chronology for the synthetic example in Table 1. The lower panel shows the final PDFs of the upper panel (red lines) with the full sigma time ranges of events (event PDFs) in each site.

	Events in final PDF 1	Events in final PDF 2
<b>Combination 1</b>	1	2
<b>Combination 2</b>	2	1

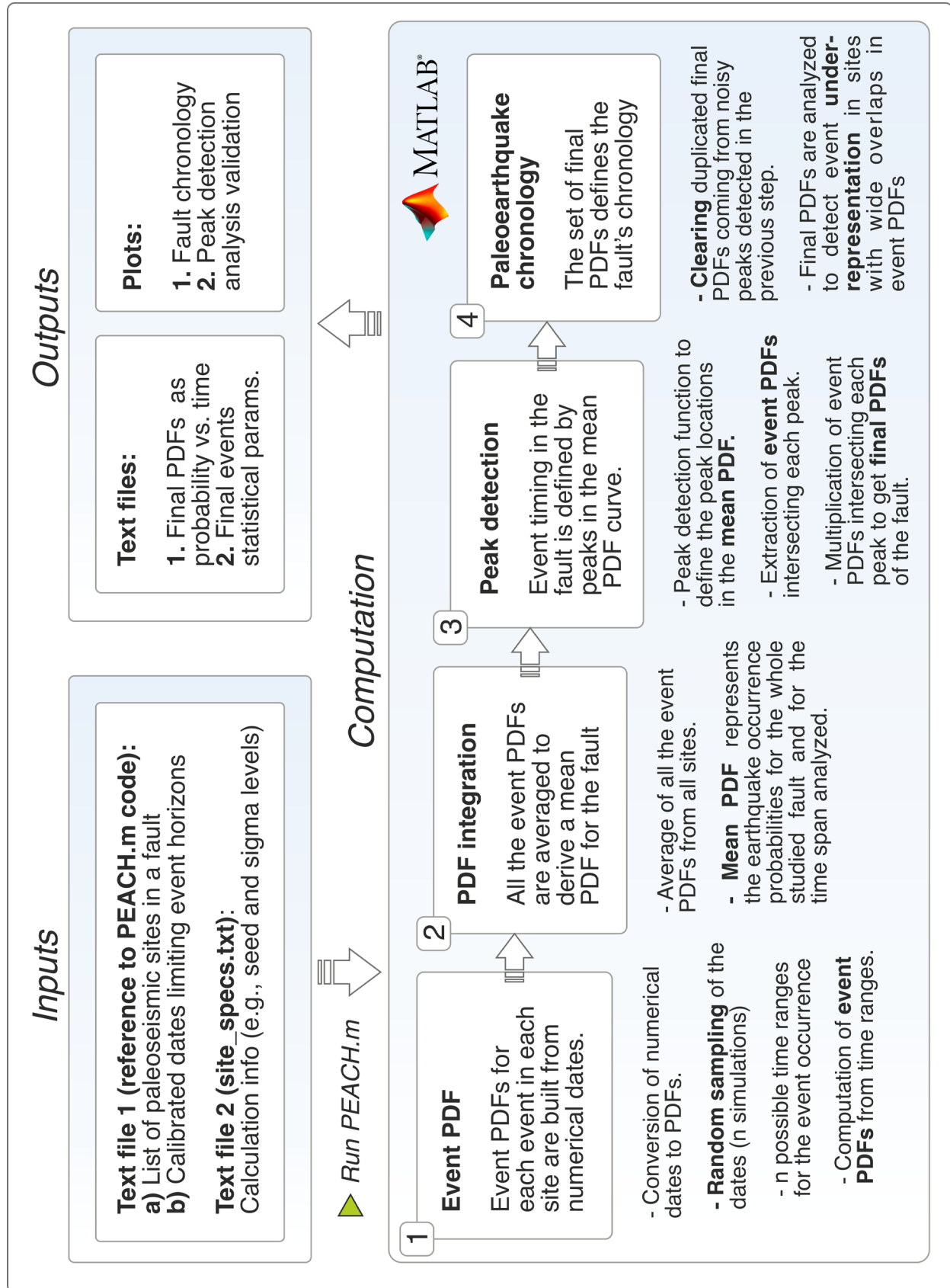
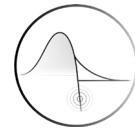
**Table 2.** Possible combinations of events in final PDFs 1 and 2 of figure 6. Note that the different combinations are mutually exclusive.



**Figure 6.** Example of Table 1 and figure 5 but with an extra overlapping old event in site A. Note that, in this case, because there are two final PDFs and 3 overlapping events, the number of events contained in each PDF is represented by the combinations in Table 2. Check the [GitHub repository](#) to access the calculation example of this figure.

Finally, the code also clears the influence of noisy minor peaks that might have been overdetect. This can happen in peak regions with jagged shapes, which are formed several minor peaks. Because these peaks belong to the same major peak, they are very close in time, meaning that they will likely intersect the same event PDFs and therefore outcome the same final PDF in the extraction. In these cases when there are repeated final PDFs the code keeps only one as representative of the true peak.

The chart in figure 7 summarizes the workflow followed in PEACH.



**Figure 7.** Workflow chart of the approach.



## 4. Input file preparation

Two input files are required to perform the computation. These inputs should be placed in the “Inputs” folder, following the structure available when downloading the code package from [GitHub](#).

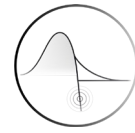
- a) A tabulated file (.txt or .xlsx) containing the calibrated numerical dates limiting each event horizon, following the format in Table 1. The limiting ages of the event should be the stratigraphically closer dates to the event horizon. Dates should also be stratigraphically coherent. Currently, the code does not allow to include phases of several numerical dates of a same unit (e.g., OxCal). The file name must be referenced at start of the PEACH.m code: `x_allsites = readtable(fullfile(pathin, “filename.txt”));`
- b) A tabulated file (site\_specs.txt) with information concerning the specific dataset that the user wants to calculate (Table 3):
  - a. Sigma level: the code allows to truncate the event PDFs to different sigma levels to compute the fault chronology. This is useful in the cases where PDFs have wide uncertainties, implying very low probabilities at the tips that might interfere the product calculation.
  - b. *Oldest unfaulted* date: to be provided only in faults where the youngest age of the last event is not available. In such cases, the user might specify a date that the code will automatically use (e.g., the oldest date of historical surface ruptures in the catalogue). It is mandatory if there is no young date available for the last event.
  - c. Standard deviation of the oldest unfaulted date.
  - d. *Oldest faulted* date: to be provided only in faults where the oldest age of the first event is not available. It can be based on regional data (e.g., the minimum age of the materials affected by the fault). It is not mandatory to provide this parameter, but please notice that if not provided (i.e., “nan”) the code automatically will assume that the oldest faulted date is the oldest date available in that trench or the oldest date in the whole dataset (all sites) in its default.
  - e. Standard deviation of the oldest faulted date.
  - f. Seed: this represents the number of Monte Carlo random simulations (samples) that the code will compute from the numerical date PDFs to build the event PDFs in each trench. Notice that the larger the number of simulations, the more detailed the shape of the PDF. However, it also means more computational time. Higher seeds (e.g., above 500) are recommended for complex datasets with a lot of events or sites to correlate.

The file name must be referenced at start of the PEACH.m code:  
`site_specs = readtable(fullfile(pathin, “site_specs.txt”));`

Calculation inputs (for the example in figure 6) are provided in the GitHub [link](#).

sigma_level	oldest_unfaulted	sd_unfaulted	oldest_faulted	sd_faulted	seed
0	1255	35	nan	nan	1000

**Table 3.** Example of the site\_specs input file. Time parameters follow the BCE/CE notation.



## 5. Output files

The outputs of the method are two types: visualization and data. These are stored in the “Outputs” folder, following the structure available when downloading the code from GitHub.

For the visualization of the model two figures are provided:

- Figure of the final paleoearthquake chronology (e.g., figures 5 or 6) where the user can visualize all the PDFs of their model and how they compare with the real data in the trenches.
- Figure of the peak detection in the mean curve (Fig. 4). The purpose of outputting this figure is so the user can visualize that all peaks corresponding to event occurrences are detected correctly and that there are not major peaks missed.

For the data two output tables (.csv) files are provided:

- A file containing all final PDFs (probability as function of time) as well as the event combinations to highlight cases in which a PDF represents more than one event (Table 4).
- A file containing the statistical values that represent each final PDF (mean and 25, 50 (median) and 75% quantiles) as well as the event combinations for each one (Table 5).

The aim of the first table is to provide raw data in case the user wants to do further statistics with the PDFs (e.g., recurrence computations, etc.). The aim with the second is to provide an easy to read and simplified representation of the PDFs to be disclosed in publications, etc.

			Time (BCE/CE)			
PDF	N. events hyp.1	N. events hyp.2	-3146	-3145	-3143...-2151	-2150
1	1	2	$2 \cdot 10^{-9}$	$3 \cdot 10^{-9}$	...	0
2	2	1	0	0	...	0
3	1	1	0	0	...	0.001

**Table 4.** Example from figure 6 of the output file containing the raw data of the final PDFs, i.e., the probabilities as a function of time. The second and third columns indicate the hypotheses of event numbers contained in each final PDF (mutually exclusive one from another). From the fourth column on the probabilities of each PDF are represented as a function of the year (BCE/CE).

PDF	Hyp. 1	Hyp. 2	Mean	1 $\sigma$	Q25	Q50	Q75
1	1	2	-1967.2	568.6	-2451.4	-1948.5	-1462.9
2	2	1	-774.4	141.8	-897.5	-774.5	-648.6
3	1	1	330.5	494.4	-103.4	322.1	756.8

**Table 5.** Example from figure 6 of the output file showcasing the final PDF statistics. Note that this output assumes a symmetric normal distribution of the PDF. Ages are in the BCE/CE notation.



## 6. Result interpretation

The earthquake chronology provided by the code can be interpreted as the minimum number of paleoearthquakes that the studied fault has produced for the observational time period analyzed. In no case the number of events constitutes a maximum because paleoseismic data always provides a minimum estimation.

### Limitations

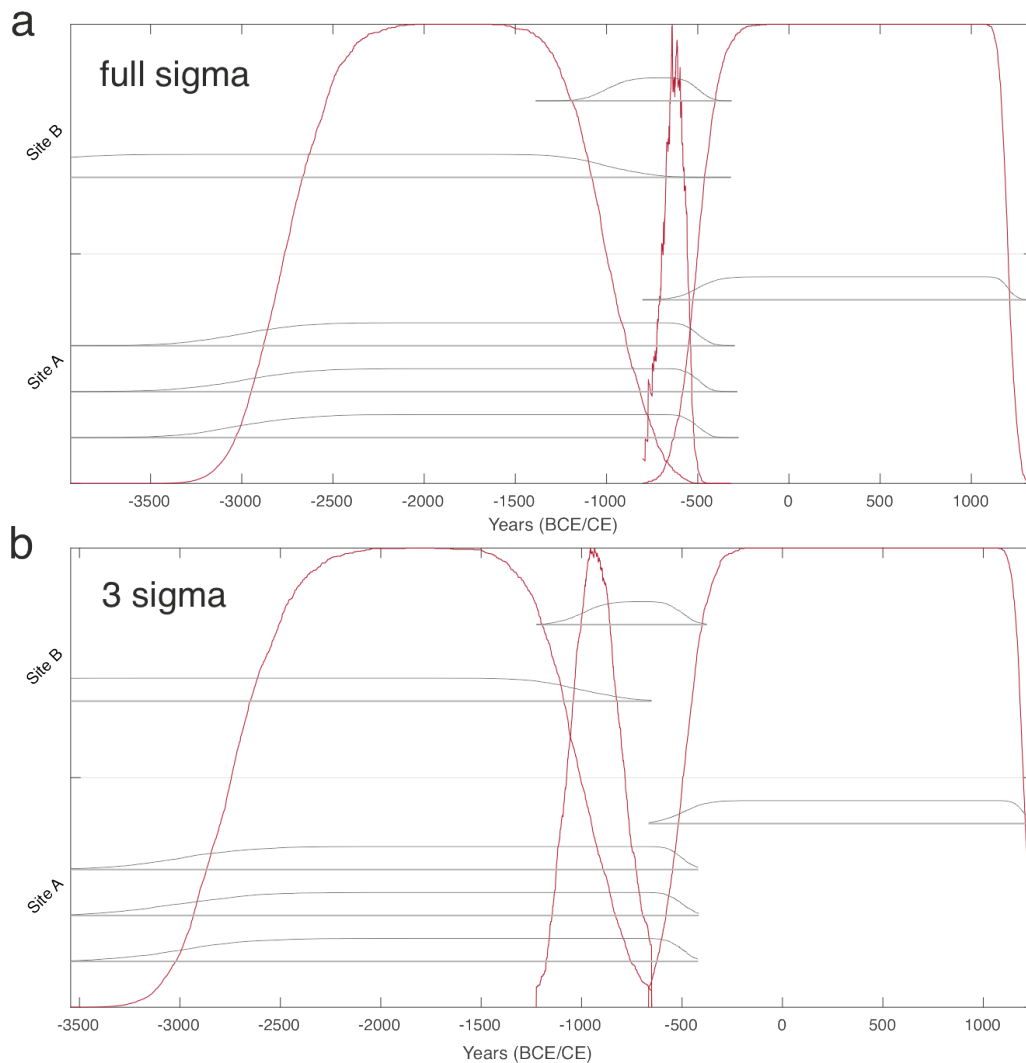
For datasets with large uncertainties (e.g., trench events with wide event PDFs due to poor age constraints), the approach can fall into overconstraining the final chronology. This is because during the product computation (part IV of the method), the coda regions of PDFs with low probabilities (near to 0) intersect the peak positions and lead to the narrowing of the final PDFs. In these cases, we recommend limiting the computation to a certain sigma level, instead of working with the full sigma range of the PDFs (usually 2-3 sigma yields good results). Thus, the user should check that the final chronology is not overconstrained and that provides a realistic interpretation of the original data.

Below we provide an example (Table 6 and Fig. 8) where overconstraining is addressed by reducing the sigma level in wide PDFs. In the first model (Fig. 8a) based on the data from Table 6, we work with the full sigma range and therefore the resulting PDFs, especially the middle one, are over-constrained. This is because the peak position of this event intersects the tips of the event PDF in site B, which have very low probabilities (close to 0). Instead, if we truncate the event PDFs to  $3\sigma$  we observe that the middle PDF in the final chronology shows a more reasonable constraining and smoother shape based on uncertainties of the data in each site (Fig. 8b).

Event ID	Site ID	Old	Error ( $1\sigma$ )	Younger date	Error ( $1\sigma$ )
a1	Site A	-500	100	1200	40
a2	Site A	-3000	300	-500	60
a3	Site A	-3000	300	-500	60
a4	Site A	-3000	300	-500	60
b1	Site B	-1000	100	-500	60
b2	Site B	-4300	350	-1000	200

**Table 6.** Input data for the example of figure 8. This case is the same as Table 1 (with one extra event), but with the uncertainties of the numerical dates artificially widened to accentuate the over-constraining feature explained in this section.

Another important limitation is related to the automatic peak detection of the code. So far, the detection threshold is set by trial-error analysis in a limited number of real and artificial datasets (between 10 and 15) of different tectonic settings. Although this threshold allowed a successful peak detection in all cases tested, we cannot discard that it might show issues in other datasets. For this reason, the user should always check the peak detection figure of the outputs to validate if there are any major peaks (e.g., peaks related to true event occurrences) that have been missed out. The user should also check for over-detection.



**Figure 8.** **a)** Earthquake chronology computed with the full sigma range of the event PDFs. Note that the second PDF is very constrained and displaced to younger ages due to influence of the low probability (around 0) region of the older event PDF in site B. **b)** Earthquake chronology computed with the 3 sigma range of the event PDFs. Note that in this case the second PDF is less constrained and shows a shape that is in better agreement with the uncertainties of the original data. The sigma truncation removes these low probability areas of the PDF in site B.

- What to do in case of underdetection?

Underdetection is the most important problem to be encountered during the analysis, as it will likely yield to an underestimated earthquake catalogue. When the total number of events is smaller than the events in the most populated site a warning appears in the Command Window. Two main causes can produce it:

- a) Too high prominence threshold. In this case the user should access the code and manually change this threshold (Fig. 9). The values should be lowered with caution, as a too small threshold can also lead to overdetection. We recommend starting by reductions of 0.01 (1% prob.).
- b) Wide uncertainties in event PDFs and important overlaps between them. In such cases the mean PDF might not show prominent enough peaks to be detectable. Therefore, we recommend working with truncated PDFs (2-





3 sigma), as they will allow better highlight the presence of each individual PDF and reduce some of the overlap.

```

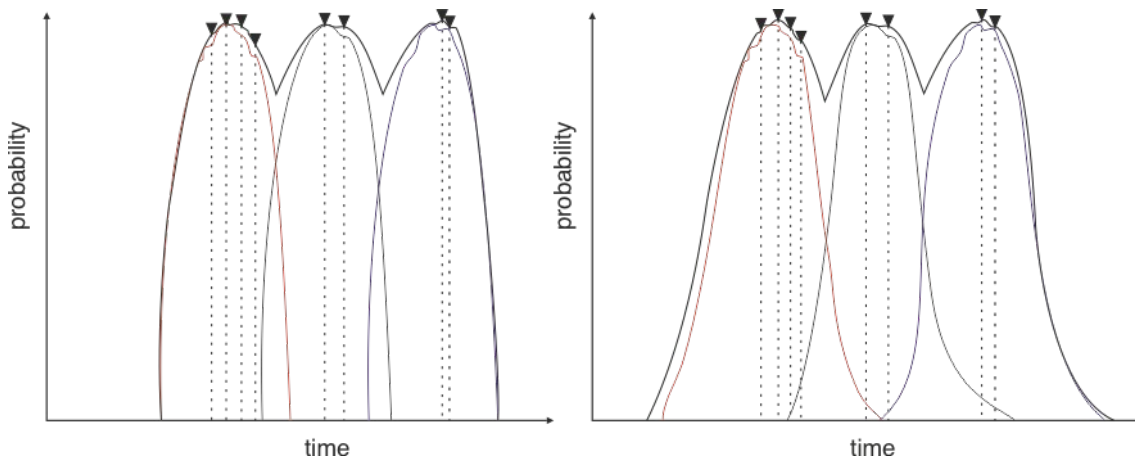
144 %% Find peaks in the mean probability curve
145
146 %Using the findpeaks function from MATLAB in both directions (left to right and right to left).
147 [peaks1, locs1, widths1, proms1] = findpeaks(tempnorm, time, "WidthReference", "halfprom", ...
148     "MinPeakProminence", 0.03);
149 pks = findpeaks(tempnorm, time, "Annotate", "extents", "WidthReference", "halfprom", ...
150     "MinPeakProminence", 0.03);
151 flipped_tempnorm = flipud(tempnorm);
152 flipped_time = fliplr(time);
153 [peaks2, locs2, widths2, proms2] = findpeaks(flipped_tempnorm, time, "WidthReference", ...
154     "halfprom", "MinPeakProminence", 0.03);

```

**Figure 9.** Section of the code (PEACH.m) where the prominence threshold is set (red squares). The numbers highlighted should all be replaced by the same value.

- What to do in case of overdetection?

Overdetection is usually a smaller problem in the approach because, as explained, the code is designed to clear the influence of noisy peak detections. However, it can happen that, in some cases, especially in datasets with large uncertainties, the clearance of noisy peaks is not exhaustive (Fig. 10). This is because, even if peaks are together in time, they can intersect different event PDFs if those have wide tips. In such cases we advise again working with truncated distributions. Bear in mind, though, that excessively truncated distributions can generate mean PDF curves with too jagged shapes that can induce overdetection. Therefore, the user should find a balance with the sigma truncation and shape of the mean PDF for optimal results. If the overdetection persists, the user may increase the threshold value in the section of the code shown in figure 9, but we do not recommend it right away.



**Figure 10.** Example of peak clearance problem in wide PDFs vs. well constrained PDFs. Note that there are 3 event PDFs (red, grey and purple) and the mean PDF (black thick line). Dashed lines represent the peak positions. The left example shows well constrained PDFs. In this case, a noisy peak detection will be successfully cleared because each group of noisy peak positions intersects the same PDFs. In the end 3 different peaks will be detected instead of 8. In the right example, because the event PDFs show wider uncertainties, not all noisy peaks in each group intersect the same PDFs; for instance, some peaks of the red PDF intersect the coda area of the grey PDF. This will lead to an unsuccessful clearance and will still return a higher number of peak values. In such cases, truncating the event PDFs can remove these undesired intersections.

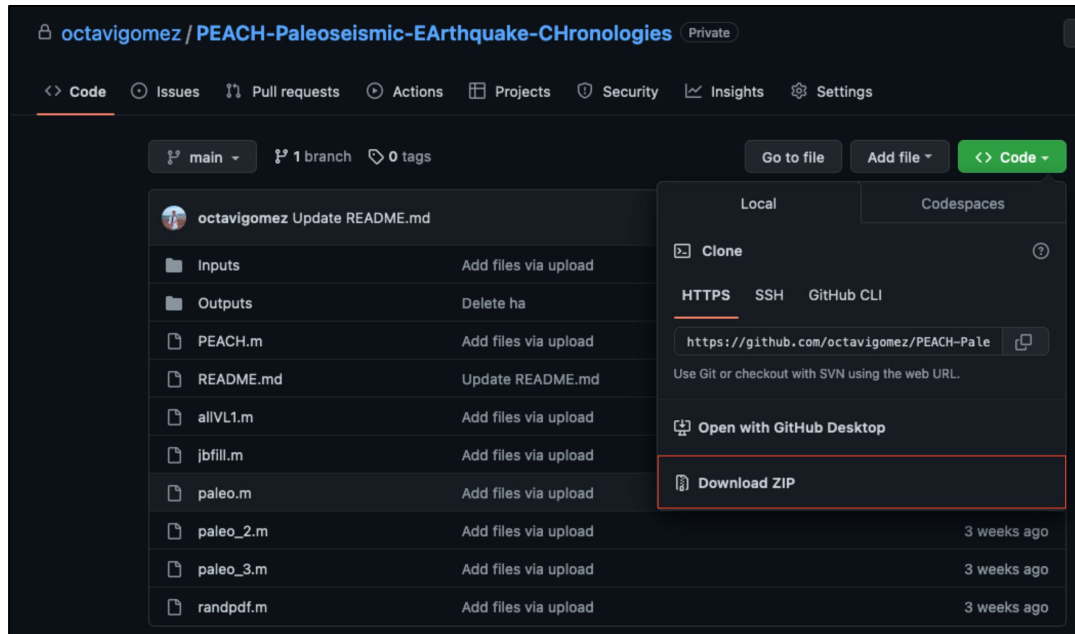


## 7. Requirements for the code

To run the code, no special requirements are needed except for a MATLAB license. The code runs easily on any laptop or computer and does not take longer than a few seconds to a couple of minutes to perform a computation. The computation time will depend on the detail in which the seed is set and the uncertainty widths of the event dates: the larger the higher the computational time.

## 8. Step by step guide to run the code

1. Go to the GitHub [link](#) and download the whole folder of the code (*PEACH-Paleoseismic-EArthquake-CHronologies-main*). Store it in your desired path. Make sure not to alter the predefined structure of the files and folders as downloaded.



2. Create the input files and store them in the *Inputs* folder. Follow the format described in this manual or take the preexisting example available in the *Inputs* folder as a reference.
3. Open the *PEACH.m* file in MATLAB to start running the code.
4. Reference the path and input files to the following variables of the code:
  - *pathin*: the path of the folder where the input files are located. By default:  
`pathin = "Inputs"`
  - *x\_allsites*: name of the file containing the event list and numerical dates.  
`x_allsites = readtable(fullfile(pathin, "filename.txt"));`
  - *site\_specs*: name of the file containing the site specifications.  
`site_specs = readtable(fullfile(pathin, "site_specs.txt"));`



```

11      %% Path and read input data from table
12
13      set(0, 'defaultFigureRenderer', 'painters')
14      set(groot,'DefaultFigureGraphicsSmoothing','off')
15
16      pathin = 'Inputs';
17      x_allsites = readtable(fullfile(pathin, "Example.xlsx"));
18      site_specs = readtable (fullfile(pathin,"site_specs.txt"));

```

5. Run the code by pressing the Run icon on top or by typing “run PEACH.m” in the Command Window.
6. When the calculation is finished, the command window will inform you. A summary of the calculation will be also printed as shown below.

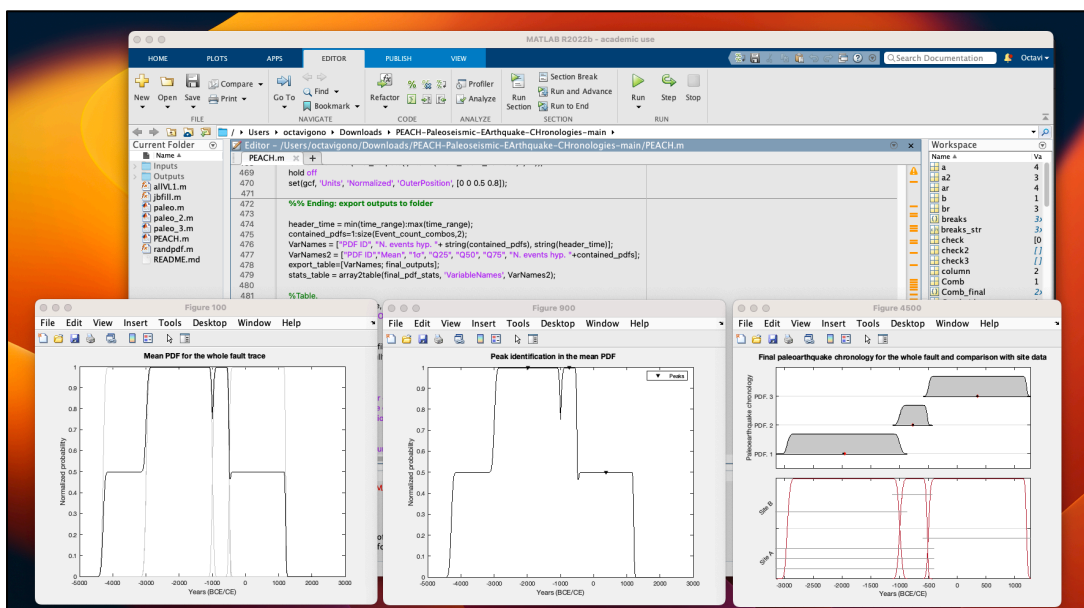
**Command Window**

```

>> run PEACH.m
Summary
- The total number of events identified in the fault is: 4
- These events are contained in a total number of final PDFs of: 3
For more information check out the outputs in the /Outputs folder.
fx >> |

```

7. When there is underdetection to the point that the total number of events is smaller than the most populated site in the inputs, a warning is printed in the command window: **Warning: There is a problem: the total number of final events is smaller than the number of events in your most populated site. Check user manual for guidance.** If this happens, please read section 6.
8. Go to the Output folders to retrieve the outputs of the calculation that describe your chronology model. Figures are outputted in .eps. Tables in .csv. When running the code, you will also be able to visualize, modify and save the output figures in MATLAB (see below).





9. Check the user manual's section 6 for result interpretation. If the results do not convince you and you need to re-adapt parameters such as the sigma truncation levels, make sure to change the input files accordingly and re-run the code. Note that the outputs will be overwritten with every new calculation, so make sure you save them separately in another folder.

\*\*\*\*\*

Please, for questions, bug reports and suggestions please feel free to write us:

Octavi Gómez-Novell: [octgomez@ub.edu](mailto:octgomez@ub.edu)

Bruno Pace: [bruno.pace@unich.it](mailto:bruno.pace@unich.it)

Francesco Visini: [francesco.visini@ingv.it](mailto:francesco.visini@ingv.it)