PEACH

Paleoseismic **EA**rthquake **CH**ronologies

User Manual

.

Version 1 (March 2023)

Octavi Gómez Novell, Bruno Pace, Francesco Visini

Update: March 2023

Disclaimer: this approach has been developed during the postdoctoral stay of Octavi Gómez Novell 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 an INGEO Department scholarship and a postdoctoral fellowship "Margarita Salas" of Octavi Gómez Novell, funded by the University of Barcelona with EU's Next Generation funds.

PEACH: Paleoseismic **EA**rthquake **CH**ronologies

User Manual – Version 1 (March 2023)

Table of contents

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 difficults 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 code also allows to use OxCal chronologies as inputs for the site correlations. The inputs and outputs of the approach are simple, easy to prepare and user friendly.

2. What are the fundaments of PEACH?

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

A. Paleoearthquakes 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 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.

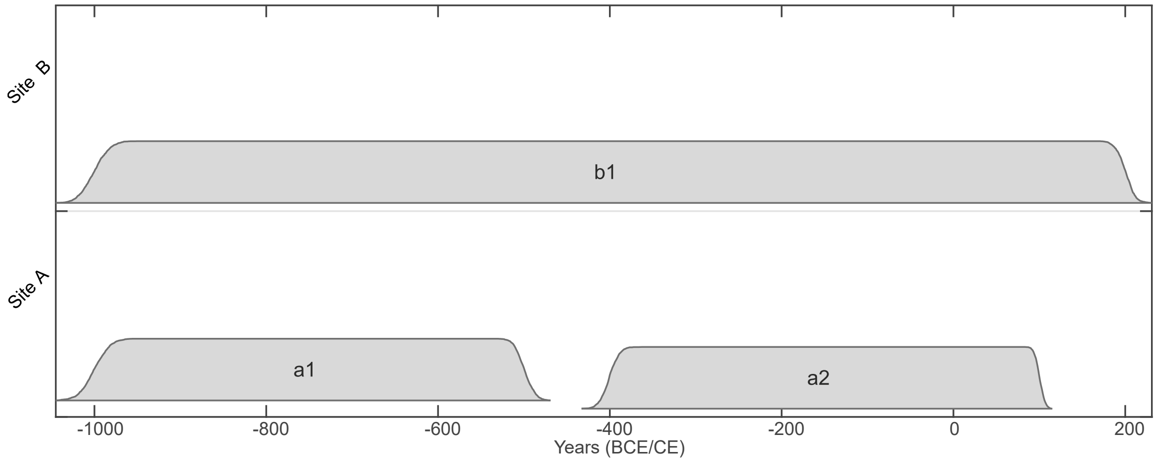


Figure 1. Synthetic example of a paleoearthquake record of two paleoseismic sites (A and B), susceptible to be correlated with the PEACH approach.

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.

Alternatively, the code also allows to use chronologies computed with OxCal as inputs for the site correlations. In such case, the input files should be prepared following the example in table 2.

The algorithm is divided into four steps that are detailed in the next page:

Event_num	Site	Event_date_old	Error	Event_date_young	Error
a1	Site A	-500	30	1200	0
a2	Site A	-3000	50	-500	30
a3	Site A	-3000	50	-500	30
a4	Site A	-3000	50	-500	30
b1	Site B	-400	50	900	10
b2	Site B	-2500	50	-900	50
b3	Site B	-4300	50	-2500	50
b4	Site B	-5600	30	-4300	50

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). Errors correspond to 1σ.

name	value	probability	site	n
E1	1200	0	Trench_1	1
E1	1195	1.58E-07	Trench_1	1
E1	1190	0	Trench_1	1
E1	1185	2.09E-07	Trench_1	1
E1	1180	1.03E-07	Trench_1	1
R1	1040	0	Trench_2	1
R1	1035	2.00E-05	Trench_2	1
R1	1030	0	Trench_2	1
R1	1025	5.15E-08	Trench_2	1
R1	1020	1.03E-07	Trench_2	1
R1	1015	5.15E-08	Trench_2	1
R1	1010	1.58E-07	Trench_2	1
R1	1005	1.58E-07	Trench_2	1
R1	1000	5.15E-08	Trench_2	1
R1	995	1.58E-07	Trench_2	1
R1	990	2.09E-07	Trench_2	1
R1	990	5.15E-08	Trench_2	1
R2	-1200	0	Trench_2	1

Table 2. Example of input file required for the computation in the case of using OxCal outputs. It corresponds to the raw data of the event PDFs in the OxCal chronology. The first three columns are directly provided by OxCal and represent the event name, year and probability, respectively. The “site” and “n” column represent the site name and number of events that the PDF in question represents. “n” should be kept as 1 unless there is a PDF that models the occurrence of more than one event. In this latter case, “n” corresponds to the number of events modelled in a same PDF.

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.

Note that this step of the methodology is overlooked in case of using OxCal chronologies.

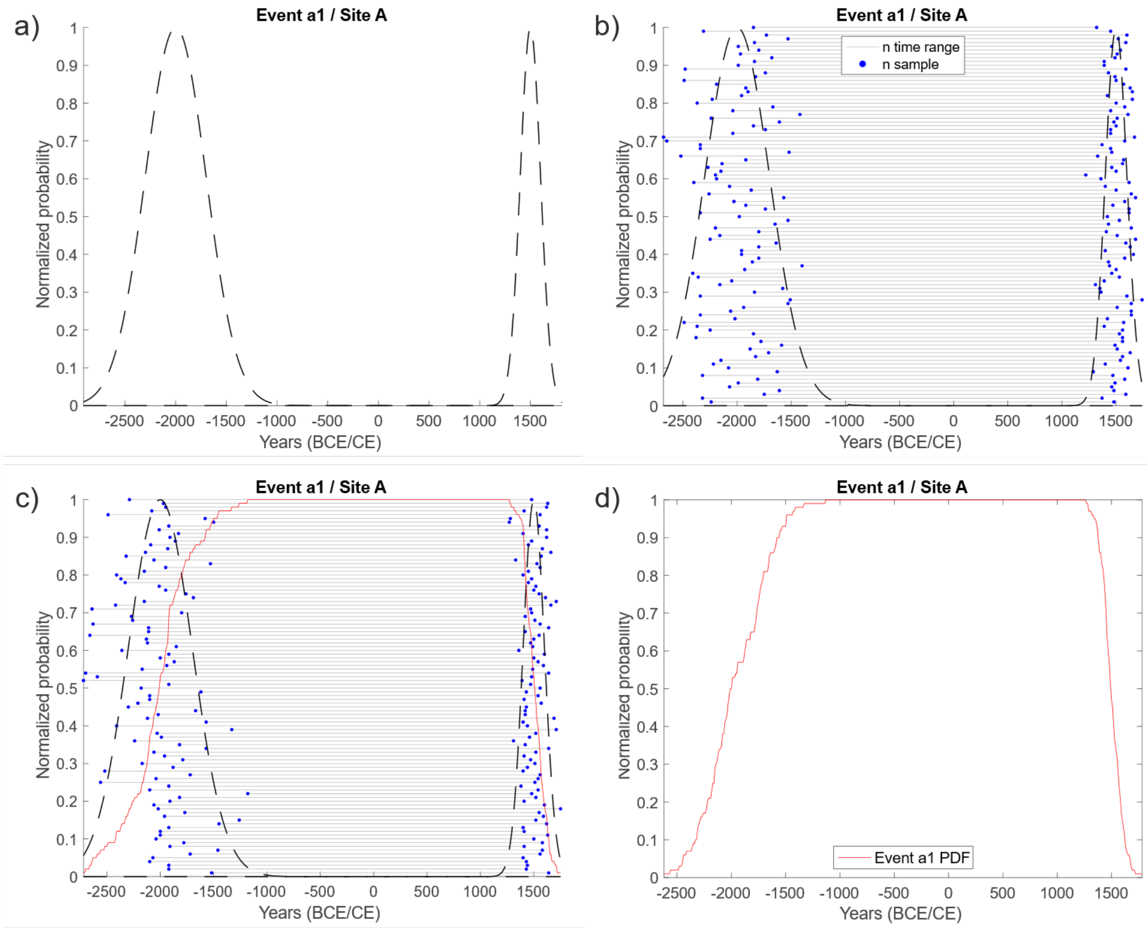


Figure 2. Visualization steps of the first part of the workflow for a hypothetical event a1: **a)** PDFs of the numerical dates provided in the inputs; **b)** random sampling of the PDFs of both numerical dates and establishment of time ranges that represent the time span for the paleoearthquake a1 occurrence. 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. 3). 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 (Fig. 4).

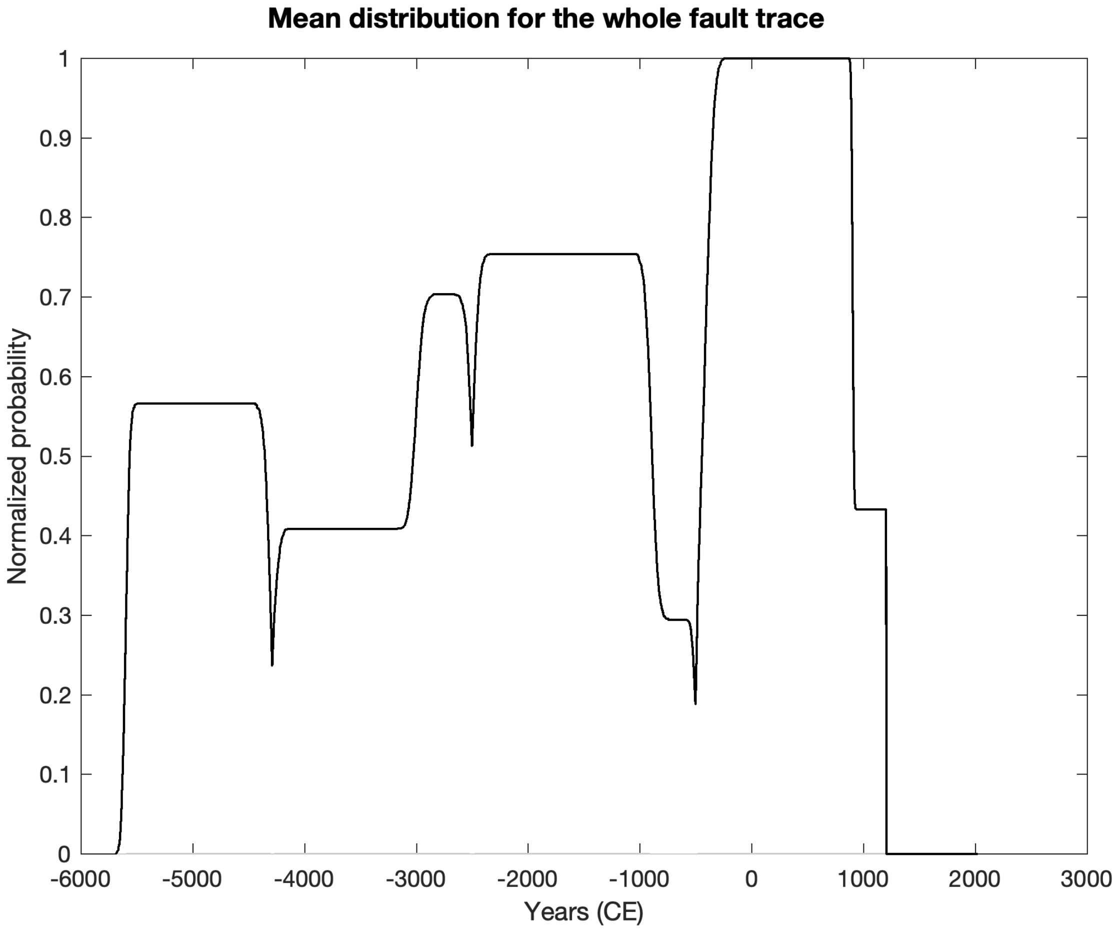


Figure 3. Normalized mean PDF obtained from averaging all the event PDFs from all sites within the studied fault trace (data from the example in Table 1).

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 by using the height of the individual PDFs in the dataset and Equation 1, where minP is the height of the flatter PDF and maxP is the height of the better constrained PDF.

$$minprom = (1/2\ minP)/(1/2\ maxP) \qquad \text{Equation 1}$$

Both parameters in Equation 1 are divided by 2 because they are measured in the mean probability distribution that averages all probabilities per time bin.

The *findpeaks* function identifies by default the peaks from left to right (older to younger ages). In plateau 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 (Fig. 4).

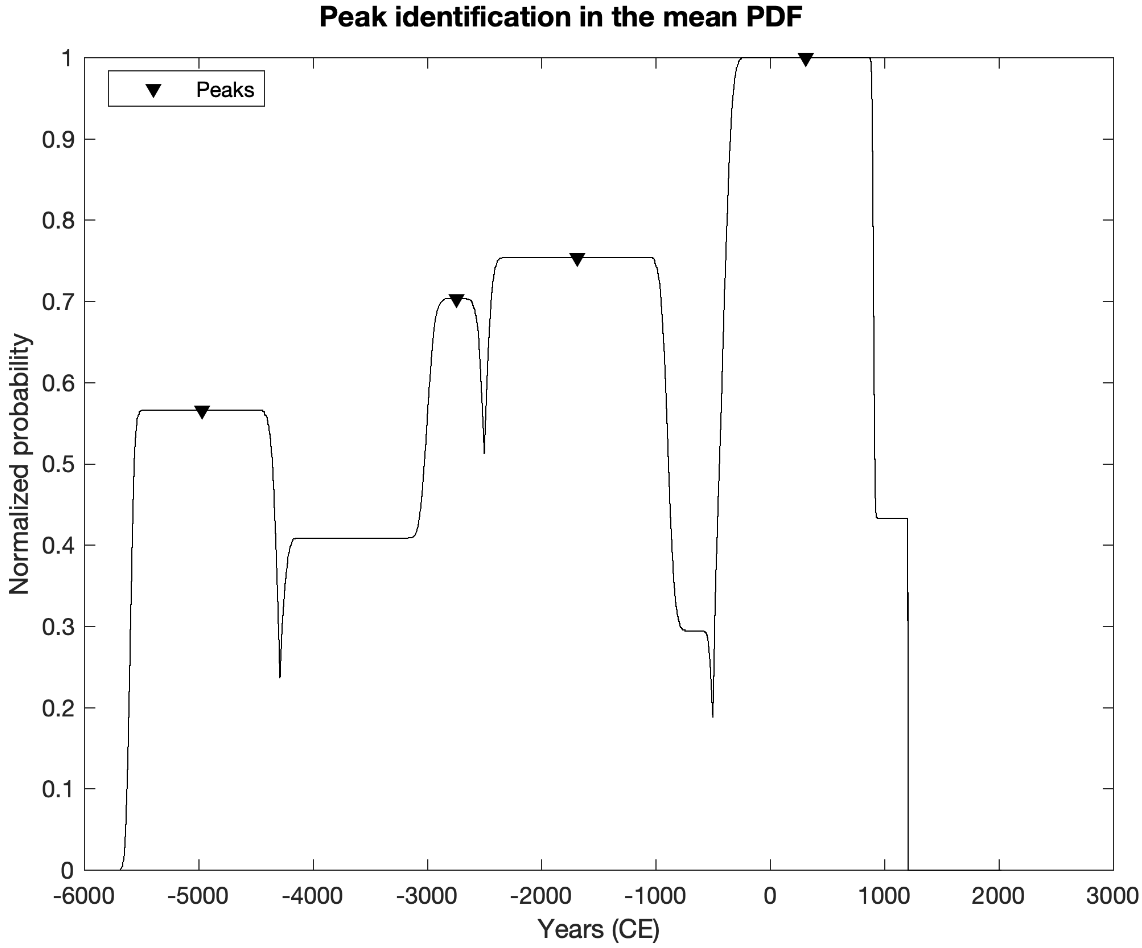


Figure 4. Peaks identified by the code in the mean PDF distribution. The time location of these peaks is used for the following steps of the approach.

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. 5).

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. If this division does not lead to whole numbers, all combinations that add to the number of overlaps are computed. For instance, if in a site there are 3 overlapping events and 2 final PDFs for that time span, 2 combinations of events are possible for each PDF (Fig. 5 and Table 3). It is the user the one that ultimately selects which combination is more feasible for the study area.

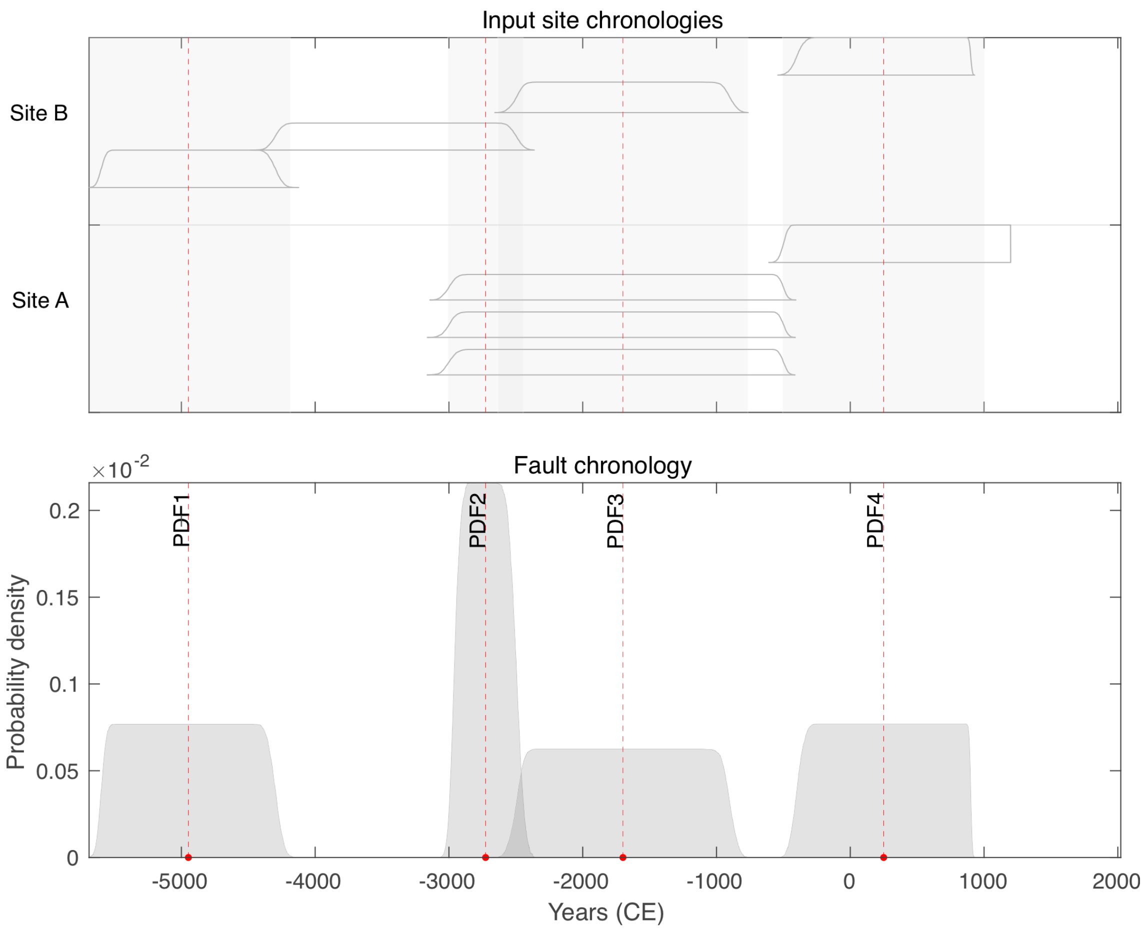


Figure 5. Example of Table 1. Note that, because there are two final PDFs and 3 overlapping events, the number of events contained in each PDF is represented by the combinations in Table 3. Check the GitHub repository to access the calculation example of this figure.

	Events in final PDF 2	**Events in final PDF 3**
Combination 1	1	2
Combination 2	2	1

Table 3. Possible combinations of events in final PDFs 2 and 3 of figure 5. Note that the different combinations are mutually exclusive.

Finally, the code also clears the influence of noisy minor peaks that might have been overdetected. 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 the code keeps only one as representative of the true peak.

The chart in figure 6 summarizes the workflow followed in PEACH.

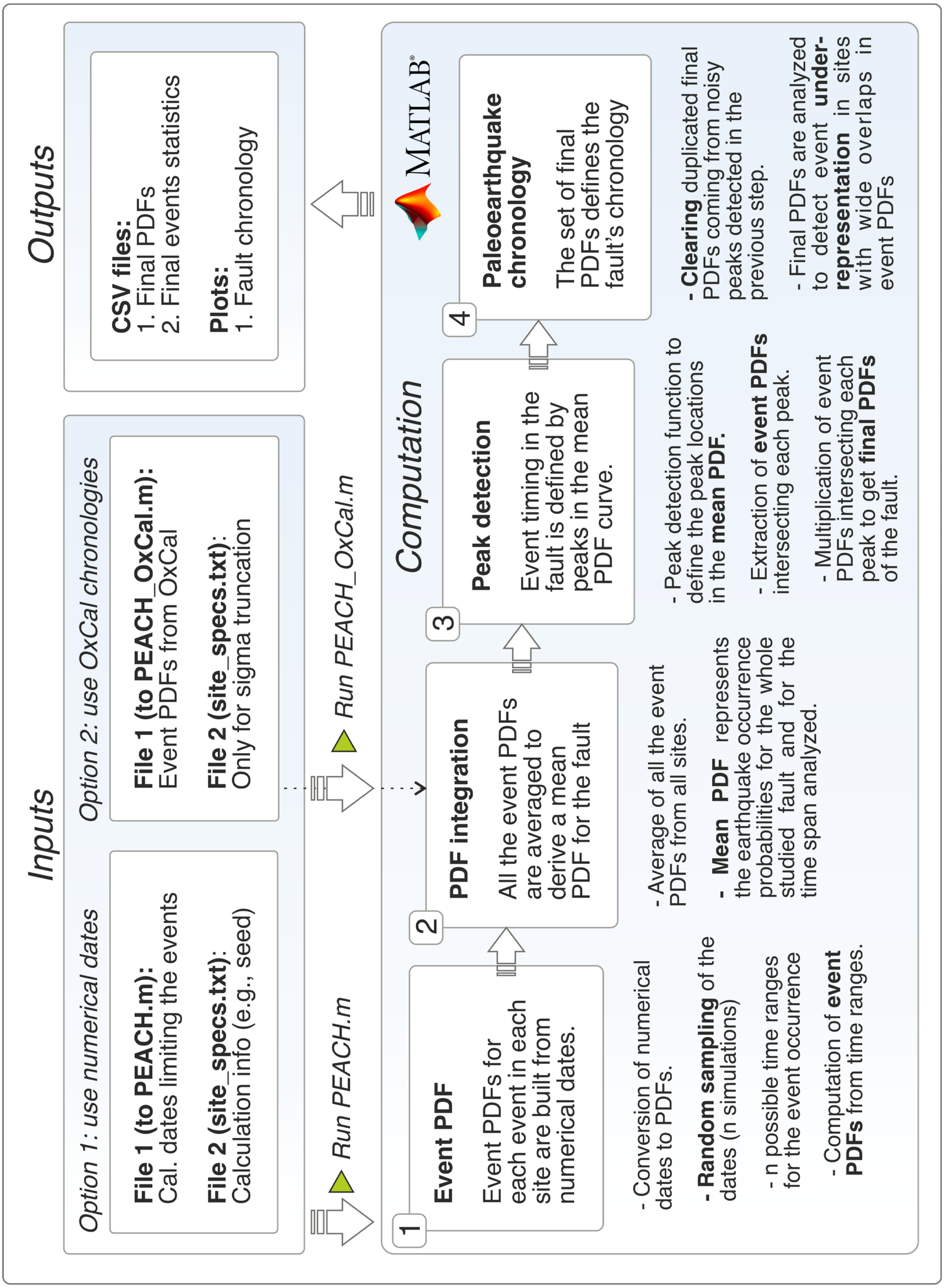


Figure 6. Workflow chart of the approach.

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 file (.csv 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.csv**"));* If you intend to use OxCal chronologies directly, you need to prepare the file (.csv or .xlsx) following the format in Table 2 and refer it to the PEACH_OxCal.m code.
b) A file (site_specs.txt) with information concerning the specific dataset that the user wants to calculate (Table 4):
    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 to the PEACH.m or PEACH_OxCal.m codes: *site_specs = readtable(fullfile(pathin, "**site_specs.txt**"));*

sigma_level	oldest_unfaulted	sd_unfaulted	oldest_faulted	sd_faulted	seed
0	1200	0	nan	nan	1000

Table 4. 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 a figure is provided containing:

- Plot of the final paleoearthquake chronology (e.g., Fig. 5) where the user can visualize all the PDFs of their model and how they compare with the real data in the trenches.
- Plot 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 5).
- 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 6).

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**	-5679	-5678	…	930
1	1	1	7.72e-07	7.72e-07	…	0
2	1	2	0	0	…	0
3	2	1	0	0	…	0
4	1	1	0	0		7.76e-07

Table 5. Example from figure 5 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	**Mean**	**1σ**	**Q25**	**Q50**	**Q75**	**Hyp. 1**	**Hyp. 2**
1	-4963.9	372.8	-5291.4	-4969.9	-4642.0	1	1
2	-2727.1	140.0	-2847.2	-2732.4	-2611.1	1	2
3	-1705.3	458.5	-2110.3	-1700.3	-1306.1	2	1
4	249.0	369.9	-60.5	244.3	566.2	1	1

Table 6. Example from figure 5 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 7 and Fig. 7) where overconstraining is addressed by reducing the sigma level in wide PDFs. In the first model (Fig. 7a) based on the data from Table 7, 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 3σ 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. 7b).

Event_num	Site	Event_date_old	Error	Event_date_young	Error
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 7. Input data for the example of figure 7. In this case we widened the uncertainties of the numerical dates to accentuate the over-constraining feature explained in this section.

Another important limitation is related to the automatic peak detection of the code. Because this threshold depends on the data itself (event PDF heights) and the shape of the mean distribution, we cannot discard that it might show issues for a correct peak detection in some 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 overdetection.

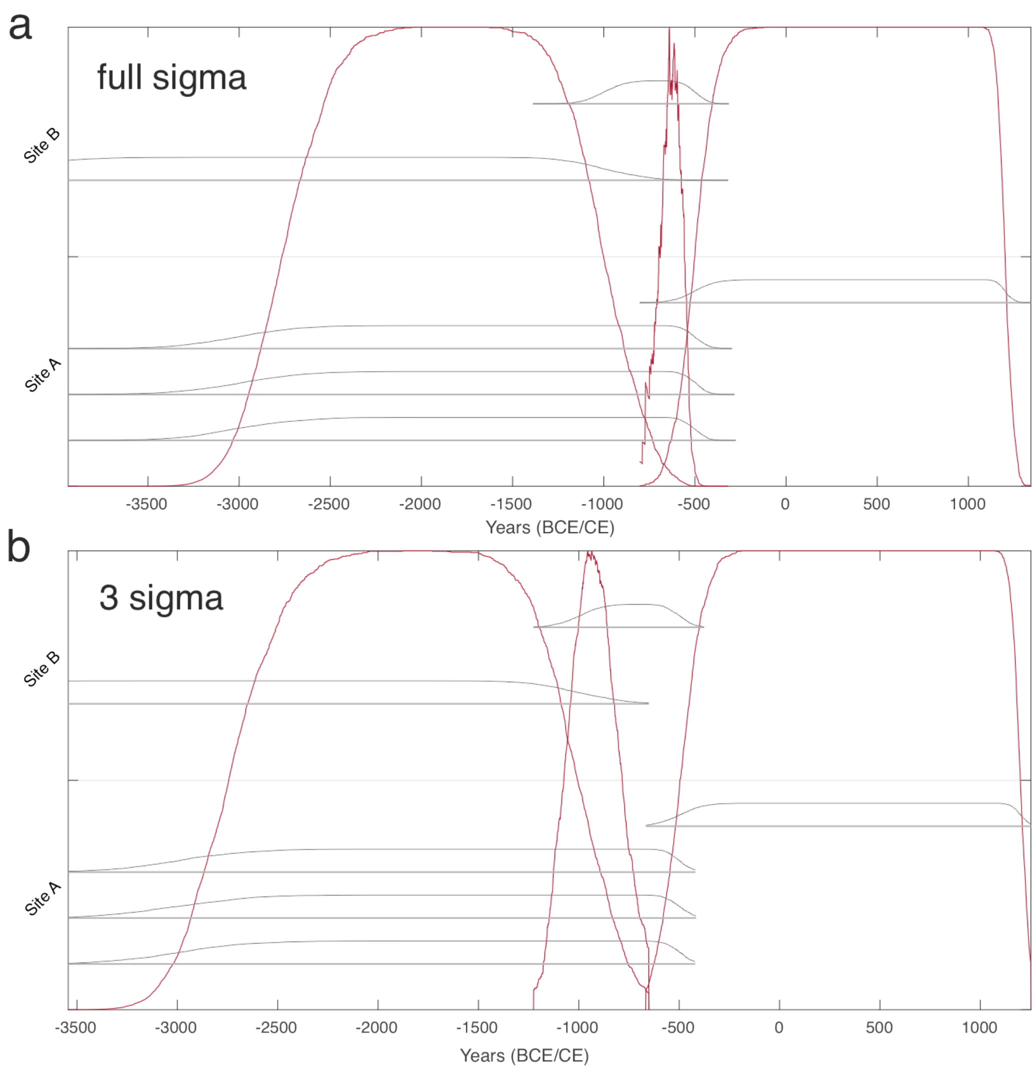


Figure 7. **a)** Earthquake chronology computed with the full sigma range of the event PDFs. Note that the second PDF is very constrained and displaced to younger ages due to influence of the low probability (around 0) region of the older event PDF in site B. **b)** Earthquake chronology computed with 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. The main cause of this is the poor peak definition in the mean probability curve due to highly overlapping events in the site chronologies. In this case, we recommend working with truncated PDFs (2-3 sigma), as they will allow to work only with the higher probability regions of the PDFs and sharpen their footprint in the mean distribution.

If underdetection persists, the user can access the code and modify the prominence threshold (Fig. 8). To do so the variable “min_prom” needs to be changed by a numeric value (a minimum probability prominence to be detected).

```
%Using the findpeaks function from MATLAB in both directions (left to right and right to
[peaks1, locs1, widths1, proms1] = findpeaks(tempnorm, time, "WidthReference","halfpr
   "MinPeakProminence",min_prom);
pks = findpeaks(tempnorm, time, "Annotate","extents", "WidthReference","halfprom", ...
   "MinPeakProminence",min_prom);
flipped_tempnrom =flipud(tempnorm);
flipped_time = fliplr(time);
[peaks2, locs2, widths2, proms2] = findpeaks(flipped_tempnrom, time, "WidthReferenc
   "halfprom", "MinPeakProminence",min_prom);
```

Figure 8. Section of the code (PEACH.m) where the prominence threshold is set (min_prom). To change the threshold the min_prom should be replaced by a numeric 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. 9). 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.

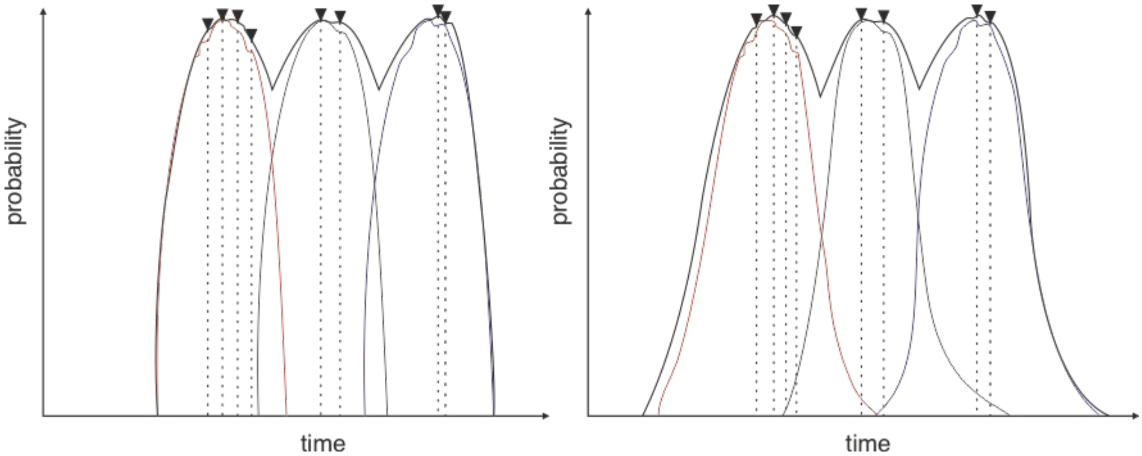


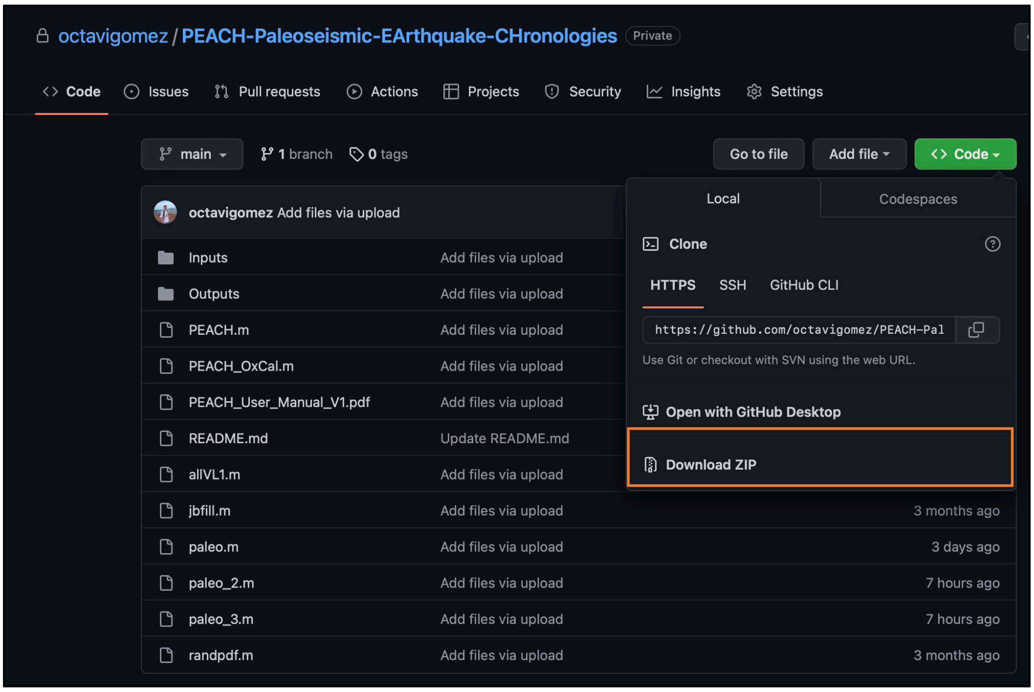
Figure 9. Example of peak clearance problem in wide PDFs vs. well constrained PDFs. Note that there are 3 event PDFs (red, grey and purple) and the mean PDF (black thick line). Dashed lines represent the peak positions. The left example shows well constrained PDFs. In this case, a noisy peak detection 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. If you use OxCal inputs open PEACH_OxCal.m instead.
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.csv**"));*
- *site_specs:* name of the file containing the site specifications.
  *site_specs = readtable(fullfile(pathin, "**site_specs.txt**"));*

```
%% Path and read input data from table

set(0, 'defaultFigureRenderer', 'painters')
set(groot,'DefaultFigureGraphicsSmoothing','off')

pathin = 'Inputs';
x_allsites = readtable(fullfile(pathin, "Example.csv"));
site_specs = readtable (fullfile(pathin,"site_specs.txt"));
```

If you use OxCal reference the following variables instead:

- *pathin:* the path of the folder where the input files are located. By default:
  *pathin = **"Inputs"***
- *Oxcal:* name of the file containing the OxCal event chronologies.
  *Oxcal = readtable(fullfile(pathin, "**filename.csv**"));*
- *site_specs:* name of the file containing the site specifications. In this case **only the sigma truncation levels** will be read by the code.
  *site_specs = readtable(fullfile(pathin, "**site_specs.txt**"));*

```
%% Path and read input data from table

set(0, 'defaultFigureRenderer', 'painters')
set(groot,'DefaultFigureGraphicsSmoothing','off')

pathin = 'Inputs';
Oxcal = readtable(fullfile(pathin, "Weber.csv"));
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"* or *"run PEACH_OxCal.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.

```
Summary
- The total number of events identified in the fault is: 5
- These events are contained in a total number of final PDFs of: 4
- The mean of 1σ's of the site event PDFs is 550 years
- The mean of 1σ's of the site event PDFs is 335 years
- The mean of the 1σ uncertainties in your final chronology are reduced by 39.1%
For more information check out the outputs in the /Outputs folder.
>>
```

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 .pdf. Tables in

.csv. When running the code, you will also be able to visualize, modify and save the output figures in MATLAB.

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.

9. References

Faure Walker, J., Boncio, P., Pace, B., Roberts, G., Benedetti, L., Scotti, O., Visini, F., & Peruzza, L. (2021). Fault2SHA Central Apennines database and structuring active fault data for seismic hazard assessment. *Scientific Data*, *8*(1), 1–20. https://doi.org/10.1038/s41597-021-00868-0

****

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

Octavi Gómez-Novell: octgomez@ub.edu
Bruno Pace: bruno.pace@unich.it
Francesco Visini: francesco.visini@ingv.it