Computers & Geosciences 182 (2024) 105496

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

A novel algorithm for seismic events multiplets search

Roberto Carluccio

Istituto Nazionale di Geofisica e Vulcanologia, via di Vigna Murata, 605, Rome, 00144, RM, Italy

ARTICLE INFO

Keywords:
Seismology
Search algorithms
Computer code
Multiple events

ABSTRACT

A common practice of seismology is to analyze earthquake occurrence in terms of events catalogues, with the aim to either find useful correlations between internal mechanisms under study and their outcome in the spatial/temporal series of the events or, more directly, to assess some statistical rules from observations. With this approach, catalogues are often searched for some recognizable patterns or behaviors: in this work we present a software tool created to reveal a particular kind of events sequences.

The idea follows from the concept of multiplets, a well known events pattern often found in seismic series. A multiplet is defined as a sequence of events, all near in space and time and exhibiting similar magnitudes. The amount of multiplets in seismic series is related, as it is for other clustering mechanisms, to underlying correlations in the physics of the events.

The software, built from scratch, scans seismic catalogues in search of events clustered as “multiplets”: this is done through the thorough application of comparison tests whose parameters thresholds are both user defined and semi-automated. The tool is however more “general” in the sense that by varying values of the filtering parameters it can reveal other kind of patterns too.

While we think that this tool can be thought as a general purpose space–time series analyzer, we have found it particularly useful when applied to the results of a seismic simulator with the purpose of assessing their adherence with the observed seismicity. It can be used as a sort of metric to quantify the simulation predictions effectiveness in terms of presence of similar multiplets distributions in simulated vs. real catalogues.

The software has been entirely developed in the Wolfram Language (Mathematica), a commercial powerful environment for scientific calculus and results report, but the main computational routine has been also ported to python for open-source, copyleft usage.

1. Introduction

1.1. Seismology as the science of complexity

In seismology, earthquakes are usually analyzed from the perspective of the relations existing among events occurrences, they are organized in temporal and spatial databases, the “seismic catalogues”. Their study and analysis are aimed to reveal every indirect information useful to shed some light on the elusive, inner physics mechanisms acting in the seismogenic process. Earthquakes are in fact the most dramatically evident and measurable effect of a complex dynamics evolution in the foremost upper, solid layer of the Earth. It is a final act in a process of stress and deformation accumulation, it is the failure of the solid rock matrix, the result of complex physics acting in a subtle, dense web of solid rock faults in perennial rewriting.

The deformation and rupture dynamics involved are of a formidable complexity and do depend on physical phenomena acting both in space and time and spanning over various orders of magnitude, from local molecular interactions to earth plate dynamics, and from milliseconds to centuries and more. The system behavior is so complex to be inherently chaotic, and is believed to be part of the wide family of “self-organized critical systems” (this term was first introduced in 1987 (Bak et al., 1987)). For this kind of systems that seem to be ubiquitous in nature, there are no ways to find predictive deterministic solutions, they can be only treated in probabilistic terms and through the fingerprint of evidence that those systems exhibit: power laws distributions and fractal behavior.

1.2. The concept of event

In this perspective, even the concept of “single seismic event” is hard to define itself: we actually wrap the “event” concept in a temporal bound, whose span is somehow related to our time perception of the energy paroxysmal release or more simply to our instrumental limitations. What we say to be “one event” it is more likely a granular

E-mail address: roberto.carluccio@ingv.it.

https://doi.org/10.1016/j.cageo.2023.105496
Received 17 January 2023; Received in revised form 10 June 2023; Accepted 20 November 2023
Available online 28 November 2023

composition of several, shorter inter-dependent sub-events, similar to those firstly studied by Mandelbrot in his early work on fractals (Berger and Mandelbrot, 1963). At the same time, the opposite perspective is also valid: our "single event" can be seen as a sub-event, a member of a cluster of events part of which already occurred and part to eventually occur in the future. In any case, it is interesting to look at the problem through the frame of "time occurrence self-similarity" and to verify if and how much these symmetries are followed.

1.3. In-dependence or inter-dependence

The earthquake dynamics act however on both time and space and the statistics need large, global systems and big datasets. From earthquakes studies it is well known that a regular, flat power law behavior in magnitude distribution over number of events (firstly evidenced by Gutenberg and Richter (1955)) generally needs big numbers, it emerges when large areas are considered. The "white spectrum" global aspect if looked locally shows "colored", uneven magnitude-frequency spectral domains, evidencing the local status of the rock fabric. Local time scale dynamics have furthermore been observed not to follow a strict time-independent Poissonian-like time sequence. Rather, events can deviate from that and show time correlations, quasi-periodical oscillations, resonances, renewal periods, recurrence times and so on. Correlations in earthquake events are sometimes evident enough to recognize patterns: they have been classified under different names (Utsu, 1970): earthquake clustering, foreshock–mainshock–aftershock sequences and swarms: in general local area earthquakes are recognized to be "space–time clustered" (Field, 2019). In summary, with independence we refer to the absence of direct influence or correlation between individual earthquake events while with interdependence we highlight the inner interconnected nature of earthquakes that emerges through their correlations and patterns in both space and time. What is seen in nature would suggest us that events are not randomly distributed but can occur (and they often do) in spatial and temporal patterns within localized areas.

1.4. Multiplets

An interesting typology of event clustering is that often known as 'doublets', 'multiplets' or 'multiple events'. It is relative to a specific type of a short-and-intermediate-term earthquake clustering, that is the occurrence of two or more earthquakes of similar and largest magnitudes in a limited space and time window.

The 'multiple events' term has however evolved through years: in the beginning it was used to identify groups of earthquakes having almost identical waveforms, originating from the same location (Poupinet et al., 1984). Here we will refer to a more modern definition of the term as a single sequence having two (or more) main shocks of similar magnitude, sometimes occurring within tens of seconds, sometimes separated by years (Beroza et al., 1995). The similarity of magnitude is the element that distinguishes multiple events from aftershock sequences, being aftershock magnitude typically thought to be smaller than that of the parent shock by about 1.2 magnitude units (Bath's law; (Vere-Jones, 1969)), and decreasing in frequency according to the Omori's law.

Multiplets (in this paper we will indifferently use the aforementioned definitions to refer to the same concept) are furthermore not rare: analyzing the distance between the epicenters of earthquakes temporally close to each other, Kagan and Jackson (1999) showed that about 20 per cent of very large earthquakes (magnitude above 7.5) are doublets, and that, in some cases, 37–75 per cent of earthquakes are multiplets.

There are different definitions of multiplets: a study concerning doublets and multiplets from the Harvard CMT catalogue in the Fiji–Kermadec-Tonga region was published by Gibowicz and Lasocki (2005). These authors defined a doublet as a pair of earthquakes (i) with a magnitude difference of no more than 0.25 units, (ii) whose centroids are separated by no more than 40 km for events with magnitude from 5.0 to 5.4, 60 km for events with magnitude from 5.5 to 5.9, and 90 km for events with magnitude equal to or greater than 6.0 and (iii) whose difference in occurrence time is no longer than 200, 300 and 450 days, respectively.

Multiplets clusters exist in nature and provide evidence of the aforementioned interdependence within dynamic local systems. Multiplets can serve, in our beliefs, as a valuable and specific fingerprint of patterns, dynamics, and hazards associated with seismic activity in specific regions.

2. The algorithm

In this work we present a filtering algorithm able to reveal seismic catalogue patterns like 'multiplets' or other clusterings types. Such a tool has been developed during the analysis of an earthquake simulator output data. The simulator tries to reproduce reality, it gets a geometric model of the simulated area faults, their associated deformation rates together with other few free parameters. It thus generates seismic catalogues that are usually compared with reality through an iterative loop in which free parameters tuning have feedback from some error metrics. The filter we are presenting has been used as part of our metric: compare multiplets distributions in both simulated and real catalogues. Some aspects of this work has already been reported in Console et al. (2020, 2022).

Our general belief is that when synthetic data is representative of the real world, at least to a certain degree, it can be really regarded as a promising way to operationally deal with earthquake forecasting and seismological risk assessment.

2.1. General features

The algorithm imports seismic catalogues in the form of CSV-like tables of events described by time, hypocenter coordinates and magnitude. Every table line is relative to an event E_i and the algorithm operates, at its innermost level, making systematic comparisons on time-ordered event couples E_i, E_j where time-ordered means that event time $t_{E_i} < t_{E_j}$ (hereafter we will always use this subscript convention).

In brief, E_i, E_j events are checked for the simultaneous satisfaction of some constitutive criteria in order to be flagged as "linked". Searched clusters are extrapolated from structures made of these linked $E_i \rightarrow E_j$ ordered couples.

The basic constitutive criteria are the following:

1. the first event magnitude of the sequence (hereafter named "pivot" event) must be greater than a given threshold;

$$M_i > M_{\mathrm{thr}} \tag{1}$$

2. the time difference between events must be less than a given functional value (interaction period: more on this later);

$$t_{E_j} - t_{E_i} < t_{GK}(M_{E_i}) \tag{2}$$

3. the distance between hypocenters must be less than a given functional distance (interaction distance: more on this later);

$$\left|\vec{r}_{E_i} - \vec{r}_{E_j}\right| \le \begin{cases} r_{GK}(M_{E_i}) & \text{(1st)} \\ \max[r_{GK}(M_{E_i}), r_{GK}(M_{E_j})] & \text{(max)} \\ r_{GK}(M_{E_i}) + r_{GK}(M_{E_j}) & \text{(sum)} \end{cases} \tag{3}$$

4. the magnitude value of any later (jth) event must lay within a given interval,

$$M_{\mathrm{ref}} - \Delta m_{(-)} < M_j < M_{\mathrm{ref}} + \Delta m_{(+)} \tag{4}$$

where positive and negative allowed magnitude differences $\Delta m_{(+)}$ and $\Delta m_{(-)}$ from M_{ref} are separately assignable and M_{ref} can be chosen between the pivot event magnitude (M_{pivot}, default case)

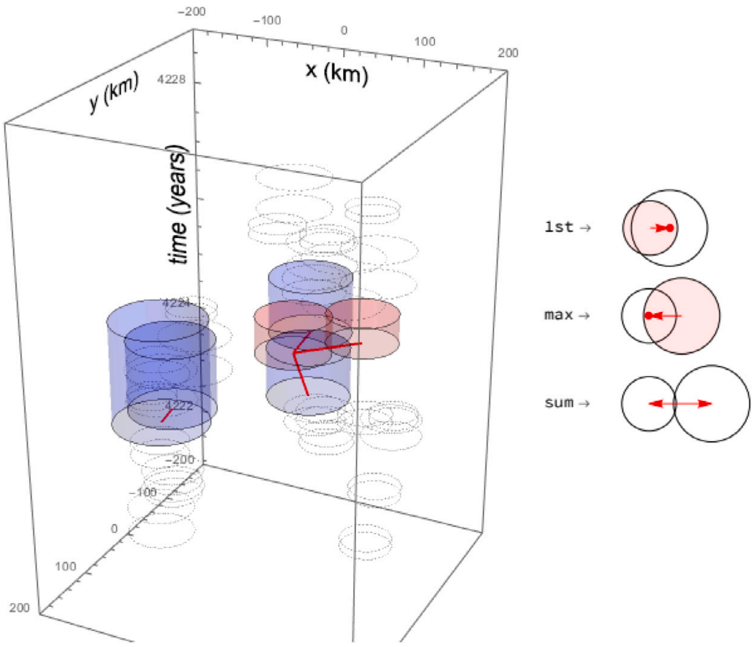


Fig. 1. Geometrical representation of criteria 2 and 3 as an intersection problem. In the shown portion of the catalogue, events are represented by cylinders of base and height respectively defined by 3 and 2. Concurrent fulfillment of both criteria is equivalent to cylinders intersection. The colored ones are those intersecting, the blue ones are potential pivots (criterion 1). In the right side of the plot the three levels of intersection of Eq. (3) are shown graphically.

or the earlier (ith) event magnitude M_i. The latter possibility is useful to investigate filter performances over aftershocks cluster search;

Threshold values in criteria 1 and 4 are arbitrarily chosen while criteria 2 and 3 values are instead calculated from event magnitude. We have used here two scalar functions, $t_{GK}(M)$ and $r_{GK}(M)$ giving, respectively, a time interval and a distance as a function of the event magnitude. They are generated by interpolation from the Gardner and Knopoff (1974) empirical table rules, but any other kind of function can be easily implemented.

Criteria 2 and 3 define a sort of overlapping region search in a 4-dimension space. Any event, originated in a $(t_i, \vec{r}_i)$ point, is extended through space (a spherical region of radius $r_{GK}(M)$) and time (a duration toward future of $t_{GK}(M)$) defining a "positioned" hyper-cylinder. Eq. (3) shows the 3 possibilities that can be used for condition on spacial interaction (levels of spheres intersection, showed in the right side of Fig. 1)

Fulfillment of criteria 2,3 among many events can be easily visualized as an intersection problem. Fig. 1 shows this in a 3D space reduced view, for clarity. Hyper-cylinders have been approximated with ordinary 3D cylinders by elimination of the z coordinate axis: the resulting view is anyway not too far from the actual case given by criteria 3 since variation of z coordinates are often much smaller than those on x and y. The plot shows a part of the event catalogue, each event being a cylinder. The colored ones are those that intersect and are eligible to be part of a multiplet if criteria 1 and 4 are also fulfilled.

The four threshold values described above physically define the characteristics and scope of the clustering analysis. They directly affect the size and extent of the search kernel used to identify correlations among elements in their multi-dimensional (time, position, magnitude) space. For our purposes, the magnitude threshold M_{thr} is chosen based on our synthetic catalogue completeness, usually around 5, and the allowed $\Delta m_{(\pm)}$ is approximately 0.5 to fit the definition of "multiplet". Time and space constraints along with their scaling with energy, on the other hand, are imported from existing literature. It is important to emphasize, anyway, that we utilized this analysis as a *comparative* metric. Independently from the actual parameters' choice, when comparing two seismic catalogues using statistics derived from multiplets, we make sure to consistently apply the algorithm using the same values sets.

2.2. Implementation

From Eq. (4) (in the default case only, where $M_{\text{ref}} = M_{\text{pivot}}$) and Eq. (1) follows that no events lower than $M_{\text{thr}} - \Delta m_{(-)}$ will be allowed. This permits to apply a pre-filtering phase to the catalogue that can sensibly reduce the number of events before entering in the main loop.

The main multiplets search loop procedure parses the catalogue in a cyclic sequence of three phases (ABC, ABC, …) that are repeated until the end of the catalogue is reached.

Fig. 2 shows this cycle on an hypothetical example and explains for each of the three A,B,C panels/phases the underlying concepts.

Phase-Panel (A) shows a scattergram of event times vs magnitudes. The process, in fact, begins by finding a pivot event, whose magnitude has to be greater than M_{thr} (criterion-1): this threshold is indicated in the plot by the red horizontal line.

Since all of the constitutive conditions criteria must be simultaneously satisfied, a pre-selection is needed to reduce computational costs. Phase A searches for a subset of events (hereafter named "pool") that begins from the pivot point and extends over following events until criterion 2 holds on a continuous adjacency basis: this is graphically shown in the (A) panel plot. A blue rectangle is drawn for each event point E_i; it extends from time t_{E_i} to $t_{E_i}+t_{GK}(M_{E_i})$: the searched "pools" are the "gapless" regions created by rectangles overlapping, indicated by orange segments on the scattergram time axis. In particular, the pool related to pivot #1 in Fig. 2 contains events #1 to #11. This reduces the next computation phase: for pools containing n events there are $n(n-1)/2$ couples to be analyzed.

Phase-Panel (B): all of the ordered couples to be systematically checked are represented in the upper inset of panel (B) of Fig. 2. The lower part shows as an arrows graph all $E_i \rightarrow E_j$ comparisons: those

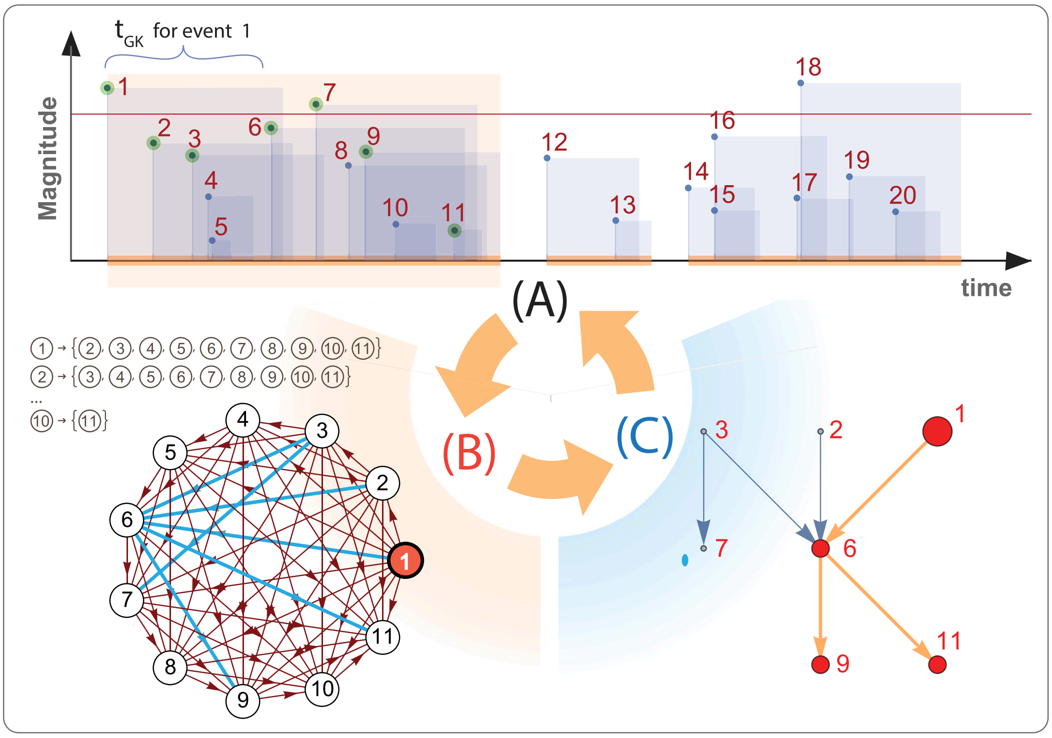


Fig. 2. Multiplets search algorithm sequentially scans a seismic catalogue passing through the three A, B, and C steps, repeating the procedure until catalogue ends. The top panel (A) is a time-magnitude-scattergram representation of the searched catalogue. The red horizontal line is the threshold magnitude for pivots search. The figure example starts from event 1 as first pivot. t_{GK} intervals (blue rectangles from dot events) form contiguous overlapping segments, shown as orange regions of time axis. The left bottom panel (B) shows how the A-to-B comparisons couples ensemble is generated and tested. A-to-B couples verifying 2, 3 and 4 criteria (blue arrows in left-bottom panel) generate the right bottom panel (C) graph from which the searched multiplet can be found. The A-B-C procedure starts again finding the next pivot, according to the search removal conditions user issued. For instance in top panel, removed events are marked with green dots and next pivot will be event #18.

that simultaneously satisfy the 2, 3 AND 4 criteria (matching couples) are the ones colored in blue.

Phase-Panel (C): how to use the matching couples information to properly choose the events forming the searched multiplet group. The algorithm can be more easily understood if the matching couples found are combined together to form the ordered acyclic graph reported in panel C of Fig. 2. In this view, the task is straightforward: it is an ordered graph traversal starting from the pivot point. Since criteria 2 and 4 can contain asymmetrical relations (i.e. t_{E_i} and t_{E_j} order cannot be reversed), the graph is searched respecting the edges orientations. More simply, starting from pivot point the algorithm searches all events that are reachable respecting arrows orientations. If such a graph branch does exist, (orange part of panel C graph) all related events are reported by the algorithm and grouped in a multiple-events cluster. A search for a new pivot (criterion 1) is thus issued and the whole procedure is repeated until the end of the catalogue.

An additional remark regards how the next pivot event search is done. In order to avoid to choose as new pivot an event already present in a previously recognized cluster, the algorithm has the possibility to remove from the next search some of the events already analyzed. Fig. 3 shows how this is operationally implemented in the algorithm through an indirect indexing scheme. An indexes vector (*id-vect*) is created upon catalogue import: the program points events only through *id-vect* components. In this way it is possible to remove catalogue events from the ongoing analysis on a random-access basis. Three choices have been foreseen:

1. no removal
2. removal of events present in the “2-3-4-connected” couples (i.e. the “matching couples” events, elements of the a-cyclic graph of Fig. 2, panel C)

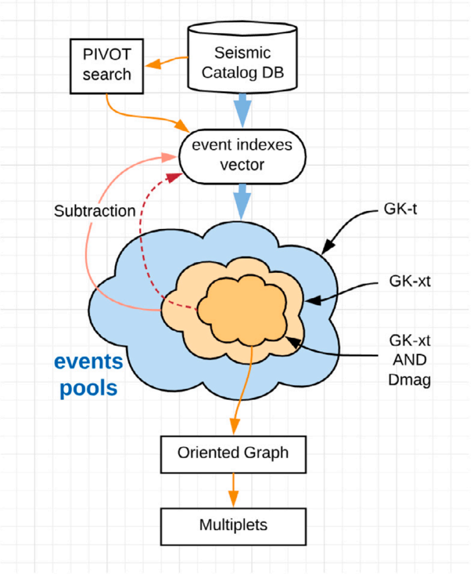


Fig. 3. Scheme of the logic used in the algorithm for the removal of already used events before the issue of a new pivot search phase.

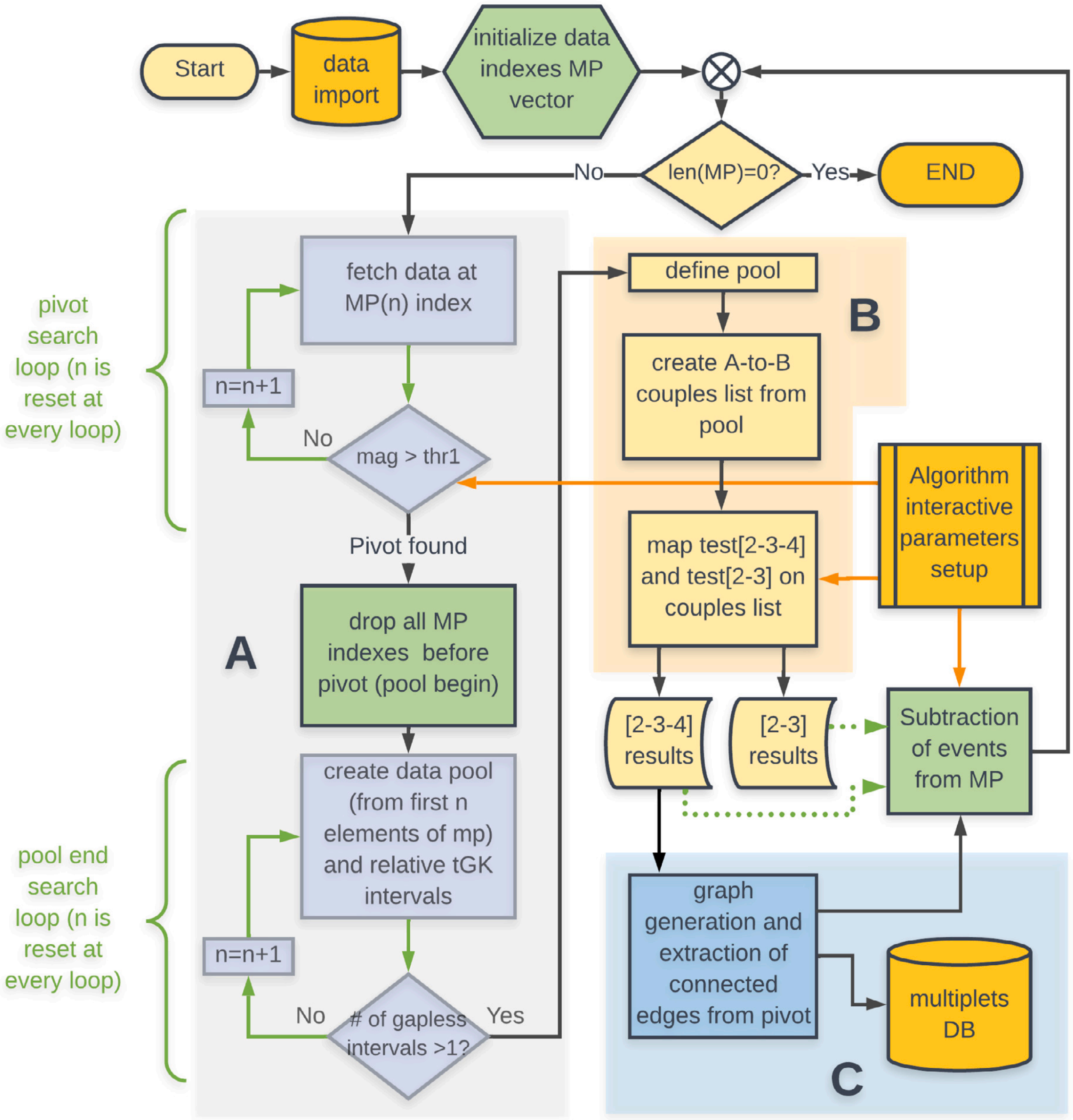

Start
data import
initialize data indexes MP vector
len(MP)=0?
No
Yes
END
pivot search loop (n is reset at every loop)
fetch data at MP(n) index
n=n+1
No
mag > thr1
Pivot found
drop all MP indexes before pivot (pool begin)
A
create data pool (from first n elements of mp) and relative tGK intervals
pool end search loop (n is reset at every loop)
n=n+1
No
# of gapless intervals >1?
Yes
define pool
B
create A-to-B couples list from pool
Algorithm interactive parameters setup
map test[2-3-4] and test[2-3] on couples list
[2-3-4] results
[2-3] results
Subtraction of events from MP
graph generation and extraction of connected edges from pivot
multiplets DB
C

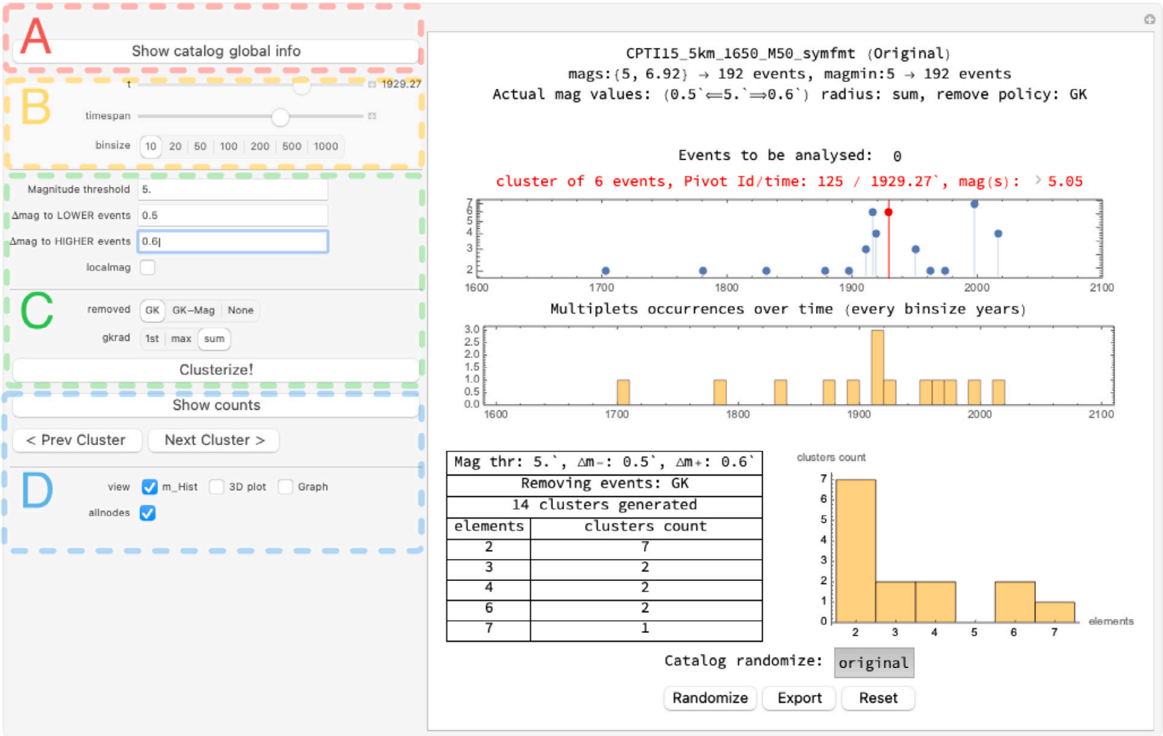
Fig. 4. Flowchart of the cluster search algorithm. A, B, C sections, for reference with Fig. 1, are indicated as colored background regions. Blocks indicated with curly brackets are the pivot and pool bounds searching macros inside the main algorithm procedure: both make use of indirect indexing through the use of a vector MP, initialized with all the integer indexes of the catalogue event list. All operations on MP components are indicated as green blocks. Algorithm takes track of data parsing completion through data reference indexes in the MP vector, gradually deleting from MP all already used data indexes. Procedure stops when MP is completely depleted.

3. removal of the “2-3-connected-only” events

These conditions usually result in a noticeable limitation of the number of found multiplets. In Fig. 2 example, the “2-3-4-connected” events (all panel (C) a-cyclic graph nodes) are marked with green dots in panel (A) to show how the next pivot search will discard event #7 to find event #18.

In Fig. 4, it is reported a concise flowchart of the whole procedure in which the (A), (B) and (C) portions are indicated with distinct background colors for reference.

Since the procedure accesses and parses the catalogue data through the use of an indexes vector MP, the flowchart in its top line loads data and creates and initializes this vector. It is filled with consecutive integers indexes $[1, 2, \dots, k]$ pointing to the k catalogue events. Underneath this row the flowchart is roughly divided in two vertical columns. The flow leads to the left one, contained in a pink “A” labeled rectangle showing the search loop for the pivot. Notice the green color used for blocks and arrows, it indicates operations made on the MP vector: as the flow progresses, green blocks gradually remove indexes from MP until it becomes empty, indicating that the catalogue scan is complete and the program can terminate. The green block in the A section drops all indexes smaller than pivot, setting it as the actual starting point. After this operation, a second loop block populates the pool event list. Operations flow to the right column, “B phase”,



t 1929.27
timespan
binsize 10 | 20 | 50 | 100 | 200 | 500 | 1000
Magnitude threshold 5.
Δmag to LOWER events 0.5
Δmag to HIGHER events 0.6
localmag □
removed GK | GK−Mag | None
gkrad 1st | max | sum
view ☑ m_Hist □ 3D plot □ Graph
allnodes ☑

```
CPTI15_5km_1650_M50_symfmt (Original)
mags:{5, 6.92} → 192 events, magmin:5 → 192 events
Actual mag values: (0.5`⟸5.`⟹0.6`) radius: sum, remove policy: GK

Events to be analysed:  0
cluster of 6 events, Pivot Id/time: 125 / 1929.27`, mag(s): › 5.05
```

Multiplets occurrences over time (every binsize years)

Mag thr: 5.`, Δm−: 0.5`, Δm+: 0.6`	
Removing events: GK	
14 clusters generated	
elements	clusters count
2	7
3	2
4	2
6	2
7	1

Catalog randomize: original

Fig. 5. Mathematica GUI of the software. Highlighted regions show parameters settings and analysis of results.

with orange background, where all event couples are created from the pool and thoroughly tested. The indexes of the events identified during this phase can be subtracted from MP in various ways (dotted green arrows), depending on the user's parameter choices (stored in the orange rightmost block with double vertical lines). Finally the "C" (light blue background) phase uses graph algorithms to reveal and store event clusters. The flow then returns to the top node ⊗ and repeats again until the end condition is encountered.

3. Code implementation

As stated above, the whole program has been written in the Wolfram language (Mathematica). It is a very powerful, high-level symbolic and compact functional language with embedded graphic that allows for very fast prototyping software creation. The program generates also a Graphic User Interface for a more rapid interaction. It is roughly divided in two columns and shown in Fig. 5, where regions have been highlighted for an easier description.

The left column contains all controls to set the parameters and start the clusters search, while the right one (uncolored base panel) contains informative graphs to show and analyze results.

The green C region is where the main search settings are set. Going from top to down, the first three input fields set Magnitude Threshold, Δ_{mag} to lower and higher events (criteria 1 and 4). The localmag button (unchecked by default) controls if in Eq. (4) M_{ref} is the pivot event magnitude or the earlier event M_i of the $\{M_i, M_j\}$ couple. Then the two toggle bars: gkrad chooses which distance is applied in Eq. (3) criterium 3 filter, and removed sets the event subtraction policy before starting a new pivot search (transition from phase C to A, see Figs. 2 and 3). After having set all previous parameters, the clusterize button launches the catalogue scan. For very large catalogues the process can last several seconds, during which a countdown to the end of the process is shown.

After catalogue parsing completion, a concise report appears in the right, lower part of the GUI. It shows a table with found multiplets classified for events number and multiplicity. The right-upper part is a scattergram timeline showing multiplets as time, number of clustered events points along the whole catalogue. The timeline is interactive and multiplets points can be selected with mouse clicks. The two buttons prev cluster and next cluster in section D allow to move cluster selection back and forth through the list. This is handy when in very large catalogues the timeline can become crowded. Information on selected cluster is given in the red row above scattergram: the small gray arrow before magnitudo of the pivot event is a control to expand a column with all cluster magnitudes.

In D region there are three checkbuttons in the view row: they toggle visibility of info graphs: m_hist is an overall timeline density bar-chart showing clusters occurrence over time. The bars width is set with the toggle bar binsize (region B). The remaining two buttons give info on the actual selected cluster: 3D plot shows the 3D cylinders overlapping plot (see Fig. 1): time axis span is adjustable with the timespan slider (region B). Finally Graph buttons toggles visibility of two versions of the oriented decision graph, shown in Fig. 6 that defined the multiplet. They are topologically equivalent but one of them is a sort of enhanced time-mag scattergram of the events, clearly showing algorithm's decisional process. The allnodes button toggles visibility of orphan unused nodes.

A last add-on, not strictly related to the scanning algorithm, is the possibility to randomize the catalogue. Given a loaded catalogue, for each and every event a new time is randomly assigned from a uniform distribution extending over the catalogue duration, and a final time sort of the catalogue is done. In this way we can check if statistic

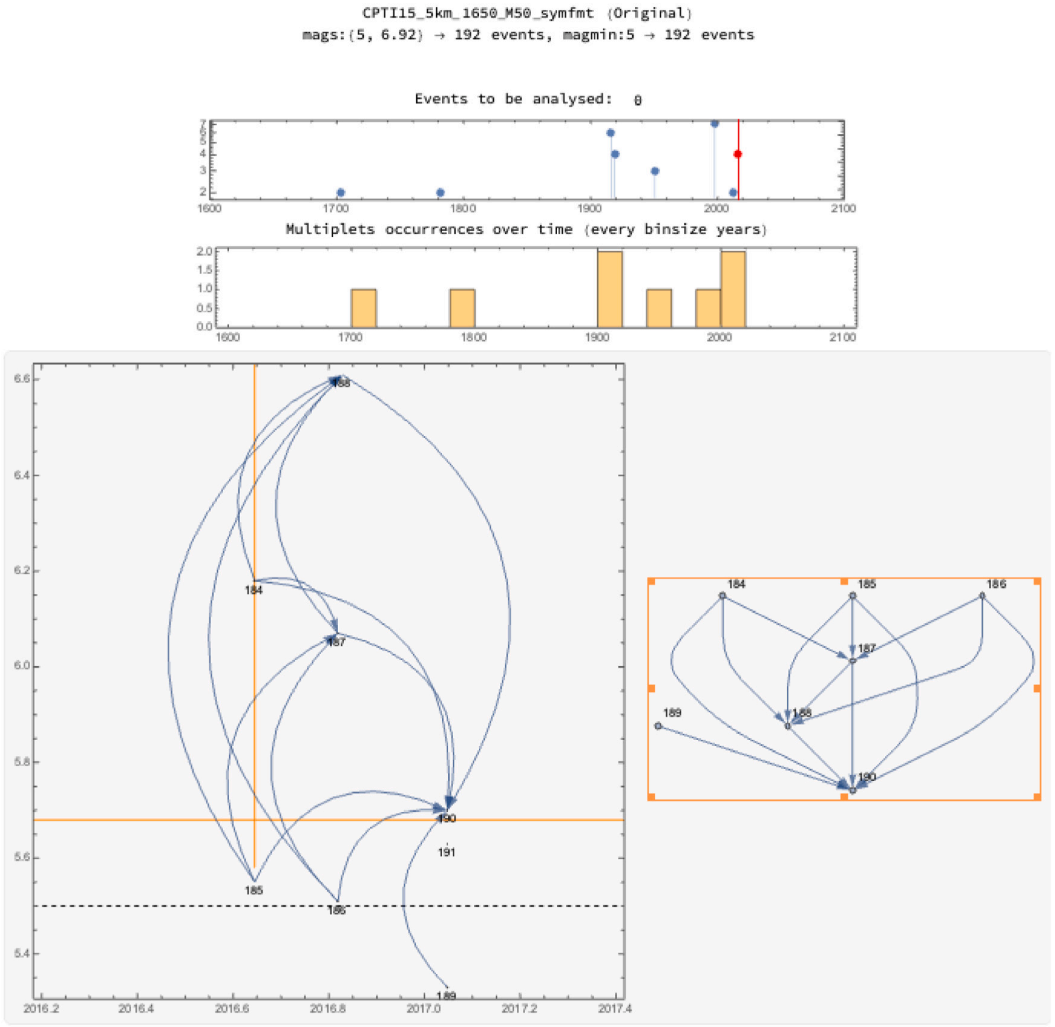


Fig. 6. For each multiplet event it is possible to pop-up the corresponding oriented acyclic decision graph (right) together with an equivalent version (left) in which nodes are plotted on a time-magnitude, timeline informative scatter-gram. Here the pivot point is the one with least ID (164) and a vertical orange bar, the dotted horizontal line is the magnitude threshold M_{thr} and the maximum and minimum allowed magnitudes are indicated by two horizontal orange lines here only the lower one is visible.

of multiplets from randomized catalogues systematically differs from the original one. This can be an empirical evidence that clustering phenomena do occur.

Since Mathematica is a commercial software, the core clustering routine, has been rewritten in python. This porting has no GUI features but has been tested to behave exactly as the original mathematica code. It is given as is, no maintenance plans are actually foreseen, but it can be freely used and improved provided that a citation is given. All code and software environment pertinent files have been made available as supplemental material through an external link. Detailed descriptions of these materials is given in Appendix.

Declaration of competing interest

This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.

The author has no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript

Data availability

Data used is freely available (seismic catalogues)

Acknowledgments

I express my gratitude to Professor R.Console and Dr. P. Vannoli for their invaluable assistance. The initial ideas for the tool stemmed from our discussions and were further developed through their profound understanding of the seismic history in the studied areas. Additionally, I appreciate the numerous seismic simulations they provided, contributing significantly to the actualization of this work.

I gratefully acknowledge INGV for their support in providing the necessary resources for this work. No special funds were required, and we appreciate the institute's commitment to our research endeavors.

Appendix. Computer code availabiliy and supplemental material

The software described in the paper is open source, not maintained and hopefully "citation-ware" i.e. it can be used completely free, and a citation to the author is more than welcome. It is available as supplementary material and is downloadable from Github at https://github.com/RobiOneKeno6i/multiplets_paper_repo.git or from zenodo at https://zenodo.org/badge/latestdoi/558425604 with assigned DOI 10.5281/zenodo.8021193. The main files of the repository are:

- *mathematica_code.nb*; the complete notebook described in the paper, with all features and the interactive GUI. It runs in a licensed Wolfram Mathematica environment or in the Wolfram Player Pro, the commercial paid version with I/O enabled.
- the python porting of the core scanning routine: it behaves as the mathematica code but with no GUI: I/O is based on disk files read-write. It is given in two distinct formats:
  - pure python code (*python_code.py*). User should have a recent version of python3 interpreter and some external packages need to be installed (using for example *pip* or *conda*).
  - the Jupyter/Colab ipython version (*multiplets_core_colab.ipynb*). It is the same code but ready to run under Jupyter, a web based IDE much more similar to the mathematica cells model. A desirable aspect of this paradigm is the possibility to let the code run online in a web environment like Google Colab, without the need to have a python interpreter locally installed.
- an example of the catalogue file (*Simulated_100k_005-02-2_geo.txt.zip*), given to test the software. The format catalogue is a table of 6 columns and each row is a seismic event. The six event parameters, in order from left to right, are:

time (yrs)	lat (°)	long (°)	depth (km)	magnitude	free event ID

  where time is in decimal years, lat and long are in decimal degrees and depth is defined with positive numbers i.e. a higher number means deeper. The last ID item is left for user reference in the catalogue, it is not used by the software, but the importer is expecting some number to be present.
- *ascii_files.zip*. The text files contained in this zip archive are the output of the python code running on the example catalogue. They are given for reference.
- *.yaml* files: two of them have been generated by the ''reproducible' ' package inside the python programs to give a snapshot on the specific aspects of the hardware and the computation. The other is the conda description of the python environment under which code has been generated and tested.
- *Wolfram_CDF_example.cdf.zip*. Since *mathematica_code.nb* cannot run without a commercial license, a CDF file has been included. This is a fully working version of *mathematica_code.nb* with no I/O capabilities: the catalogue data is completely contained in the CDF. It can be used freely with the CDF player from Wolfram and is provided here for GUI demonstration.

In both python executables, input variables must be set editing the program code. The filenames of the actual released code have no pre-assigned path so they default to the same directory where code is. Running the procedure is straighforward: user must put the catalogue file and the code in the same directory and setup the input parameters. M_{thr} is the code variable magthresh, ΔM_+ and ΔM_- are dmplus and dmminus: they expect a numeric value. Flags for pool removal and space distance behavior (see Section 2.1, Eq. (3)) are the removal variable whose value can be: ''GK'', ''GK-Mag'' and ''None'' and gkrad that can be set to ''sum'', ''max'' or ''1st''. After run is finished, output files are written on the same directory.

The jupyter version behaves the same when it is run locally. When, instead, Google Colab environment is used, some disk space is temporarily assigned and made available for read/write operations. Upon disconnection, however this space is wiped out and output results files must first be downloaded. In addition to this procedure, Google allows connection between Colab IDE and Google Drive in order to offer a persistent repository for files. The first cell of ipynb code initiates this connection, and the procedure is anyway well documented online. File paths in the code should be accordingly modified.

On a modern Mac arm M1 processor the rather big synthetic catalogue example file contains 337254 events from which a subset of 95807 events is prefiltered. Search time on this subset, for the parameters reported above, gives out 2123 clusters found by the python code in about 2 min and half of processing time, while Mathematica time is around 20 s. Google Colab time varies, but in the free version computation is usually slower, lasting around 4 min.

References

Bak, P., Tang, C., Wiesenfeld, K., 1987. Self-organized criticality: An explanation of the 1/f noise. Phys. Rev. Lett. 59 (4), 381.

Berger, J.M., Mandelbrot, B., 1963. A new model for error clustering in telephone circuits. IBM J. Res. Dev. 7 (3), 224–236.

Beroza, G.C., Cole, A.T., Ellsworth, W.L., 1995. Stability of coda wave attenuation during the Loma Prieta, California, earthquake sequence. J. Geophys. Res.: Solid Earth 100 (B3), 3977–3987.

Console, R., Murru, M., Vannoli, P., Carluccio, R., Taroni, M., Falcone, G., 2020. Physics-based simulation of sequences with multiple main shocks in central Italy. Geophys. J. Int. 223 (1), 526–542. http://dx.doi.org/10.1093/gji/ggaa300.

Console, R., Vannoli, P., Carluccio, R., 2022. Physics-based simulation of sequences with foreshocks, aftershocks and multiple main shocks in Italy. Appl. Sci. 12 (4), 2062.

Field, E.H., 2019. How physics-based earthquake simulators might help improve earthquake forecasts.

Gardner, J., Knopoff, L., 1974. Is the sequence of earthquakes in Southern California, with aftershocks removed, Poissonian? Bull. Seismol. Soc. Am. 64 (5), 1363–1367.

Gibowicz, S.J., Lasocki, S., 2005. Earthquake doublets and multiplets in the Fiji-Tonga-Kermadec region. Acta Geophys. Pol. 53 (3), 239.

Gutenberg, B., Richter, C., 1955. Magnitude and energy of earthquakes. Nature 176 (4486), 795.

Kagan, Y., Jackson, D., 1999. Worldwide doublets of large shallow earthquakes. Bull. Seismol. Soc. Am. 89 (5), 1147–1155.

Poupinet, G., Ellsworth, W., Frechet, J., 1984. Monitoring velocity variations in the crust using earthquake doublets: An application to the Calaveras Fault, California. J. Geophys. Res.: Solid Earth 89 (B7), 5719–5731.

Utsu, T., 1970. Aftershocks and earthquake statistics (1): Some parameters which characterize an aftershock sequence and their interrelations. J. Faculty Sci. Hokkaido Univ. Ser. 7, Geophys. 3 (3), 129–195.

Vere-Jones, D., 1969. A note on the statistical interpretation of Båth's law. Bull. Seismol. Soc. Am. 59 (4), 1535–1541.