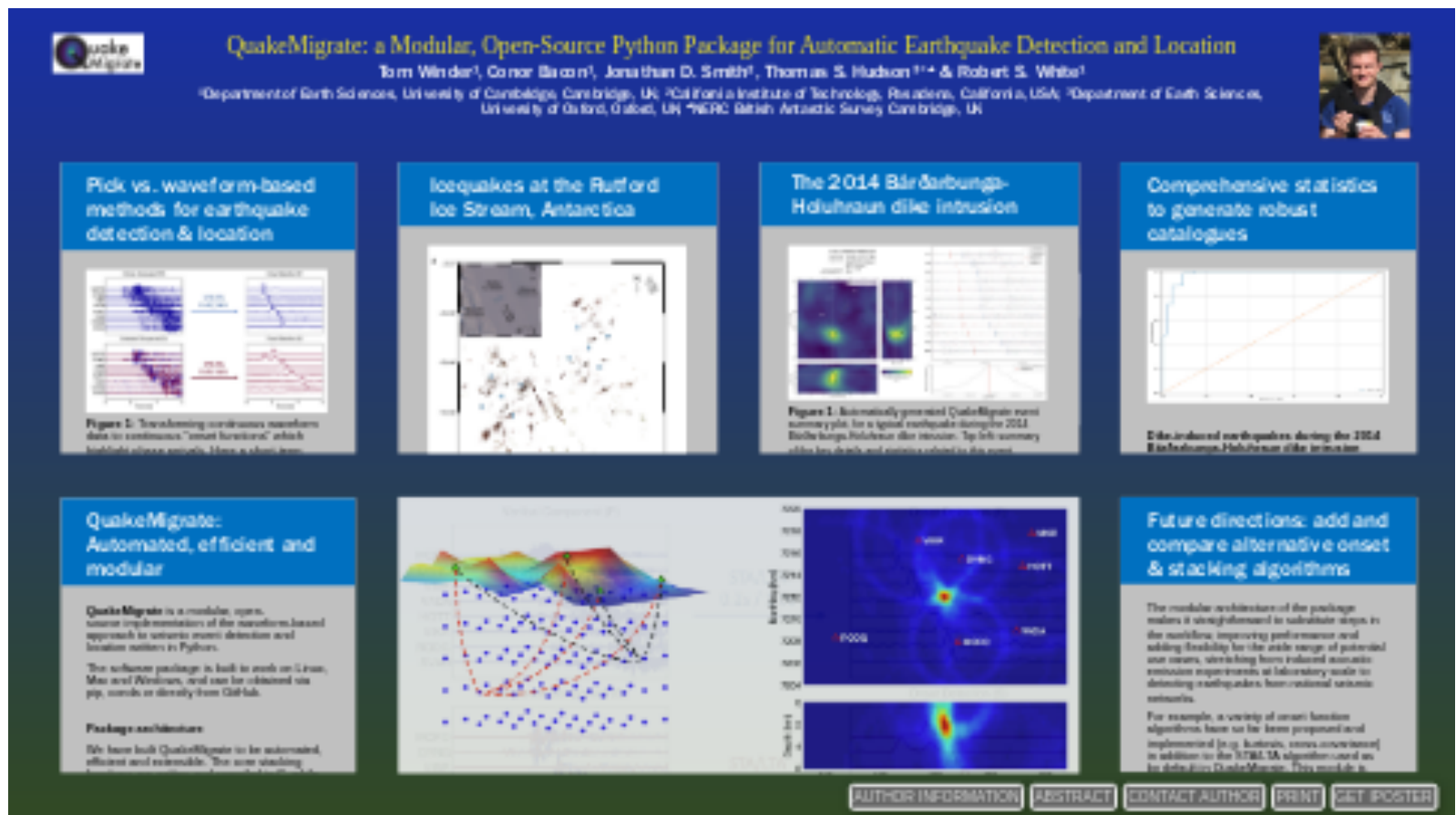


# QuakeMigrate: a Modular, Open-Source Python Package for Automatic Earthquake Detection and Location

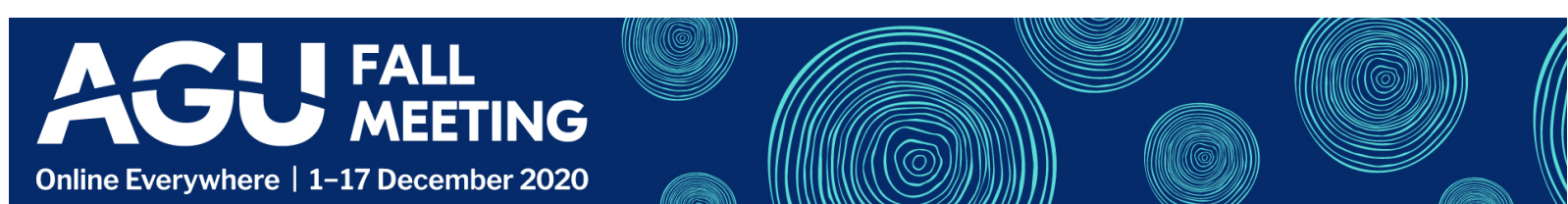


Tom Winder<sup>1</sup>, Conor Bacon<sup>1</sup>, Jonathan D. Smith<sup>2</sup>, Thomas S. Hudson<sup>3,4</sup> and Robert S. White<sup>1</sup>

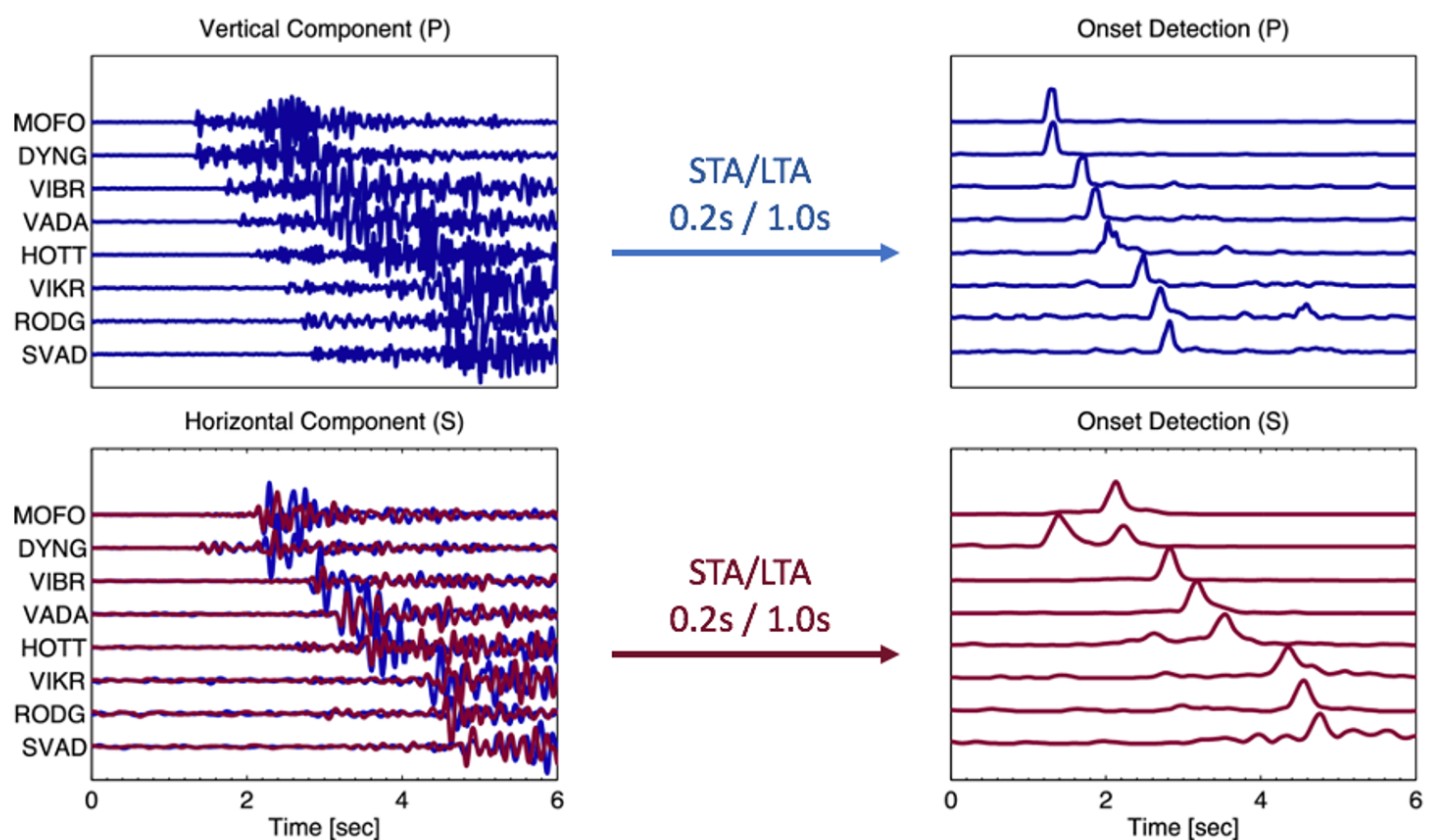
<sup>1</sup>Department of Earth Sciences, University of Cambridge, Cambridge, UK; <sup>2</sup>California Institute of Technology, Pasadena, California, USA; <sup>3</sup>Department of Earth Sciences, University of Oxford, Oxford, UK; <sup>4</sup>NERC British Antarctic Survey, Cambridge, UK



PRESENTED AT:



# PICK VS. WAVEFORM-BASED METHODS FOR EARTHQUAKE DETECTION & LOCATION



**Figure 1:** Transforming continuous waveform data to continuous "onset functions" which highlight phase arrivals. Here a short-term average / long-term average algorithm is applied to the vertical component waveform from each station to identify P phase onsets, and to the horizontal components for S. Modified after [Drew et al. \(2013\)](https://doi.org/10.1093/gji/ggt331) (<https://doi.org/10.1093/gji/ggt331>).

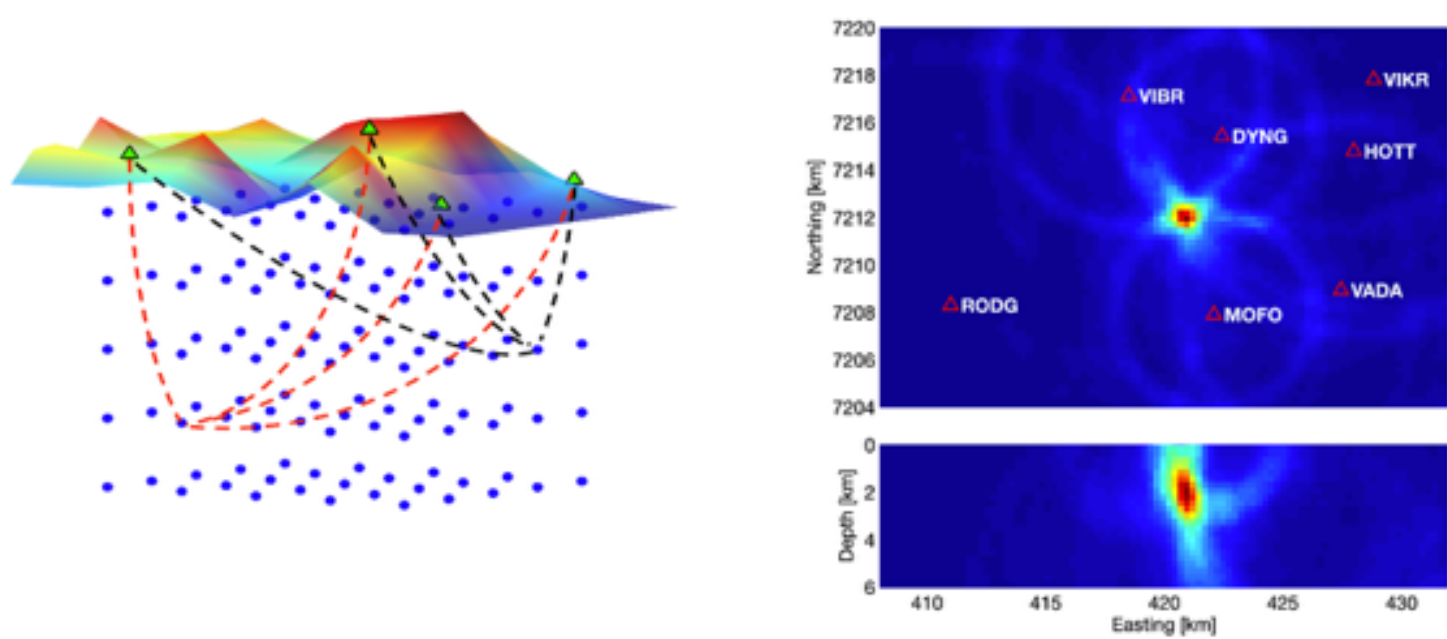
## Pick-base earthquake detection

The standard and most widely-used approach: first detect phase arrivals by considering the seismic traces recorded at each station individually, before comparing picks made across the network.

1. Choose a "characteristic function", or trigger algorithm, to identify phase arrivals within the continuous data recorded at each station. These algorithms may analyse changes in one or a combination of the amplitude, phase or frequency content of the signal.
2. Select a ***pick threshold***: triggers which exceed this threshold will be retained as "picks"; ***the remaining information is discarded***.
3. Identify groups of picks which correspond to a single event using a phase association algorithm.
4. Use traditional pick-based techniques to calculate absolute (e.g. NonLinLoc) and relative (e.g. HypoDD) locations.

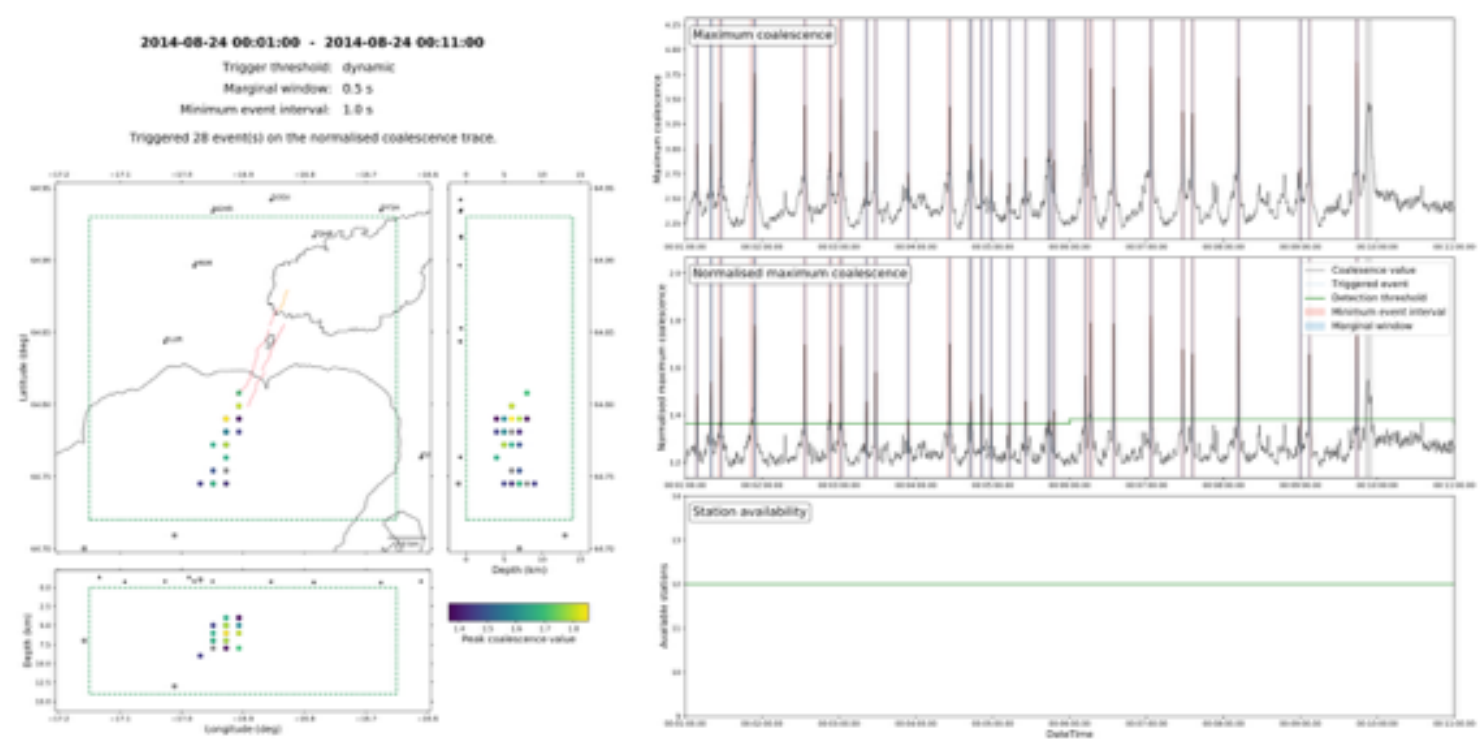
## Waveform-based earthquake detection

More recently developed approach, derived from the waveform migration and stacking techniques used in active-source seismology. Instead of reducing the information recorded at each seismometer to discrete time picks, the waveforms recorded across the entire network are combined to perform a grid-search for coherent sources of energy in the subsurface. Here we describe the "partial waveform stacking" approach, where the raw waveform is transformed prior to migration.



**Figure 2:** Schematic diagram illustrating the 3D subsurface grid, and a slice through the grid at the time of an event: warm colours represent a high stacking value; triangles mark seismic stations. Modified after [Drew et al. \(2013\)](https://doi.org/10.1093/gji/ggt331) (<https://doi.org/10.1093/gji/ggt331>).

1. Analogous to step (1) for pick-based detection, but here *we do not specify a threshold*. Instead *all information in the transformed input waveform is retained for the network-based analysis*. We denote these continuous measures of the likelihood of a particular seismic phase arrival "onset functions" (Figure 1).
2. For each seismic phase, calculate a lookup table of traveltimes from each station to a 3D subsurface grid of nodes.
3. Scan over these nodes, treating each as a possible earthquake hypocentre; at each timestep (each candidate event origin time) migrate the phase onset functions from each station into the 3D grid, and stack them at each grid node.
4. Repeat the process at each timestep, generating a 4-D image of the subsurface. At the location and origin time of each true earthquake, peaks in the onset functions for each seismic phase and from each station across the network will stack coherently or "coalesce" at that grid node.
5. Identify true events by **setting a threshold coalescence value**; the location & time of the coalescence peak gives the event hypocentre & origin time (Figure 3).



**Figure 3:** QuakeMigrate summary plot illustrating the process of triggering events from the continuous coalescence image by setting a threshold coalescence value (see second box for more details). Left shows map and cross sections with dots representing the location of coalescence peaks in the grid. Right panels show time series of the raw and normalised maximum coalescence values in the grid, with peaks above the threshold identified as candidate events.

### Benefits of the waveform-based approach:

- Avoids the strong trade-off between catalogue completeness and the proportion of mis-picks when selecting a "pick threshold".
- Even in low-noise conditions, and using the most state-of-the-art algorithms (e.g. PhaseLink, [Ross et al. 2018](https://doi.org/10.1093/gji/ggy111)

(<https://doi.org/10.1029/2018JB016674>); SCANLOC, [Gempa Gmbh](https://docs.gempa.de/scanloc/current/) (<https://docs.gempa.de/scanloc/current/>) - see comparison by [Grigoli et al., 2018](https://doi.org/10.1093/gji/ggy019) (<https://doi.org/10.1093/gji/ggy019>)) phase association can be computationally expensive and error-prone. Missed picks can lead to missed detections, while false-picks can lead to artefacts or erroneous event locations. This is particularly the case in seismic swarms, where events are separated by very short time intervals; a common feature of microseismic datasets.

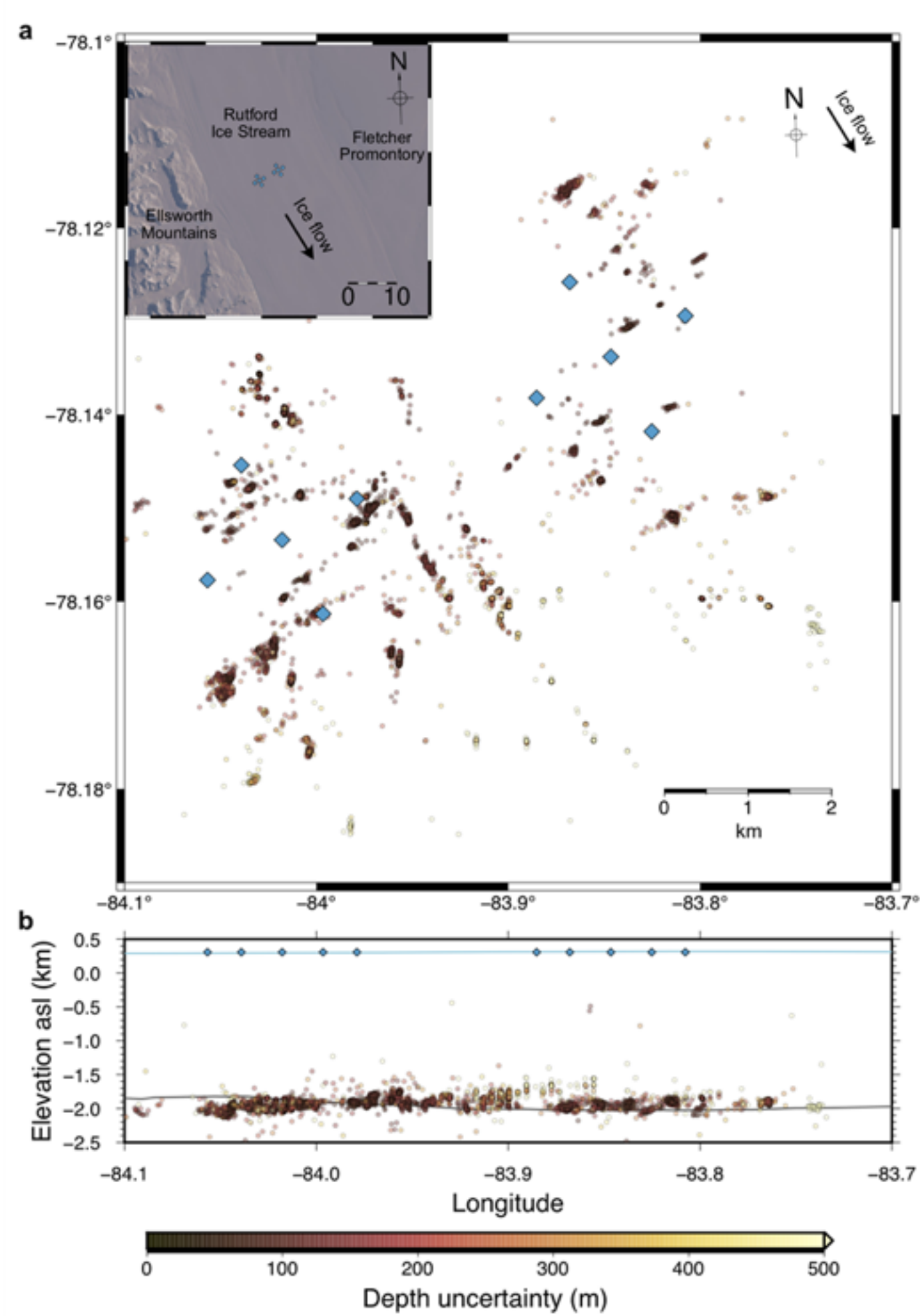
- By retaining all recorded information until the point of comparing the waveforms across the whole network (and thereby calculating the event location!), even extremely weak arrivals on individual stations can be associated with true events - where they are found to correspond to a network-consistent peak in coalescence - thereby helping constrain the location.
- Furthermore, exploiting this coherency between information recorded at stations across the network also improves the detection capability for the smallest events.



# ICEQUAKES AT THE RUTFORD ICE STREAM, ANTARCTICA

TRY IT OUT ONLINE!\*

([https://mybinder.org/v2/gh/QuakeMigrate/QuakeMigrate/3f6f9ab030109e32c1c68d267bc456bb1?filepath=examples%2FIcequake\\_Rutford%2Ficequakes\\_rutford.ipynb](https://mybinder.org/v2/gh/QuakeMigrate/QuakeMigrate/3f6f9ab030109e32c1c68d267bc456bb1?filepath=examples%2FIcequake_Rutford%2Ficequakes_rutford.ipynb))



**Figure 1:** Icequakes detected at the Rutford Ice Stream, Antarctica between 20-21st January 2009. a: epicentres shown by circles, coloured by depth uncertainty; stations are marked by blue diamonds. Inset shows location context and ice flow direction. b: the ice surface and ice-bed interface are indicated by blue and grey lines, respectively; icequakes and stations marked as in (a).

Summary:

- Extremely high event rate: *on average* 1 event every 10 seconds
- Icequakes generated by stick-slip sliding at the ice-bed interface
- QuakeMigrate detects over 10x more events than resolved using a previous waveform-based detection algorithm (CMM - [Drew et al., 2013](https://doi.org/10.1093/gji/ggt331) (<https://doi.org/10.1093/gji/ggt331>); [Smith et al., 2015](https://doi.org/10.1002/2015JF003587) (<https://doi.org/10.1002/2015JF003587>))
- Well-located events are tightly confined to the ice-bed interface, indicating excellent location performance and realistic location uncertainties.
- Extremely clean catalogue with few false triggers, after filtering by location uncertainty statistics output by QM (see top-right box for more details!).
- Process this data yourself: check out the [web-hosted version here](https://mybinder.org/v2/gh/QuakeMigrate/QuakeMigrate/3f6f9ab030109e32c1c68d267bc456bbf79d82c9?filepath=examples%2FIcequake_Rutford%2Ficequakes_rutford.ipynb) ([https://mybinder.org/v2/gh/QuakeMigrate/QuakeMigrate/3f6f9ab030109e32c1c68d267bc456bbf79d82c9?filepath=examples%2FIcequake\\_Rutford%2Ficequakes\\_rutford.ipynb](https://mybinder.org/v2/gh/QuakeMigrate/QuakeMigrate/3f6f9ab030109e32c1c68d267bc456bbf79d82c9?filepath=examples%2FIcequake_Rutford%2Ficequakes_rutford.ipynb)) (same link as at top) or the [Jupyter Notebook](https://github.com/QuakeMigrate/QuakeMigrate/blob/development/examples/Icequake_Rutford/icequakes_rutford.ipynb) ([https://github.com/QuakeMigrate/QuakeMigrate/blob/development/examples/Icequake\\_Rutford/icequakes\\_rutford.ipynb](https://github.com/QuakeMigrate/QuakeMigrate/blob/development/examples/Icequake_Rutford/icequakes_rutford.ipynb)) on GitHub.

**\*Note:** this should open a Jupyter Notebook in your browser, and allow you to run a small portion of seismic data from the Rutford dataset through QuakeMigrate.

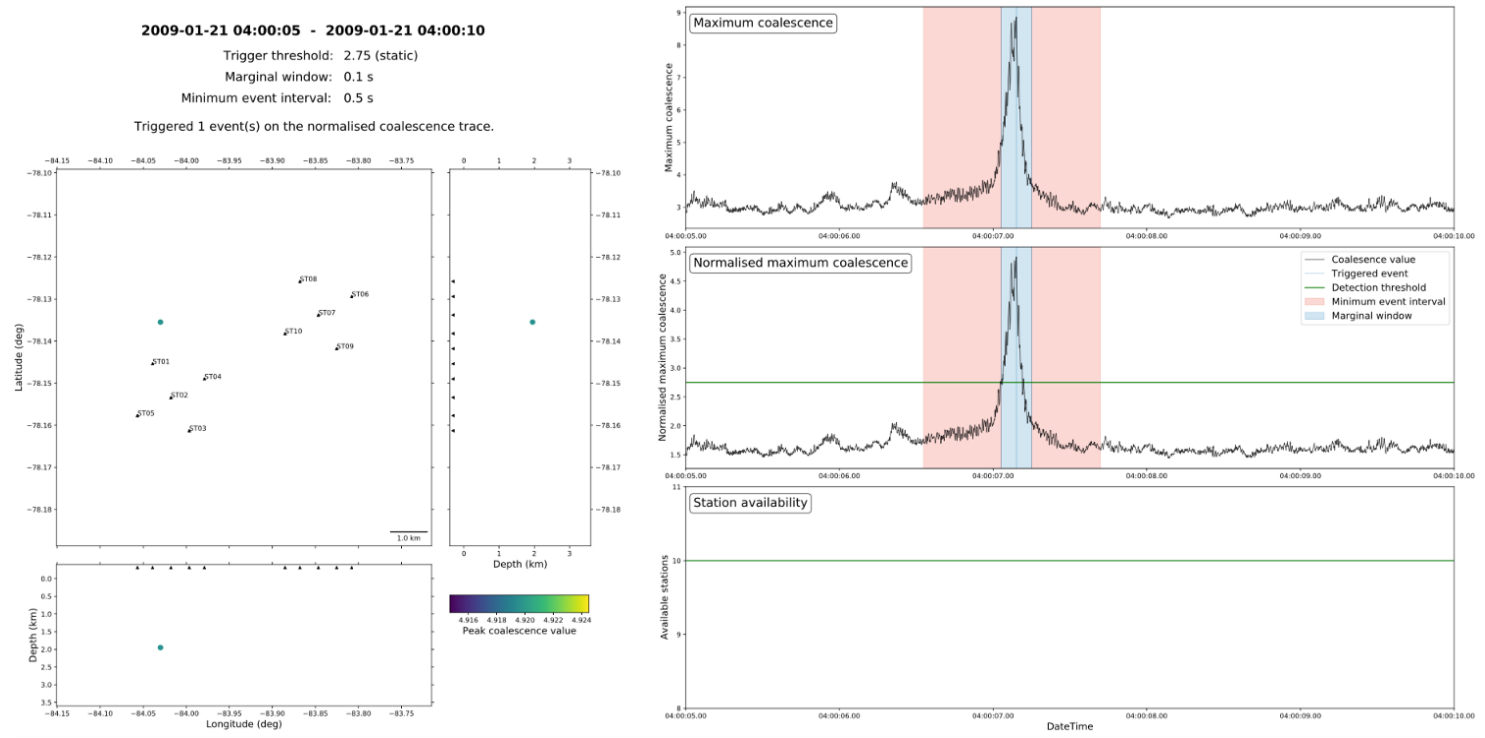
You can also try out a second icequake example detecting icequakes at the Skeiðarárjökull outlet glacier, Iceland ([https://mybinder.org/v2/gh/QuakeMigrate/QuakeMigrate/AGU\\_2020\\_binder?filepath=examples%2FIcequake\\_Iceland%2Ficequakes\\_iceland.ipynb](https://mybinder.org/v2/gh/QuakeMigrate/QuakeMigrate/AGU_2020_binder?filepath=examples%2FIcequake_Iceland%2Ficequakes_iceland.ipynb))

**Step-by-step workflow (for a detailed discussion of parameter choices see [Hudson et al., 2019](https://doi.org/10.1017/aog.2019.18) (<https://doi.org/10.1017/aog.2019.18>))**

1) Generate the **lookup-table**. We use a homogeneous velocity model for P and S phases. Grid node spacing 100m x 100m x 100m; grid extent 10 x 10 x 3.5 km ( = ~ 350,000 nodes).

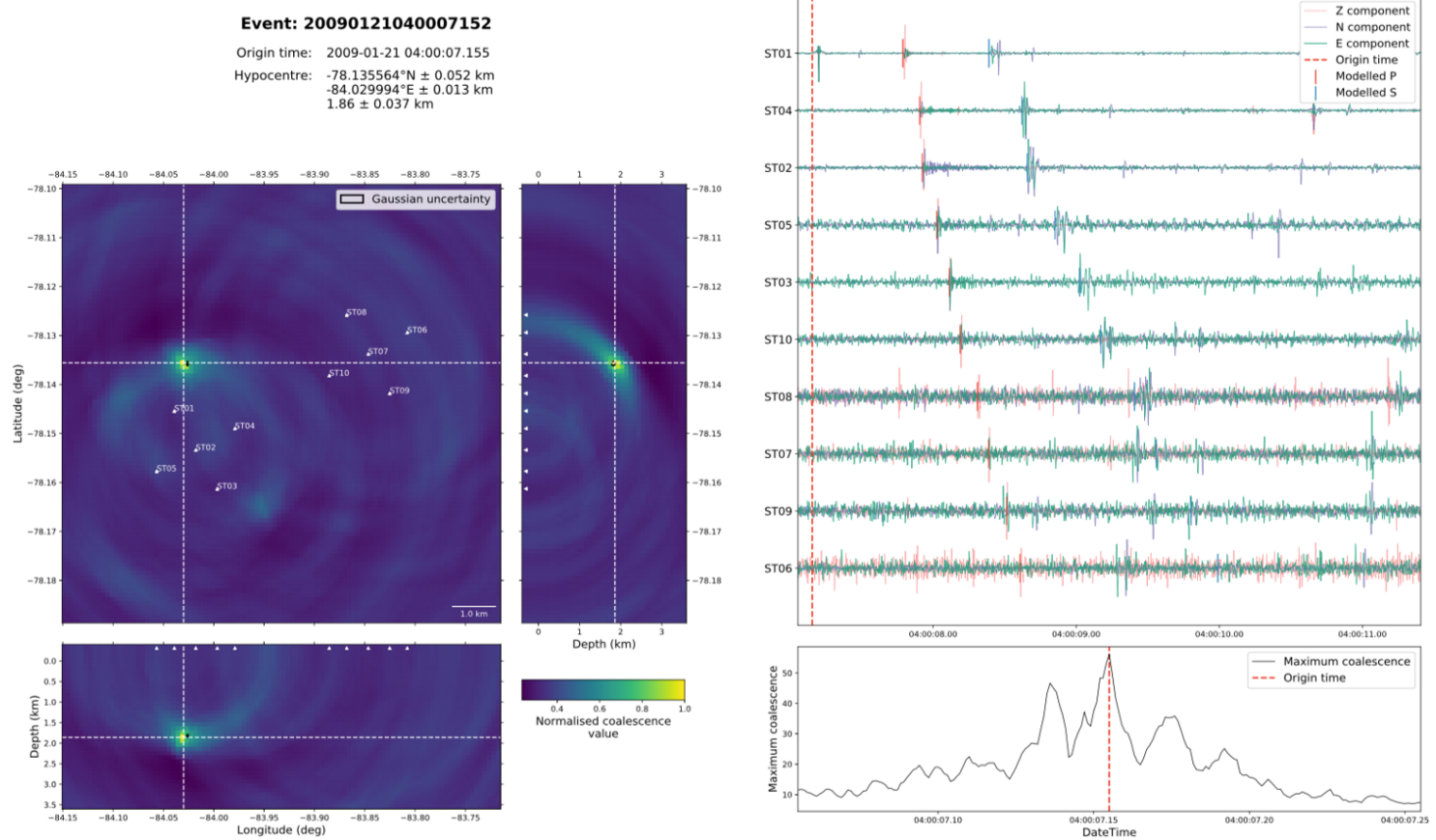
2) **Detect**. Sampling rate: 500 Hz (decimated from 1000 Hz), no decimation owing to the extremely sharp onset functions and risk of spatial aliasing. Onset function parameters: P [20 Hz, 200 Hz] bandpass filter; S [10 Hz, 150 Hz] bandpass filter, P [0.01 s, 0.25 s] STA/LTA windows; S [0.05 s, 0.5s] STA/LTA windows. "Classic" (overlapping) STA/LTA algorithm. 10 stations. Compute time: runs at ~ real time running on 16 CPU cores.

3) **Trigger**. Static threshold (owing to the consistent data quality, noise levels and short duration) of 2.0, operating on the normalised coalescence trace. Trigger threshold choice was made by inspection of the automatically generated QM trigger summary plots. Compute time ~ tens of minutes (single CPU).



**Figure 2:** QM Trigger summary plot displaying 10 seconds of coalescence data output by **detect**. One event is identified above the trigger threshold - in this example using a more strict threshold of 2.75.

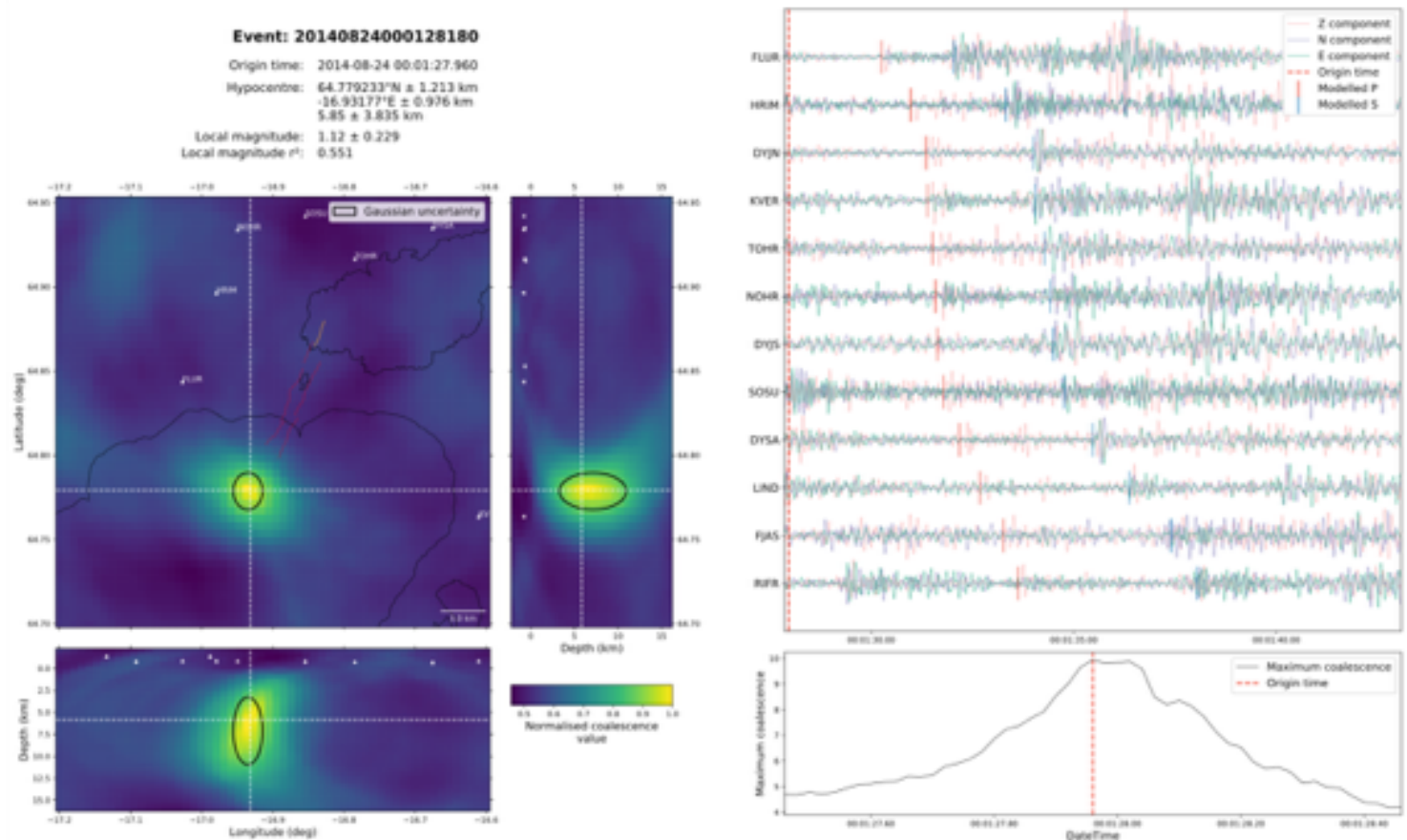
4) **Locate**. Identical parameters to detect, but now running at 1000 Hz, and using the "centred" STA/LTA onset function, which gives a more accurate approximation of a probability density function describing the timing of a possible phase arrival. Compute time  $\sim 7$  seconds per event running on 4 CPU cores.



**Figure 3:** QM Location summary plot displaying left: the location probability density function (or coalescence map) and right: a waveform gather displaying modelled phase arrival times for the location estimate with the input waveforms. See B-H dike intrusion Figure 1 for more details. Notice the extremely tight location constraint, achievable in part due to the extremely impulsive phase arrivals and clean data.

5) Filter by QM covariance error statistic, and remove scattered events shallower than 1500m depth (an artefact of the station geometry). See top-right box for more details!

# THE 2014 BÁRÐARBUNGA-HOLUHRAUN DIKE INTRUSION



**Figure 1:** Automatically generated QuakeMigrate event summary plot, for a typical VT earthquake during the 2014 Bárðarbunga-Holuhraun dike intrusion. Top left: summary of the key details and statistics related to this event, including local magnitude and location uncertainties reported by QM. Left: map and cross-section views through the 3D coalescence map for this event: light colours indicate high coalescence = high probability of the hypocentre being at this location. Maximum coalescence location indicated by white dashed cross-hairs on each panel, and gaussian uncertainty described by the black ellipse. Right: waveform gather, with modelled P and S phase arrivals based on the best estimate of the event location and origin time marked by red and blue ticks. Bottom right: max coalescence value in 3D grid at each timestep for a short time window around the event origin time. Sharply peaked coalescence map and coalescence-through-time plot indicate a well-constrained event location. Note following event (S-wave arrivals visible ~ 5 seconds later than for the first event!).

## Summary:

- Very high event rate (> 4000 events per day).
- Volcanotectonic earthquakes induced on pre-existing faults around the base of the dike as it intruded northwards (Woods et al., 2019 (<https://doi.org/10.1016/j.epsl.2018.10.032>); Ágústssdóttir et al., 2019 (<https://doi.org/10.1029/2018JB016010>)).
- QuakeMigrate detects ~ 2 x as many events as found by Ágústssdóttir et al. using CMM.
- Raw QM event locations agree exceptionally well with the relative relocations calculated by Woods et al. (2019) (<https://doi.org/10.1016/j.epsl.2018.10.032>).
- Detection and location performance prove to be resilient to a heterogeneous network geometry and a wide variety of event types and magnitudes ( $M_L0$  to  $M_W5$ ; VT events, hybrid events, and LP's).
- Process the data yourself! Try out the [dike example](https://github.com/QuakeMigrate/QuakeMigrate/tree/development/examples/Volcanotectonic_Iceland) ([https://github.com/QuakeMigrate/QuakeMigrate/tree/development/examples/Volcanotectonic\\_Iceland](https://github.com/QuakeMigrate/QuakeMigrate/tree/development/examples/Volcanotectonic_Iceland)) on our GitHub.

## Step-by-step workflow:

1) Calculate the **lookup-table**. We use a built-in wrapper for the NonLinLoc eikonal ray-tracer to calculate P and S wave traveltimes for a 1D gradient velocity model.

Traveltimes can alternatively be calculated purely in Python using the Fast-Marching Method, making use of the scikit-fmm package. Grid node spacing 0.5 x 0.5 x 0.5 km; grid extent ~ 30 x 30 x 20 km (= ~ 80,000 nodes).



2) **Detect**. Sampling rate 50 Hz (downsampled from 100 Hz), decimate the grid by a factor of 2 to 1 km spacing; location uncertainty is greater than this so coherent sources still stack successfully above the background noise.

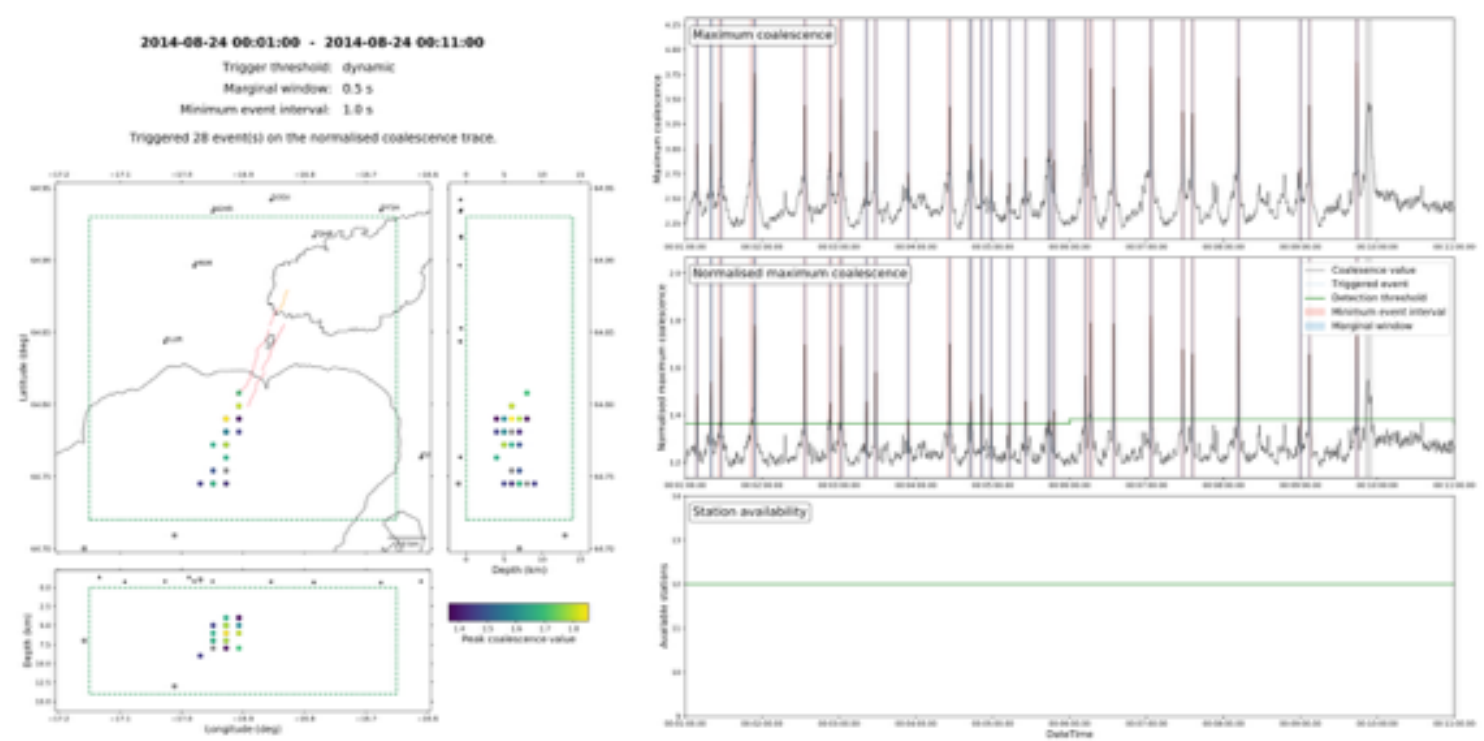
Onset function parameters: P [2 Hz, 16 Hz], S [2 Hz, 14 Hz] bandpass filter; P [0.2 s, 1 s], S [0.2 s, 1 s] STA/LTA windows, using the "classic" (overlapping) STA/LTA algorithm. 12 stations.

Compute time: 150 x faster than real-time running on 12 CPU's.

3) **Trigger**. Dynamic trigger threshold (owing to variable noise levels, station dropouts and event rates), at 1.1x the median of the normalised coalescence trace (Figure 2).

This provides excellent robustness in the presence of station dropouts (particularly where part of the network is telemetered, for example) and variable noise levels (e.g. due to storms, earthquake swarms, passing surface wave trains from regional or teleseismic events, or even volcanic tremor!).

Compute time: ~ minutes (single CPU).

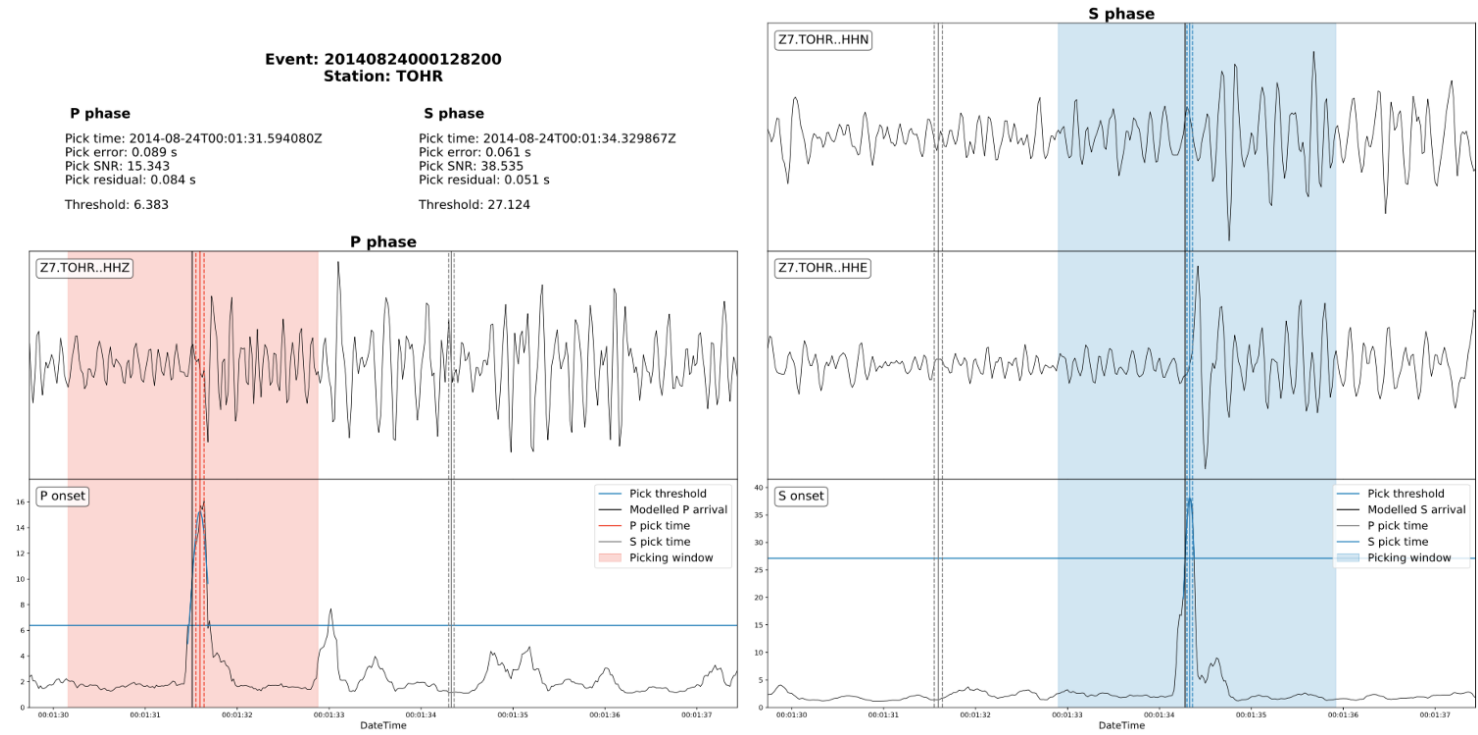


**Figure 2:** QuakeMigrate summary plot illustrating the process of triggering events from the continuous coalescence image by setting a threshold coalescence value (see second box for more details). Left shows map and cross sections with dots representing the location of coalescence peaks in the grid. Red lines show the location of the graben which formed above the opening dike, orange the 2014-15 eruption fissure, and black the glacier margin (south) and lava flow outline (north-east). Right panels show time series of the raw and normalised maximum coalescence values in the grid, with peaks above the threshold identified as candidate events.

4) **Locate**. Identical settings to **detect**, but without decimating the search grid, and using the "centred" STA/LTA onset function.

An instrument response inventory, response removal parameters and an attenuation function specific to this region were supplied to also calculate local magnitudes, and some associated statistics useful for filtering out false triggers (see top-right box for more details!).

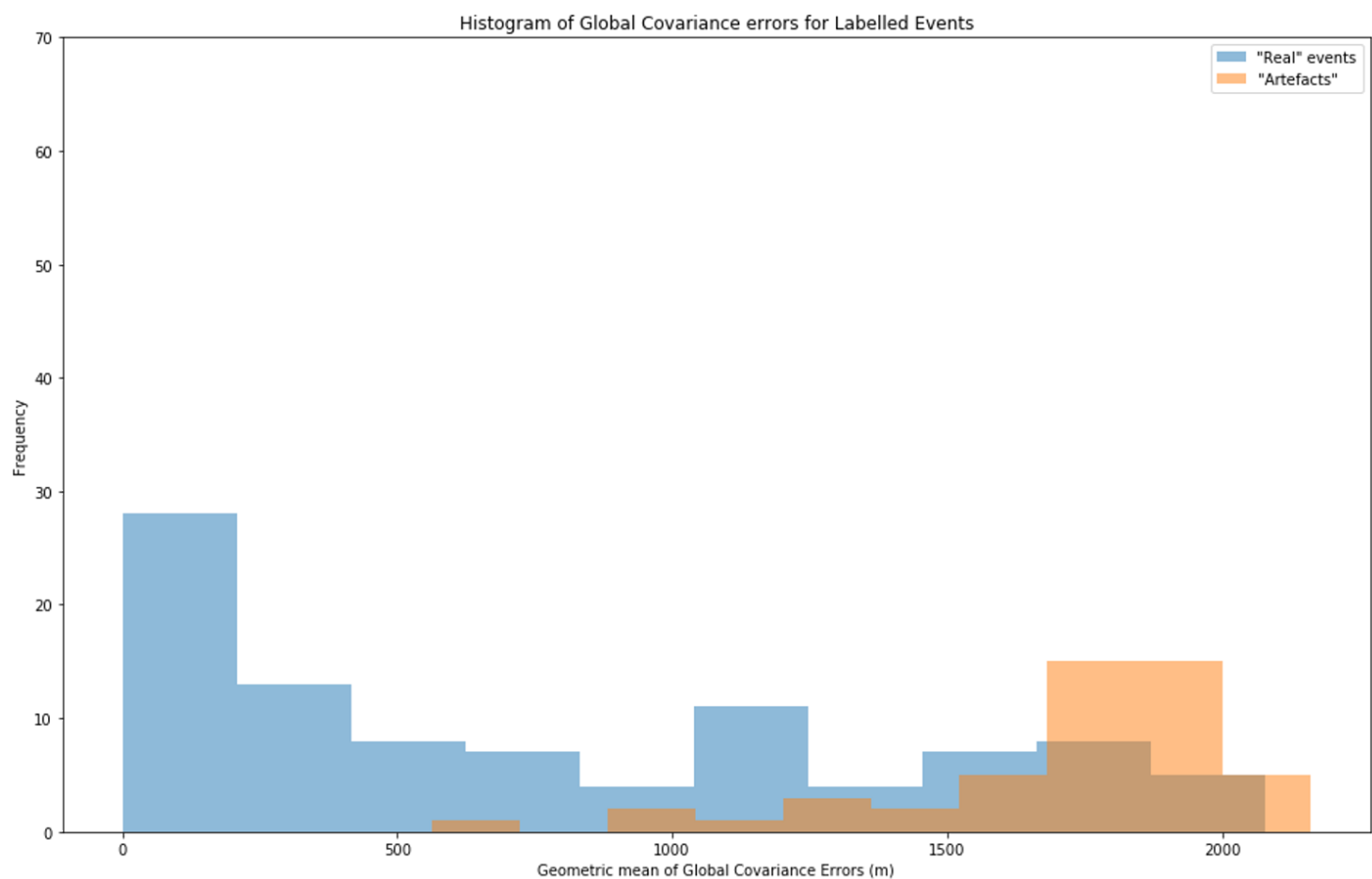
Compute time ~ 7 seconds per event including magnitude calculation (~ 1 second) and generating summary plots (~ 3 seconds) running on 2 CPU cores.



**Figure 3:** Optional pick summary plot automatically output by QM. This shows the input waveforms, onset functions calculated from them, and automatic picks made by fitting a gaussian function to peaks above the dynamically determined pick threshold within the picking window. Here we benefit from the 'locate then pick' approach, being able to use a-priori information about the expected timing of the phase arrivals (red window = P, blue = S) both in making the pick and determining the background noise level. These picks are reported with information on their SNR and timing uncertainty, derived from the gaussian fit amplitude and width.

The option to automatically output phase picks is useful for integrating QM into existing research frameworks, including, for example, pipelines to use waveform cross-correlation to calculate precise differential travel times for relative relocation, or to measure shear-wave splitting.

# COMPREHENSIVE STATISTICS FOR ROBUST CATALOGUES



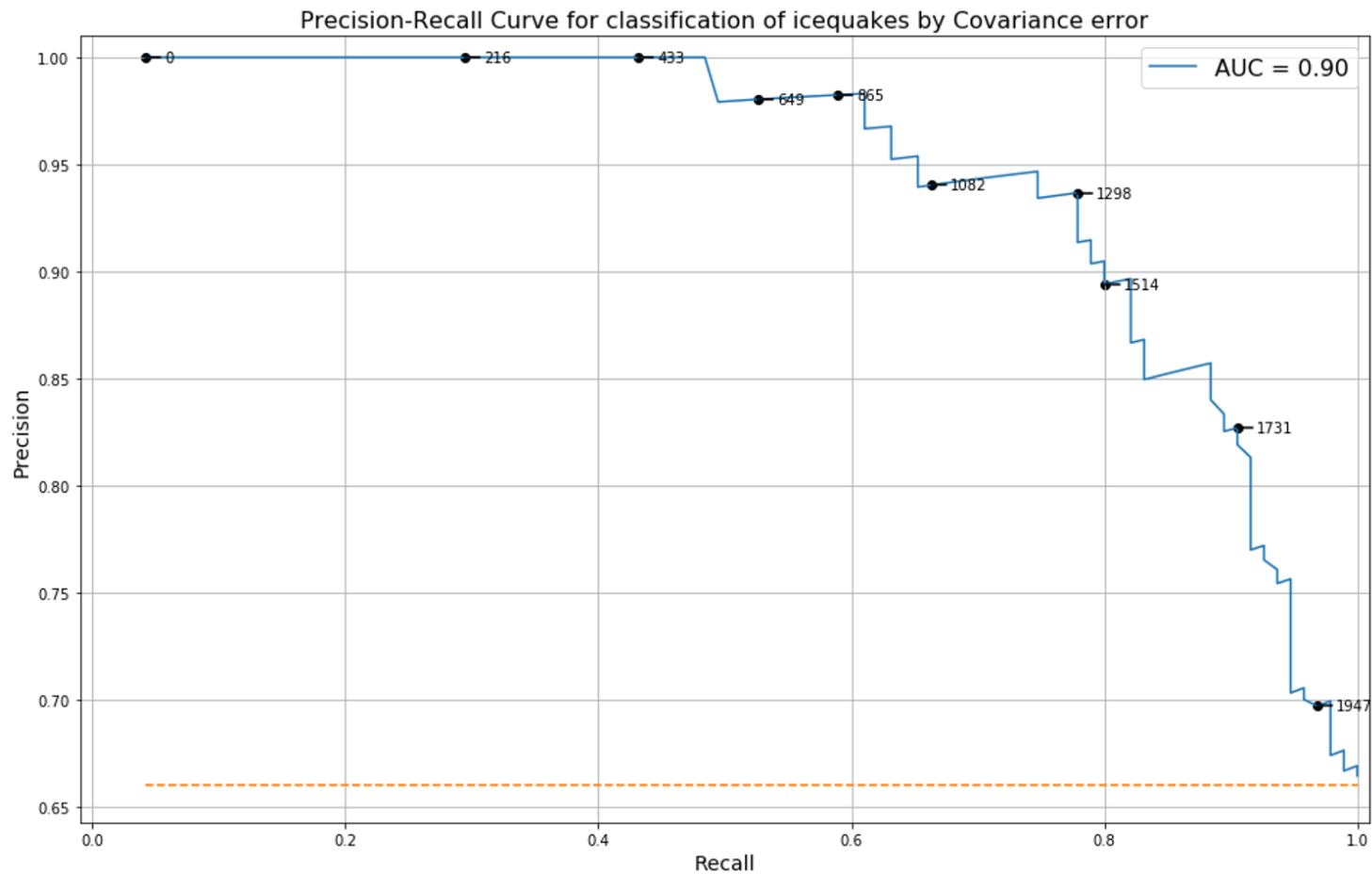
**Figure 1:** Histogram of icequake covariance errors - a characterisation of how sharp the peak in the 3D coalescence map is - for a sample of 144 events from the full catalogue which have been labelled as "real" or "artefact" by an analyst, based on inspection of waveforms and the QuakeMigrate event summary plot.

## Icequakes at the Rutford Ice Stream, Antarctica

We seek to make an objective assessment of filtering performance, and choice of filter value, based on manual labelling and statistical analysis of a subset of the catalogue. By choosing a representative sample, the insights gained here can be applied to the catalogue as a whole, which would be otherwise impractical to assess, leading to a subjective (and likely sub-optimal) choice of filter.

To generate the catalogue for this example, true events were distinguished from artefacts based on the location uncertainty statistics reported by QuakeMigrate.

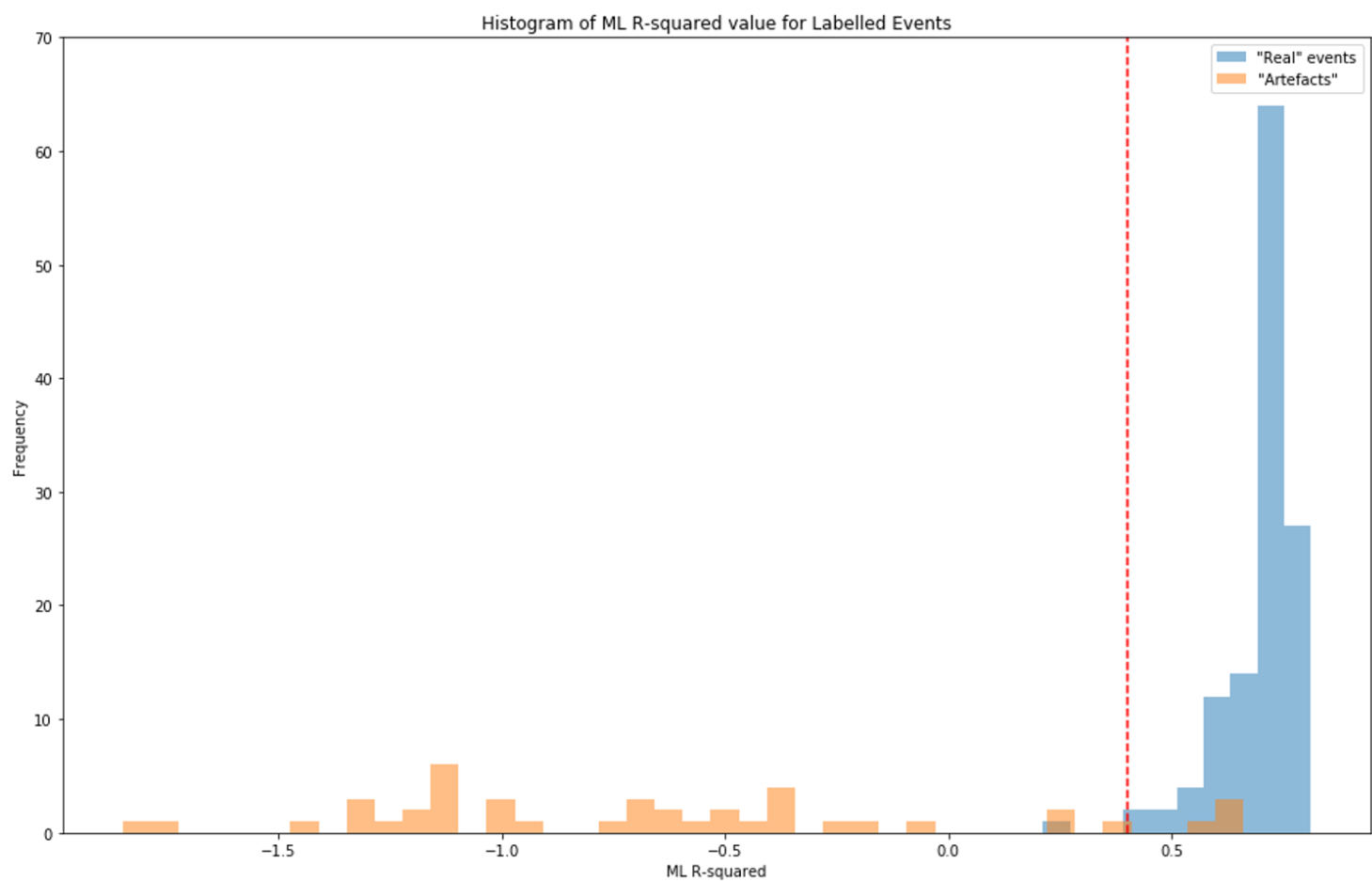
It is immediately apparent in Figure 1 that the covariance error statistic does a good job of separating real events (blue) from artefacts (orange). To select a filtering threshold, a precision & recall analysis was undertaken to find a threshold level with an acceptable ratio of true to false-positives (Figure 2). This also provides a measure of the classification power, characterised by the area under the curve (AUC) statistic - here 0.89 (0.5 = random chance; 1.0 = perfect classification accuracy).



**Figure 2:** Precision-recall curve illustrating the classification performance when filtering by covariance error at a range of filtering thresholds (black labels - value is covariance error filter threshold in metres). Precision = True Positives (TP) / (TP + FP) (proportion of retained events which are real). Recall = TP / TP + False Negatives (proportion of real events correctly retained)

A fairly strict filter of 500 m was chosen, corresponding to a recall rate of ~ 50%, with a precision of > 95% (i.e. < 5% false-positive rate). However it is important to note that this type of analysis gives the user the power to select a filter that suits their needs: they may instead choose to prioritise recall (i.e. minimising the number of false negatives).

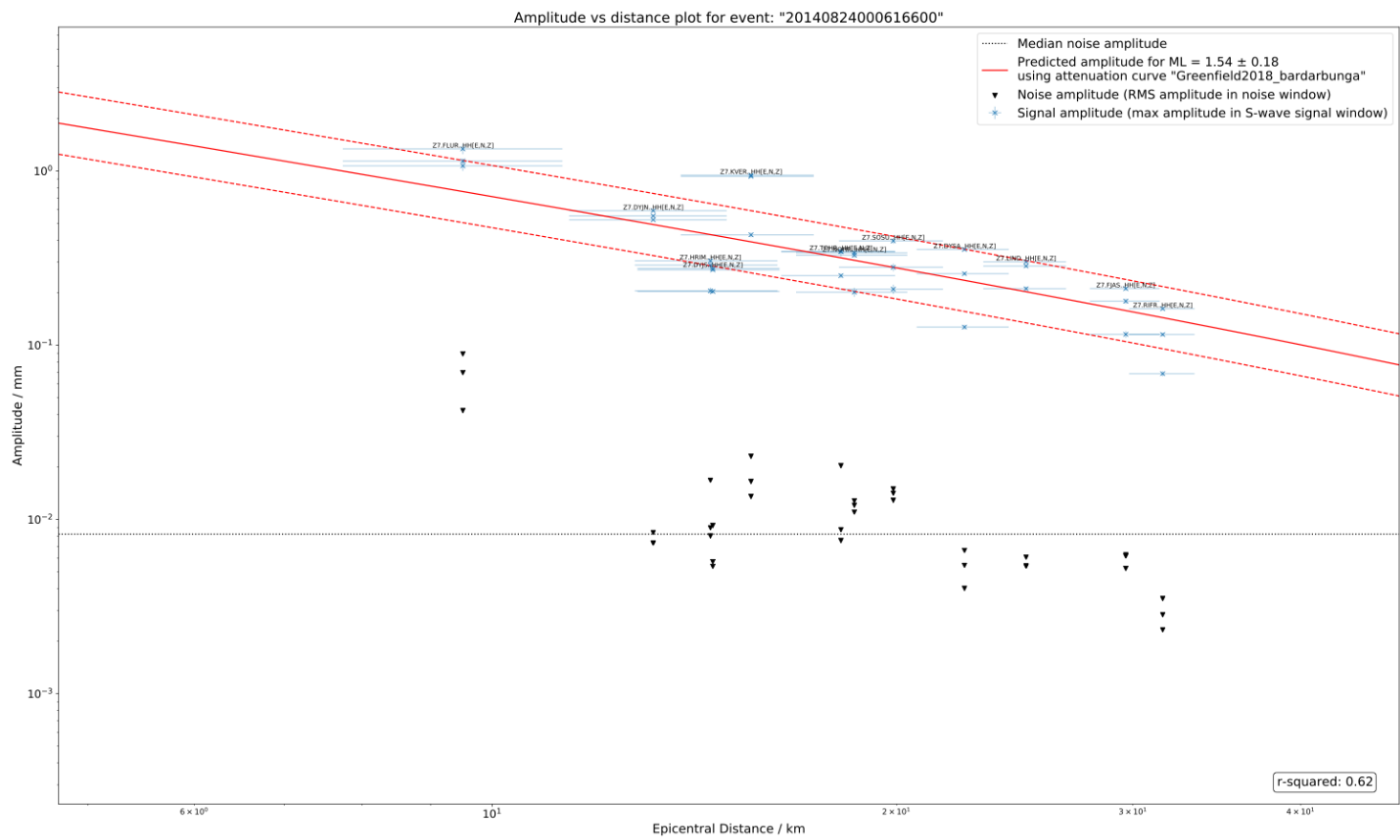
### Dike-induced earthquakes during the 2014 Bárðarbunga-Holuhraun dike intrusion



**Figure 3:** Histogram of Local Magnitude  $r^2$  for a subset of 169 labelled events identified by QM during the 2014 dike intrusion. Example filter threshold illustrated in red.

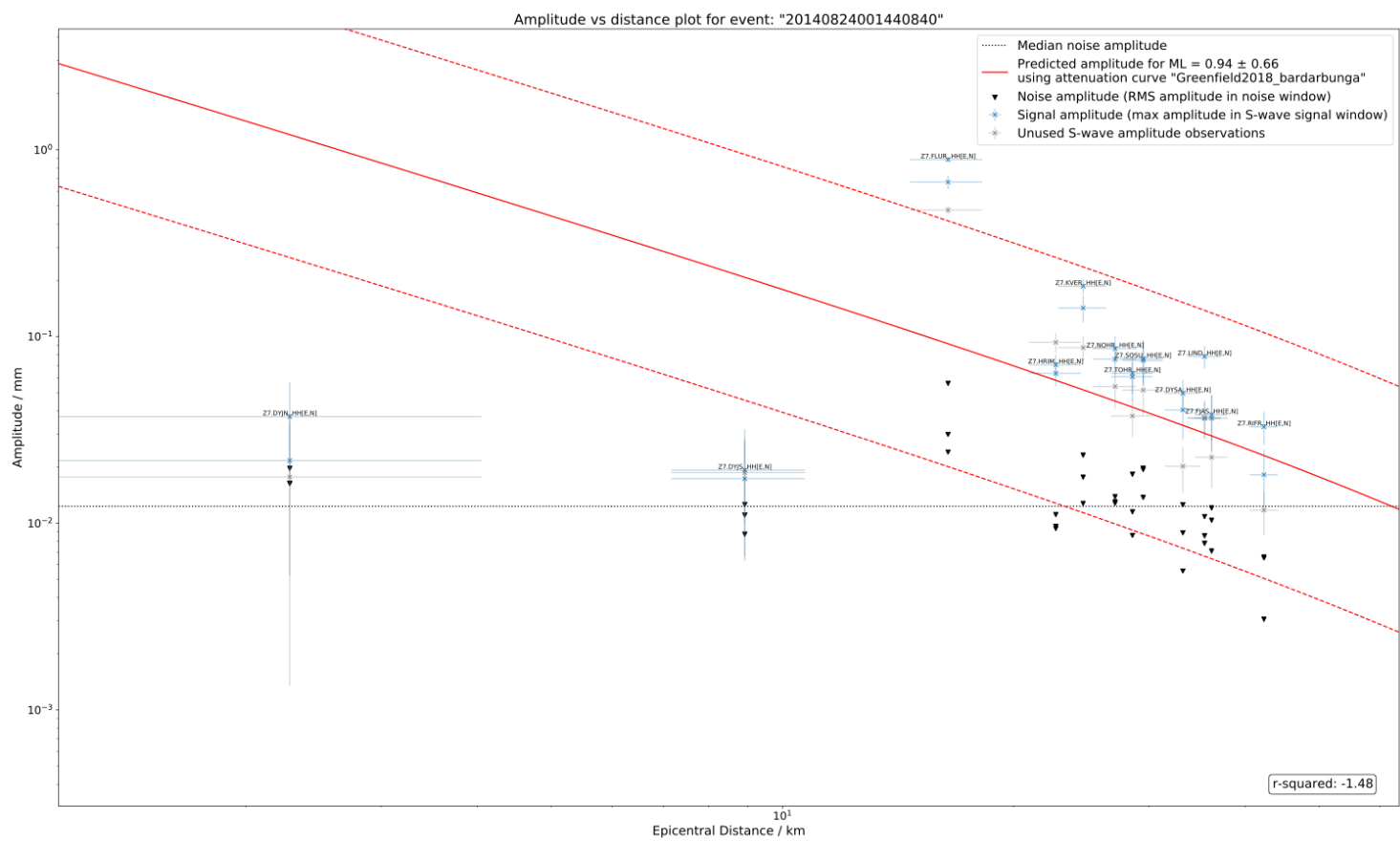


For this example a novel statistic was used to distinguish between real events and artefacts. Wood-Anderson simulated amplitudes were measured for S waves on the horizontal components for local magnitude calculation. The network-averaged local magnitude and region-specific attenuation curve were then used to compare predicted to observed amplitudes at each station (Figure 4).



**Figure 4:** Amplitude vs. distance plot (automatically generated by QuakeMigrate) illustrating the good agreement between predicted and observed S-wave amplitudes for a "real" event.

For real events, the agreement is good, leading to a high  $r^2$  value. For artefacts (Figure 5) caused by mislocations, or misidentification of noise as phase arrivals, the amplitude distribution across the network does not correspond to the expected S-wave attenuation curve. A precision-recall analysis reveals that this provides an exceptionally powerful tool to generate robust and highly complete event catalogues - with an AUC score (classification performance) of 0.98. The covariance error statistic performed similarly to the icequake case, with an AUC of 0.91.



**Figure 5:** Amplitude vs. distance plot for a mislocated event. The event was correctly located at its true origin time (around 5 seconds later).



# QUAKEMIGRATE: AUTOMATED, EFFICIENT AND MODULAR

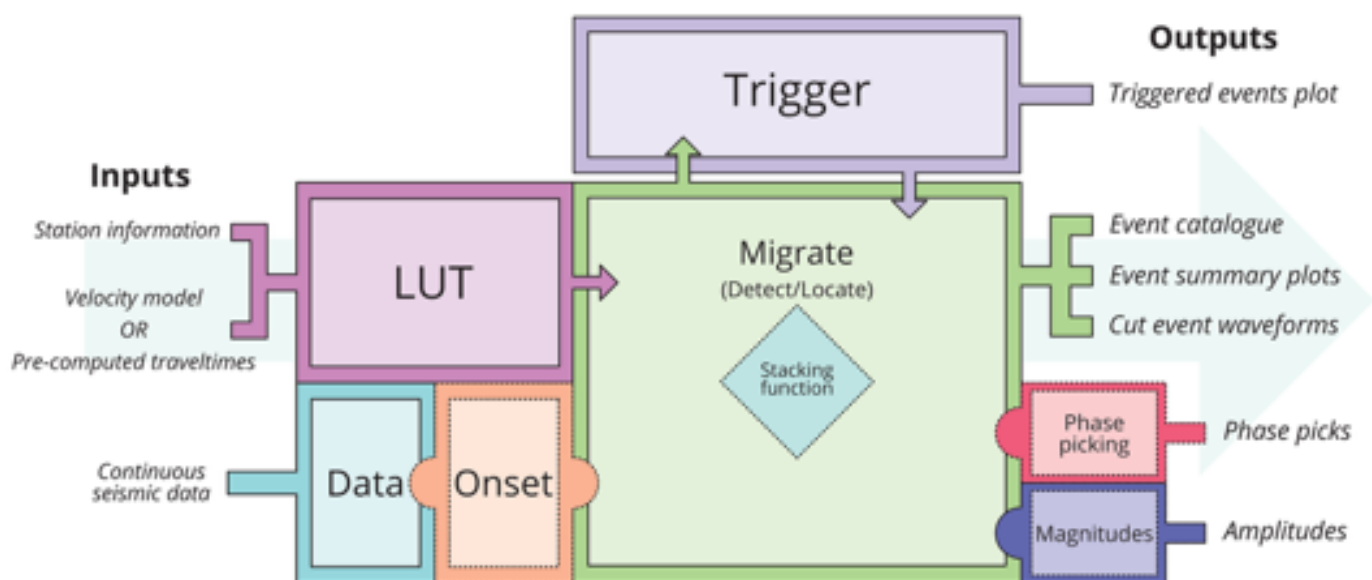
**QuakeMigrate** (<https://github.com/QuakeMigrate/QuakeMigrate/tree/development>) is a modular, open-source implementation of the waveform-based approach to seismic event detection and location written in Python.

The software package is built to work on Linux, Mac and Windows, and can be obtained via pip, conda or directly from GitHub.

## Package architecture

We have built QuakeMigrate to be automated, efficient and extensible. The core stacking functions are written and compiled in C, while the majority of the package is written in Python, all comprehensively [documented](https://quakemigrate.readthedocs.io/en/development/index.html) (<https://quakemigrate.readthedocs.io/en/development/index.html>).

The process of detection and location has been subdivided to increase the flexibility and efficiency of the workflow. This allows the user to specify different onset function and migration parameters for detection and location, where the objectives are subtly different.



**Figure 1:** Schematic diagram illustrating the package architecture and (from left to right) the workflow to generate a catalogue from archived continuous seismic data.

For **detection**, robustness to noise (including environmental and instrumental noise), computational efficiency and the sensitivity to a wide range of event types is desirable. The full, continuous detect output is then saved in mini-SEED format and available for the user to inspect.

From this, candidate events can be **triggered** using a range of built-in algorithms, including spatial filters. Automatic plots give great power to the user to determine the most suitable parameters for their dataset.

Subsets of candidate events are then **located**, by re-computing the 3D migration for a short time window around the candidate event origin time. The objectives here are to calculate accurate and precise locations and robust location uncertainties. As such, a denser grid may be used, with onset function parameters tuned for this specific subset of events (based, for example on the source impulsiveness and frequency content).

In addition, **locate** outputs:

- a range of plots to further interrogate the results and optimise parameter choices
- local magnitudes
- cut waveforms
- automatic phase picks.

A suite of utility functions provide an interface between QuakeMigrate and a selection of widely-used software packages for further analysis (e.g. Snuffler, for manual arrival time and polarity picking, or MFast, for shear-wave splitting analysis) to make it straightforward to include into existing research workflows.

## How to run QuakeMigrate

There are 3 key inputs required to use QuakeMigrate:

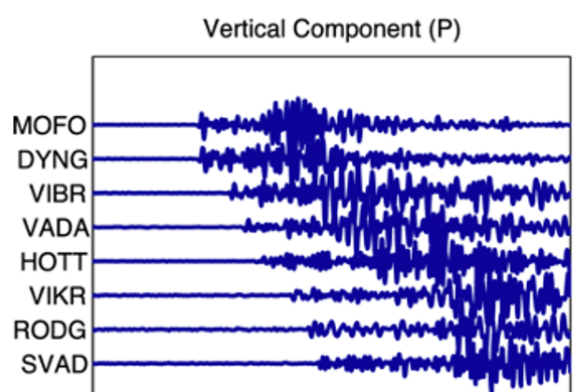
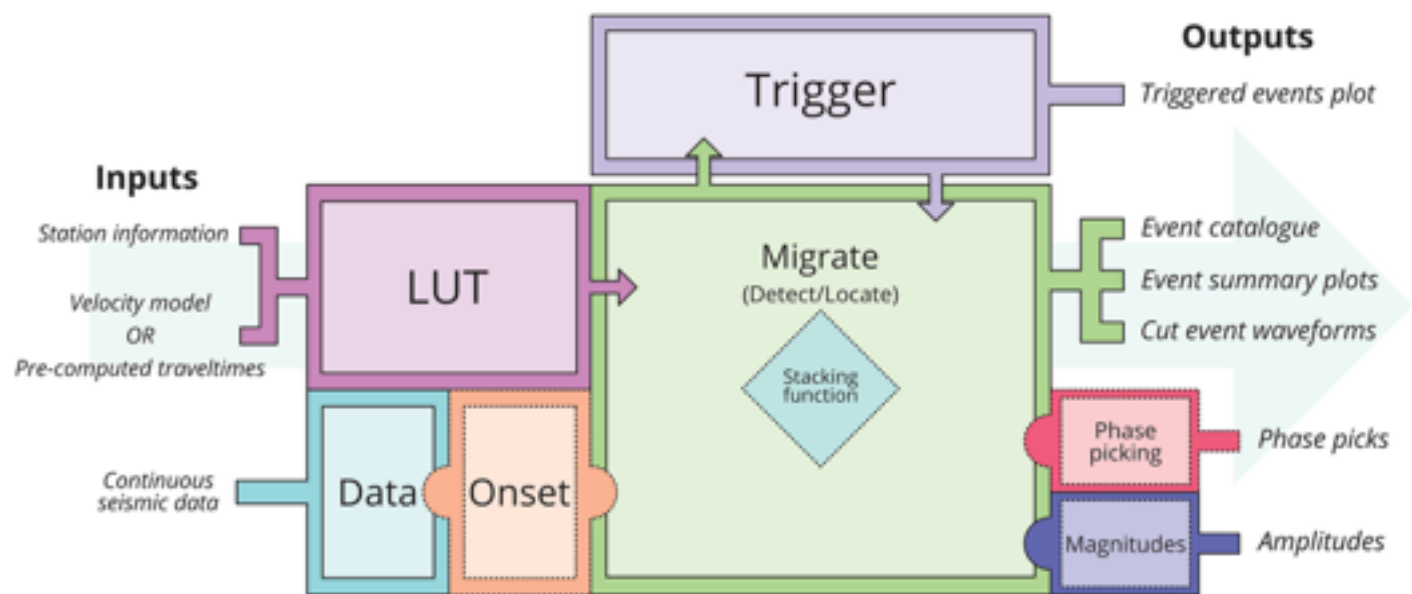
1. Station locations (lat, lon, elevation)
2. A velocity model (homogeneous, 1D, or 3D) **OR** a pre-computed lookup-table in the format used by NonLinLoc
3. Continuous waveform data, stored in some form of archive structure

## Workflow:

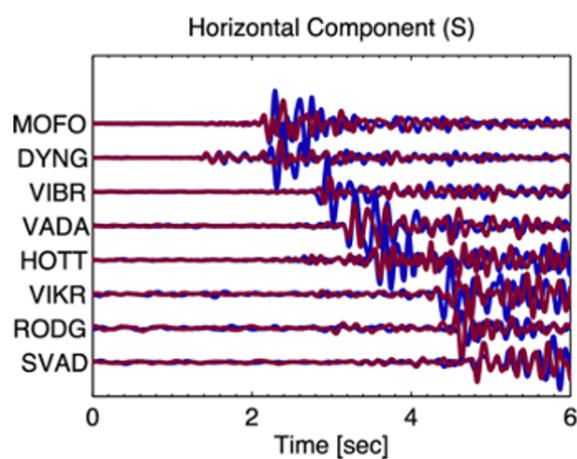
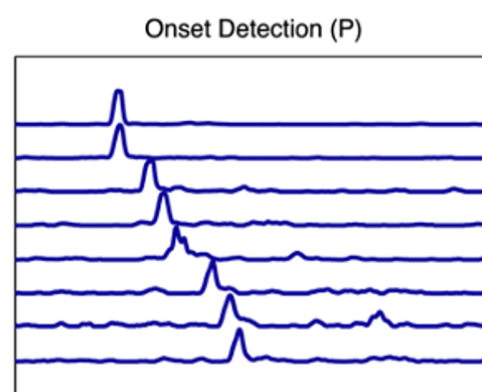
1. Import or calculate your lookup-table. This implicitly includes selecting the 3D grid over which you wish to search for possible event hypocentres.
2. Run **detect**. As described above, this consists of continuously migrating onset functions through the 3D grid throughout the specified time window. The user specifies the onset function algorithm and associated parameters (e.g. STA / LTA windows and bandpass filters). The output is a continuous stream containing the amplitude and location of the coalescence peak in the 3D grid at each timestep. This is saved in mini-SEED format (taking advantage of its excellent compression ratio) to afford maximum flexibility to the user.
3. Identify candidate events from the detect output using **trigger**. By default, this will identify peaks in the coalescence-through-time stream above a static trigger threshold specified by the user. A range of more sophisticated options are available - please see the tutorials and GitHub examples for details!
4. Calculate refined locations for the candidate events identified in trigger using **locate**. Here we re-compute the migration, but with the option to use a denser search grid than used for detect, potentially also with specially tuned onset function parameters, and only for a short time window around the candidate event origin time. The user can here select a phase picking algorithm and specify an instrument response inventory and response removal parameters to calculate local magnitudes.
5. The array of statistics output by **locate** can now be used to filter out false triggers, and generate a robust earthquake catalogue.



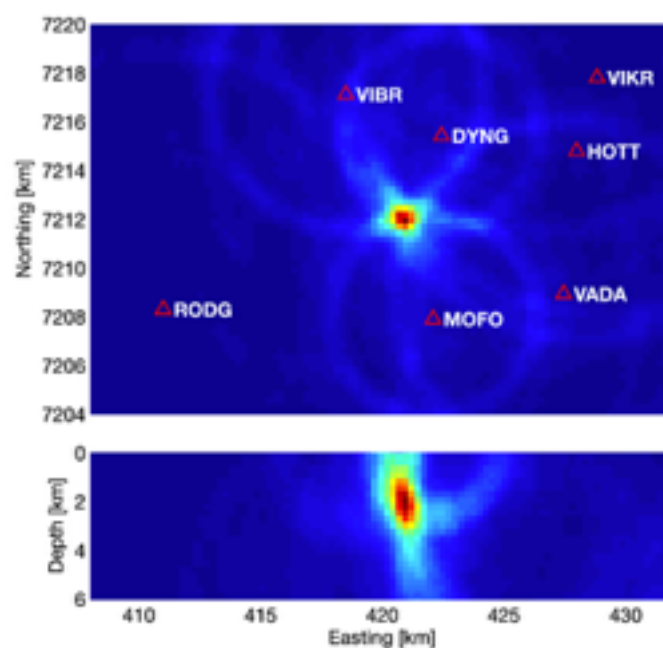
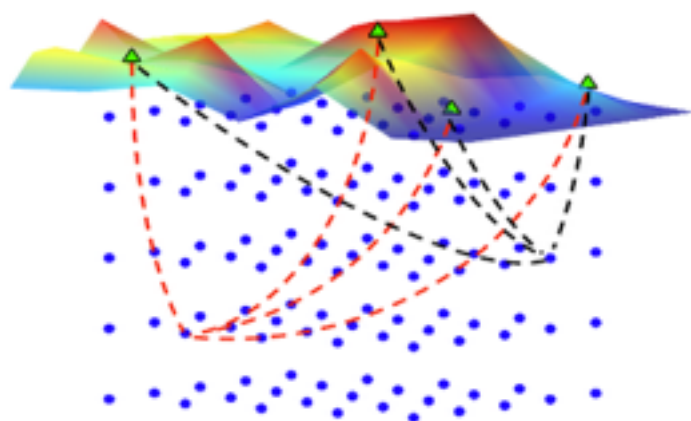
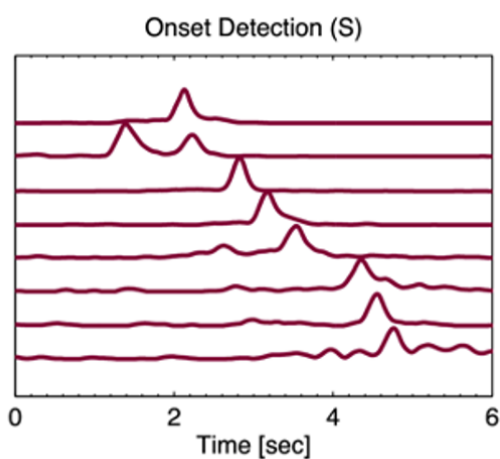


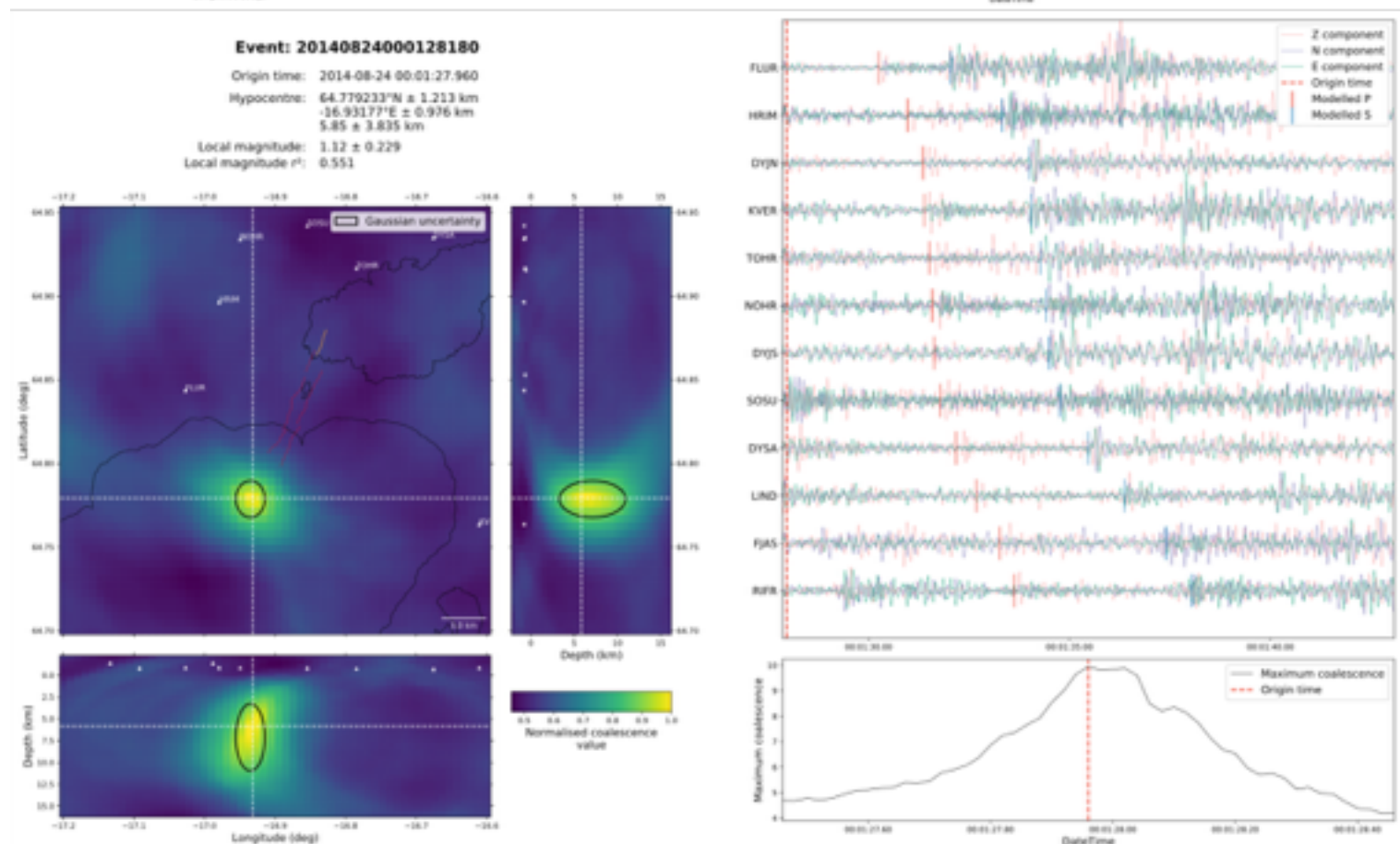
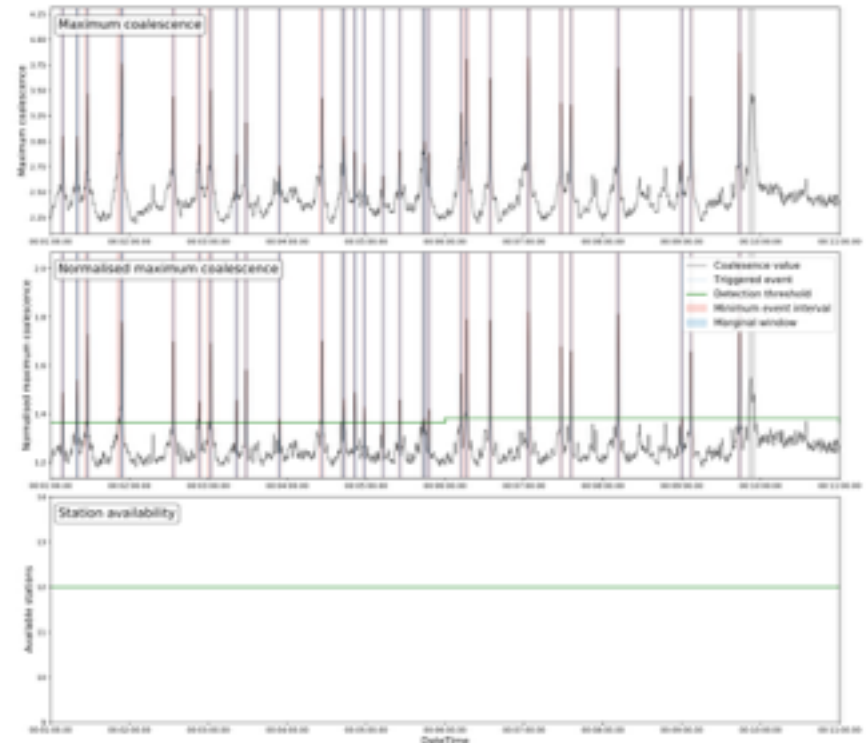
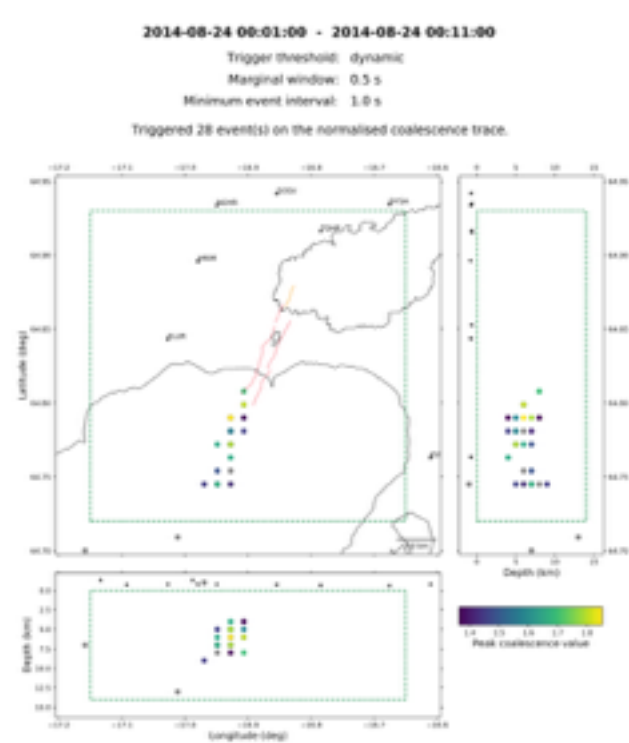


STA/LTA  
0.2s / 1.0s

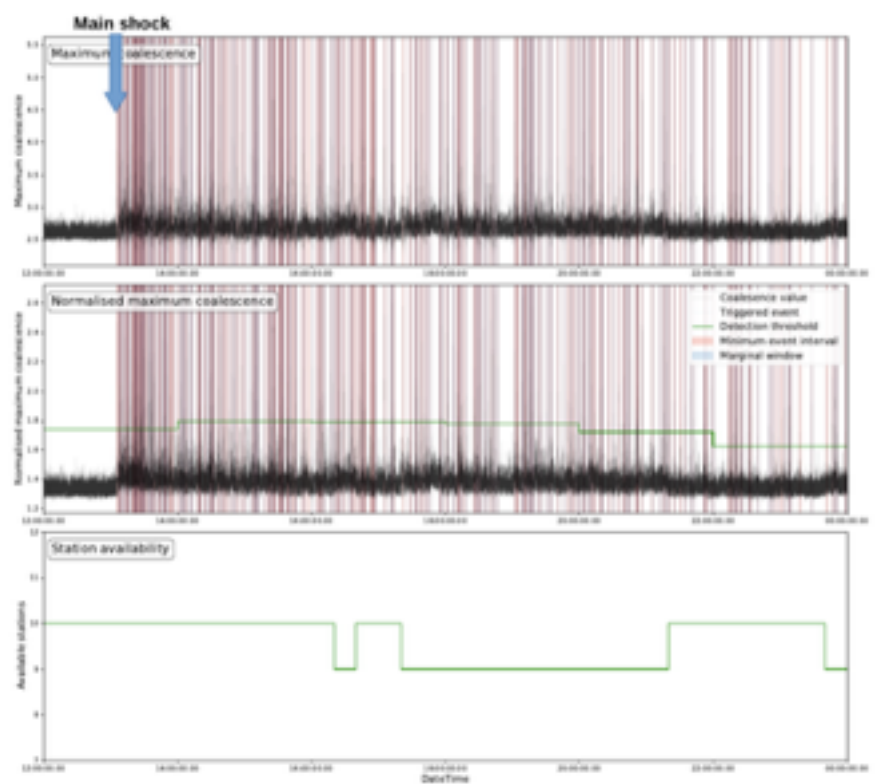
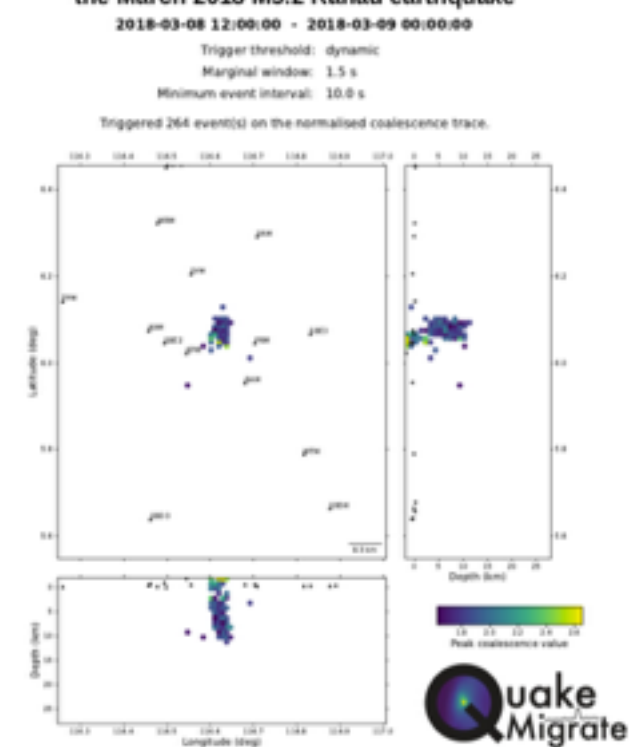


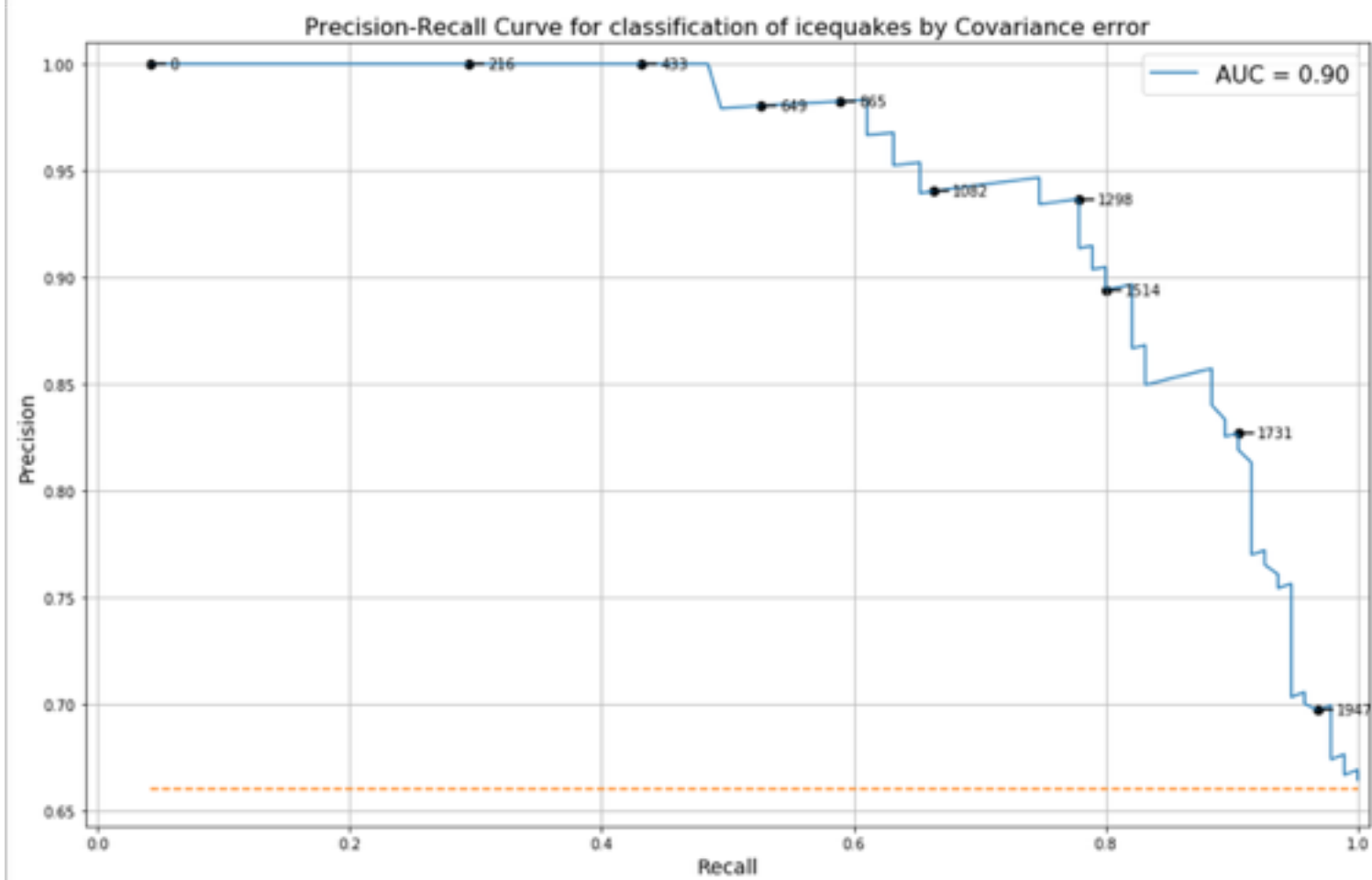
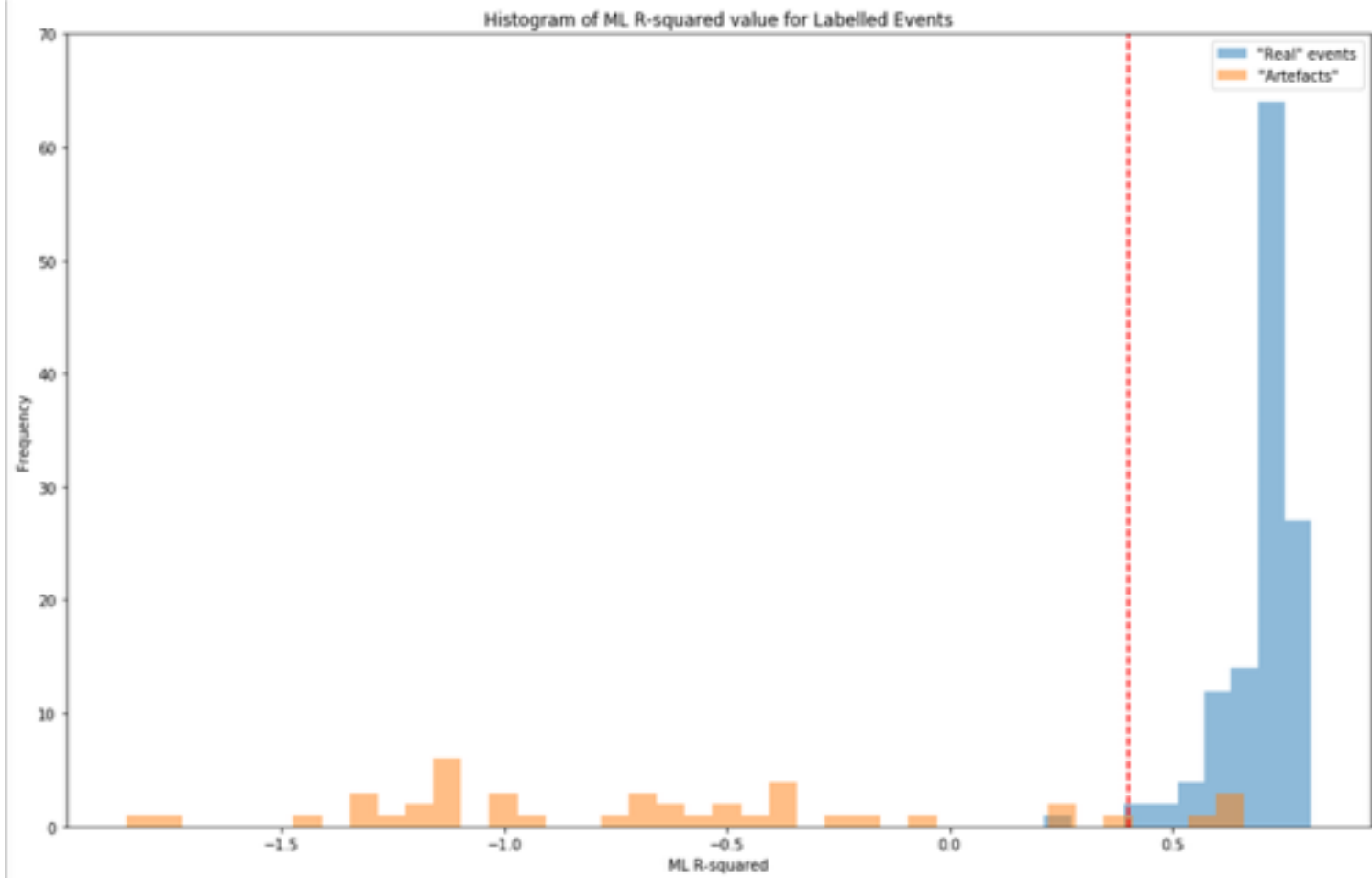
STA/LTA  
0.2s / 1.0s





### Trigger plot for the aftershock sequence of the March 2018 M5.2 Ranau earthquake





# FIND US ON GITHUB!

If you have comments, requests or ideas, please drop us an email or find us on

[GitHub](https://github.com/QuakeMigrate/QuakeMigrate/tree/development) (<https://github.com/QuakeMigrate/QuakeMigrate/tree/development>)!

## Future: comparing alternative onset and stacking algorithms

The modular architecture of the package makes it straightforward to substitute steps in the workflow, improving performance and adding flexibility for the wide range of potential use cases, stretching from induced acoustic emission experiments at laboratory scale to detecting earthquakes using national seismic networks.

For example, a variety of onset function algorithms have so far been proposed and implemented (e.g. kurtosis - [Langet et al., 2014](#) (<https://doi.org/10.1785/0120130107>), cross-covariance - [Grigoli et al, 2014](#) (<https://doi.org/10.1093/gji/ggt477>)) in addition to the simple STA/LTA algorithm used by default in QuakeMigrate. This module is implemented with an abstract base class, encouraging users to contribute plug-and-play alternatives, the performance of which can then be rigorously compared within a consistent architecture.

The picker module is similarly implemented, and there are likewise a multitude of phase picking algorithms which might be leveraged here. In all cases, they can take advantage of the benefits afforded by the "locate then pick" approach, where the timing of the true phase arrival is known to occur within some short time interval around the modelled phase arrival time.

In addition to these peripheral modules, the migration module at the core of the QuakeMigrate package is also written such that a range of alternative stacking functions may be implemented and their performance rigorously compared - for example the coherence stacking technique used by Grigoli et al.

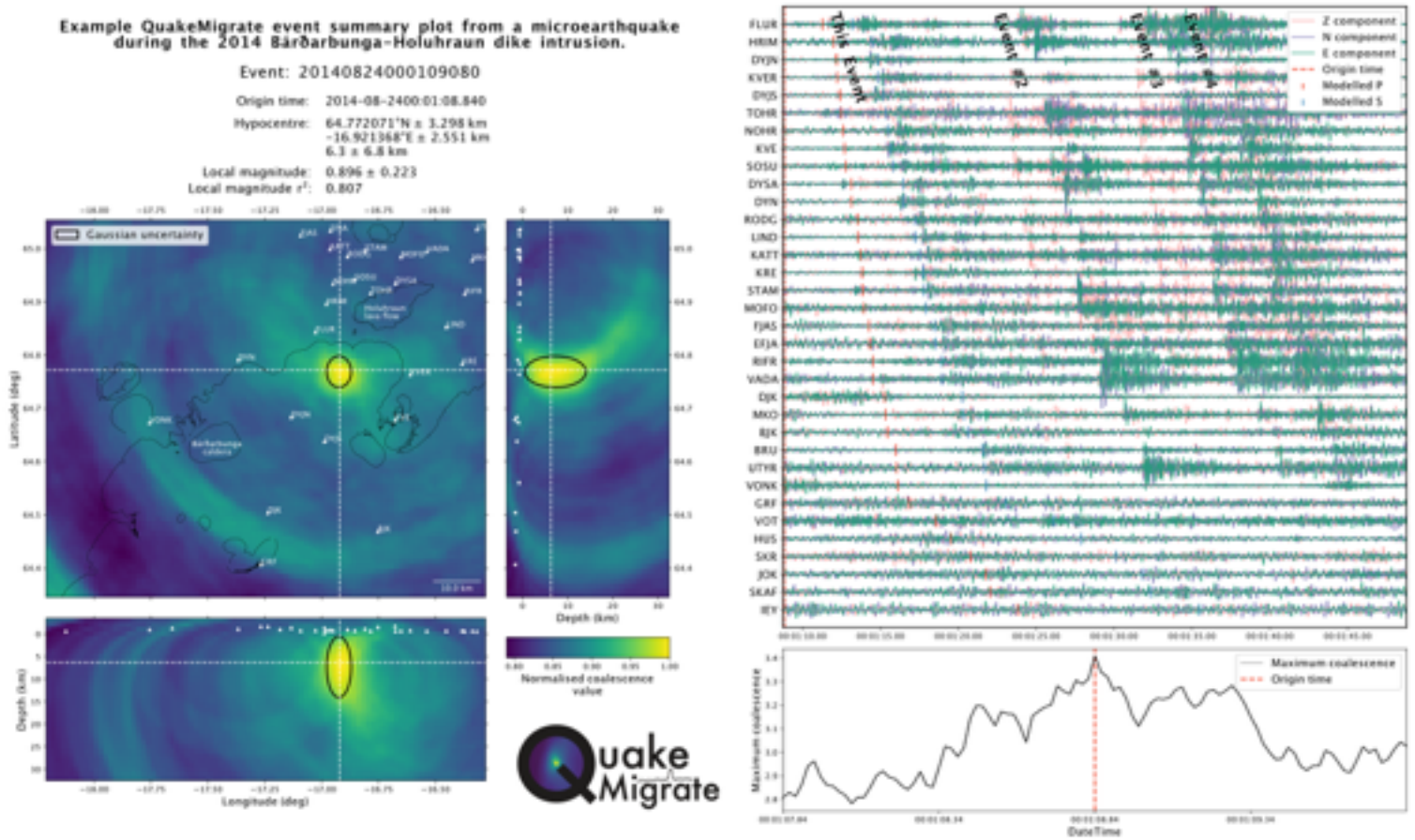


# ABSTRACT

Detecting and locating microearthquakes from continuous waveform records is the fundamental step in microseismic processing. Dense local networks and arrays have introduced the possibility to detect large numbers of far weaker events, but when viewed on seismic records from individual stations their waveforms are often obscured by noise. Furthermore, areas of interest for microseismic monitoring often feature extremely high event rates, highlighting the limitations of traditional techniques based on phase picking and association. In order to maximise the new insights gained, we require fully automated techniques which can exploit modern recordings to produce highly complete earthquake catalogues containing few artefacts.

QuakeMigrate is a new modular, open-source python package providing a framework to efficiently, automatically and robustly detect and locate microseismicity. The user inputs continuous seismic data, a velocity model or pre-calculated look-up table and list of station locations. Instead of reducing the raw waveforms to discrete time picks, they are transformed (by amplitude, frequency and/or polarisation analysis) to continuous functions representing the probability of a particular phase arrival through time. These ‘onset functions’ from stations across the network are then migrated according to a travel-time look-up table and stacked to perform a grid-search for coherent sources of energy in the subsurface. This enables detection of earthquakes at close to or below the signal-to-noise ratio at individual stations, and implicitly associates phase arrivals even at very small inter-event times.

We demonstrate the flexibility and power of this approach with examples of basal icequakes detected at the Rutford Ice Stream, Antarctica, dike- and caldera-collapse induced seismicity at Bárðarbunga central volcano, Iceland, and the aftershock sequence from a M5 earthquake at Mt. Kinabalu, North Borneo. The modular nature of the workflow and wide range of automatic plotting options makes parameter choice straightforward, and robust event location uncertainty statistics facilitate filtering to produce a robust catalogue. QuakeMigrate also outputs phase picks and local magnitude estimates, with an architecture designed to promote further community-driven extension in future.



([https://agu.confex.com/data/abstract/agu/fm20/2/1/Paper\\_763312\\_abstract\\_738095\\_0.png](https://agu.confex.com/data/abstract/agu/fm20/2/1/Paper_763312_abstract_738095_0.png))