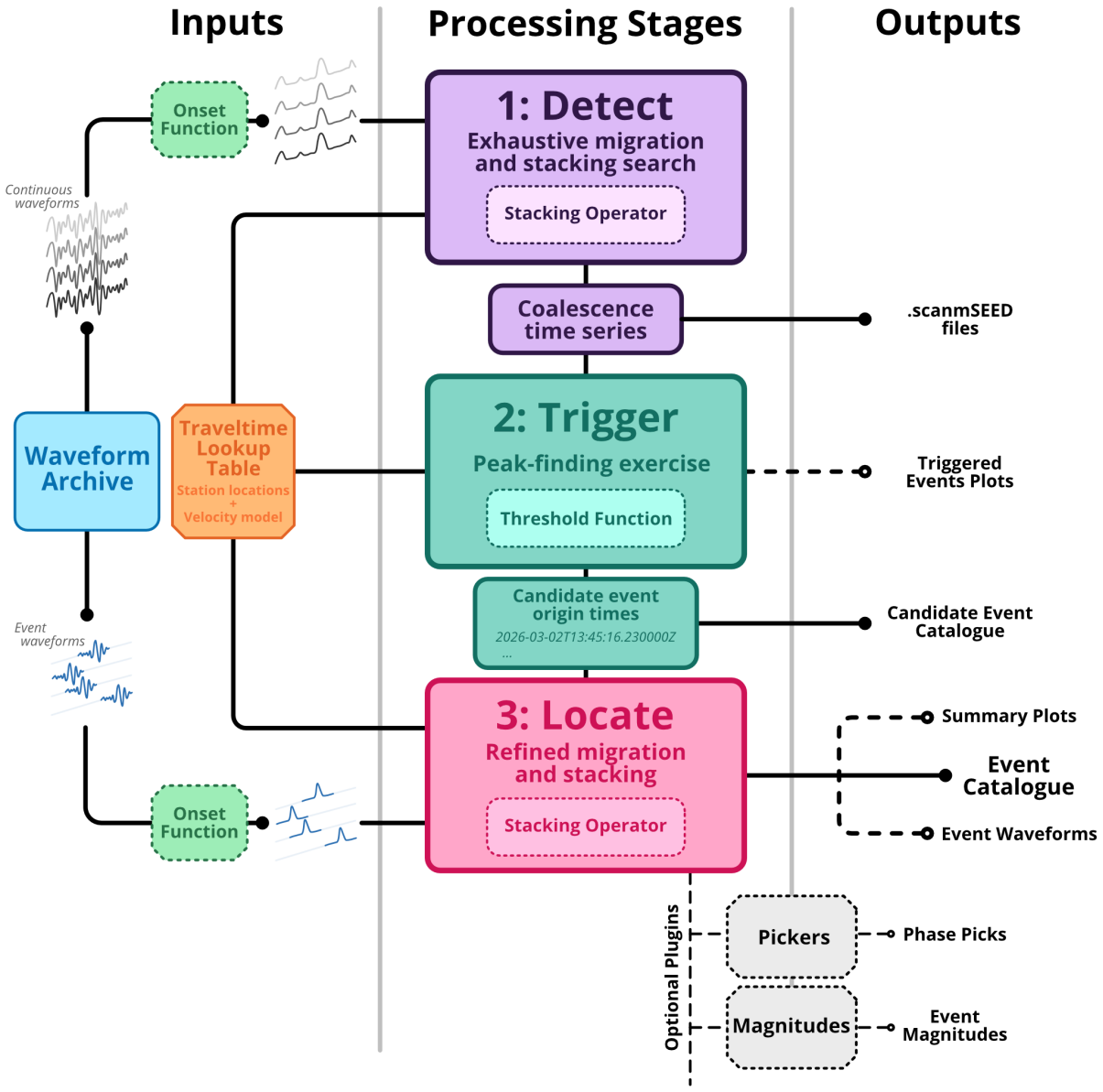
QuakeMigrate: a Python Package for Automatic Earthquake Detection and Location Using Waveform Migration and Stacking

Tom Winder* [*1,2], **Conor A. Bacon*** [1,3], **Jonathan D. Smith** [4], **Tom Hudson** [5], **Robert S. White** [1]

[1]Department of Earth Sciences, University of Cambridge, Cambridge, United Kingdom, [2]Institute of Earth Sciences, University of Iceland, Reykjavík, Iceland, [3]Lamont-Doherty Earth Observatory, Columbia University, New York, United States, [4]British Antarctic Survey, Cambridge, United Kingdom, [5]Department of Earth and Planetary Sciences, ETH Zurich, Zurich, Switzerland



Caption on next page.

*Corresponding author: tomwinder@hi.is

Figure S1 (Previous page) Schematic illustrating the workflow of a QuakeMigrate run. Sections with dashed outlines (Onset Function, Stacking Operator, Threshold Function, Pickers, Magnitudes) are plugin/extension modules that can be customised or substituted for alternatives by the user. The workflow runs from top to bottom, through the Detect, Trigger and Locate stages. Inputs and outputs for each of these are illustrated in the left column, with the Waveform Archive and Traveltime Lookup Table (LUT) shared across multiple stages. Outputs are shown on the right, with dashed lines and open circles indicating optional outputs. Note that—where these outputs are used as inputs for the following QuakeMigrate processing stage—they can be viewed, edited or pre-processed by the user as they see fit, so long as the format is maintained. In the extreme case, Locate can be run independently of the prior two stages, utilising a list of candidate event origin times provided from an alternative source.

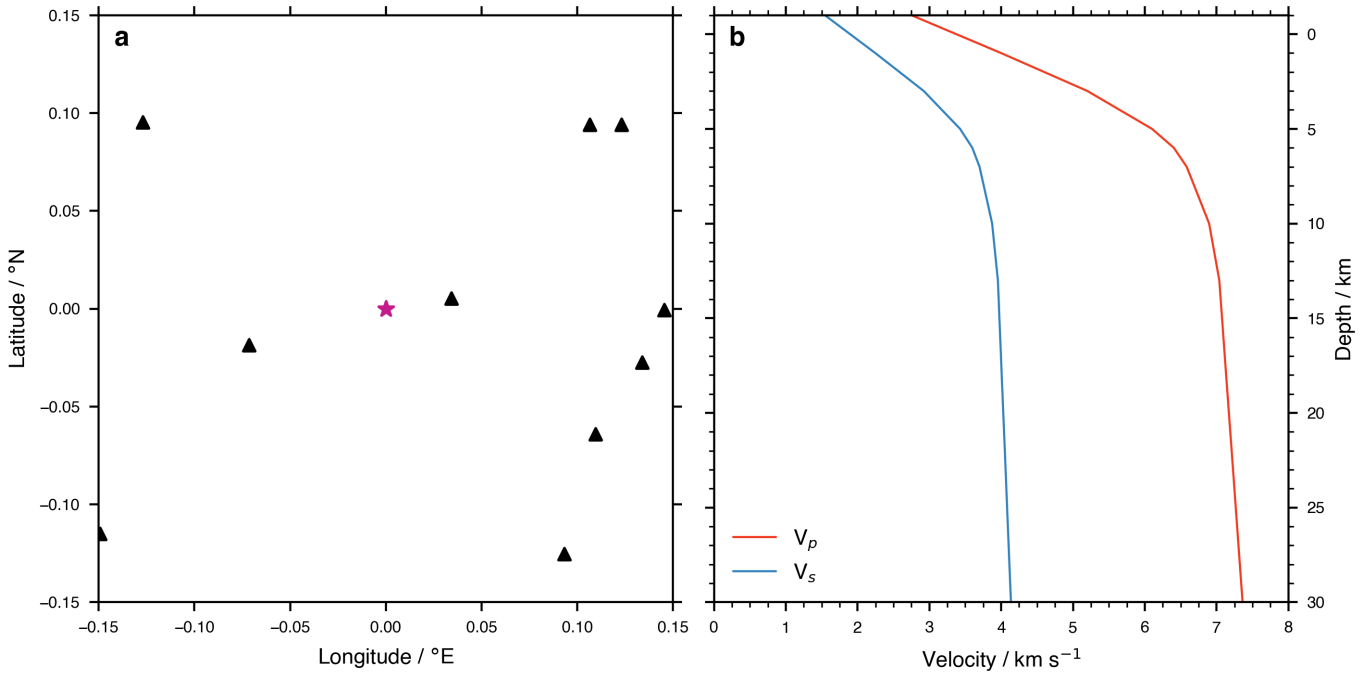


Figure S2 Network geometry and velocity model for the synthetic test. a) Map view of the seismic network, with black triangles denoting the seismic stations, and the synthetic source denoted by the pink star. b) The 1-D P- and S-wave velocity model.

S1 Generating the synthetic waveforms

The synthetic waveforms used in Section 3 are based on the simple Gaussian-derivative wavelet function. An initial wavelet is positioned within a 10 minute trace starting at 2022-02-18T12:00:00.0 such that the onset of the wavelet (i.e., the origin time of the earthquake) is at 2022-02-18T12:05:00.0. This template waveform is then time-shifted by a computed traveltime for each station and phase (stored in the traveltime lookup table). Noise, pulled from a normal distribution with a standard deviation of 0.02 s, is added to each of these traveltimes to simulate the effect of velocity model inaccuracies. The P- and S-phase waveforms are assigned to the vertical and horizontal components of the 3-component synthetics. A 3-D rotation is then applied to this system of components to account for the azimuthal distribution of stations around the earthquake source. The elements of this rotation operator for each station are computed using the back-azimuth between the station and the source, and an angle-of-incidence to the surface computed from the gradient of the traveltime grid in the vicinity of the station. The angle-of-incidence as a function of radial distance from the source to the receiver is shown in Figure S3. The effect of attenuation as a function of distance is artificially added using the empirical equation of Hutton and Boore (1987), with a source local magnitude of 1 M_L. Finally, noise, pulled from a normal distribution with a standard deviation of 0.1, is added to the waveform amplitudes to simulate the ambient noise field of the Earth. The result of the two noise-adding processes are portrayed in Figure S4.

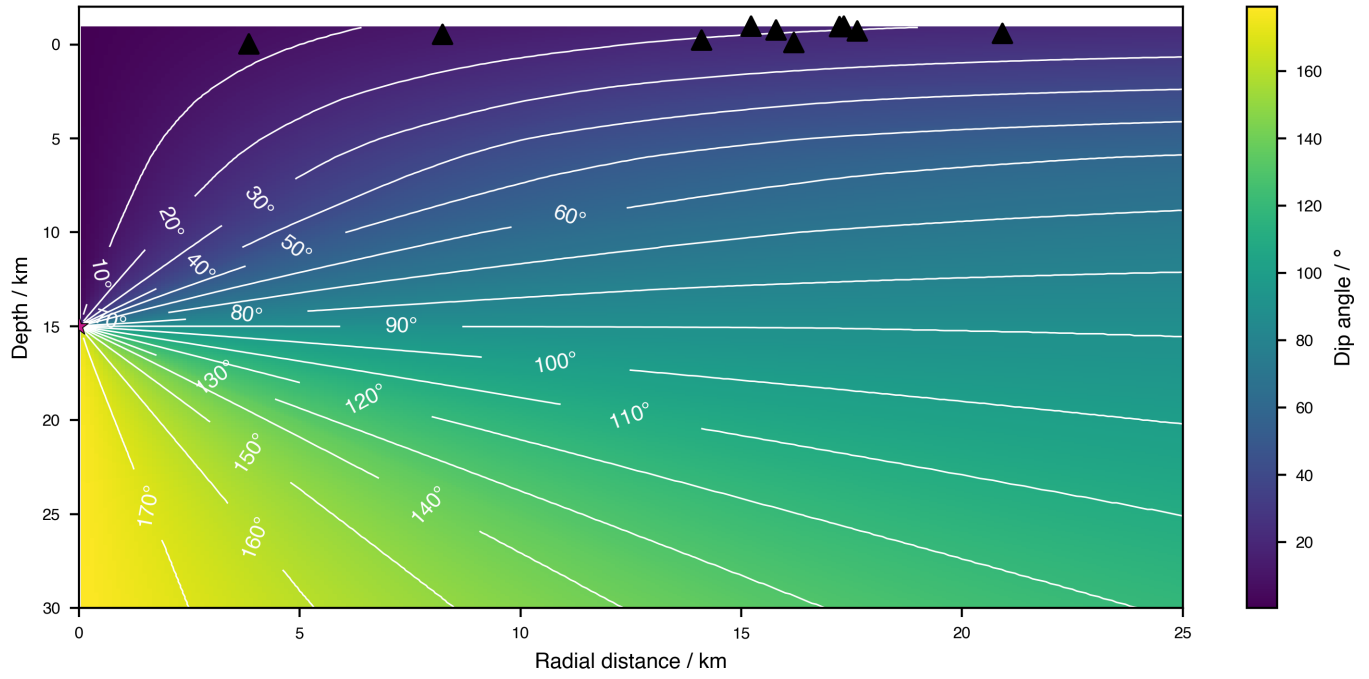


Figure S3 Behaviour of the angle-of-incidence (i.e., the dip angle of the ray vector with the vertical) as a function of receiver depth and radial distance from source to receiver for a source at 15 km depth. 10° contour lines are indicated by the white lines. The source and receivers are indicated by the pink star and black triangles, respectively.

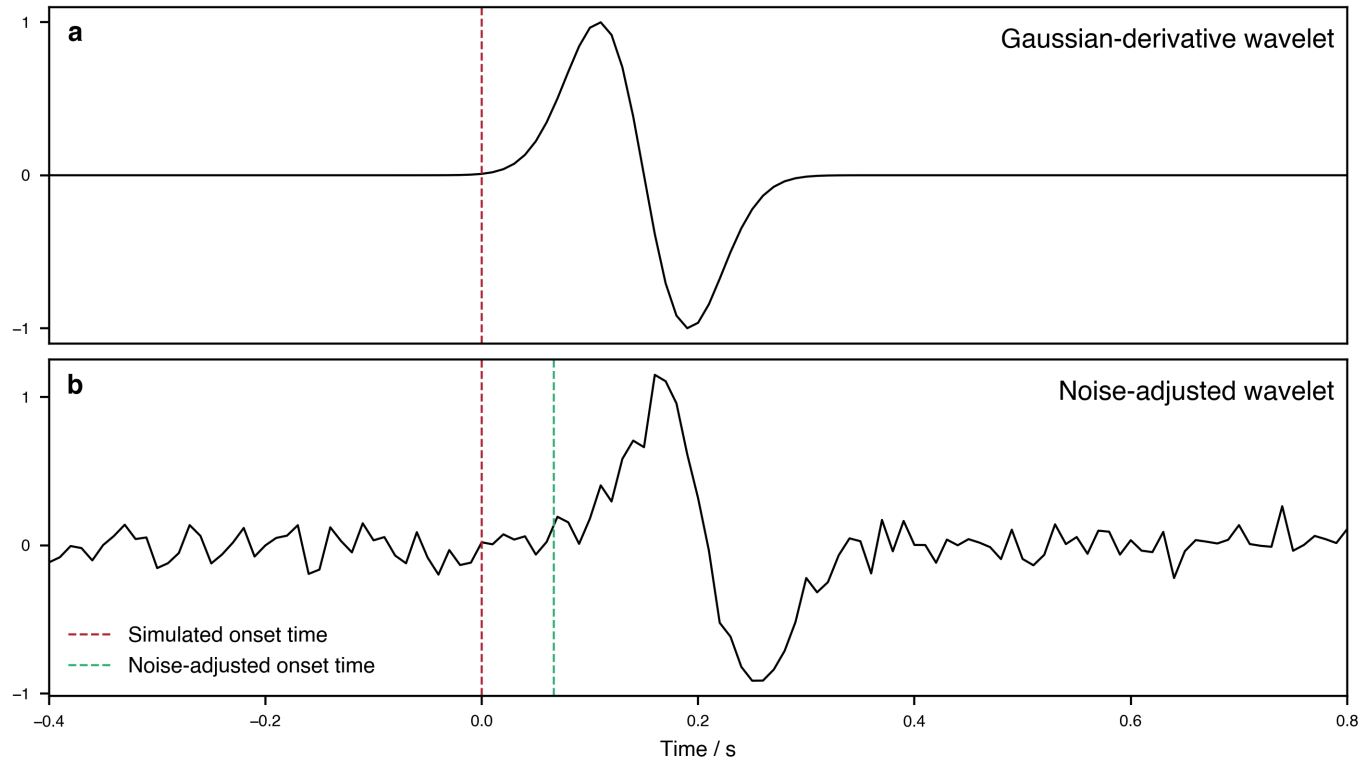


Figure S4 Illustration of: a) the Gaussian-derivative wavelet and b) the effect of the synthetic noise added to the station-phase traveltimes and the waveform amplitudes.

S2 Impact of P-phase only waveform migration and stacking

To support the assertion that using both P- and S-phase onset functions leads to better detection and location performance (Section 3.8 of the main text), we here show the results for P-phase only side-by-side with the P- and S-phase derived results (for the synthetic example, in Figure S5, and for one event from the Askja example, in Figure S6). For the synthetic example the location difference is only 0.05 km laterally and 0.2 km in depth, and the origin time is identical, which highlights that for very well constrained event locations (with clear phase arrivals and excellent net-

work geometry) P-phase only locations can perform well, though the location uncertainty is still larger. In contrast, excluding the S-phase onset functions makes a substantial difference for the event from the Askja example (Figure S6), where the location is offset by 7.5 km laterally, and 0.3 km in depth, as well as a 0.72 s difference in origin time. The estimated location uncertainty is also significantly larger in all three dimensions—and particularly in depth—as well as the geometric mean covariance, corresponding to the significantly worse location constraint. This example makes it clear that for smaller events, in particular, using all available phases is essential to obtain a reliable location.

Any combination of phases can be passed into the core migration and stacking engine, so long as the corresponding traveltime lookup tables exist. The only difference between the two panels in each Figure is the phases used to perform the migration and stacking.

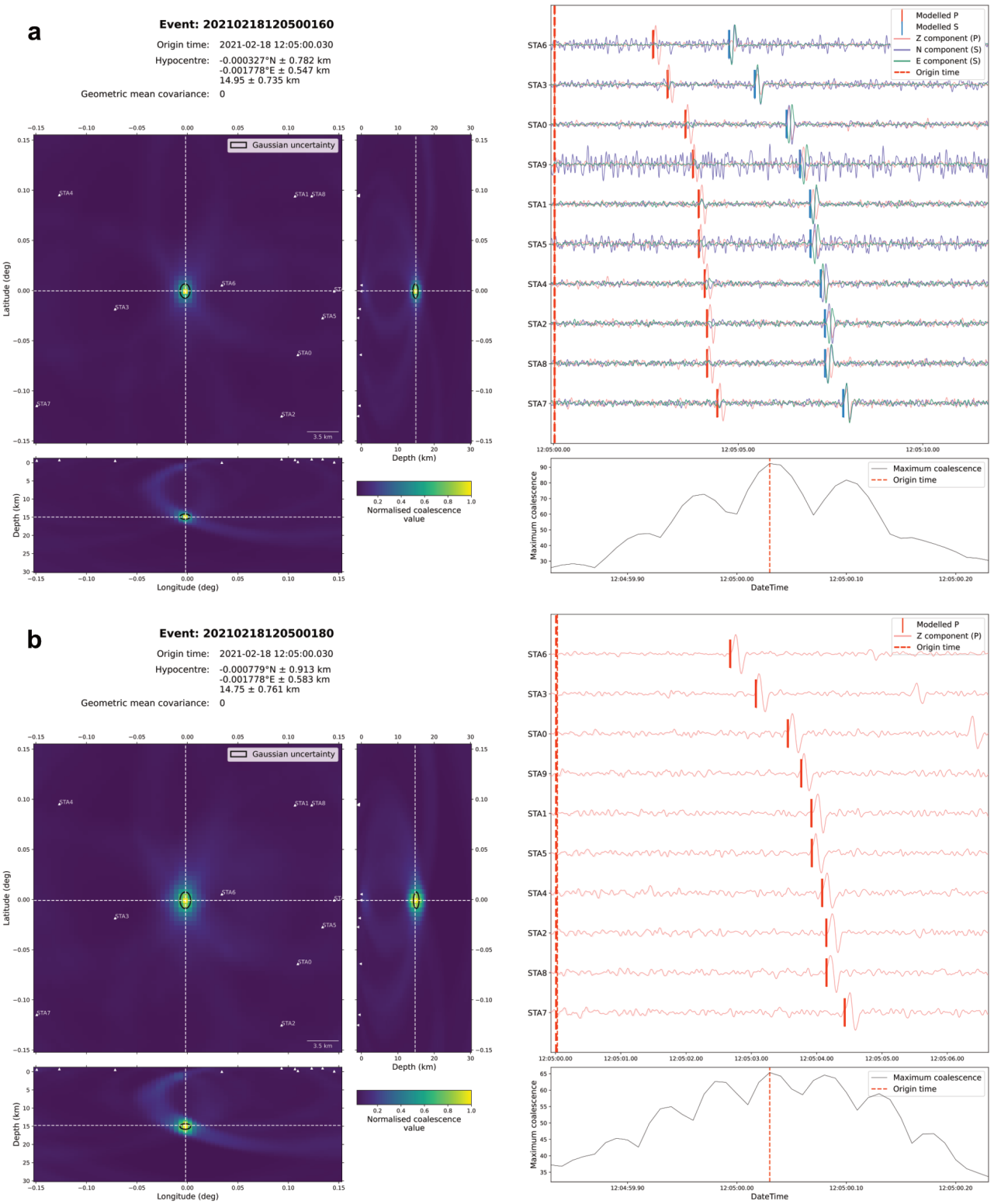


Figure S5 Comparison of the locations derived from **a** using both the P- and S-phase arrivals and **b** using purely P-phase arrivals (i.e., only calculating, migrating, and stacking the P-onset function from the Z component) for the synthetic example. These are examples of the built-in event summary generated as part of the standard QuakeMigrate Locate function.

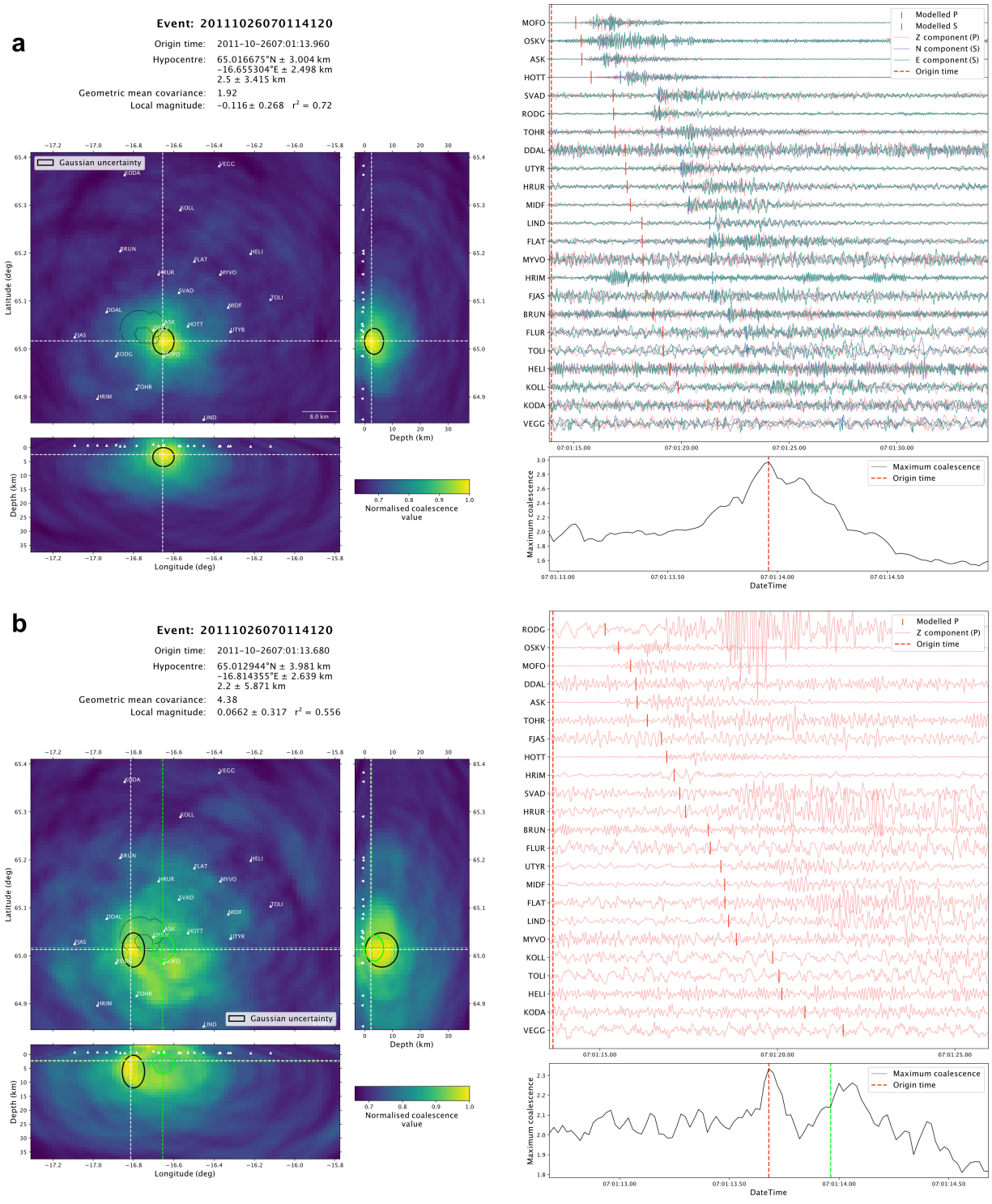


Figure S6 Comparison of the locations derived from **a** using both the P- and S-phase arrivals and **b** using purely P-phase arrivals (i.e., only calculating, migrating, and stacking the P-onset function from the Z component) for event 20111026070114120 from the Askja example. In **b** the location estimate and uncertainty ellipse from **a** (i.e., the reference location, from migrating both P- and S-phase onset functions) are annotated in light green on the map-view and cross-section slices, as well as the origin time in the bottom-right panel. These are examples of the built-in event summary generated as part of the standard QuakeMigrate Locate function.

S3 Profiling of QuakeMigrate

We provide here an overview of the computational memory and runtime scaling of QuakeMigrate as a function of tuning parameters by profiling the Askja volcano-tectonic and deep long period earthquake example presented in Section 4.2 of the main manuscript. In this analysis we look at the memory usage through time for each stage of QuakeMigrate, as well as how both runtime and peak memory consumption (known as the "memory high watermark") vary as a function of some of the key tuning parameters: the timestep, number of computational threads, and traveltime lookup table decimation. Code for reproducing these profiles is provided in the GitHub repository that accompanies this manuscript.

A single illustrative hour of waveform data (12:00–13:00 UTC on 2011-10-26) is used across all profiling to illustrate the runtime and peak memory usage. These profiling results presented in Figure S7 highlights the key trade-offs between a higher resolution grid (i.e., a lower decimation factor), the detect timestep (i.e., how much waveform data is processed, migrated, and stacked at a time), and the number of theads (i.e., how much the migration and stacking stage can be parallelised).

The profiling results presented below were generated using a MacBook Pro 14 (2021) with an Apple M1 Max CPU and 32 GB of RAM (512-bit LPDDR5 SDRAM memory). The Apple M1 Max has 8 high-performance "Firestorm" cores (3228 MHz clockspeed) and 2 energy-efficient "Icestorm" cores (2064 MHz). The 8 Firestorm cores are split into two clusters, with each cluster sharing 12 MB of L2 cache.

S3.1 Memory usage through time for each stage

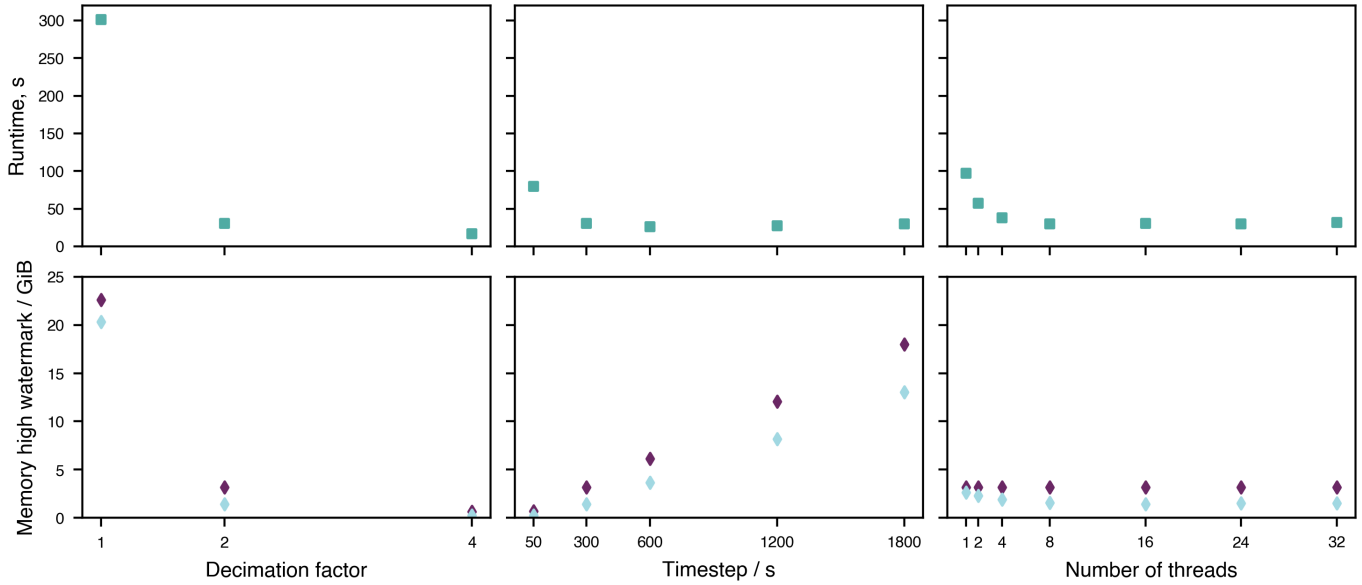


Figure S7 Spatial and temporal profiling of the Detect stage of QuakeMigrate using 1 hour of data (12:00–13:00 UTC on 2011-10-26) from the Askja example presented in Section 4.2. The top and bottom rows show the variation in runtime and peak memory usage, respectively. From left to right in these rows, we vary the decimation factor, the timestep, and the number of threads. In the memory usage panels, the purple diamonds indicate the peak memory usage, whereas the blue diamonds show the average memory usage over the course of the run.

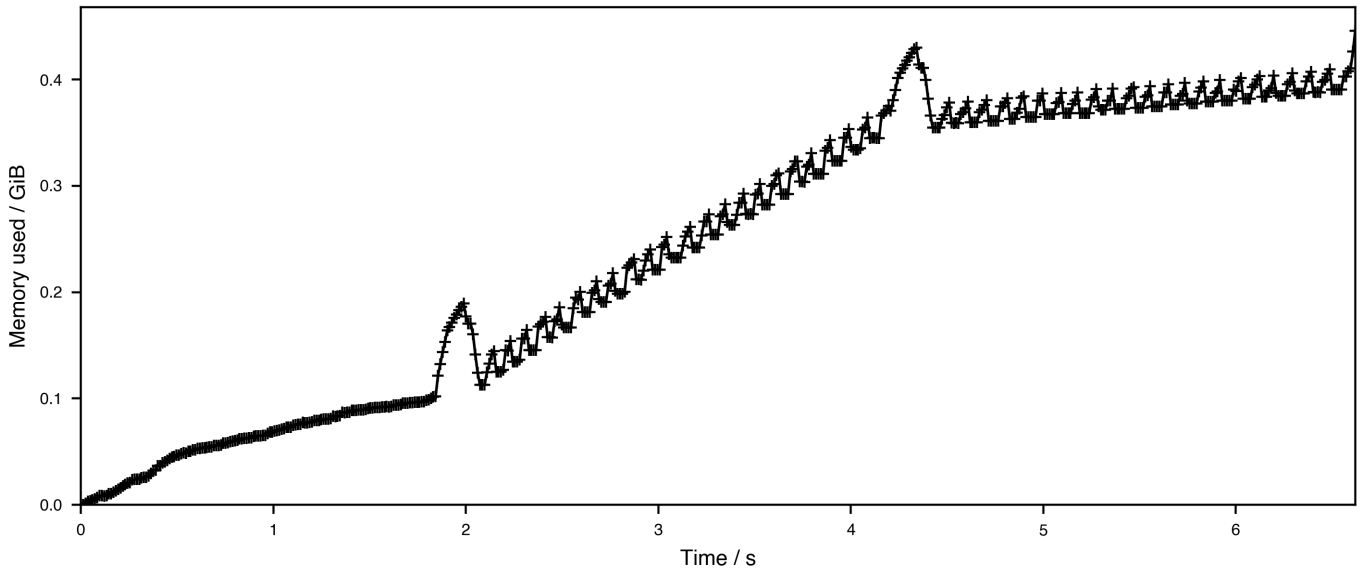


Figure S8 Overview of the memory usage through time for the creation of the traveltime lookup tables.

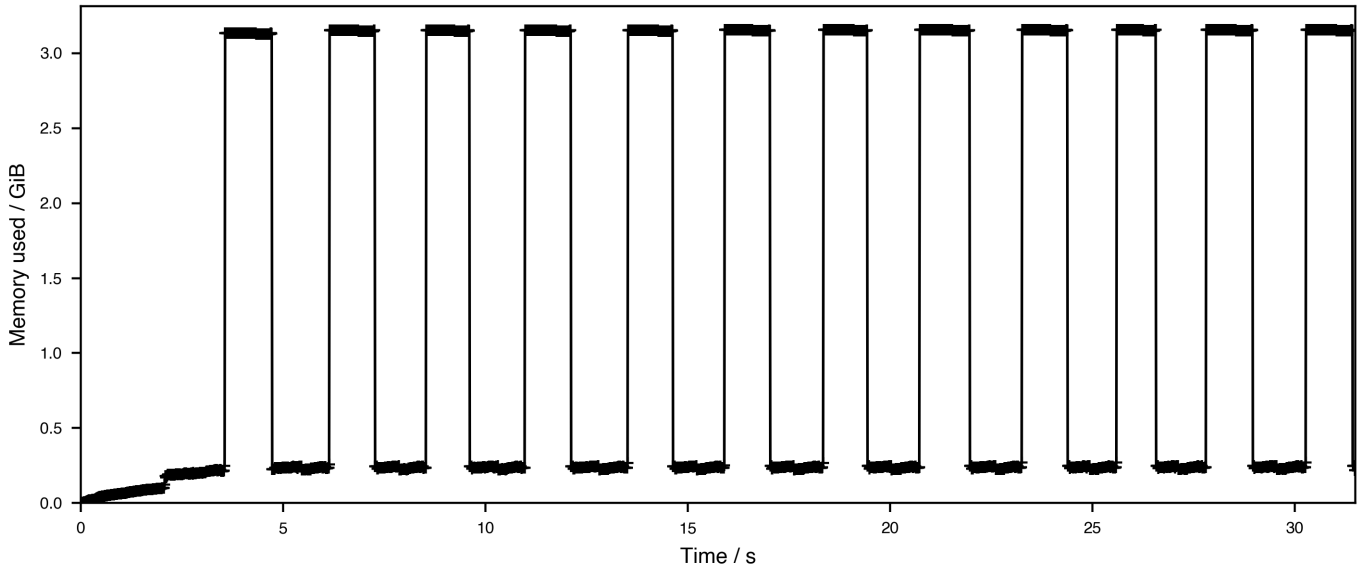


Figure S9 Overview of the memory usage through time for the Detect stage. It can be seen that the program cycles between low memory usage (waveform pre-processing) and high memory usage (during the migration and stacking).

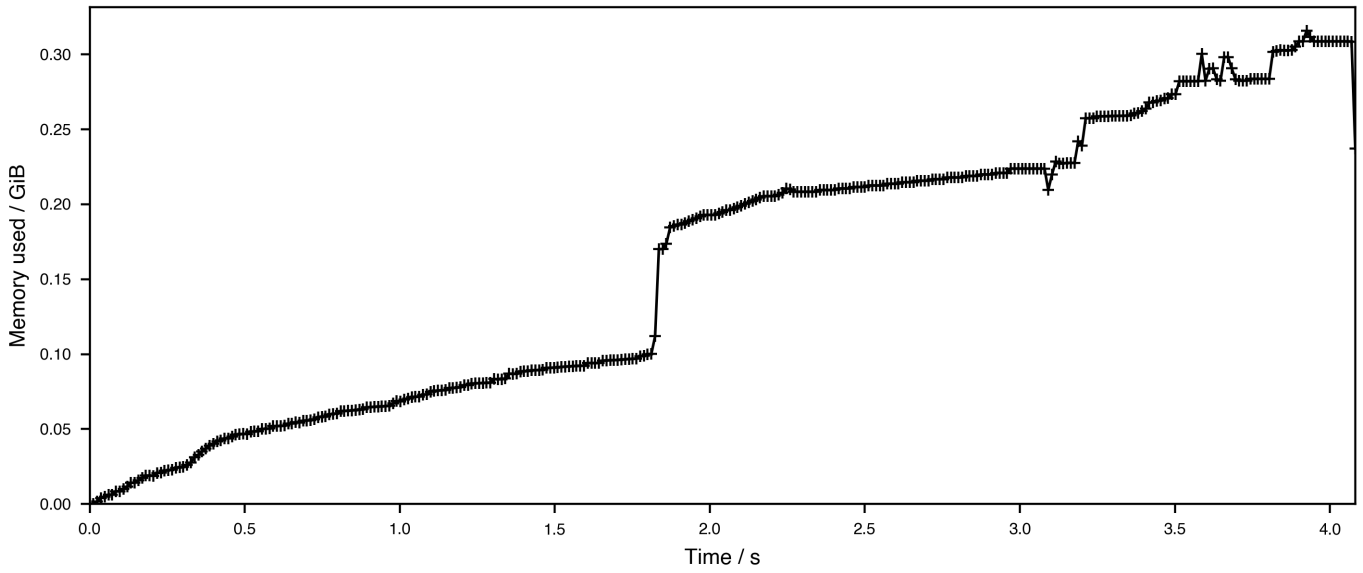


Figure S10 Overview of the memory usage through time for the Trigger stage.

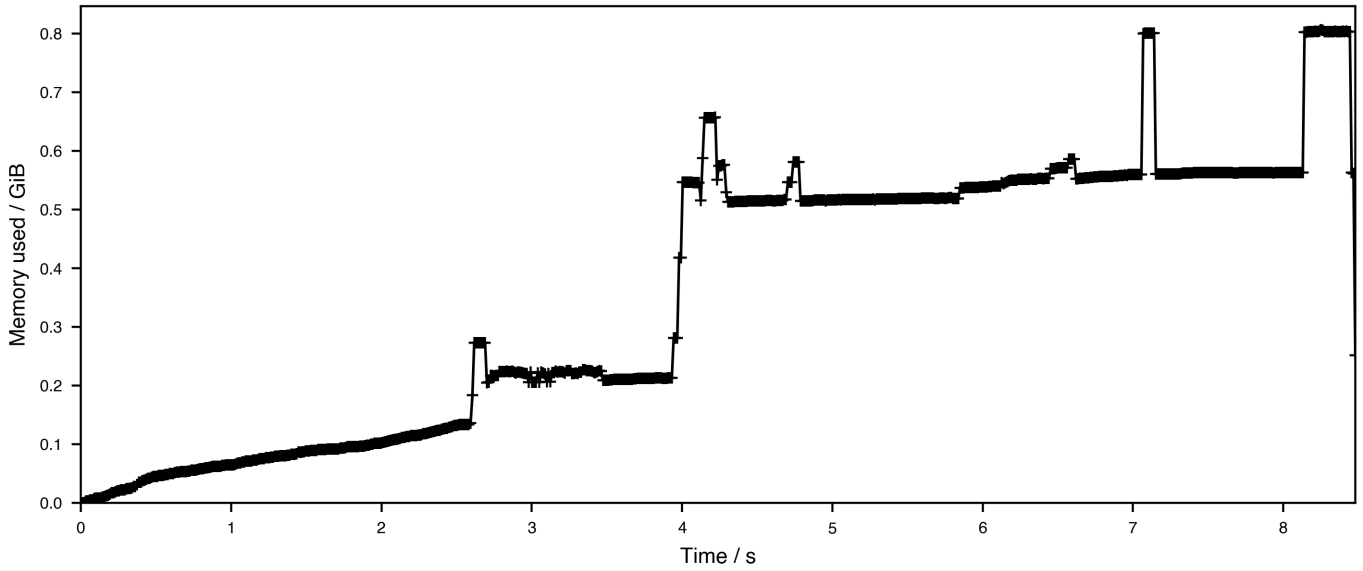


Figure S11 Overview of the memory usage through time for the Locate stage. The profile shows the location of a single event.

S4 Filtering strategy for the Icequake example

In order to make a quantitative assessment of the available filtering options, a subset of events from the QuakeMigrate catalogue (the first 10.5 minutes of icequakes on 2009-01-20; comprising 149 events) were inspected manually and labelled as either "real" events or "artefacts", or "ambiguous" if it was not definitive. This classification was made based on inspection of the event summary plots automatically generated by Locate; mainly through verifying that coherent P- and S-wave arrivals were visible across multiple stations in the waveform gather, and with the expected SNR variation with distance. 117 events were labelled "real"; 22 as "artefacts", and 10 were unclassified. Considering only the classified events, this indicates the initial catalogue consists of 84% true events, upon which any applied filtering should aim to improve.

Supplementary Figure S12 presents a Precision-Recall analysis of the performance of two available statistics; S12a shows the Global Covariance statistic (after taking the geometric mean across each axis, reported in the standard Locate output as `COV_Err_XYZ`), S12b shows the performance of the Coalescence value. For each case an AUC (area under the curve) statistic is calculated as

$$AUC = \frac{AUC_0 - PREC_i}{1 - PREC_i}, \tag{1}$$

where AUC_0 is the full trapezoidal area under the Precision-Recall curve, and $PREC_i$ is the precision of the input catalogue (the proportion of "real" events) corresponding to the expected score for a completely unskilled (random) classifier). The AUC score therefore indicates the filtering performance above random chance. An unskilled classifier would return an AUC score of 0, while a perfect performance would give a score of 1, representing 100% of true positives retained (100% precision), and equivalently no false negatives (100% recall), with no false positives.

The Global Covariance statistic returns a very high AUC score of 0.93, while the Coalescence filter returns a lower, though still good, value of 0.72. The Precision-Recall analysis also affords the opportunity to make a deliberate choice of filter value: in each case the Precision-Recall curve is annotated with sample filter values. In this case we wish to demonstrate location performance for the best located events, so choose a filter value with a very high precision at the cost of slightly lower recall; 0.15 km for the Global Covariance filter is estimated to give 99% precision at around 75% recall, based on this sample analysis.

The choice of these values can be further qualified by inspecting the statistics for the entire located catalogue (Supplementary Figure S13). Here it is clear that the chosen Global Covariance filter value falls relatively high up the knee of the histogram—corresponding to the choice of a conservative filtering strategy. For the Coalescence values (Supplementary Figure S13b) the chosen value of 5.5 falls just above the roll-off point where most events were removed by the Global Covariance filter, represented by the difference between the blue and green shaded bars.

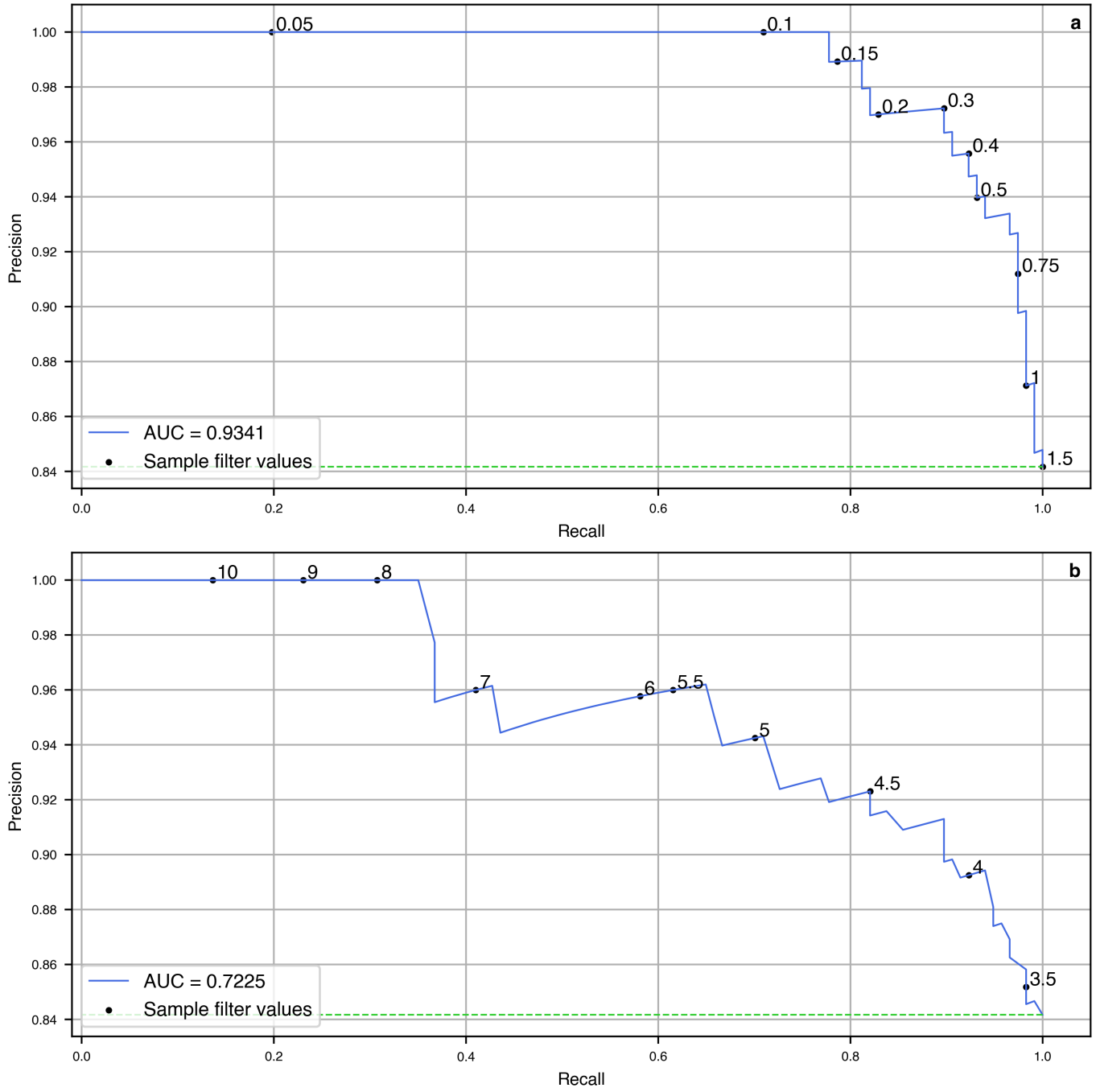


Figure S12 Precision Recall analysis of filtering performance of the **a** Global Covariance and **b** Coalescence statistics. Precision is defined as the ratio of true positives to the sum of true positives and false positives; Recall is the ratio of true positives to true positives and false negatives. Blue curve shows filtering performance across the full range of possible filter values in the sample dataset; green dashed line shows the precision of the starting catalogue, i.e., the proportion of positive samples ("real" events) in the set of labelled events, corresponding to the expected Precision-Recall curve for a completely unskilled (random) classifier. Annotated dots show Precision and Recall for a range of sample filter values.

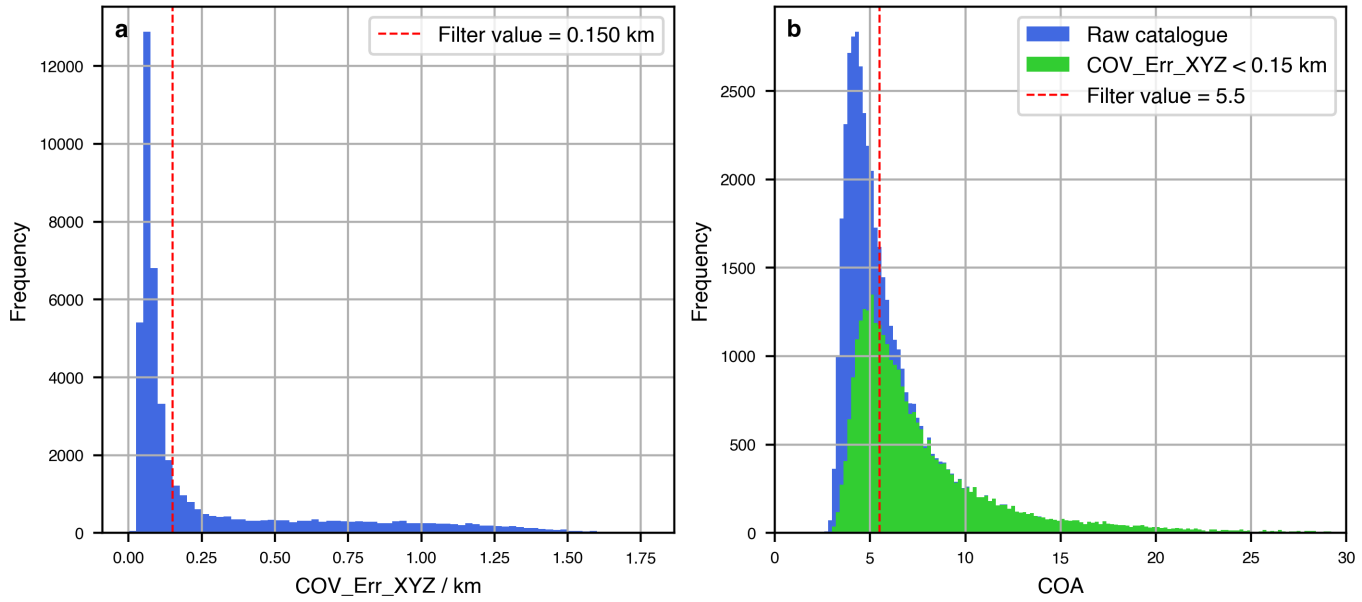


Figure S13 Histograms of **a** Global Covariance and **b** Coalescence values in the raw icequake catalogue. Red dashed lines show selected filter values. In **b**, blue bars show the histogram for the raw catalogue, while green show the histogram for the catalogue after removing events with the Global Covariance filter.

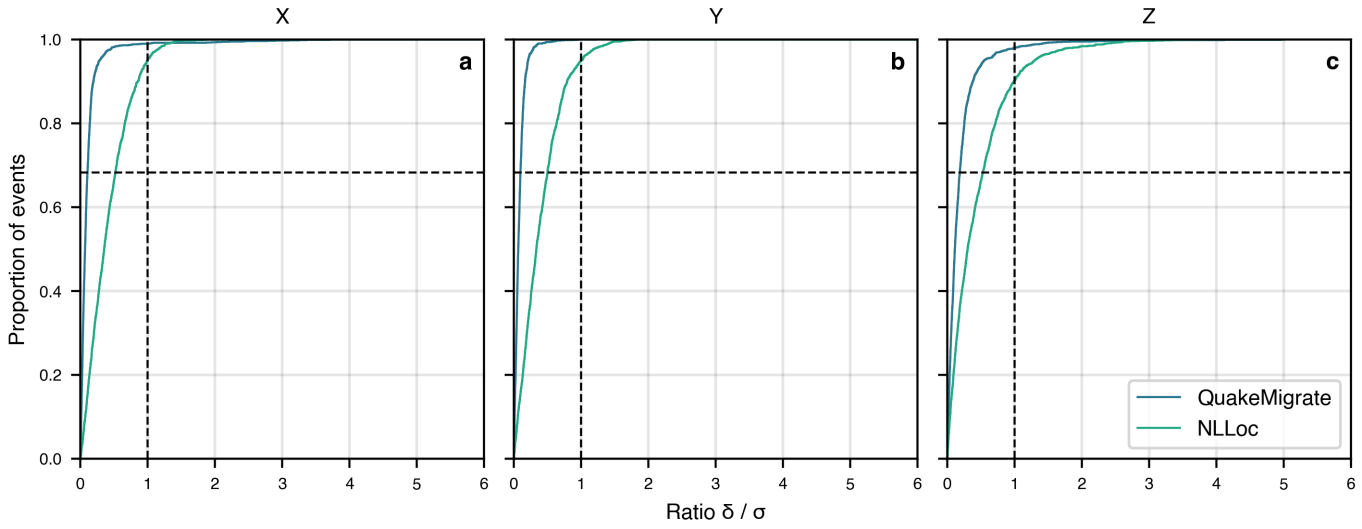


Figure S14 Comparison between location differences and quoted uncertainties for individual events in the Askja manual pick/QuakeMigrate location benchmark. Panels **a**-**c** show cumulative density functions for the ratio of the location difference (δ) to the quoted 1σ uncertainty in a given direction (X, Y or Z) for each individual event. Green line shows the ratio calculated against the location uncertainty estimate reported by NonLinLoc for the manual location; blue line is for the QuakeMigrate gaussian location uncertainty. Black cross-hairs highlight the 1σ percentile (68.3%); if lines stay to the upper-left of this point this indicates that location differences are less than the reported location uncertainty for the expected proportion of events. Note that x-axes are curtailed at a ratio of 6 for clarity; a very small minority of points exceed this ratio.