SCOTER

Multiple-Earthquake Location by Using
Static and Source-Specific **S**tation **CO**rrection **TER**ms

Nima Nooshiri
GFZ German Research Centre for Geosciences

Potsdam, Germany
February, 2019

Contents

Abstract

There has been growing recognition of the importance of the accurate seismic locations in quantitative seismological studies, such as seismic hazard analyses, fault zone characterization, and Earth's deformation. Accurate estimation of seismic locations is critical since a wrong estimate of the seismic source location will result in wrong interpretations in the subsequent analyses. We present SCOTER, an open-source Python program package that is designed to relocate multiple seismic events by using P- and S-wave *s*tation *co*rrection *ter*ms. The package implements static and shrinking-box source-specific station terms techniques extended to regional and teleseimic distances and adopted for probabilistic, non-linear, global-search location for large-scale multiple-event location. This program provides robust relocation results for seismic event sequences over a wide range of spatial and temporal scales by applying empirical corrections for the biasing effects of 3-D velocity structure. Written in the Python programming language, SCOTER is run as a stand-alone command-line tool (requiring no knowledge of Python) and also provides a set of sub-commands to develop inputs (dataset, configuration etc) and export results (hypocenter parameters, travel-time residuals etc) – routine but non-trivial tasks that can consume much user time. This package can be used for relocation in local, regional, and teleseimic scales. We describe SCOTER's functionality, design and technical implementation, accompanied by an overview of its use cases. As an illustration, we demonstrate the applicability of this tool through two examples based on (1) a catalogue of several hundred events in the Arctic plate boundary region using regional and teleseismic arrival times and (2) a small dataset of low-magnitude seismic events recorded by dense, local stations at the western Iberia, central Portugal. The relocated datasets highlight the future potential for applying the SCOTER relocation tool to greatly improve the relative location accuracy among nearby events.

1 Introduction

Earthquake hypocentre location using observations of seismic phase arrival times is a classic inverse problem in seismology that has a rich history of conceptual and methodological advancements dating back more than a century. Catalogues of seismic locations are one of the most important and widely-used forms of seismological data providing important information on a number of seismotectonic problems, from the imaging of fault-zone structure to the improvement of seismic-hazard assessment. The degree to which these problems can be resolved depends on the quality of the seismic catalogues, which themselves turn on the reliability of the seismic location methodology.

Improvement in absolute seismic location accuracy is always limited by imperfect knowledge of 3-D variations in the Earth structure (Thurber, 1992; Thurber and Eberhart-Phillips, 1999). However, numerous methods have been developed yielding significant improvements to relative location accuracy through the joint relocation of pairs or clusters of events (Douglas, 1967; Jordan and Sverdrup, 1981; Got et al., 1994; Richards-Dinger and Shearer, 2000; Waldhauser and Ellsworth, 2000; Lin et al., 2007; Myers et al., 2007; Grigoli et al., 2016; Trugman and Shearer, 2017). Although initial catalogues are routinely determined from the phase picks and associated travel times of each event in isolation, relative relocation methods are primarily based on travel-time corrections refined by station terms for a group of events along with differential travel times of pairs of events observed at common stations. This formulation helps mitigate common-mode errors introduced by the biasing effects of unmodelled velocity structure.

Here we present and describe a computer program package designed for relocating seismicity using static and source-specific station terms, which we term SCOTER program and which we are making available to the community as an integrated software package. The program can greatly improve the relative location accuracy among nearby events by applying empirical corrections for the biasing effects of unmodelled velocity heterogeneities. Details about the relocation algorithm adopted in SCOTER are contained in Nooshiri et al. (2017). This package can be used for relocations in local, regional, and global scales. It is an easy-to-use command-line tool that is configured with a plain text file in a human-readable structured data format. SCOTER is an integrated and time-saving program that provides the user with several subcommands for a number of routines, from observation phase file preparation, to travel-time binary grid files computation, to exporting and plotting the relocation results. A strong motivation for designing SCOTER has been to provide a simple and flexible environment that shields the user from the complexities of programming and, therefore, can be used not only by seismologists but also other research geoscientists. SCOTER is written in Python programming language and runs on Linux and Mac OS platforms. As a demonstration of the new software functionality, we use SCOTER to relocate two earthquake data sets contained in the package to show location improvements of the algorithm relative to the typical seismic network locations. The manuscript is a concise but near-complete tour that aims to turn the reader into a productive SCOTER user very quickly. The package is licensed by GNU General Public License, Version 3.0 (GPLv3) and freely available to all users at https://gitext.gfz-potsdam.de/nooshiri/scoter.

2 Overview of software functionality

In a style similar to Unix shells, SCOTER is invoked as a command-line program by calling `scoter` from the Unix/Linux shell. Its command-line interface uses standard Unix/Linux conventions for its options and arguments. The command-line tool also offers several subcommands. The first command below gives a list of available subcommands and a brief summary of them, and the second one provides more information about a subcommand and its different option flags:

```
$ scoter --help
$ scoter <subcommand> --help
```

Options are recognized by their leading double-dashes (e.g. `--format`). In the program documentation, *required* and *optional* arguments are denoted by `<···>` and `[···]` punctuation marks, respectively.

SCOTER is part of the Pyrocko ecosystem (Heimann et al., 2017), an open-source community project that develops Python tools for a variety of geophysical tasks hosted by the GFZ Helmholtz Centre Potsdam. SCOTER uses many of Pyrocko's utility functions, as well as Python's numpy, scipy and matplotlib libraries, and relies on NonLinLoc code (Lomax et al., 2000) to do the actual event location part of the calculation, combining them into a more specialized piece of software. SCOTER's functionality can be summarized as follows:

– Parsing earthquake bulletin files in a few different formats (currently supported formats are `QuakeML` and SeisComP3 `autoloc`) into NonLinLoc phase files.
– Creating 3-D binary grid files of seismic travel times.
– Parallelising the location calculation by distributing the input data across multiple processes to reduce the runtime of the execution.
– Performing separate location processes; single-event, static station term and source-specific station term (SSST) methods.
– Exporting results in convenient formats for statistical analysis and plotting.
– Diagnostic plots to visualise the performance of the algorithm.

There are several steps that should be taken when using SCOTER in a relocation project, from preparing required input files to exporting results. The basic work-flow can be summarised as follows:

– Setting up a project folder containing required input files.
– Setting up a configuration file for SCOTER.
– Running the program to relocate multiple events.
– Exporting the relocation results.
– Creating result plots for analysis and interpretations.

To use SCOTER with a specific data set, we suggest the folder structure shown in Figure 1. Details on required files and data formats, as well as data preparation, are given in the following sections by showing the applicability of the program through examples. Instructions on how to download and install SCOTER is given in Appendix A1.

3 Input files and data

The fundamental input data for the SCOTER algorithm are phase pick data. It also requires input event and station data files that uniquely identify each seismic event (e.g. with an event identifier and possibly reference hypocentre) and each station (e.g. with a station code

```
project_dir/
  ├─ config/ % contains configuration file
  │   └─ config_arctic.sf
  ├─ data/
  │   ├─ phase/ % contains observation phase files
  │   │   ├─ eid_001.nll
  │   │   ├─ eid_002.nll
  │   │   └─ ...
  │   └─ events_arctic.pf
  ├─ meta/ % contains station files
  │   ├─ stations-ims.sf
  │   ├─ stations-reg.sf
  │   └─ ...
  ├─ time/ % contains travel-time grid files
  │   ├─ fescan.P.DEFAULT.time.buf
  │   ├─ fescan.P.DEFAULT.time.hdr
  │   └─ ...
  └─ run_dir/ % created at runtime, contains relocation results
```

Figure 1: Example folder structure when using SCOTER with a specific data set.

and optionally a network code). Additionally, a pre-computed travel-time grid file needs to be provided for each seismic phase considered. Previously determined station terms (either static or source-specific) can be included as a starting point for the location process. Finally, the various inputs to the program, along with algorithm control parameters, are combined in an input configuration file read by the SCOTER upon initial computation.

3.1 Observation phase files

The multiple-event relocation algorithm applied in the SCOTER program is adopted for probabilistic, non-linear, global-search location method implemented in the software package NonLinLoc (Lomax et al., 2000). The SCOTER tool relies on this back-end (external program) to do the actual event location part of the calculation (see Nooshiri et al. (2017) for more details). Therefore, the arrival-time data should be provided in the NonLinLoc observation phase file format (`NLLOC_OBS`) for the SCOTER. For each seismic event in the bulletin data set, a separate phase pick file needs to be provided whose name must include the corresponding event name (i.e. resource identifier of the event or eventID). The program offers a simple unified interface to handle and process earthquake bulletin files through `dump-obs` subcommand. The current implementation of SCOTER has the ability of parsing bulletin files in `QuakeML` and SeisComP3 `autoloc` formats into `NLLOC_OBS` phase files.

3.2 Station data files

An example of the format of the station data file is given in Table 1. It consists of a single line per station providing, in order, the network code, the station code, the latitude in degree (between $-90°$ and $90°$), the longitude in degree (from $-180°$ to $180°$), and the elevation in metre. Any line starting with **#** is considered as a comment line and ignored. The first column, i.e. network code, is optional and a given station data file has either four or five columns, although it is advised to use both network and station codes as sometimes station code alone may be ambiguous. The program can also be provided with multiple station files.

Table 1: The station data file

```
# NetCode (optional), StaCode, Lat, Lon, Elev [m]
    GE    MLR      45.4917    25.9437   1360.0
    GE    SUMG     72.5763   -38.4539   3240.0
    IM    ARCES    69.5348    25.5057    403.0
    IM    SPITS    78.1777    16.3700    323.0
    NO    NORES    60.7353    11.5414    302.0
    NS    HOPEN    76.5084    25.0109    125.0
              ⋮                   ⋮
```

3.3 Travel-time grid files

For each seismic phase considered, a travel-time grid file needs to be provided. Each grid file is stored with a small, simple ascii header file and a binary data buffer file. Both files have identical names except for the extension. SCOTER does not do the actual computation of the seismic travel times. A separate travel-time modelling code is required to do this. However, SCOTER provides some external software supports in order to make it simple to build a database of pre-calculated travel-time grids. The program includes a front-end tool, `grist`, to create and inspect travel-time grid files:

```
$ grist --help                   # list of available subcommands
$ grist <subcommand> --help      # subcommands with built-in help
```

Two modelling codes, including command-line interfaces, required for `grist` to build and store time grids are: (1) a modified version of the *iaspei-tau* routine (Snoke, 2009), and (2) *LLNL-Earth3D* computer code (Matzel et al., 2014; Simmons et al., 2015). The former computes travel times through a 1-D, spherically-layered earth model and the latter allows for fast calculation of 3-D seismic travel times through global-scale 3-D tomography model LLNL-G3Dv3 (Simmons et al., 2012).

The distribution of *iaspei-tau* code included in the SCOTER package is an extended version of the original routine (Steven Gibbons and Johannes Schweitzer, personal communication, 2018) that calculates the family of IASPEI standard seismic phases, such as *Pg*/*Sg*, *Pb*/*Sb*, *Pn*/*Sn*, *PKPab*, *PKPbc*, *PKPdf* etc. (Storchak et al., 2003). Currently, there are two variations of 1-D layered velocity model files that can be used. The first format is `tvel` ascii file format used by the IASPEI *ttimes* package (Kennett et al., 1995). This format has two comment lines, followed by lines composed of depth, V_p, V_s and density, all separated by white-space (Table 2). The second format is `nd` format for "named discontinuities" used by the *TauP* program (Crotwell et al., 1999). The file consists of two types of lines, those that specify velocity at a depth, and those that specify the name of a discontinuity. The former provides, in order, depth, V_p, V_s and density, and the latter specifies one of the three labels, *mantle*, *outer-core* and *inner-core* placed on a line by themselves (Table 3). It should be noted that, for both formats, only depth, V_p and V_s are required and the remaining parameter is not needed for travel-time calculations.

The user can create their own custom 1-D model in one of the valid formats to the program described before and use model from saved file to compute the travel times. Alternatively, a model name that is associated with a standard model as part of the program can be specified. Several standard models are included within the distributed package, such as ak135 (Kennett et al., 1995), iasp91 (Kennett and Engdahl, 1991), prem (Dziewonski and Anderson, 1981).

Table 2: The `tvel` velocity model file

```
ak135 - P
ak135 - S
   00.0   5.80    3.46    2.72
   20.0   5.80    3.46    2.72
   20.0   6.50    3.85    2.92
   35.0   6.50    3.85    2.92
   35.0   8.04    4.48    3.3198
   77.5   8.045   4.49    3.3455
  120.0   8.050   4.50    3.3713
  165.0   8.175   4.509   3.3985
  210.0   8.300   4.518   3.4258
  210.0   8.300   4.523   3.4258
  260.0   8.482   4.609   3.4561
  310.0   8.665   4.696   3.4864
            ⋮       ⋮
```

Table 3: The `nd` velocity model file

```
    00.0    5.80    3.2      2.6
    15.0    5.80    3.2      2.6
    15.0    6.80    3.9      2.9
    24.4    6.80    3.9      2.9
mantle
    24.4    8.111   4.491    3.381
     ⋮              ⋮
  2891.0   13.72    7.265    5.566
outer-core
  2891.0    8.065   0.000    9.903
     ⋮              ⋮
  5150.0   10.36    0.000   12.170
inner-core
  5150.0   11.03    3.504   12.760
     ⋮              ⋮
```

Following command gives a complete list of the built-in 1-D layered velocity models:

```
$ grist list-models-1D
```

Finally, 1-D model travel-time grid file for a specific seismic phase can be easily built by invoking the following subcommand given the epicentral distance range, depth range, a file prefix, and desirably a custom velocity model file or a built-in model name:

```
# 1-D model travel-time calculation
$ grist raytrace-1D <--phase=phase> <--model=name|filename>
↪ <--distances=Xmin:Xmax:deltaX> <--depths=Zmin:Zmax:deltaZ>
↪ <--prefix=file_prefix> [--out-dir=dir_name]
```

Travel-time grid files built through 1-D, spherically-layered earth models are used for multiple stations, meaning that the calculated arrival time at a specific station is a function of only source depth and source-receiver surface distance. As mentioned before, the program provides the user with an interface for station-specific travel-time calculation through the 3-D tomography model LLNL-G3Dv3 (Simmons et al., 2012), which is a global-scale model of the crust and mantle P-wave velocity with regional-scale details. It should be noted that computation of travel times through the 3-D model can take 0.1 to 1.0 seconds per grid node depending on complications due to ray multipathing and, therefore, creating a 3-D model travel-time grid file for a specific phase at a single station would be computationally expensive. If the user decides they need for 3-D travel times for a P-wave type phase (e.g. *Pg*, *Pb*, *Pn* etc) at a specific station, the command to create such grid files would be:

```
# 3-D P-wave model travel-time calculation
$ grist raytrace-3D <--phase=phase> <--slons=west:east:step>
↪ <--slats=south:north:step> <--sdepths=bottom:top:step>
↪ <--rloc=lat,lon> <--prefix=file_prefix> <--station=sta_code>
↪ [--network=net_code] [--rdepth=elev_km] [--out-dir=dir_name]
```

Table 4: The station correction terms file

# NetCode (optional),	StaCode,	Phase,	StaTerm [s]
GE	SUMG	P1	-0.6178
GE	SUMG	S1	0.6345
IM	ARCES	P1	2.1970
IM	SPITS	P1	-0.8363
IM	SPITS	S1	-0.3579
NS	HOPEN	P1	-0.4289
NS	HOPEN	S1	-1.1252
	⋮		⋮

In the command above, `--slons`, `--slats` and `--sdepths` options specify longitude, latitude and depth (in kilometres) ranges of the source region, respectively. These values help the program define the size, origin and node spacing of the location search grid. For a specific P-wave type phase given via `--phase` option, seismic travel times are calculated from all the nodes in the 3-D grid (i.e. potential source points) to the receiver specified by the `--rloc` option. In this case, since the grid file is created for a specific station, the station code and optionally the network code (if it is available in the station file) are also needed in addition to the file prefix to help the program name the output files properly. It should also be mentioned that, the file prefix must remain identical to that of any other files created through 1-D velocity model if a mixture of both 1-D and 3-D model travel times are used in the location process. As an optional argument, the receiver elevation in kilometres can be given via `--rdepth` flag.

3.4 Previously determined station term files

Although we do not choose this option in the examples provided here, SCOTER has the ability to start the relocation process using pre-determined station terms (either static or source specific). This functionality can be critical for application in routine near real-time processing where relocating a single-event using previously determined source-specific station terms (SSSTs) for neighbouring events is desired. We can also consider another special case in which the user is interested in rerunning the program for only the SSST location step (e.g. to test a different set of algorithm control parameters) and can make use of this functionality to apply the previously determined static station terms as a starting point for the SSST calculation. In these cases, SCOTER is flexible in its accommodation of pre-existing station terms. Table 4 shows an example of the format of the station terms file. Like the station data file (Table 1), any line starting with **#** is considered as a comment line and ignored.

3.5 Configuration file

SCOTER is configured with a plain text file in YAML format. The YAML format has been chosen because it can represent arbitrarily nested data structures built from mappings, lists, and scalar values. It also provides an excellent balance between human and machine readability. When working with such files, it is good to know that the indentation is part of the syntax and that comments can be introduced with the **#** symbol.

By invoking the subcommand below, a project structure is deployed into a directory and a template configuration file is generated that can be set up and used to relocate multiple events using SCOTER:

```
# Initialise new project structure
$ scoter init <project_dir>
```

Table 5 presents the overall structure of a SCOTER configuration file. It has a top-level container (mapping) that is introduced with the line `--- !scoter.Config` and has several child elements, e.g. `path_prefix`, `dataset_config` etc. Some of these entries may again contain their own child elements (indented blocks of lines) or lists (lines introduced with dashes).

The detailed description of the configuration file, parameters used and some tips to set those parameters are provided in Appendix A2. In summary, main configuration sections in a SCOTER configuration file are:

- `DatasetConfig`: the core of input and data configuration. Here, the user defines where all input data such as observation phase files, input event and station data files, travel-time grid files are stored.
- `StationTermsConfig`: handles the parameters valid for static and source-specific station terms calculation, such as number of iterations, search radius for and minimum and maximum number of nearby events, misfit norm. It also consists of parameters to weight and control the location quality of neighbouring events during station terms calculation.
- `NetworkConfig`: provides the flexibility of using a subset of observed arrival times recorded within a specific surface distance range.
- `NLLocConfig`: the core of event location part of the calculation that provides a unified interface to configure and run the NonLinLoc program.

Table 5: Example SCOTER configuration file (YAML format)

```yaml
%YAML 1.1
--- !scoter.Config

# All dataset file paths (input data) referenced below are
# treated relative to the location of this configuration file.
path_prefix: ".."

# Path, where to store output files (run directories)
rundir: "./out"

# --------------------------------------------------------
# Configuration section for dataset (input data)
# --------------------------------------------------------
dataset_config: !scoter.DatasetConfig
  bulletins_template_path: "data/phase/${event_name}.nll"
  stations_paths:
  - "meta/stations.sf"
    ⋮
# --------------------------------------------------------
# Configuration section for station terms
# --------------------------------------------------------
station_terms_config: !scoter.StationTermsConfig
  # Subsection on static station terms (STATIC)
  static_config: !scoter.StaticConfig
    niter: 0
    ⋮
  # Subsection on source-specific station terms (SSST)
  ssst_config: !scoter.SourceSpecificConfig
    niter: 20
    ⋮
  # Subsection on location quality control
  locqual_config: !scoter.LocationQualityConfig
    standard_error_max: 2.0
    ⋮
# --------------------------------------------------------
# Configuration section for station network selection
# --------------------------------------------------------
network_config: !scoter.NetworkConfig
  station_selection: false
  ⋮
# --------------------------------------------------------
# Configuration section for location parameters
# --------------------------------------------------------
nlloc_config: !scoter.NLLocConfig
  trans: !scoter.NLLocTrans
    # Subsection on geographic transformation
    trans_type: "GLOBAL"
    ⋮
```

4 Guided tour of use cases

In this section, we describe the computational details underlying the current implementation of SCOTER. Two examples including the Arctic plate boundary region and the western Iberia, Portugal are provided as a demonstration of the package. As we intend this code to be open-source, future releases of the code may include minor modifications and more functionalities to the algorithm presented here. All the location results shown in this section are obtained by setting the configuration parameters to case-specific values. Although our method requires only the phase picks, the station locations, and a set of pre-computed travel-time grid files, careful tuning of some of the parameters is required for different applications to achieve optimal results.

4.1 Application to the seismicity along the Arctic plate boundary

4.1.1 Data set

Our first data set is a bulletin of 614 seismic events in the Arctic boundary region. The earthquakes occurred from 1998 to 2015 between 70°N−84°N and 9°W−10°E. The magnitude of the events ranges from M2.0 to M6.0. The bulletin has been generated by using high-quality P- and S-wave arrival times. The primary selection criterion was that the events had clear regional P and S onsets at several of the permanent regional stations on both sides of the spreading ridge which are most sensitive to low-magnitude events. A minimum requirement was six phases. The phase arrivals for events with larger magnitudes were augmented by teleseismic P arrivals from the Reviewed Event Bulletin (REB) of the International Data Centre (IDC) with signal-to-noise ratio exceeding 3.0. Supplementary teleseismic phases have been recorded by the distant seismic arrays and three-component stations of the International Monitoring System (IMS). The teleseismic azimuthal coverage provided by the IMS for this source region in the Northern Hemisphere is superb and this increases the likelihood of high epicenter accuracy based on seismic network criteria (Bondár et al., 2004). Fig. 2 shows the seismicity in the region of interest together with the distribution of the permanent regional network and IMS stations at far-regional and teleseismic distances. Following is a step-by-step guide explaining how to obtain improved event locations for the seismicity in the Arctic plate boundary region using SCOTER.

4.1.2 Data preprocessing

The Arctic earthquake data set is composed of 614 events for which the pick information is included and stored in a single file in `QuakeML` format (Schorlemmer et al., 2011). Following command is issued to load and parse the bulletin file and will write the observation phase files in the `NLLOC_OBS` file format for each seismic event contained in the bulletin file:

```
# Parse QuakeML and dump into NLLOC_OBS
$ scoter dump-obs ./data/Bulletin_Arctic.xml --format=QuakeML
↪ --prefix=eid_ --output-dir=./data/phase/
↪ --output-events=./data/events_arctic.pf
```

The output phase files are automatically named according to the event names (i.e. resource identifiers of the events). Optionally, a prefix and a suffix can be provided for the above subcommand to customise the names of the output phase files by specifying `--prefix` and `--suffix` options. In this example, the name of each output observation file begins

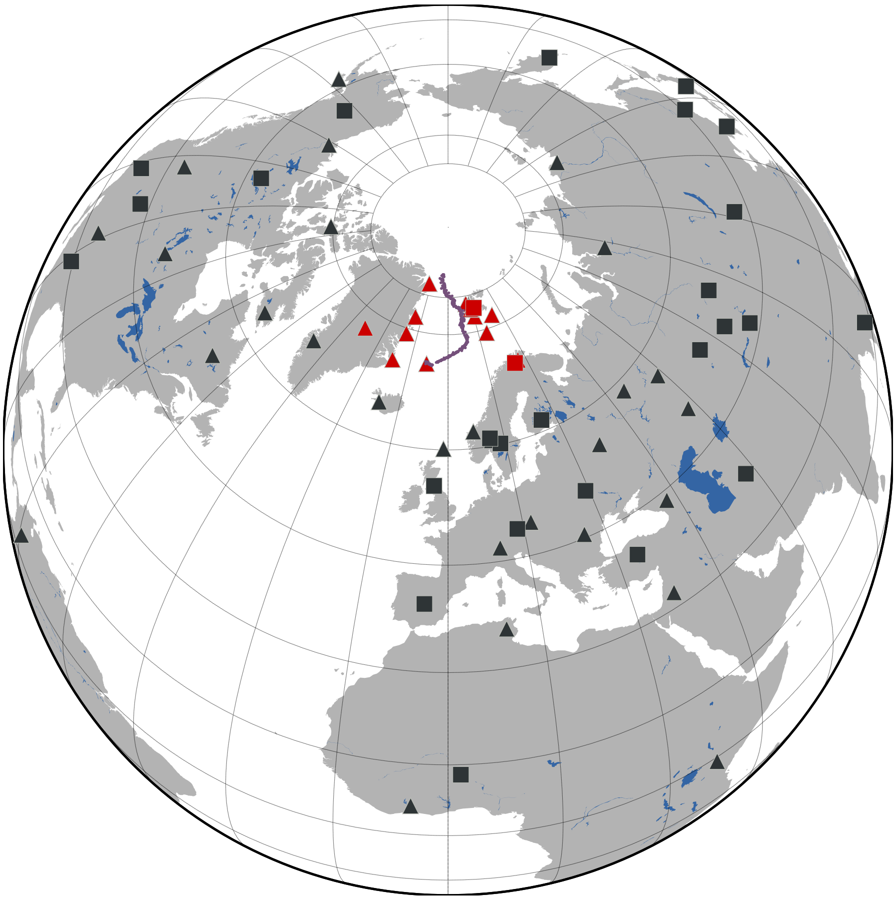

Figure 2: Global map showing ridge seismicity in the region of interest from 70°N–84°N together with the regional network displayed in red and the International Monitoring System (IMS) stations at far-regional and teleseismic distances shown in black. Triangle and square symbols indicate permanent three-component stations and seismic arrays, respectively.

with `eid_` prefix followed by the event identifier (unique name). The other file output by running the command above, `events_arctic.pf`, is in `Pyrocko` event file format (Heimann et al., 2017) and contains information and possibly reference solutions for all events available in the data set. During the relocation procedure, the program can access to the available observation files by looking into possible expansions of a template string containing `${event_name}` placeholder, which is substituted with the event names defined in the events information file mentioned before. It should be noted that SCOTER also lets the user to relocate events with a subset of arrival times recorded by stations at a specific epicentral distance range. To take up this option, the user should make sure that their initial bulletin data set provides a reference hypocentre solution for each event, since the station data selection (filtering) in SCOTER is based on the surface distances between the reference (initial) epicentre of the events and seismic stations. In this example, such information is included in the initial bulletin data and will be automatically saved into the events file.

Our data set for the Arctic plate boundary region consists of the first regional P phases labelled as *P1*, the first regional S phases labelled as *S1*, and teleseismic P arrivals. For the velocity model, here we create a custom 1-D model Fennoscandian (FESCAN, Mykkeltveit and Ringdal, 1981) in `tvel` format and use model from saved file to compute the seismic travel times. Travel-time grid files for these phases are easily built by invoking the following commands given the epicentral distance range, depth range, a file prefix, and our custom velocity model file:

```
# First regional P and S travel times
$ grist raytrace-1D --phase=P1,S1 --model=model/fescan.tvel
↪ --distances=0:14.95:0.05 --depths=0:35:2.5
↪ --prefix=fescan --out-dir=./time/

# Teleseismic P travel times
$ grist raytrace-1D --phase=P --model=model/fescan.tvel
↪ --distances=13:124.75:0.25 --depths=0:35:2.5
↪ --prefix=fescan --out-dir=./time/
```

It should also be noted that the file prefix must be the same when travel-time grid files are created for different phases. Surface distance and depth values are defined by giving the corresponding minimum, maximum and step-size values in the format of `min:max:step`. In the example command above, the teleseismic P-wave travel times are calculated from sources at a depth range of $0-35$ km with a depth interval of 2.5 km, to the receivers within the epicentral distance range of $13° - 124.75°$ with a distance spacing of $0.25°$.

4.1.3 Configuration set-up

To initiate a relocation project for the Arctic earthquake data set, we can run the command below that will create a project directory called `arctic` in the current directory:

```
# Initialise the Arctic project directory
$ scoter init arctic
```

The project folder already contains a configuration file for multiple-event location with SCOTER. Before going further, we need to put the observation phase files and travel-time grid files prepared in Section 4.1.2, as well as the station files, in our project directory following the folder structure suggested in Figure 1.

The complete configuration file is quite extensive and, for the sake of brevity, the reader is referred to the generated configuration file itself to get familiar with variables needed to be set while it has been tried to provide it as a self-explanatory file. Moreover, Appendix A2 explains in some detail the many configuration options available in SCOTER. In general, the settings for different configuration sections (see Section 3.5) are quite straightforward and the user can easily follow the documentation in the provided configuration file to modify it for their own future purpose.

To refine the event locations in the Arctic bulletin data set, we will first relocate all seismicity with the 1-D velocity model FESCAN (referred to as single-event locations) and then, in an iterative manner, we estimate the direct P- and S-wave SSSTs for each source-receiver ray paths to reduce the biasing effects of the true 3-D earth structure and relocate the events by applying the obtained correction terms. Although we could obtain our initial locations using the static station term method and then use these station terms as a starting point for SSST calculation, we perform a generalisation of this approach that is to continuously reduce the cut-off distance for searching nearby events between first and final iterations. In other words, we start the search radius with a large value to include a large group of events from which we compute the SSSTs, then decrease it to some specified minimum distance to calculate station terms using only the closest events. To relocate events in the Arctic plate boundary region, we perform 20 iterations of SSST computation and gradually reduce the distance cut-off during the iterations from 500 to 50 km. We also use all available neighbouring events found within a search radius at each iterations.

4.1.4 Checking data and configuration

Before beginning the actual relocation process, we can now execute the following subcommand to run some sanity checks:

```
# Check data and configuration
$ scoter check config/config_arctic.sf
```

In particular, SCOTER exercises a subset of application functions needed to determine whether it can access and read the input data and to assure that the configuration parameters are set carefully and the relocation analysis works roughly as expected. It is basically a quick, broad and shallow testing that SCOTER offers to avoid wasting the user's time and effort. The sanity-check information is reported to the standard output.

4.1.5 Starting the relocation process

When the configuration file is ready, we can run the relocation procedure by invoking the subcommand `go`:

```
# Run scoter multiple-event location
$ scoter go config/config_arctic.sf --steps=A,C --parallel=8
```

The location process can be run in parallel using multiple CPUs at the same time, which makes the runtime faster. The number of events to process in parallel can be set by specifying `--parallel` option. If -1 is given, all CPUs in the system are used. As mentioned before, SCOTER can be used for single-event relocation without applying any station terms (Single), static station terms relocation (Static), and source-specific station terms relocation (SSST). For simplicity, these steps are named as A, B, C, respectively. Therefore, any desired relocation step(s) can be re-run, for instance, to play around with different configuration parameters. In this example, the relocation procedure starts with single-event location and then, in an iterative manner, estimates the P- and S-wave SSSTs for each source-receiver ray paths to reduce the biasing effect of the true 3-D earth structure and relocates the events by applying the obtained correction terms to P and S phases.

4.1.6 Exporting the results

To export the final results (e.g. hypocentre parameters, travel-time residuals), first we need to execute the subcommand `harvest` to cache the results:

```
# Run harvesting to cache scoter results
$ scoter harvest config/config_arctic.sf --parallel=8
```

For iterative steps (i.e. Static and SSST steps), this stores the results for all available iterations. If the user decides that they only need the outputs for the last iteration, they can pass the flag `--last-iteration` (or `--last-iter` as an alias) to the subcommand `harvest`. Specifying the `--weed` flag makes it possible to weed out only those events having at least one phase arrival time corrected by a station term value in each iteration.

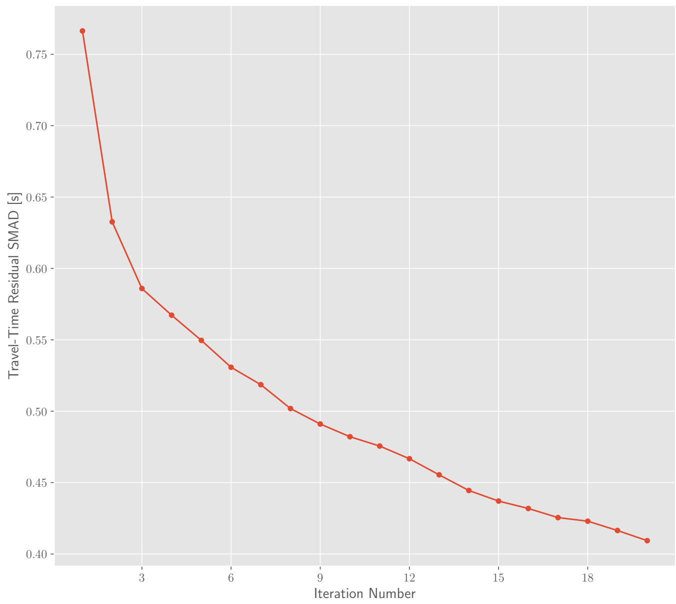

Figure 3: Reduction of travel-time residual SMAD for the 614 events in the Arctic plate boundary region with iteration numbers of SSST calculations.

The convergence of the algorithm can be evaluated in terms of the median absolute deviation (MAD) or scaled median absolute deviation (SMAD, a robust measure of the spread of a distribution which is equal to the standard deviation for Gaussian distributions) of the travel-time residuals from one iteration to the next by running the following command:

```
# Plot convergence curve of the SSST calculation
$ scoter plot-convergence config/config_arctic.sf --step=C
↪ --statistic=SMAD --save --format=pdf
```

which creates Figure 3 and saves it as a pdf file. The figure shows the reduction of the travel-time residual SMAD from the 614 events with iteration number in our SSST calculation. The SMAD of the residuals drops from 0.135 to 0.1 s.

Moreover, SCOTER offers several subcommands to export some values of interest that can save much user time, such as applied station correction terms, initial and final hypocenter parameters, travel-time residuals before and after applying the station correction terms. The output files are in convenient formats for statistical analysis and plotting and can be easily loaded and used in some other libraries and tools such as numpy and GMT. For example, we can issue the following command to export an ascii file in column format that provides the first P phase (labelled as *P1* in the catalogue) station term corrections for a particular regional station SPITS. The station terms for each event recorded at the receiver in question are reported at each event location in the output file:

```
# Export source-specific station terms
$ scoter export-ssst config/config_arctic.sf --phase=P1
↪ --network=IM --station=SPITS --output=export/SSST_SPITS_P1.dat
```

Figure 4 shows the final calculated first regional P- and S-wave SSST values relative to the 1-D velocity model FESCAN for the regional array SPITS. The station terms for each event recorded at the receiver are plotted at each event location. Spatial coherence of the pattern of the SSSTs indicates the existence of heterogeneities in the real velocity structure. Positive

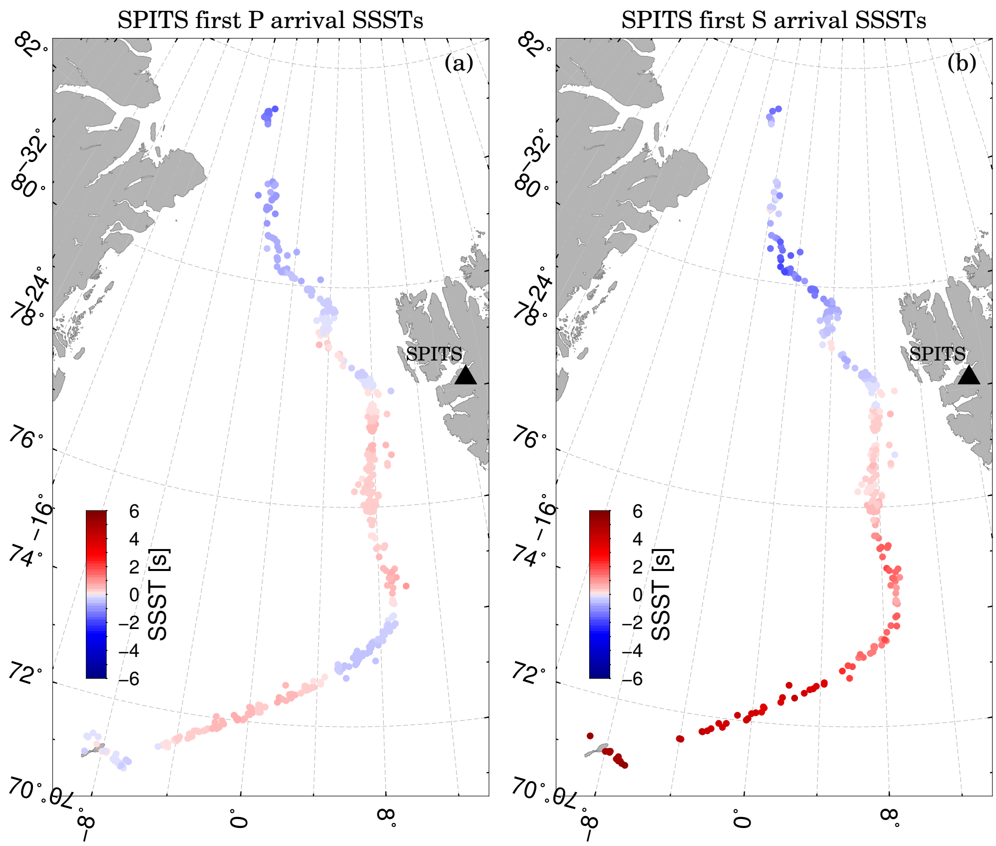

Figure 4: Calculated source-specific station terms (SSST) for regional seismic array SPITS for (a) first P-wave arrivals and (b) first S-wave arrivals. The SSST values are color coded and plotted at the event locations.

correction terms indicate low velocities and negative values reflect high velocity anomalies on the way of the seismic rays travelling from sources to the receiver.

Hypocentre parameter files for each location step can be exported easily by invoking the following commands:

```
# Export hypocentre parameter files
# Single-event location (step: A)
$ scoter export-events config/config_arctic.sf --step=A
↪ --format=columns --output=export/events_Single_loc.dat

# SSST location (step: C)
$ scoter export-events config/config_arctic.sf --step=C
↪ --format=columns --output=export/events_SSST_loc.dat
```

The output files can be exported in either column or `Pyrocko` event file format. Figure 5a-b shows in map view a comparison of SSST locations and single-event locations. Here, the spatial scattering and alignment of seismicity into planar features can be considered as a qualitative measure of the improvement in the SSST relocations. This is visible in the decreased scatter and improved clustering in the new hypocentre locations, which better define the Arctic plate boundary. Figure 5c reveals that the SCOTER algorithm provides relative location improvements comparable to the BayesLoc program by Myers et al. (2007) shown by previous studies (Gibbons et al., 2017). The BayesLoc algorithm does provide

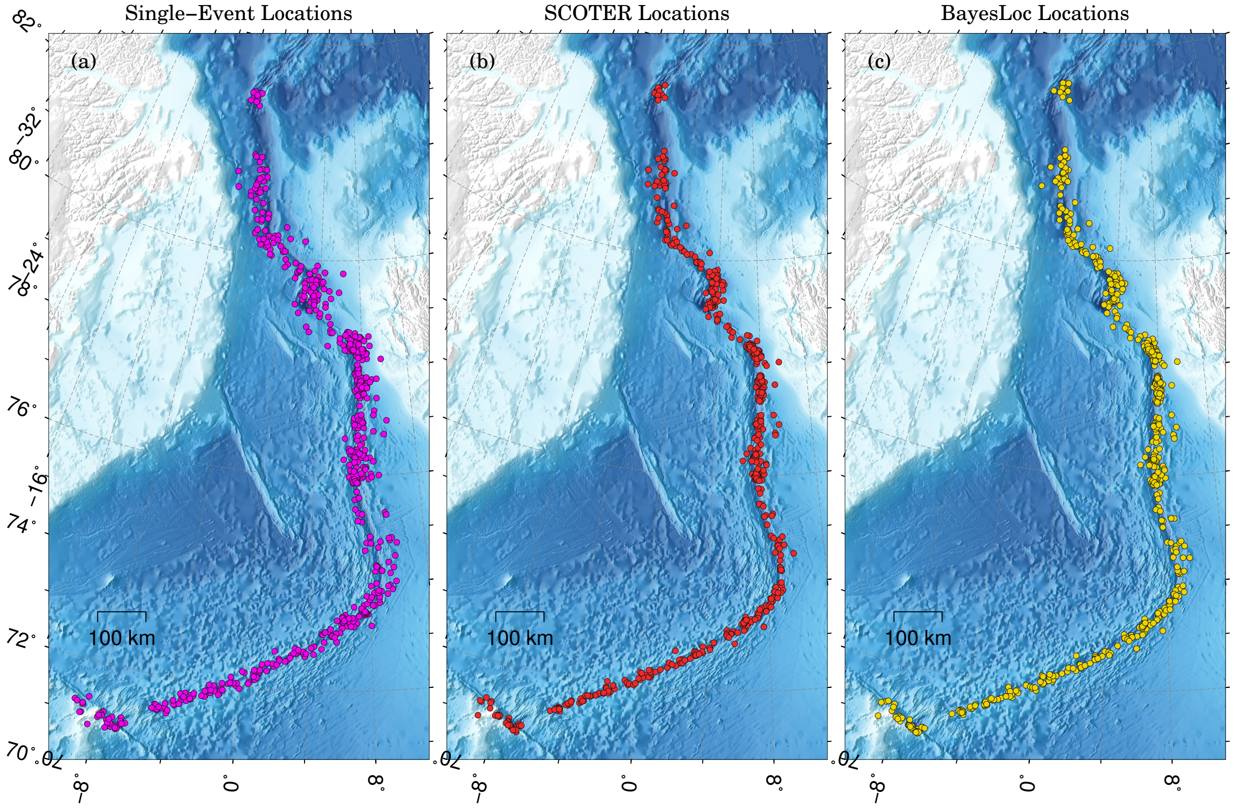

Figure 5: Relocated seismicity in the Arctic Ridge using (a) a single-event, non-linear algorithm and (b) source-specific station correction terms (SSSTs) implemented in the SCOTER program, and (c) BayesLoc locations presented in Gibbons et al. (2017).

an extensive list of algorithm control parameters relating BayesLoc probability model and MCMC sampler and the optimal parameter choices required to obtain the best locations may not be always obvious to the user. Although SCOTER has its own set of parameter choices, we designed them to be as few and as straightforward as possible.

The travel-time residuals for a particular seismic phase can be simply obtained for each recording station in the dataset and stored as plain-text files by issuing the subcommand `export-residuals`, which extracts the residuals at a given seismic station before and after application of computed station correction terms:

```
# Export travel-time residuals
$ scoter export-residuals --phase=P1 --network=IM --station=SPITS
↪ --output=export/residuals_SPITS.dat
```

The output file is in column format and can be comfortably used in different plotting tools to make and compare histograms of residual distributions. Figure 6 compares histograms of first P and first S travel-time residuals at the regional network SPITS from single-event and SSST locations and depicts significant improvements to residual reduction achieved by using station correction terms. Applying SSST values leads to the SMAD reduction of 68 per cent and 58 per cent for P and S arrival times, respectively, at the regional network SPITS.

The program also provides the user with the subcommand `plot-residuals` to make figures that summarize residual analysis as a function of event-station distance. As an example, figures similar to the sub-plots on the left column in Figure 7 can be created by invoking the following command:

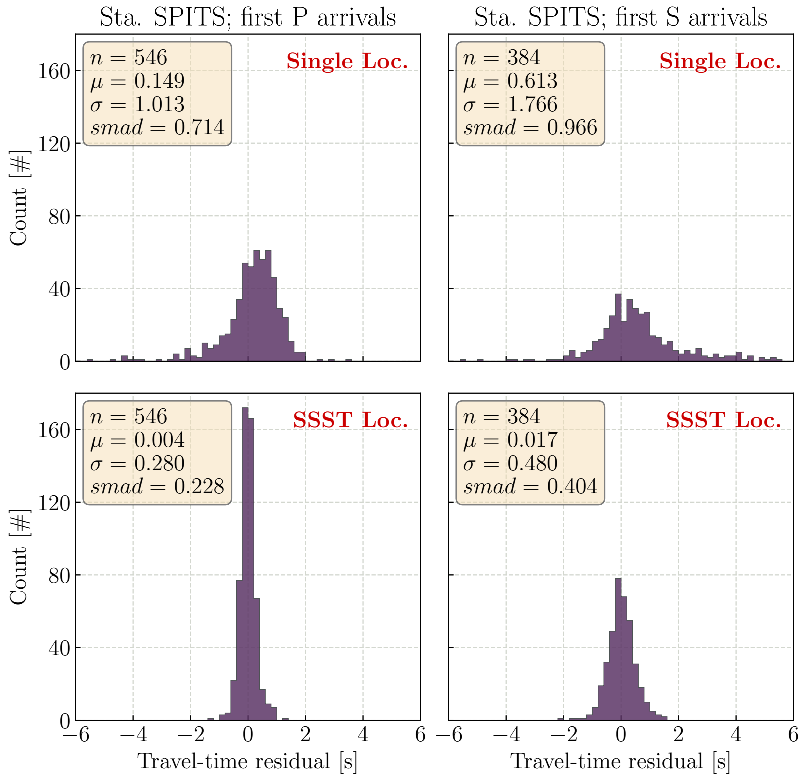

Figure 6: Histograms of the travel-time residuals from (top) single-event and (bottom) the SSST relocations for the station shown in Fig. 4. The scaled median absolute deviation (smad), mean, and sample standard deviation of each distribution are indicated on each panel.

```
# Plot travel-time residual heat maps (2-D histograms)
$ scoter plot-residuals --phase=P1 --steps=A,C --distances=0:15:1
↪ --residuals=-10:10:0.2 --colormap=gist_heat
```

It should be mentioned that the frequency of occurrence (heat plots) of the residuals are normalized in the given distance range so that the residual bin with the largest number of hits (the scaling cell) is assigned a value of 1. In the command above, distance and residual ranges are specified by giving the corresponding minimum, maximum and bin-width values in the format of `min:max:bin-width`. Figure 7 reveals that strong variations in initial P- and S-wave residuals are smoothed out in the final residuals from events and peaks in the near-regional and regional distances where the effect of lithospheric velocity anomalies is strong.

4.2 Application to the seismicity in western Iberia, central Portugal

As the second demonstration of the program, we use SCOTER to relocate low-magnitude, local-scale earthquake clusters in western Iberia, Portugal. The seismic data were recorded by a local broadband array (DOCTAR array) deployed in the Western Ossa Morena Zone (WOMZ) between June 2011 and September 2012 (Matos et al., 2018). The data set was also complemented with data recorded by nearby permanent and temporary stations.

In this example, we relocate the earthquakes using a 3-D tomographic velocity model. Specifically, we use the 3-D Preliminary Reference Iberia Seismic Model (PRISM3D) (Arroucau et al., 2017), which is a three-dimensional P- and S-wave velocity model for Iberia

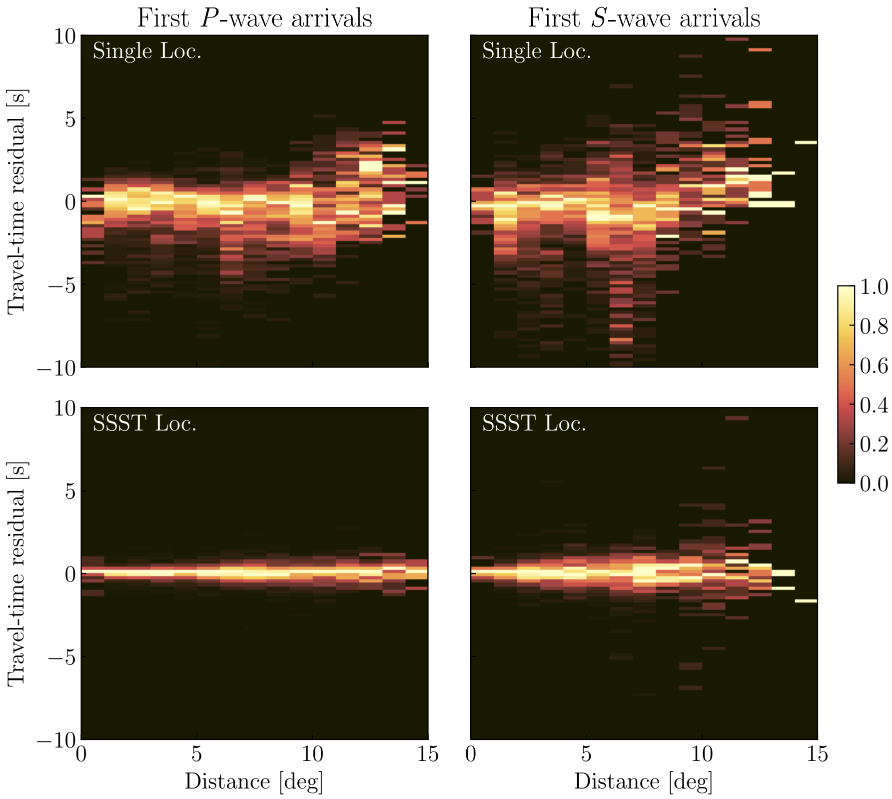

Figure 7: Frequency of occurrence (heat plots) of travel-time residuals for (top) single-event and (bottom) the SSST locations. The heat plots are normalized in each distance range so that the residual bin with the largest number of hits (the scaling cell) is assigned a value of 1. All other bins in that distance range are linearly scaled from zero to one by dividing the number of hits by the number of hits in the scaling cell.

and north Africa based on several previously published geophysical models (e.g. tomographic models, receiver functions and active source studies). Although 3-D velocity information provides enhanced absolute hypocentre locations, the tomography model is relatively smooth and cannot account for small-scale velocity structure that can also introduce bias and scatter in event locations. Thus, to further refine the event locations, we combine the shrinking-box SSST method and 3-D model location. Our strategy is to first relocate all events using the PRISM3D model. Then, to constrain the absolute locations, we calculate a set of well-constrained static station terms using a group of 3-D relocated events with at least four recording stations, an azimuthal gap $< 180°$ and at least one station within 20 km of the event. These *a priori* static station terms are then included in the SSST location process in which the SSST values for P- and S-wave picks are computed from the travel-time residuals of nearby events and subtracted from the arrival-time picks, and the 3-D relocation process is repeated with the adjusted arrival-time data. Here, we perform five iterations of 3-D location and SSST computation. The distance cut-off for SSST calculation is reduced gradually during the iterations from 50 km to 10 km. The relocation results are shown in Figure 8. The notable differences between two sets of locations are that the newly located hypocentres show improved clustering and sharpen seismogenic source zones in the region. For the new catalogue relocated using SSSTs, average shifts relative to single-event locations are ∼4.5 km in epicenters and ∼3.2 km in depth. Overall, the location uncertainties of SSST locations are about $30 - 40$ per cent those of the single-event locations (Figure 9). Furthermore, Figure 9 reveals that the new locations appear to be more clustered and some substructure can be seen.

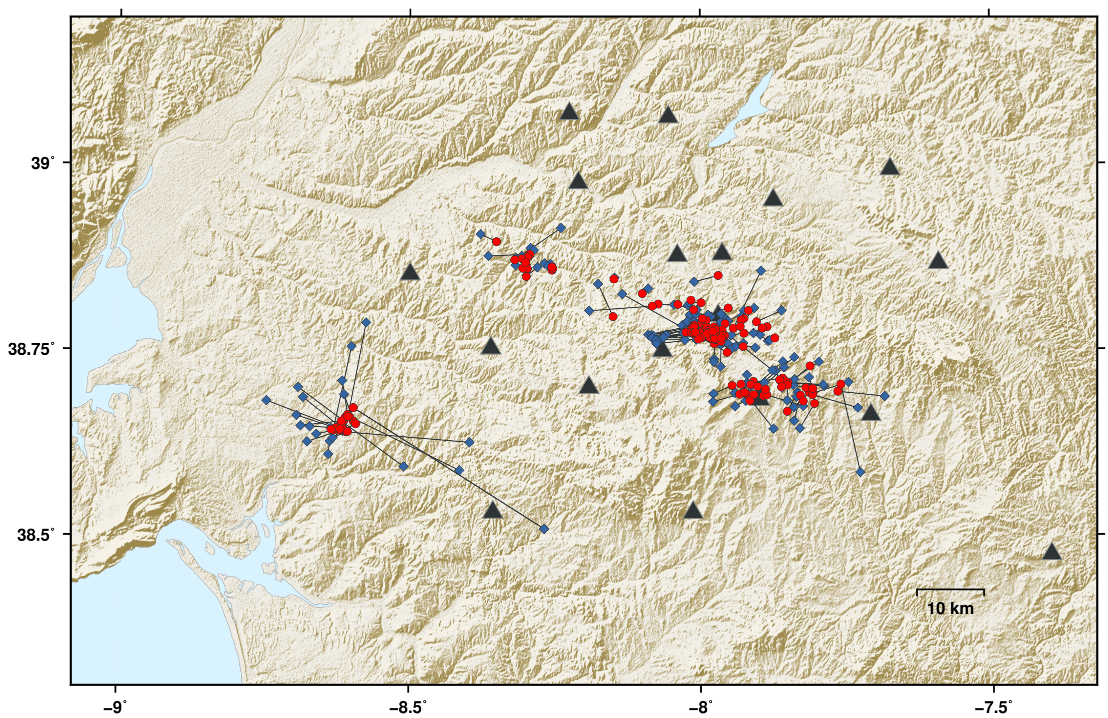

Figure 8: Map view of the seismicity in the Iberia region, Portugal. Non-linear, single-event locations and SSST locations are shown by blue diamonds and red circles, respectively. Solid lines indicate mislocations between corresponding hypocentres. Triangles represent seismic stations.

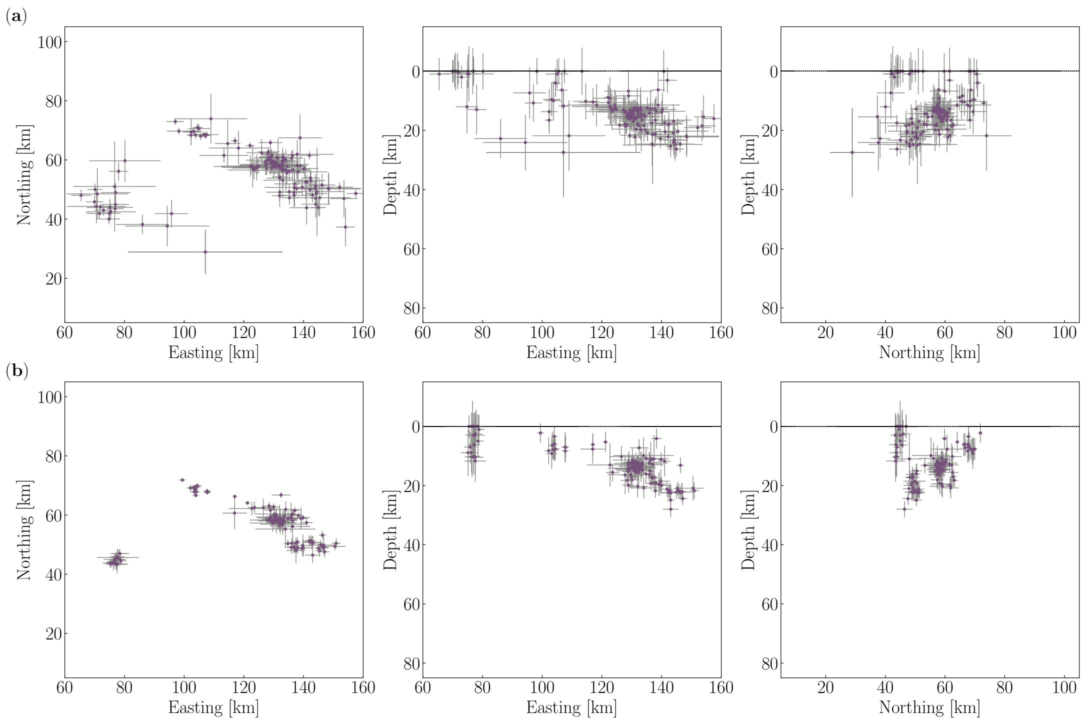

Figure 9: (a) Horizontal view and vertical cross sections of the non-linear, single-event locations and the associated 68% uncertainty bounds. (b) The same as Figure 9a but for SSST locations. The reference point $(X, Y) = (0, 0)$ corresponds with the point $(\lambda, \phi) = (-9.50, 38.25)$ in the geographic coordinates system.

5 Summary

SCOTER is an open-source earthquake relocation tool that applies deterministic static and source-specific station terms to improve the relative location accuracy among nearby events. The algorithm is adopted for probabilistic, non-linear, global-search location and enables more robust relocation results. This package can be used for multiple-event relocations in local, regional, and global scales. It provides the user with a number of powerful tools that are useful and handy for preparing the input data and post-processing and visualising the relocation results. Functionality, design and technical implementation of the new tool is in the pursuit of providing a simple and flexible environment for the user. SCOTER's main functionalities and performance were demonstrated through examples. The encouraging results for these examples showed the scientific potential for large-scale relocation efforts using the SCOTER tool.

Release notes

The SCOTER relocation codes described in this article comprise an open-source software package under the GNU General Public License, Version 3.0 (GPLv3). The latest version of the SCOTER source distribution, which includes source codes, a user guide, an example data set, and a change history, is publicly available for download via the project repository at https://gitext.gfz-potsdam.de/nooshiri/scoter.

Bibliography

Arroucau, P., Custódio, S., Civiero, C., Dias, N. A., and Silveira, G. (2017). PRISM3D: a preliminary 3D reference seismic model of the crust and upper mantle beneath Iberia. In *EGU General Assembly Conference Abstracts*, volume 19 of *EGU General Assembly Conference Abstracts*, page 16801.

Bondár, I., Myers, S. C., Engdahl, E. R., and Bergman, E. A. (2004). Epicentre accuracy based on seismic network criteria. *Geophysical Journal International*, 156(3):483–496. doi:10.1111/j.1365-246X.2004.02070.x.

Crotwell, H. P., Owens, T. J., and Ritsema, J. (1999). The TauP Toolkit: Flexible seismic travel-time and ray-path utilities. *Seismological Research Letters*, 70(2):154. doi:10.1785/gssrl.70.2.154.

Douglas, A. (1967). Complex morphology of subducted lithosphere in the mantle beneath the Tonga trench. *Nature*, 215:47–48. doi:10.1038/374154a0.

Dziewonski, A. M. and Anderson, D. L. (1981). Preliminary reference Earth model. *Physics of the Earth and Planetary Interiors*, 25(4):297 – 356. doi:10.1016/0031-9201(81)90046-7.

Gibbons, S. J., Harris, D. B., Dahl-Jensen, T., Kværna, T., Larsen, T. B., Paulsen, B., and Voss, P. H. (2017). Locating seismicity on the Arctic plate boundary using multiple-event techniques and empirical signal processing. *Geophysical Journal International*, 211(3):1613–1627. doi:10.1093/gji/ggx398.

Got, J.-L., Fréchet, J., and Klein, F. W. (1994). Deep fault plane geometry inferred from multiplet relative relocation beneath the south flank of Kilauea. *Journal of Geophysical Research*, 99(B8):15375–15386. doi:10.1029/94JB00577.

Grigoli, F., Cesca, S., Krieger, L., Kriegerowski, M., Gammaldi, S., Horalek, J., Priolo, E., and Dahm, T. (2016). Automated microseismic event location using Master-Event Waveform Stacking. *Scientific Reports*, 6:25744. doi:10.1038/srep25744.

Heimann, S., Kriegerowski, M., Isken, M., Cesca, S., Daout, S., Grigoli, F., Juretzek, C., Megies, T., Nooshiri, N., Steinberg, A., Sudhaus, H., Vasyura-Bathke, H., Willey, T., and Dahm, T. (2017). Pyrocko - An open-source seismology toolbox and library. doi:10.5880/GFZ.2.1.2017.001.

Jordan, T. H. and Sverdrup, K. A. (1981). Teleseismic location techniques and their application to earthquake clusters in the south-central pacific. *Bulletin of the Seismological Society of America*, 71(4):1105–1130.

Kennett, B. L. N. and Engdahl, E. R. (1991). Traveltimes for global earthquake location and phase identification. *Geophysical Journal International*, 105(2):429–465. doi:10.1111/j.1365-246X.1991.tb06724.x.

Kennett, B. L. N., Engdahl, E. R., and Buland, R. (1995). Constraints on seismic velocities in the earth from traveltimes. *Geophysical Journal International*, 122(1):108–124. doi:10.1111/j.1365-246X.1995.tb03540.x.

Lin, G., Shearer, P. M., and Hauksson, E. (2007). Applying a three-dimensional velocity model, waveform cross correlation, and cluster analysis to locate southern California seismicity from 1981 to 2005. *Journal of Geophysical Research: Solid Earth*, 112(B12). doi:10.1029/2007JB004986.

Lomax, A. (2005). A reanalysis of the hypocentral location and related observations for the great 1906 California earthquake. *Bulletin of the Seismological Society of America*, 95(3):861–877. doi:10.1785/0120040141.

Lomax, A. (2019). The NonLinLoc software guide. http://alomax.free.fr/nlloc/index.html. last accessed January 2019.

Lomax, A., Virieux, J., Volant, P., and Berge-Thierry, C. (2000). Probabilistic earthquake location in 3-D and layered models. In Thurber, C. H. and Rabinowitz, N., editors, *Advances in Seismic Event Location*, volume 18 of *Modern Approaches in Geophysics*, pages 101–134. Springer Netherlands. doi:10.1007/978-94-015-9536-0_5.

Matos, C., Custódio, S., Batlló, J., Zahradník, J., Arroucau, P., Silveira, G., and Heimann, S. (2018). An active seismic zone in intraplate west Iberia inferred from high-resolution geophysical data. *Journal of Geophysical Research: Solid Earth*, 123(4):2885–2907. doi:10.1002/2017JB015114.

Matzel, E., Simmons, N. A., and Myers, S. (2014). *LLNL-Earth3D User Manual*. Lawrence Livermore National Laboratory. https://www-gs.llnl.gov/content/assets/docs/LLNL-Earth3D_User_Manual_032814.pdf.

Myers, S. C., Johannesson, G., and Hanley, W. (2007). A Bayesian hierarchical method for multiple-event seismic location. *Geophysical Journal International*, 171(3):1049–1063. doi:10.1111/j.1365-246X.2007.03555.x.

Mykkeltveit, S. and Ringdal, F. (1981). Phase identification and event location at regional distance using small-aperture array data. In Husebye, E. S. and Mykkeltveit, S., editors, *Identification of Seismic Sources – Earthquake or Underground Explosion*, pages 467–481, Dordrecht. Springer Netherlands. doi:10.1007/978-94-009-8531-5_23.

Nooshiri, N., Saul, J., Heimann, S., Tilmann, F., and Dahm, T. (2017). Revision of earthquake hypocentre locations in global bulletin data sets using source-specific station terms. *Geophysical Journal International*, 208(2):589–602. doi:10.1093/gji/ggw405.

Richards-Dinger, K. B. and Shearer, P. M. (2000). Earthquake locations in southern California obtained using source-specific station terms. *Journal of Geophysical Research*, 105(B5):10939–10960. doi:10.1029/2000JB900014.

Schorlemmer, D., Euchner, F., Kästli, P., and Saul, J. (2011). QuakeML: status of the XML-based seismological data exchange format. *Annals of Geophysics*, 54(1):59–65. doi:10.4401/ag-4874.

Simmons, N. A., Myers, S. C., Johannesson, G., and Matzel, E. (2012). LLNL-G3Dv3: Global P wave tomography model for improved regional and teleseismic travel time prediction. *Journal of Geophysical Research: Solid Earth*, 117(B10). doi:10.1029/2012JB009525.

Simmons, N. A., Myers, S. C., Johannesson, G., Matzel, E., and Grand, S. P. (2015). Evidence for long-lived subduction of an ancient tectonic plate beneath the southern Indian Ocean. *Geophysical Research Letters*, 42(21):9270–9278. doi:10.1002/2015GL066237.

Snoke, J. A. (2009). Traveltime tables for iasp91 and ak135. *Seismological Research Letters*, 80(2):260. doi:10.1785/gssrl.80.2.260.

Storchak, D. A., Schweitzer, J., and Bormann, P. (2003). The IASPEI standard seismic phase list. *Seismological Research Letters*, 74(6):761. doi:10.1785/gssrl.74.6.761.

Tarantola, A. and Valette, B. (1982). Inverse problems = quest for information. *Journal of Geophysics*, 50:159–170.

Thurber, C. H. (1992). Hypocenter-velocity structure coupling in local earthquake tomography. *Physics of the Earth and Planetary Interiors*, 75(1):55 – 62. doi:10.1016/0031-9201(92)90117-E.

Thurber, C. H. and Eberhart-Phillips, D. (1999). Local earthquake tomography with flexible gridding. *Computers and Geosciences*, 25(7):809 – 818.

Trugman, D. T. and Shearer, P. M. (2017). GrowClust: A hierarchical clustering algorithm for relative earthquake relocation, with application to the Spanish Springs and Sheldon, Nevada, earthquake sequences. *Seismological Research Letters*, 88(2A):379. doi:10.1785/0220160188.

Waldhauser, F. and Ellsworth, W. L. (2000). A double-difference earthquake location algorithm: Method and application to the northern Hayward fault, California. *Bulletin of the Seismological Society of America*, 90(6):1353–1368. doi:10.1785/0120000006.

A1 Installation and updating

The latest version of SCOTER is freely available to all users at https://gitext.gfz-potsdam.de/nooshiri/scoter. After installing `git`, the SCOTER's source code can be downloaded and installed using:

```
$ cd ~/src/    # or wherever you keep your source packages
$ git clone https://gitext.gfz-potsdam.de/nooshiri/scoter.git
$ cd scoter
$ sudo python setup.py install
```

When SCOTER is installed, its dependencies are automatically installed except for NonLinLoc program. The original software is developed and maintained by Anthony Lomax (Lomax, 2019). The latest version of NonLinLoc modified and packaged as SCOTER backend can be downloaded and installed by:

```
$ cd ~/src/    # or wherever you keep your source packages
$ git clone https://gitext.gfz-potsdam.de/nooshiri/scoter_NLL.git
$ cd scoter_NLL/src
$ sudo make -R all
```

For updating an existing manual installation of SCOTER, the source code needs to be updated using:

```
$ cd scoter    # change to the directory where SCOTER was cloned
$ git pull origin master
```

and then reinstalled as described above.

A2 The SCOTER configuration in more detail

A2.1 The `DatasetConfig` section

`path_prefix`: Defines a prefix which is prepended to all paths in the configuration.

`${event_name}`: Placeholder that is substituted with the event names defined in `events_path` file.

`events_path`: File with the hypocenter information and possibly reference solutions.

`bulletins_template_path`: Template path for the files with phase pick observations.

`stations_paths`: List of the files with station information.

`traveltimes_path`: Directory of the travel-time grid files.

`traveltimes_prefix`: File prefix for the travel-time grid files.

`takeoffangles_template_path`: Template path for the files with pre-computed takeoff angles.

`starting_delays_path`: File(s) with the previously determined station correction terms.

A2.2 The `StationTermsConfig` section

`StaticConfig` subsection:

`niter`: Number of iterations for static station terms calculation.

`phase_list`: List of direct P- and S-wave type phases for which static terms are computed.

`nresiduals_min`: Minimum number of residuals required to compute static term for each station.

`SourceSpecificConfig` subsection:

`niter`: Number of iterations for source-specific station terms (SSST) calculation.

`phase_list`: List of direct P- and S-wave type phases for which SSST values are calculated.

`start_cutoff_dist`, `end_cutoff_dist`: Starting and ending distance cut-off (R_{max}) for the SSST calculation. For intermediate iterations, R_{max} is set to values spaced evenly in logarithmic scale *Unit: m*

`start_nlinks_max`, `end_nlinks_max`: Starting and ending maximum number of nearby events (N) for the SSST calculation. For intermediate iterations, N is set to values spaced evenly in logarithmic scale.

`nlinks_min`: Minimum number of nearby events (i.e. residuals) required to compute SSST values for each source-receiver path.

`ndelays_min`: Minimum number of adjusted picks (i.e. station terms) for an event required to consider it as a neighbouring event in SSST iterations.

`WeightConfig` subsection:

`distance_weighting`: How to weight nearby events located around a target event. This option is valid only for SSST computation. Available choices are `"uniform", "distance", "effective_distance"`. *Default: "uniform"*

`apply_outlier_rejection`: Whether to apply residual outliers rejection. Used to detect large travel-time residuals that should not be included in station terms calculation. *Default: true*

`outlier_rejection_type`: Cut-off threshold type for residual outliers. Available choices are `"static", "dynamic"`. *Default: "dynamic"*

`outlier_rejection_level`: Cut-off threshold level for residual outliers. For "static" cut-off, it is the absolute threshold in seconds. For "dynamic" cut-off, it is a factor to multiply the scaled median absolute deviation (SMAD) of the residuals. *Default:* 6

`LocationQualityConfig` subsection:

Events with low-quality locations (i.e. do not satisfy one or more of the following conditions) are not used in station terms calculation.

`standard_error_max`: Maximum location RMS. *Unit: s*

`secondary_azigap_max`: Maximum secondary azimuthal gap *Unit: deg*

`largest_uncertainty_max`: Maximum semi-major axis of the confidence ellipsoid (corresponding to the largest location uncertainty). *Unit: m*

A2.3 The `NetworkConfig` section

`station_selection`: Whether to select (filter) station data based on epicentral (surface) distances. The distances are calculated based on the initial locations of the events defined in `events_path` file. *Default: false*

`station_dist_min`, `station_dist_max`: Minimum and maximum source-receiver surface distance. *Unit: deg*; *Default:* 0, 180

A2.4 The `NLLocConfig` section

`NLLocTrans` subsection:

Geographic transformation parameters.

`trans_type`: Sets geographic to working coordinates transformation parameters. Available choices are `"GLOBAL"` (spherical mode, referred to as *Global*) and `"SIMPLE", "NONE", "SDC", "LAMBERT"` (rectangular mode, referred to as *Non-Global*).

The following three additional parameters need to be included and set if the `trans_type` is one of `"SIMPLE", "SDC", "LAMBERT"` options:

`lat_orig`, `lon_orig`: Latitude (between $-90°$ and $90°$) and longitude (from $-180°$ to $180°$) of the rectangular coordinates origin. *Unit: deg*

`rot_angle`: Rotation angle of geographic north measured clockwise relative to the rectangular coordinates system Y-axis. *Unit: deg*

The following three additional parameters need to be included and set only if the `trans_type` is set to `"LAMBERT"`:

`ref_ellips`: Reference ellipsoid name. Available choices are "WGS-84", "GRS-80", "WGS-72", "Australian", "Krasovsky", "International", "Hayford-1909", "Clarke-1880", "Clarke-1866", "Airy", "Bessel", "Hayford-1830", "Sphere".

`first_paral`, `second_paral`: First and second standard parallels (between $-90°$ and $90°$). *Unit: deg*

`NLLocGrid` subsection:

Describes location search grid.

`x_num`, `y_num`, `z_num`:
Non-Global: Number of grid nodes in the x/y/z directions.
Global: Number of grid nodes in the long./lat./depth directions.

`x_orig`, `y_orig`:
Non-Global: x/y location of the grid origin relative to the geographic origin. *Unit: km*
Global: long./lat. of the south-west corner of the grid. *Unit: deg*

`z_orig`:
Non-Global: z location of the grid origin relative to the geographic origin (positive z-axis points down). *Unit: km*
Global: depth of the south-west corner of the grid (positive z-axis points down). *Unit: km*

`dx`, `dy`:
Non-Global: Grid node spacing along the x-/y-axis. *Unit: km*
Global: Grid node spacing along long./lat. directions. *Unit: deg*

dz: Grid node spacing along the z-axis/depth direction. *Unit: km*

NLLocSearchOcttree subsection:

Specifies the oct-tree global search parameters.

init_num_cells_x, init_num_cells_y, init_num_cells_z: Initial number of oct-tree cells in the x/y/z or long./lat./depth directions.

min_node_size: Smallest oct-tree node side length to process. The oct-tree search is terminated after a node with a side smaller than this length is evaluated. *Unit: km*

max_num_nodes: Total number of nodes to process.

use_sta_density: Available choices are 0, 1. If 1, weights oct-tree cell probability values used for subdivide decision in proportion to number of stations in oct-tree cell. Gives higher search priority to cells containing stations, stabilises convergence to local events when global search used with dense cluster of local stations. *Default:* 0

stop_on_what: Available choices are 0, 1. If 1, the oct-tree search is terminated when first min_node_size reached. If 0, the oct-tree search continues until max_num_nodes is evaluated, but cell subdivision is only done to min_node_size. *Default:* 1

NLLocMeth subsection:

Specifies the location algorithm (i.e. how to construct the posterior PDF) and corresponding parameters.

method: Likelihood function. Available choices are:

"GAU_ANALYTIC": Least-squares L2 norm (Tarantola and Valette, 1982).
"EDT": Equal Differential Time (EDT) likelihood function (Lomax, 2005).
"EDT_OT_WT": Weights the sum of the EDT probabilities by the variance of origin-time estimates over all pairs of readings. Downweights locations with inconsistent origin-time estimates.

min_dist_sta_grid, max_dist_sta_grid: Minimum and maximum distance between a station and *the center* of the location search grid. Stations that are not in this distance range will not be used for event location. Use -1 for no minimum and very large value for no maximum (e.g. 1.0e6). *Unit: km*

min_num_phases, max_num_phases: Minimum and maximum number of phase picks that must be accepted for event location. Use -1 for no maximum.

min_num_Sphases: Minimum number of S phases that must be accepted for event location. Use -1 for no maximum number.

vp_vs_ratio: V_p/V_s ration. If > 0.0, only P-wave travel-time grids are read and vp_vs_ratio is used to calculate S-wave travel-times. If < 0.0, S-wave travel-time grids are directly used (if available). Cannot be used in Global mode.

reject_duplicate_arrivals: Whether to leave out duplicate arrivals (observations having the same station and phase labels). If false, then duplicate arrival times will be used if $\Delta t < 0.5 \times$ sigma_time.

NLLocGau subsection:

Specifies parameters for theoretical, Gaussian model errors described by the covariance matrix $\mathbf{C_T}$ (Tarantola and Valette, 1982):

$$[\mathbf{C_T}]_{ij} = \sigma_{\mathbf{T}}^2 . \exp\left\{-\frac{1}{2}\frac{D_{ij}^2}{\Delta^2}\right\}, \tag{A1}$$

where D_{ij} is the distance between the station i and the station j, $\sigma_{\mathbf{T}}$ is some theoretical error for travel time to one station due to model errors, and Δ is the correlation length of errors (the wavelength or the length of lateral heterogeneities of the medium).

sigma_time: $\sigma_{\mathbf{T}}$ in Equation A1. *Unit: s*

corr_len: Δ in Equation A1. *Unit: km*

NLLocGau2 subsection:

Specifies parameters for travel-time dependent model errors. Sets the travel-time error in proportion to the travel-time, thus giving effectively a station-distance weighting, which was not included in the Equation A1 by Tarantola and Valette (1982). This gives improvements in hypocenter clustering.

sigma_tfraction: Fraction of travel time to use as error (between 0 and 1). Travel-time error is calculated as sigma_tfraction $\times T^{calc.}$ where $T^{calc.}$ is the model-predicted travel time.

sigma_tmin, sigma_tmax: Minimum and maximum acceptable values for travel-time error.

NLLocPhaseid subsection:

Specifies the mapping of a set of phase codes (e.g. *PN, PG*) to a standardized phase code (e.g. *P*). Can be present multiple times in the configuration file as members of phaseid_list entry.

std_phase: Standardized phase code.

phase_code_list: List of phase codes that may be present in a observation files and should be mapped to (i.e. considered as) the std_phase in the location process.

NLLocElevcorr subsection:

Simple elevation correction using the travel time of a vertical ray from elevation zero to the elevation of the station.

apply_elevcor: Whether to apply elevation correction. *Default: false*

vel_p: P-wave velocity used to calculate the elevation correction for P-wave type phases. *Unit: km/s*; *Default:* 5.80

vel_s: S-wave velocity used to calculate the elevation correction for S-wave type phases. *Unit: km/s*; *Default:* 3.46

NLLocStawt subsection:

Station distribution weighting. Helps to correct for irregular station distribution, i.e. a high density of stations in regions such as Europe and North America and few or no stations in regions such as oceans. The relative weight for station i is:

$$W_i = \frac{1}{\sum_j \exp\left\{-\frac{d_{ij}^2}{c^2}\right\}^2}, \tag{A2}$$

where d_{ij} is the surface distance between the station i and the station j, and c is the cut-off distance.

`apply_stawt`: Whether to apply station distribution weighting. *Default: false*

`cutoff_dist`: c in Equation A2. If < 0.0, it is automatically set to the mean distance between all pairs of stations. *Unit: deg*; *Default:* -1.0