User manual

for STADIUM-py

Version 1.0

By Utpal Kumar and Cédric P Legendre

Last update: May 4th, 2020

Contents

Chapter 1

Information

Receiver Function and SKS automatic measurement - Seismological Tools Automated Download, processing & Imaging Using Mostly Python (STADIUM - Py)

- By Cédric P Legendre (@*cplegendre*) and Utpal Kumar (@*earthinversion*) on github
- Based on RF analyses by: Tom Eulenfeld (@*trichter*); `https://github.com/trichter/rf`
- Based on SKS analyses by: Jack Walpole (@*JackWalpole*); `https://github.com/JackWalpole/splitwavepy`
- Based on ObsPy: `https://github.com/obspy/`

If you use STADIUM - Py, please cite the following:

Kumar et al., (2020)
STADIUM-Py: Automated receiver functions and shear-wave splitting measurements beneath seismic stations in the USArray and Germany
Journal name
(submitted?)

Chapter 2

Installation

2.1 Requirements

In a first step, we shall create a separate environment (we arbitrary chose "rfsksenv"). Mote: it requires preinstalled anaconda) To install anaconda, please check: `https://docs.anaconda.com/anaconda/install/` To create this environment, a yml file has been prepared to ease the process, and to ensure all the required package have the proper version, as evolution of the packages may affect the available functions.

Then type the following command:

For OSX:

```
conda env create -f environment\_osx\_10\_14\_6.yml
```

For Linux:

```
conda env create -f environment\_UbuntuXIX.yml
```

Then:

```
conda activate rfsksenv
```

Known issues

If having some issues with Cartopy, one simple fix may be:

```
pip uninstall shapely; pip install --no-binary :all: shapely
```

With some UBUNTU distributions, it was impossible to call pip from conda. Therefore, some of the necessary dependencies were not installed and needed to be installed manually:

- cartopy

```
conda install -c conda-forge cartopy
```

- h5py

```
conda install h5py
```

- obspyh5

```
pip install obspyh5 or
conda install obspyh5
```

- rf

```
pip install rf
```

- splitwavepy

```
pip install splitwavepy
```

Then run:

```
python stadium.py
```

Chapter 3

User's input

Most of the code is automated. However, several parameters shall to be set by the user. A total of four files can be modified safely by the user:

- input_file.yaml
- stepwise.yaml
- advRFparam.yaml
- advSKSparam.yaml

Below, you will find a list of the parameters, with possible values and description for each file.

3.1 input_file.yaml

input_file.yaml

```yaml
# PARAMETERS: VALUES
project_name: Simple_RF_test/

fresh_start: 1

makeRF: 1
makeSKS: 0

##  Define the geographical region
mnlong: 6 #minimum longitude
mxlong: 15 #maximum longitude
mnlat: 47 #minimum latitude
mxlat: 56 #maximum latitude
```

Figure 3.1: Basic input files.

This file has 4 parameters that can be modified to run STADIUM-Py:

- project_name: define the name of the folder where all files will be stored.
- fresh_start: (0/1), toggle to start fresh, previous files in the project_name directory will be removed.
- makeRF: (0/1), toggle receiver functions part of STADIUM-Py
- makeSKS: (0/1), toggle SKS measurements part of STADIUM-Py

In addition, 4 parameters are used to select the region of interest:

- mnlong: minimum longitude of the user-selected region,
- mxlong: maximum longitude of the user-selected region,
- mnlat: minimum latitude of the user-selected region,
- mxlat: maximum latitude of the user-selected region.

3.2 Settings/stepwise.yaml

Settings > ! stepwise.yaml

```yaml
data_settings:
  client: IRIS #enter all the clients separated by commas
  network: all
  station: GRFO
  locations: {"","00"} #List all locations, e.g "","00"

plot_settings:
  plot_stations: 1 #plot the stations map of all available stations and all retrieved stations
  plot_events: 1 #plot the stations map of all retrieved events for each station
  plot_all_retrieved_events_stations: 1

rf_stepwise:
  obtain_inventory_RF: 1
  download_data_RF: 1
  compute_plot_RF: 1
  plot_ppoints: 1
  plot_RF_profile: 1

sks_stepwise:
  obtain_inventory_SKS: 0
  download_data_SKS: 0
  plot_traces_ENZ: 0
  plot_traces_RTZ: 0
  plot_SKS_measure: 1
  plot_SKS: 1
  picking_SKS: 1
  plot_traces: 1
  plot_trigger: 1
  plot_data_nodata_map: 1
```

Figure 3.2: Stepwise input files.

This file is provided in case user want to run specific part of the code (for testing or parameteres adjustments). By default, all the values listed below are set to 1, for fully automated run.

Data retrieval setting:

- client: enter all the *Obspy* clients separated by commas. The updated client list can be found there: `https://docs.obspy.org/packages/obspy.clients.fdsn.html`. Current list – 24 items: BGR, EMSC, ETH, GEONET, GFZ, ICGC, INGV, IPGP, IRIS, ISC, KNMI, KOERI, LMU, NCEDC, NIEP, NOA, ODC, ORFEUS, RASPISHAKE, RESIF, SCEDC, TEXNET, USGS, USP.
- network: list specific station network (* by default). If the user wants to perform RF or SKS studies on a selected station networks, those can be defined there for a fully automated run. In addition,

using the stepwise mode allows to edit the station list manually before downloading the data.

- station: list specific station code (* by default). If the user wants to perform RF or SKS studies on a selected station list, those can be defined there for a fully automated run. In addition, using the stepwise mode allows to edit the station list manually before downloading the data.
- locations: list all locations, e.g "",“00". Selecting all these location codes makes it relatively easy to acquire any and all available data.
- plot_stations: plot the stations map of all available stations and all retrieved stations. Possible values: (0/1), default (1).
- plot_events: plot the stations map of all potential events for each station. Possible values: (0/1), default (1).
- plot_all_retrieved_events_stations: plot the stations map of all retrieved events for each station. Possible values: (0/1), default (1).

RF setting:

- obtain_inventory_RF: (0/1), List all the stations available. Possible values: (0/1), default (1).
- download_data_RF: (0/1),Download the waveforms to calculate the Reciever Functions. Possible values: (0/1), default (1).
- compute_plot_RF: (0/1), Plot the results (for Reciever Functions). Possible values: (0/1), default (1).
- plot_ppoints: (0/1), Plot the piercing points (for Reciever Functions). Possible values: (0/1), default (1).
- plot_RF_profile: (0/1), Plot the vertical profiles (for Reciever Functions). Possible values: (0/1), default (1).

SKS setting:

- obtain_inventory_SKS:List all the stations available (for SKS). Possible values: (0/1), default (1).
- download_data_SKS: Download the waveforms to calculate the shear-wave splitting of SKS phase. Possible values: (0/1), default (1).
- plot_traces_ENZ: Plot the waveforms (for SKS). Possible values: (0/1), default (1).
- plot_traces_RTZ: Plot the rotated waveforms (for SKS). Possible values: (0/1), default (1).
- plot_SKS_measure: Plot the grid search for phase and delay time. Possible values: (0/1), default (1).
- plot_SKS: Plot the results (for SKS). Possible values: (0/1), default (1).
- picking_SKS: Picking of the SKS phase. Possible values: (0/1), default (1).
- plot_traces: Plot the waveforms of the SKS phase. Possible values: (0/1), default (1).
- plot_trigger: Plot the automatic picking of the SKS phase. Possible values: (0/1), default (1).
- plot_data_nodata_map: Plot the map with all stations (color code = data availability). Possible values: (0/1), default (1).

3.3 Settings/advRFparam.yaml

Settings > ! advRFparam.yaml

```yaml
filenames:
  invRFfile: rf_stations.xml #station xml
  RFsta: all_stations_RF.txt #station text catalog
  retr_stations: all_stations_rf_retrieved.txt #retrived stations list file
  data_rf_suffix: rf_profile_data #rf data file name: {net}-{stn}-rf_profile_data.h5
  events_map_suffix: RF-events_map #events map filename suffix {net}-{stn}-RF-events_map.png
  retr_station_prefix: RF_stations #retrieved stations prefix
  rf_compute_data_suffix: rf_profile_rfs #rf computation result file name: network-station-rf_profile_rfs.h5
  rfprofile_compute_result_prefix: rf_profile_profile #rf profile computation result file name: rf_profile_profile{azimuth}_*.h5
  h_kappa_settings:
    h_kappa_res_file: h-kappa-values.txt
    plot_h: 1
    plot_kappa: 1

rf_profile_settings:
  num_profile_divs_lat: 2
  num_profile_divs_lon: 3
  ppdepth: 70 #piercing points depth

rf_event_search_settings:
  minradiusRF: 30 #min radius from each station for events search
  maxradiusRF: 90 #max radius from each station for events search
  minmagnitudeRF: 6.5
  maxmagnitudeRF: 9.5

rf_filter_settings:
  minfreq: 0.5 #stream minfreq for bandpass
  maxfreq: 2 #stream maxfreq for bandpass

rf_display_settings:
  trace_height: 0.1 #height of one trace in inches
  trim_min: -5 #trim stream relative to onset before plotting
  trim_max: 20 #trim stream relative to onset before plotting
  rf_info: default #additional axes for RF plot, None for no additional axes
```

Figure 3.3: Input files for RF.

This file is provided for advanced users. The default values we provide are set after some testing with limited dataset. With other data, it is possible that the data quality is different and the parameters may requires some adjustments.

The first items in this file define the names of output files created by STADIUM-py. They do not affect the computation of receiver functions, and are mostly cosmetics.

- invRFfile: name of the station list file (default: rf_station.xml),
- RFsta: name of the station list text file (default:all_stations_RF.txt),
- retr_stations: name of the retrieved station list text file (default: all_stations_rf_retrieved.txt),
- data_rf_suffix: rf data file name. (default: net-stn-rf_profile_data.h5),
- events_map_suffix: events map filename suffix. (default: net-stn-RF-events_map.png),
- retr_station_prefix: retrieved stations prefix. (default: RF_station),
- rf_compute_data_suffix: rf computation result file namex. (default: net-stn-rf_profile_rfs.h5),
- rfprofile_compute_result_prefix: rf profile computation result file name. (default: rf_profile_profileazimuth_*.h5),
- h_kappa_res_file: file to store h-kappa values (default:h-kappa-values.txt).

Changing the following parameters will affect the computation process.

- plot_h: Plot Moho thickness. Possible values: (0/1), default (1),
- plot_kappa: Plot Vp/Vs ratio. Possible values: (0/1), default (1),
- num_profile_divs_lat: Number of EW profiles. Possible values: (0/100+), default (2),
- num_profile_divs_lon: Number of NS profiles. Possible values: (0/100+), default (2),
- minradiusRF: min radius from each station for events search (default: 30 degrees),
- maxradiusRF: max radius from each station for events search (default: 90 degrees),
- minmagnitudeRF: Minimum magnitudes of events (for Reciever Functions, default 6.5),
- maxmagnitudeRF: Maximum magnitudes of events (for Reciever Functions, default 9.5),
- minfreq: stream minfreq for bandpass (default 0.5),
- maxfreq: stream maxfreq for bandpass (default 2.0),
- trace_height: height of one trace in inches (default 0.1),
- trim_min: trim stream relative to onset before plotting (default -5s),
- trim_max: trim stream relative to onset before plotting (default +20s),
- ppdepth: piercing points depth (for plotting, default 70 km),
- rf_info: additional axes for RF plot, None for no additional axes.

3.4 Settings/advSKSparam.yaml

Settings > ! advSKSparam.yaml

```yaml
filenames:
  invSKSfile: sks_stations.xml #station xml
  SKSsta: stations_SKS.txt #station text catalog
  retr_stations: all_stations_sks_retrieved.txt #retrived stations list file
  data_sks_suffix: sks_profile_data #sks data file name: {net}-{stn}-sks_profile_data.h5
  events_map_suffix: SKS-events_map #events map filename suffix {net}-{stn}-SKS-events_map.png
  retr_station_prefix: SKS_stations #retrieved stations prefix
  sks_meas_indiv: sks_measurements.txt #sks measurements file suffix for individual stations
  sks_measure_map: SKS_station_Map #filename of sks measurements map

sks_event_search_settings:
  minradiusSKS: 90
  maxradiusSKS: 120
  minmagnitudeSKS: 6.5
  maxmagnitudeSKS: 9.5

sks_filter_settings:
  minfreq: 0.01 #stream minfreq for bandpass
  maxfreq: 0.6 #stream maxfreq for bandpass

sks_picking:
  trimstart: 30 #trim the traces for sks picking trace starttime+trimstart  to starttime+trimend
  trimend: 110 #trim the traces for sks picking trace starttime+trimstart  to starttime+trimend
  picking_algo:
    sks_picking_algo: recursive_sta_lta #picking algorithm for sks phase...other options are classic_sta_lta, z_detect, carl_sta_trig, delayed_sta_lta
    sks_picking_algo_thr0: 2.5 #starting threshold for sks picking algorithm
    sks_picking_algo_thr1: 0.65 #end threshold for sks picking algorithm

sks_measurement_contrains:
  sel_param: lam12 #options: snr, lam12; selection parameter of the measurements: either use signal to noise ratio, snr or use the eigenvalue ratio (lambda1/lambda2), lam12
  sel_param_settings:
    snr_ratio: 2 #minimum signal to noise ratio of the traces for filtering good measurements
    lam12fast_threh: 1.1 #threshold for the lambda1/lambda2 for fast direction pick
    lam12lag_threh: 1.1 #threshold for the lambda1/lambda2 for lag time pick
  lag_settings:
    minlag: 0 #minimum allowed lag time in sks measurements
    maxlag: 3 #maximum allowed lag time in sks measurements
    maxdlag: 1.5 #maximum allowed error in the lag time
  fast_dir_settings:
    maxdfast: 7 #maximum allowed error in the fast direction

sks_measurement_plot:
  show_no_measurement: 1 #1 if wish to plot stations with no measurement
  segregate_measurements: 0 #1 if wish to segregate measurements
  meas_seg_points: {lev1: 0, lev2: 8, lev3: 15} #segregate data to plot sks measurements in different color (3 levels)

error_plot_toggles:
  error_plot_indiv: 0 #make 1 to plot the error profiles of fast direction and lag time for each measurements
  error_plot_all: 1 #make 1 to plot the error profiles of fast direction and lag time for each measurements
```

Figure 3.4: Input files for SKS.

This file is provided for advanced users. The default values we provide are set after some testing with limited dataset. With other data, it is possible that the data quality is different and the parameters may requires some adjustments.

The first items in this file define the names of output files created by STADIUM-py. They do not affect the computation of receiver functions, and are mostly cosmetics.

- invSKSfile: station xml file (default: sks_stations.xml),
- SKSsta: station text catalog (default: stations_SKS.txt),
- retr_stations: retrived stations list file (default: all_stations_sks_retrieved.txt),
- data_sks_suffix: sks data file name (default: net-stn-sks_profile_data.h5),

- events_map_suffi: events map filename suffix (default: net-stn-SKS-events_map.png),
- retr_station_prefix: retrieved stations prefix (default: SKS_stations)
- sks_meas_indiv: sks measurements file suffix for individual stations (default: sks_measurements.txt),
- sks_measure_map: filename of sks measurements map (default: SKS_station_Map).

Changing the following parameters will affect the computation process.

- minradiusSKS: Minimum epicentral distance (for SKS, default: 90 degrees),
- maxradiusSKS: Maximum epicentral distance (for SKS, default: 120 degrees),
- minmagnitudeSKS: Minimum magnitudes of events (for SKS, default: 6.5),
- maxmagnitudeSKS: Minimum magnitudes of events (for SKS, default: 9.5),
- minfreq: stream min frequency for bandpass (default 0.01),
- maxfreq: stream max frequency for bandpass (default 0.6),
- trimstart: trim the traces for sks picking trace starttime+trimstart to starttime+trimend (default: 30s),
- trimend: trim the traces for sks picking trace starttime+trimstart to starttime+trimend (default: 110s),
- sks_picking_algo: picking algorithm for sks phase...other options are classic_sta_lta, z_detect, carl_sta_trig, delayed_sta_lta (default: recursive_sta_lta),
- sks_picking_algo_thr0: starting threshold for sks picking algorithm (default 2.5),
- sks_picking_algo_thr1: end threshold for sks picking algorithm (default 0.65),

- sel_param: options: snr, lam12; selection parameter of the measurements: either use signal to noise ratio, snr or use the eigenvalue ratio (lambda1/lambda2), lam12 (default: lam12),
- snratio: minimum signal to noise ratio of the traces for filtering good measurements (default 2.0),
- lam12fast_threh: threshold for the lambda1/lambda2 for fast direction pick (default 1.1),
- lam12lag_threh: threshold for the lambda1/lambda2 for lag time pick (default 1.1),
- minlag: minimum allowed lag time in sks measurements (default 0),
- maxlag: maximum allowed lag time in sks measurements (default 3.0),
- maxdfast: Maximum allowed error in the fast direction (default 7.0),
- maxdlag: maximum allowed error in the lag time (default 1.5),
- show_no_measurements: (0/1) if wish to plot stations with no measurement (default 1),
- show_null_measurements: (0/1) plot the stations with only null measurements on the map (default 1),
- segregate_measurements: (0/1) outout the segregated measurements to text file (default 1),
- meas_seg_points: { lev1: 0, lev2: 8, lev3: 15} segregate data to plot sks measurements in different color (3 levels)
- error_plot_indiv: (0/1), make 1 to plot the error profiles of fast direction and lag time for each measurements (default 1),
- error_plot_all: (0/1), make 1 to plot the error profiles of fast direction and lag time for each measurements (default 1).

Chapter 4

Example test run

4.1 Fresh start

First, set the parameter fresh_start and run STADIUM-py:

```
Are you sure you want to start fresh? (Input 'yes' to continue): yes
2020/04/28 11:02:09|INFO| --> Successfully created the directory Simple_RF_test/dataRF/
2020/04/28 11:02:09|INFO| --> Successfully created the directory Simple_RF_test/InfoRF/
2020/04/28 11:02:09|INFO| --> Successfully created the directory Simple_RF_test/ImagesRF/RFplots/
2020/04/28 11:02:09|INFO| --> Successfully created the directory Simple_RF_test/ImagesRF/STA_EV/
2020/04/28 11:02:09|INFO| --> Successfully created the directory Simple_RF_test/ImagesRF/Profile/
2020/04/28 11:02:09|INFO| --> Successfully created the directory Simple_RF_test/tmp/
```

Figure 4.1: Fresh start.

Figure 4.1 Fresh start.

A fresh start is advised for the first run, to ensure all folders are empty and that no remnant data from other projects are present. However, starting fresh may result in data loss.

4.2 Receiver functions

Search all stations for which data are available (defined input_files.txt). For each station, search the events that satisfy some critera (defined input_files.txt):

```
2020/04/28 11:02:12|INFO| Retrieving station information
2020/04/28 11:02:12|INFO| from IRIS
2020/04/28 11:02:12|INFO| Obtaining events catalog


2020/04/28 11:02:12|INFO| Retrieving event info for IU-GRFO
2020/04/28 11:02:12|INFO| Obtaining catalog: RF: IU-GRFO-1994-2006
2020/04/28 11:02:23|INFO| Writing the event data into a text file
2020/04/28 11:02:23|ERROR| Unable to write for 2001-11-14T09:26:10.410000Z
2020/04/28 11:02:23|ERROR| Unable to write for 2000-07-11T01:32:26.560000Z
2020/04/28 11:02:23|INFO| Finished writing the event data into a text and xml file


2020/04/28 11:02:23|INFO| Retrieving event info for IU-GRFO
2020/04/28 11:02:23|INFO| Obtaining catalog: RF: IU-GRFO-2006-2009
2020/04/28 11:02:31|INFO| Writing the event data into a text file
2020/04/28 11:02:31|INFO| Finished writing the event data into a text and xml file


2020/04/28 11:02:31|INFO| Retrieving event info for IU-GRFO
2020/04/28 11:02:31|INFO| Obtaining catalog: RF: IU-GRFO-2009-2013
2020/04/28 11:02:38|INFO| Writing the event data into a text file
2020/04/28 11:02:38|ERROR| Unable to write for 2011-03-11T05:51:20.500000Z
2020/04/28 11:02:38|INFO| Finished writing the event data into a text and xml file


2020/04/28 11:02:38|INFO| Retrieving event info for IU-GRFO
2020/04/28 11:02:38|INFO| Obtaining catalog: RF: IU-GRFO-2015-2599
2020/04/28 11:02:51|INFO| Writing the event data into a text file
2020/04/28 11:02:51|INFO| Finished writing the event data into a text and xml file
2020/04/28 11:02:51|INFO| Catalog xml/txt files saved at Simple_RF_test/InfoRF/
2020/04/28 11:02:51|INFO| Downloading the RF data
2020/04/28 11:02:51|INFO| all_stations_RF.txt exists!
2020/04/28 11:02:51|INFO| Simple_RF_test/InfoRF/all_stations_rf_retrieved.txt does not exist...obtaining events ca
talog..
2020/04/28 11:02:51|INFO|
```

Figure 4.2: Station and event selection.

Download the waveforms.

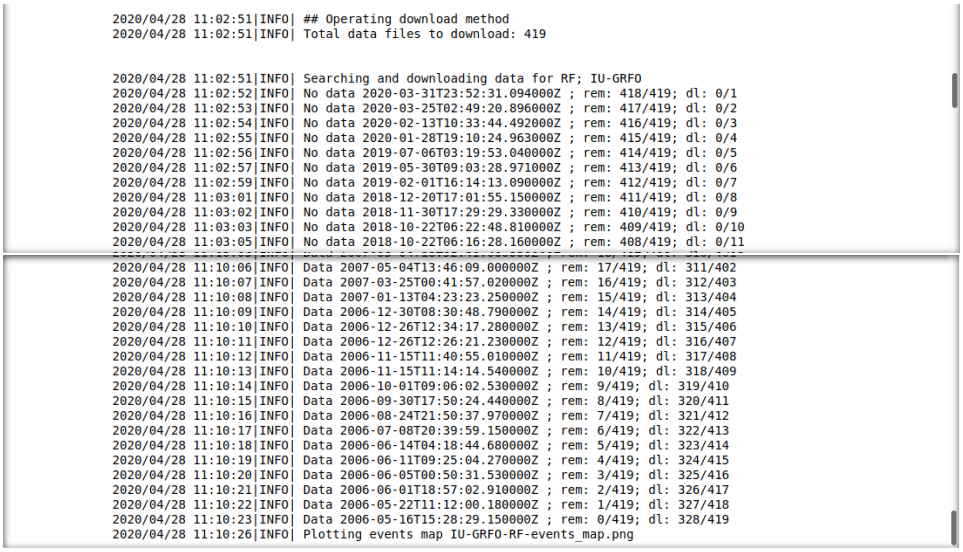

Figure 4.3: Station and event selection.

Data processing following:

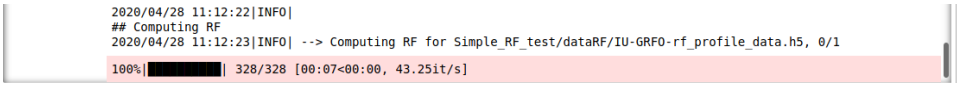

Figure 4.4: Processing starts.

The processing of receiver function follows:

- Filter and rotate the trace into the LQ domain (Figure 4.5).

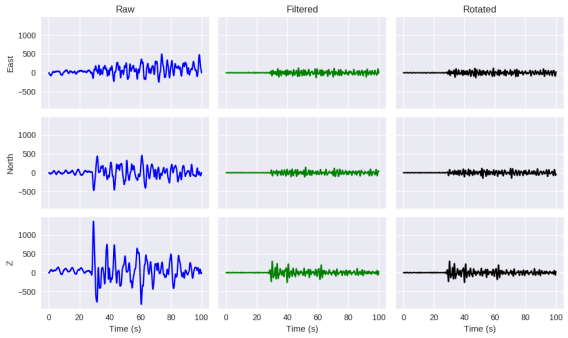

Figure 4.5: Rotation and filtering.

- Deconvolve the radial and tangential components by the vertical component (Figure 4.6).

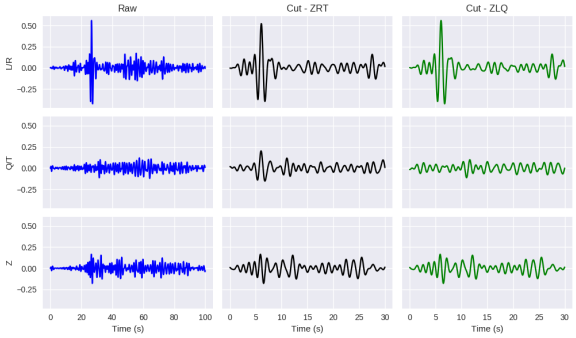

Figure 4.6: Deconvolution, trim and rotate the receiver functions.

- Calculate the piercing points for each event (Figure 4.8).

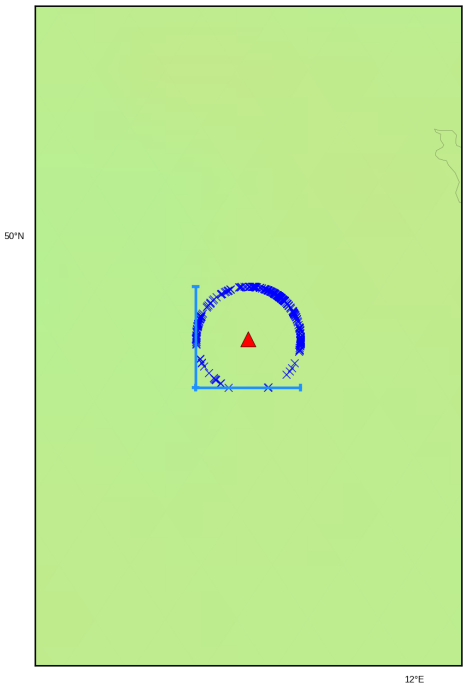

Figure 4.7: Locate the piercing points for each event.

- Stacks the reciever functions before plotting (Figure 4.8).

```
2020/04/28 11:12:43|INFO| ## Operating plot_piercingpoints_RF method
2020/04/28 11:12:43|INFO| --> Plotting the piercing points on map
Simple_RF_test/dataRF/ppoints_df.pkl does not exist
2020/04/28 11:12:45|INFO| Calculating profile
2020/04/28 11:12:45|INFO| Calculating the boxes
2020/04/28 11:12:45|INFO| For az: 0 startlat: 49.5433, startlon: 10.9752, endlat: 50.0334, length: 54
100%|██████████| 984/984 [00:03<00:00, 273.70it/s]
2020/04/28 11:12:49|INFO| ------> Calculated profile for azimuth 0: Simple_RF_test/dataRF/rf_profile_profile0_49_
50_54_0.h5; Number of traces in the box: 18


2020/04/28 11:12:49|INFO| Calculating the boxes
2020/04/28 11:12:49|INFO| For az: 90 startlat: 49.5433, startlon: 10.9752, endlon: 11.2818, length: 34
100%|██████████| 984/984 [00:02<00:00, 384.44it/s]
```

Figure 4.8: Stacks all receiver functions for a specific station.

- Plot the reciever functions for L and Q components, sorted by back azimuth (or distance) (Figure 4.9).

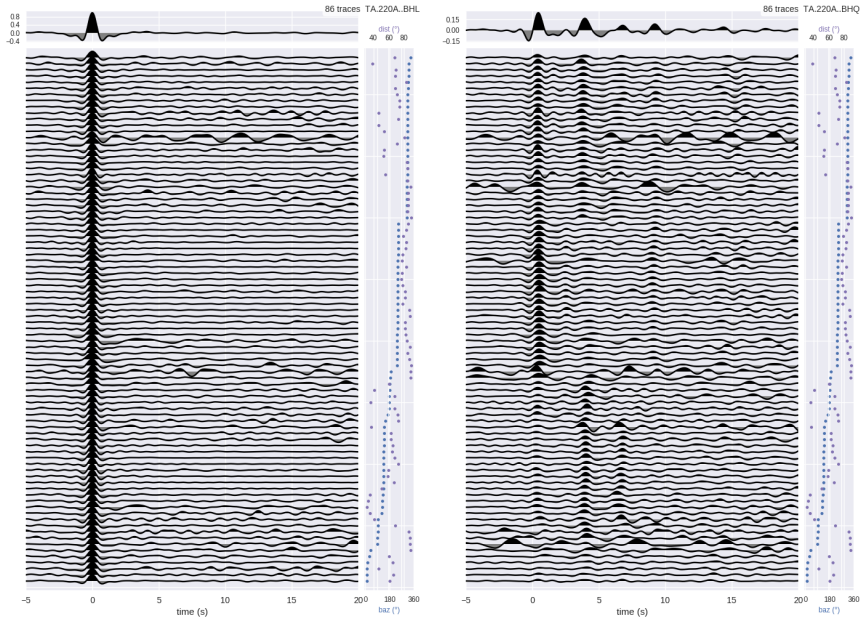

Figure 4.9: Receiver functions profiles for L and Q components.

- Create some vertical profile for all stations in selected regions (Figure 4.10).

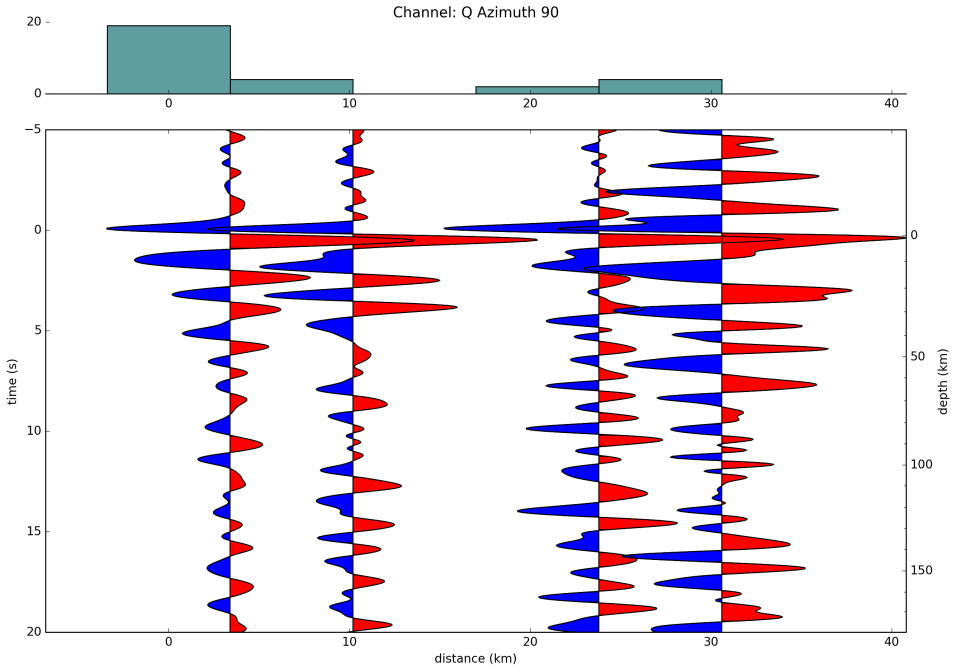

Figure 4.10: Vertical profiles within the region. With one station it is not really convincing.

Finally, compute the Moho depth and V_p/V_s ratio using the $H - \kappa$ method. First, RF stacks for individual stations are picked:

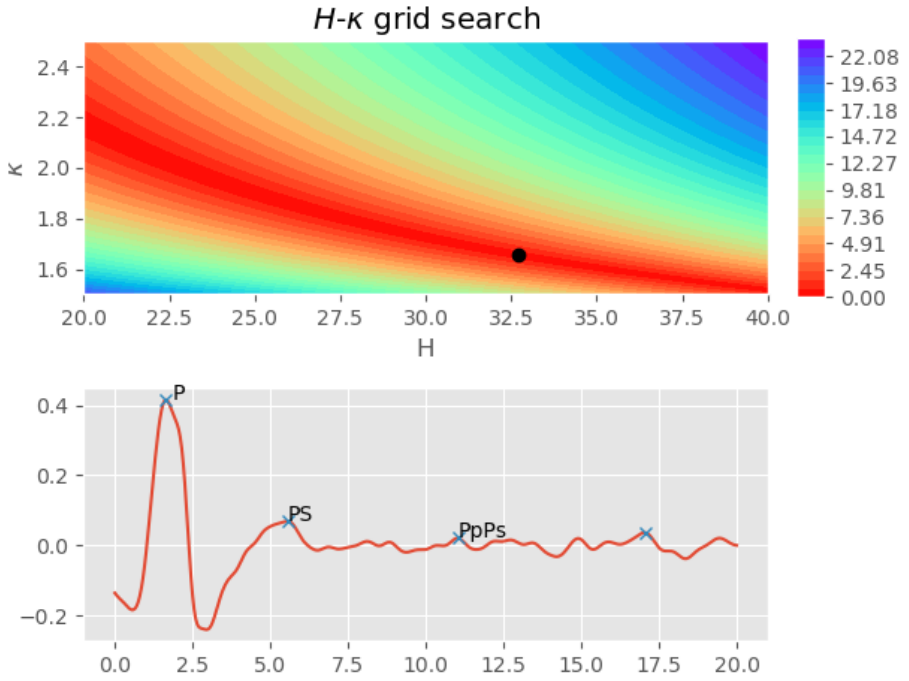

Figure 4.11: Automated picking of phases on stack receiver function.

Then a grid search is performed to fins the best fitting parameters, which are stored and plotted:

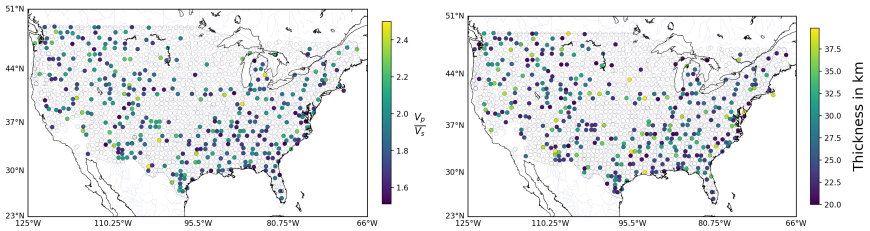

Figure 4.12: Map of Vp/Vs ratio (left) and Moho depth (right).

4.3 Shear-wave splitting

The processing of SKS measurements follows: Search all stations for which data are available (defined input_files.txt). For each station, search the events that satisfy some critera (defined input_files.txt):

```
2020/04/28 11:20:15|INFO| Running the program for makeRF: 0; makeSKS: 1
2020/04/28 11:20:15|INFO|
WORKING ON SKS
2020/04/28 11:20:15|INFO|
# Initializing the downloadDataclass
2020/04/28 11:20:15|INFO| Obtaining Inventory
2020/04/28 11:20:15|INFO| Trying to operate the get_stnxml method
2020/04/28 11:20:15|INFO| ## Operating get_stnxml method
```

Figure 4.13: Station and event selection.

Download the waveforms.

```
2020/04/28 11:20:15|INFO| Retrieving station information
2020/04/28 11:20:16|INFO| from IRIS
2020/04/28 11:20:16|INFO| Obtaining events catalog


2020/04/28 11:20:16|INFO| Retrieving event info for IU-GRFO
2020/04/28 11:20:16|INFO| Obtaining catalog: SKS: IU-GRFO-1994-2006
2020/04/28 11:20:20|INFO| Writing the event data into a text file
2020/04/28 11:20:20|ERROR| Unable to write for 2002-04-18T05:02:45.840000Z
2020/04/28 11:20:20|INFO| Finished writing the event data into a text and xml file


2020/04/28 11:20:20|INFO| Retrieving event info for IU-GRFO
2020/04/28 11:20:20|INFO| Obtaining catalog: SKS: IU-GRFO-2006-2009
2020/04/28 11:20:26|INFO| Writing the event data into a text file
2020/04/28 11:20:26|INFO| Finished writing the event data into a text and xml file


2020/04/28 11:20:26|INFO| Retrieving event info for IU-GRFO
2020/04/28 11:20:26|INFO| Obtaining catalog: SKS: IU-GRFO-2009-2013
2020/04/28 11:20:30|INFO| Writing the event data into a text file
2020/04/28 11:20:30|INFO| Finished writing the event data into a text and xml file


2020/04/28 11:20:30|INFO| Retrieving event info for IU-GRFO
2020/04/28 11:20:30|INFO| Obtaining catalog: SKS: IU-GRFO-2015-2599
2020/04/28 11:20:39|INFO| Writing the event data into a text file
2020/04/28 11:20:39|INFO| Finished writing the event data into a text and xml file
2020/04/28 11:20:39|INFO| Catalog xml/txt files saved at Simple_RF_test/InfoSKS/
2020/04/28 11:20:39|INFO| Downloading the SKS data
2020/04/28 11:20:39|INFO| Simple_RF_test/InfoSKS/all_stations_sks_retrieved.txt does not exist...obtaining invento
ry!
2020/04/28 11:20:39|INFO|


2020/04/28 11:20:39|INFO| ## Operating download method
2020/04/28 11:20:39|INFO| Total data files to download: 378


2020/04/28 11:20:39|INFO| Searching and downloading data for SKS; IU-GRFO
2020/04/28 11:20:39|INFO| Reading events catalog file
2020/04/28 11:20:41|INFO| Data 2009-02-11T17:34:51.380000Z ; rem: 377/378; dl: 0/1
2020/04/28 11:20:43|INFO| Data 2009-01-03T22:33:42.660000Z ; rem: 376/378; dl: 1/2
2020/04/28 11:20:45|INFO| Data 2009-01-03T19:43:55.840000Z ; rem: 375/378; dl: 2/3
2020/04/28 11:20:46|INFO| Data 2008-11-20T23:29:07.000000Z ; rem: 374/378; dl: 3/4
2020/04/28 11:20:48|INFO| Data 2008-11-16T17:02:32.810000Z ; rem: 373/378; dl: 4/5
2020/04/28 11:20:49|INFO| No data 2008-09-11T00:00:02.660000Z ; rem: 372/378; dl: 5/6
2020/04/28 11:20:52|INFO| Data 2008-06-30T06:17:43.820000Z ; rem: 371/378; dl: 5/7
2020/04/28 11:20:54|INFO| Data 2008-05-09T21:51:31.570000Z ; rem: 370/378; dl: 6/8
2020/04/28 11:20:55|INFO| Data 2008-03-03T14:11:15.120000Z ; rem: 369/378; dl: 7/9
2020/04/28 11:20:57|INFO| Data 2008-02-25T21:02:19.810000Z ; rem: 368/378; dl: 8/10
2020/04/28 11:21:00|INFO| Data 2008-02-25T18:06:05.040000Z ; rem: 367/378; dl: 9/11
2020/04/28 11:21:02|INFO| Data 2008-02-25T08:36:33.910000Z ; rem: 366/378; dl: 10/12
2020/04/28 11:21:04|INFO| Data 2008-02-24T14:46:23.530000Z ; rem: 365/378; dl: 11/13
2020/04/28 11:21:05|INFO| Data 2008-02-23T15:57:19.210000Z ; rem: 364/378; dl: 12/14
2020/04/28 11:21:07|INFO| Data 2008-02-10T12:22:03.340000Z ; rem: 363/378; dl: 13/15
2020/04/28 11:21:09|INFO| Data 2007-12-16T08:09:20.020000Z ; rem: 362/378; dl: 14/16
2020/04/28 11:21:11|INFO| Data 2007-11-25T19:53:08.260000Z ; rem: 361/378; dl: 15/17
2020/04/28 11:21:12|INFO| Data 2007-11-25T16:02:18.320000Z ; rem: 360/378; dl: 16/18
2020/04/28 11:21:14|INFO| Data 2007-11-16T03:13:00.430000Z ; rem: 359/378; dl: 17/19
2020/04/28 11:21:16|INFO| Data 2007-11-15T15:06:00.820000Z ; rem: 358/378; dl: 18/20
```

Figure 4.14: Station and event selection.

Plot the database.

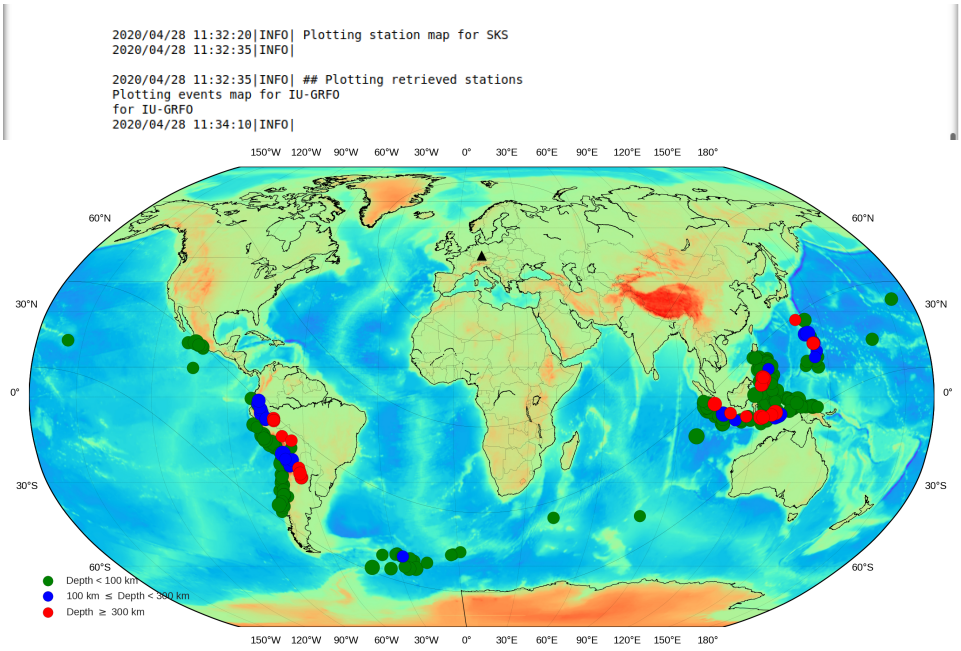

Figure 4.15: Station and event selection.

Filter and rotate the trace into the radial/tangential referencial.

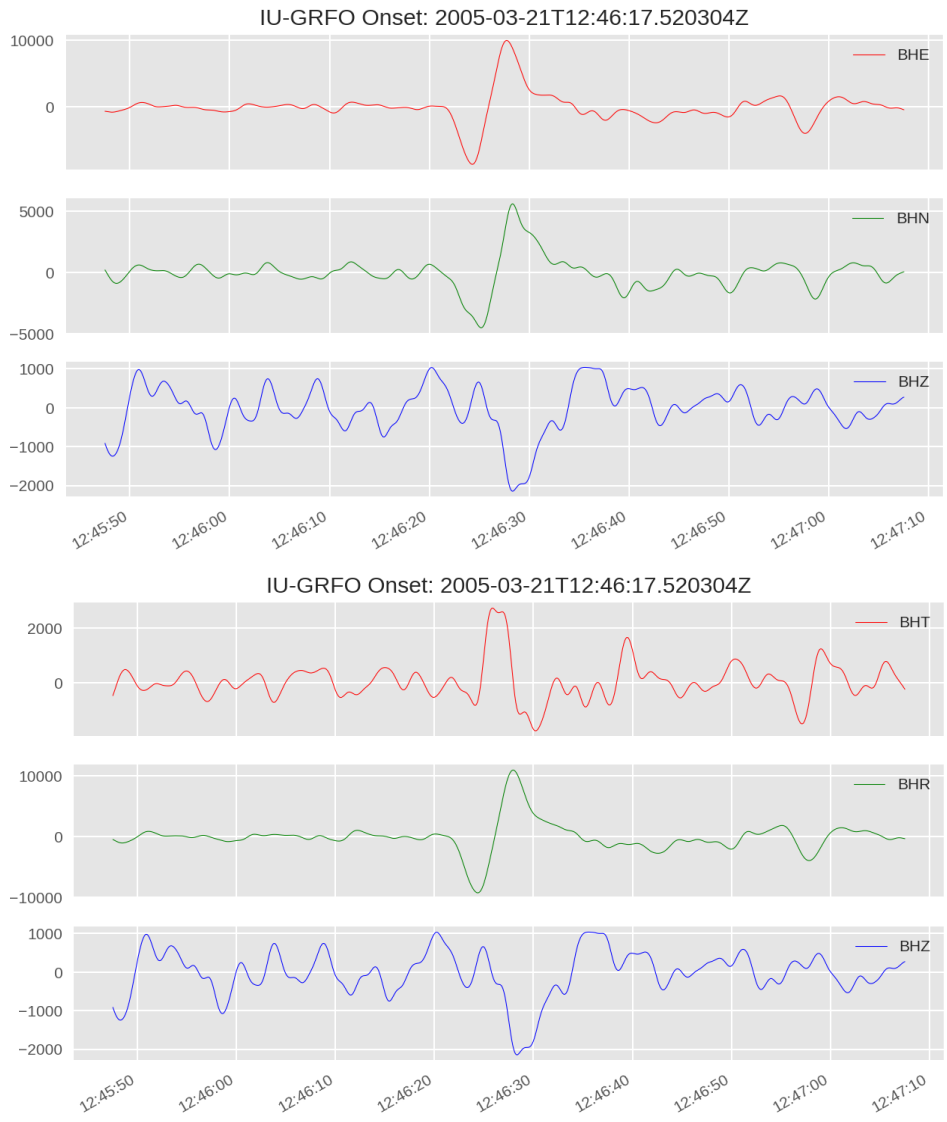

Figure 4.16: Traces in ZNE reference (top) and rotated (bottom).

Pick the SKS phase on the radial component:

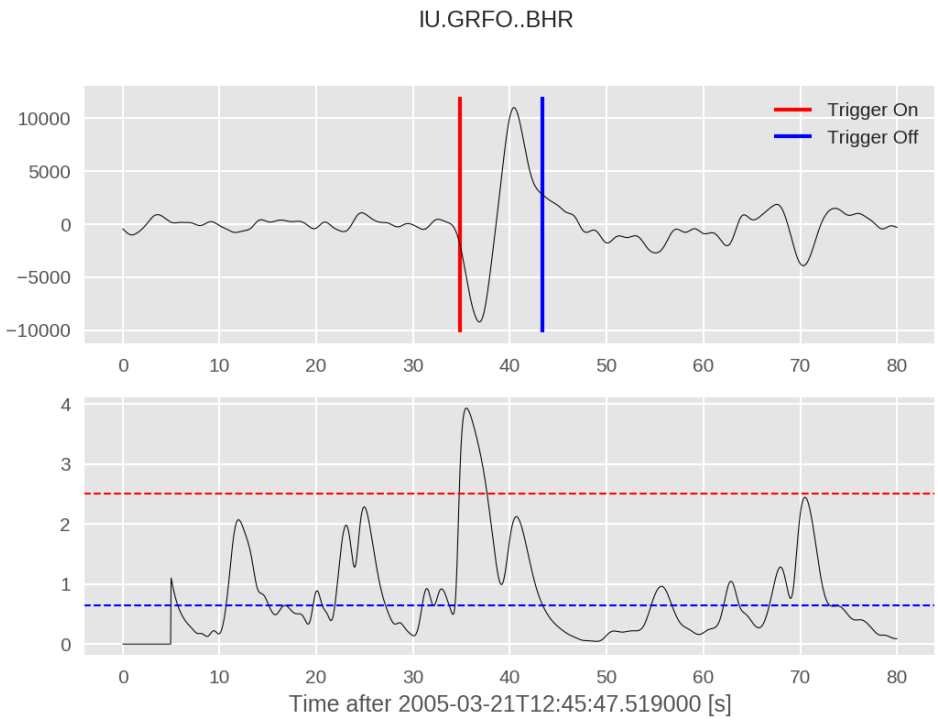

Figure 4.17: Automatic picking of the SKS phase.

Invert for phase and delay time.

```
2020/04/28 11:34:10|INFO| ## Pre-processing
2020/04/28 11:34:10|INFO| Cut the traces around the SKS arrival
2020/04/28 11:34:11|INFO| Calculating SKS arrival times for Simple_RF_test/dataSKS/IU-GRFO-sks_profile_data.h5

2020/04/28 11:34:14|INFO| 1/277 Bad measurement: 1994-07-25T22:00:24.460000Z! dfast = 21.5, dlag = 0.5, snr: 2.9
2020/04/28 11:34:15|INFO| 2/277 Bad phase pick: 1994-08-19T10:02:51.610000Z
2020/04/28 11:34:17|INFO| 3/277 Bad phase pick: 1994-09-28T16:39:53.250000Z
2020/04/28 11:34:19|INFO| 4/277 Null measurement 1994-10-08T21:44:07.710000Z
2020/04/28 11:34:21|INFO| 5/277 Bad phase pick: 1994-11-14T19:15:30.530000Z
2020/04/28 11:34:24|INFO| 6/277 Bad measurement: 1994-11-15T20:18:11.460000Z! dfast = 16.5, dlag = 0.3, snr: 2.2
2020/04/28 11:34:27|INFO| 7/277 Good measurement: 1994-12-10T16:17:39.240000Z; fast = 30.00+-4.00, lag = 1.70+-0.
23
```

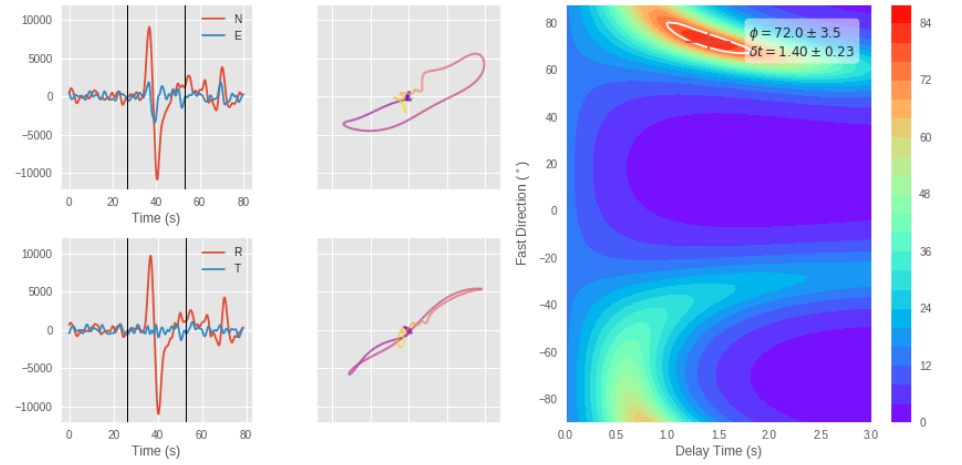

Figure 4.18: Grid search for best fitting phase and delay.

Produce a station specific summary file.

```
1 Stlon Stlat Stbaz
2 11.2203 49.6909 201.7683
3 EventTime EvLong EvLat Evdp FastDirection(degs) deltaFastDir(degs) LagTime(s) deltaLagTime(s)
4 1994-12-10T16:17:39.240000Z -101.4289  18.1152 53.8   30.0    4.0 1.7 0.2
5 1995-10-03T01:51:28.240000Z -77.8971  -2.7332 59.7    0.0    1.5 3.0 0.1
6 1995-12-25T04:43:23.160000Z 129.2035  -6.9207 133.0   74.0    4.5 1.2 0.2
7 1997-01-23T02:15:19.790000Z -65.6922 -22.0498 247.3   78.0    1.5 1.5 0.1
8 2000-05-12T18:43:15.160000Z -66.7400 -23.7790 193.5   76.0    2.0 1.6 0.2
9 2000-06-07T23:45:24.790000Z 101.9360  -4.6150 20.2   84.0    2.5 2.0 0.4
10 2002-10-10T10:50:22.230000Z 134.2585  -1.7511 24.8  -38.0    5.5 3.0 0.2
11 2004-11-26T02:25:00.440000Z 135.4096  -3.6543  5.1   62.0    3.5 2.0 0.5
12 2005-03-21T12:23:53.070000Z -63.4567 -24.9411 576.6   72.0    3.5 1.4 0.2
13 2005-09-26T01:55:37.710000Z -76.4755  -5.7362 129.1   84.0    2.0 2.3 0.4
14 2005-11-17T19:26:54.800000Z -67.9421 -22.3676 161.7   74.0    2.0 1.6 0.2
15 2006-04-30T19:17:15.700000Z -71.2350 -27.0991 13.7   78.0    2.5 1.3 0.2
16 2006-04-30T21:41:01.270000Z -71.0139 -27.2318 30.3   70.0    3.0 1.5 0.3
17 2007-10-24T21:02:51.810000Z 100.9873  -3.9514 29.9    6.0    5.0 2.0 0.4
18 2007-11-15T15:06:00.820000Z -70.0773 -22.7389 35.0  -30.0    4.0 3.0 0.1
19 2008-03-03T14:11:15.120000Z 125.7162  13.4239 26.0  -34.0    5.0 3.0 0.1
20 2010-03-05T11:47:07.530000Z -73.4491 -36.6675 21.8  -34.0    3.0 3.0 0.2
21 2010-08-12T11:54:15.020000Z -77.3694  -1.2779 206.5   86.0    1.5 2.2 0.4
22 2010-08-13T21:19:34.220000Z 141.5621  12.4748 16.4  -54.0    5.0 1.7 0.3
23 2011-01-01T09:56:58.460000Z -63.2373 -26.8513 584.3   72.0    4.5 1.5 0.2
```

Figure 4.19: Station summary for non null measurements.

Average all measurements for all stations and plot the results.

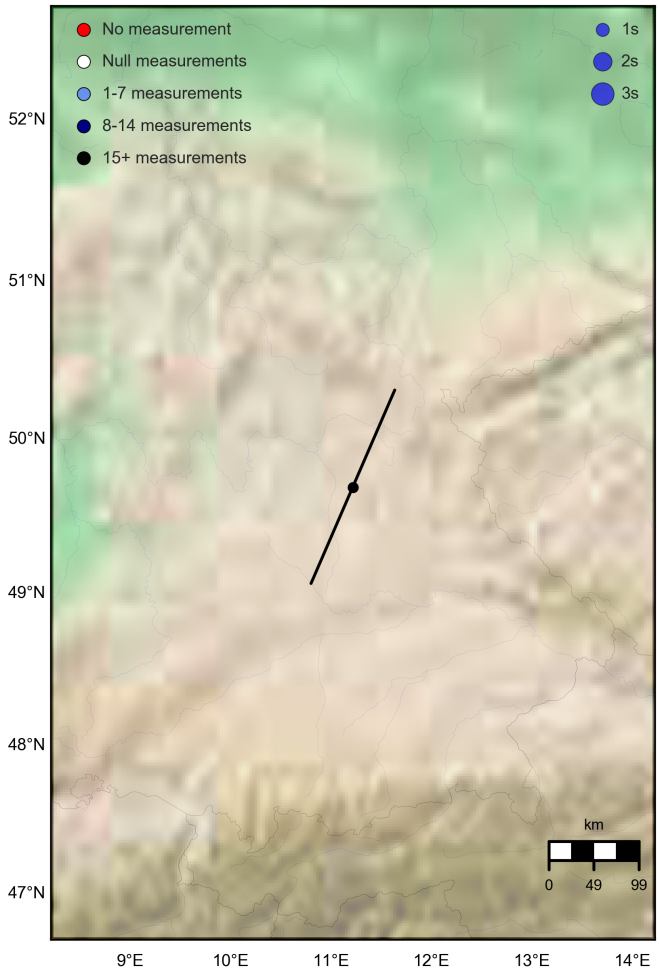

Figure 4.20: Final map of average measurements at all stations.

Prepare summary files.

- Summary file "all_sks_measure.txt"

NET STA lon lat AvgFastDir AvgLagTime NumMeasurements
IU GRFO 11.2203 49.6909 56.9800 1.9364 22

- Map of available data

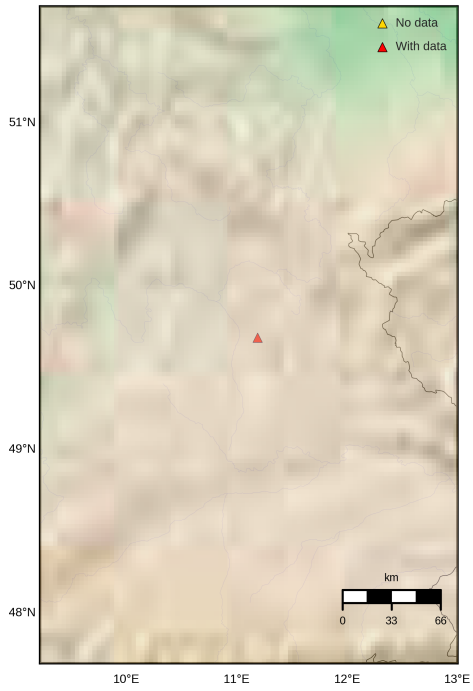

Figure 4.21: Map of available data.