Example of running SeisLoc for an icequake detection

Overview:

This notebook shows how to run SeisLoc for icequake detection, using a 2 minute window of continuous seismicity from Hudson et al (2019). Please refer to this paper for details and justification of the settings used.

Here, we detail how to:

1. Create a travel-times lookup table for the example seismometer network
2. Run a stage to coalesce energy through time
3. Run a trigger stage, to trigger and therefore detect an icequake
4. Outline of some of the key outputs

1. Create a travel-times lookup table (LUT)

In [4]:

```
# Import neccessary modules:
import SeisLoc.core.model as cmod   # Velocity model generation functions
import SeisLoc.signal.scan as cscan  # Detection and location algorithms
import SeisLoc.io.mseed as cmseed # MSEED data processing
import pandas as pd
```

In [7]:

```
# Set the parameters for the travel-times lookup table (LUT):
lut = cmod.LUT(center=[0.0,0.0,0.0], cell_count=[20,20,140], cell_size=[100,100,20], azimuth=0.0) # Create an empty LUT with a centre, cell count (x,y,z) and cell size (x,y,z in metres) specified
lut.set_lonlat(-17.224,64.328) # Set the lat and lon of the centre of the LUT
lut.lcc_standard_parallels=(64.32,64.335) # Set the LUT standard parallels
lut.setproj_wgs84('LCC') # Set the LUT projection
STATIONS = pd.read_csv('SeisLoc_inputs/Stations.txt',delimiter=',') # Read in a file containing the station information
lut.set_station(STATIONS.as_matrix(),units='lat_lon_elev') # Set the station parameters for the LUT
lut_path = 'SeisLoc_outputs/LUT/Icequake.LUT' # Set the path to save the LUT to
v_p_homo_model = 3630
v_s_homo_model = 1833
```

In [9]:

```
# And compute and save the LUT:
lut.compute_Homogeous(v_p_homo_model,v_s_homo_model) # Compute for a homogeneous velocity model
lut.save(lut_path)
```

2. Coalesce the seismic energy through time

In [10]:

```
# Read in the continuous seismic data:
DATA = cmseed.MSEED(lut_path,HOST_PATH='SeisLoc_inputs/MSEED/Icequake') # Imports the continuous seismic data in
DATA.path_structure(TYPE='YEAR/JD/STATION')
```

In [11]:

```
# Set the parameters for running the coalescence through time:
# Setup the coalescence object:
scn = cscan.SeisScan(DATA,lut_path,output_path='SeisLoc_outputs/RUN
S/Icequake',output_name='Icequake_example')
# Specify key detect/trigger parameters:
scn.sample_rate     = 500 # Sampling rate of data, in Hz
scn.bp_filter_p1    = [10, 125, 4] # The band-pass filter parameter
s for the P-phase (10 to 125 Hz, with 4th order corners)
scn.bp_filter_s1    = [10, 125, 4] # The band-pass filter parameter
s for the P-phase (10 to 125 Hz, with 4th order corners)
scn.onset_win_p1    = [0.01, 0.25] # Length of the STA and LTA time
windows for the P-phase
scn.onset_win_s1    = [0.05, 0.5] # Length of the STA and LTA time
windows for the S-phase
scn.time_step       = 0.75 # The length of the time-step
scn.CoalescenceGrid = False
scn.Decimate        = [1,1,1] # Decimation factors in x,y,z (no dec
imation here)
scn.NumberOfCores   = 12 # Number of cores/processors to use
```

```
Path = 'SeisLoc_outputs/RUNS/Icequake', Name = 'Icequake_example'
```

In [13]:

```
# Run SeisLoc to find the coalescence of energy through time:
# (Note: Outputs a .scn file with the overall coalesence value for
each timestep)
start_time_str = '2014-06-29T18:41:55.0'
end_time_str = '2014-06-29T18:42:20.0'
scn.Detect(start_time_str,end_time_str) # Finds the coalescence of
energy over the start and end times specified
```

```
==================================================================
============================================================
   SeisLoc - Coalescence Scanning : PATH:SeisLoc_outputs/RUNS/Iceq
uake - NAME:Icequake_example
==================================================================
====
   Continious Seismic Processing for 2014-06-29T18:41:55.000000 to
2014-06-29T18:42:20.000000
==================================================================
============================================================
~~~~~~~~~~~~~ Processing - 2014-06-29T18:41:54.350000 to 2014-06-2
9T18:41:57.750000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:41:55.100000 to 2014-06-2
9T18:41:58.500000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:41:55.850000 to 2014-06-2
9T18:41:59.250000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:41:56.600000 to 2014-06-2
9T18:42:00.000000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:41:57.350000 to 2014-06-2
9T18:42:00.750000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:41:58.100000 to 2014-06-2
9T18:42:01.500000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:41:58.850000 to 2014-06-2
```

```
9T18:42:02.250000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:41:59.600000 to 2014-06-2
9T18:42:03.000000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:00.350000 to 2014-06-2
9T18:42:03.750000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:01.100000 to 2014-06-2
9T18:42:04.500000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:01.850000 to 2014-06-2
9T18:42:05.250000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:02.600000 to 2014-06-2
9T18:42:06.000000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:03.350000 to 2014-06-2
9T18:42:06.750000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:04.100000 to 2014-06-2
9T18:42:07.500000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:04.850000 to 2014-06-2
9T18:42:08.250000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:05.600000 to 2014-06-2
9T18:42:09.000000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:06.350000 to 2014-06-2
9T18:42:09.750000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:07.100000 to 2014-06-2
9T18:42:10.500000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:07.850000 to 2014-06-2
9T18:42:11.250000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:08.600000 to 2014-06-2
9T18:42:12.000000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:09.350000 to 2014-06-2
9T18:42:12.750000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:10.100000 to 2014-06-2
9T18:42:13.500000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:10.850000 to 2014-06-2
9T18:42:14.250000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:11.600000 to 2014-06-2
9T18:42:15.000000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:12.350000 to 2014-06-2
9T18:42:15.750000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:13.100000 to 2014-06-2
9T18:42:16.500000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:13.850000 to 2014-06-2
9T18:42:17.250000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:14.600000 to 2014-06-2
9T18:42:18.000000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:15.350000 to 2014-06-2
9T18:42:18.750000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:16.100000 to 2014-06-2
9T18:42:19.500000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:16.850000 to 2014-06-2
9T18:42:20.250000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:17.600000 to 2014-06-2
9T18:42:21.000000 ~~~~~~~~~~~~~
~~~~~~~~~~~~~ Processing - 2014-06-29T18:42:18.350000 to 2014-06-2
9T18:42:21.750000 ~~~~~~~~~~~~~
```

3. Run the trigger stage, to detect and output individual icequakes

In [15]:

```
# Set any trigger parameters that may be different/additional to the initial coalescence stage:
scn.DetectionThreshold  = 1.5 # SNR threshold for the coalescence through time. Will detect an event if the coalescence goes above this for a given timestep
scn.MarginalWindow      = 2.75 # The length of the time-step window, + pre and post padding (i.e. 0.75 sec time-step window + 1s padding either side)
# Various output boolian switches:
scn.CoalescenceVideo    = False
scn.CoalescenceGrid     = False
scn.CoalescencePicture  = True
scn.CoalescenceTrace    = False
scn.PickingType         = 'Gaussian' # Defines type of pick error uncertainty estmiation method
```

In [16]:

```
# And run event detection/triggering:
start_time_str = '2014-06-29T18:41:55.0'
end_time_str = '2014-06-29T18:42:20.0'
scn.Trigger(start_time_str,end_time_str) # Triggers events, outputing .event, .stn and .pdf for each event in the directory SeisLoc_outputs/RUNS/Icequake
```

```
SeisLoc_outputs/RUNS/Icequake/Icequake_example.scn
--Processing for Event 1 of 1 - 20140629184210336
Elapsed time: 3.108747 seconds.

Elapsed time: 3.130824 seconds.

Creating Seismic Picture
Elapsed time: 2.098314 seconds.
```

4. Some of the key outputs

In [21]:

```
# Show the .event file, containing event origin time and location:
icequake_event_fname = "SeisLoc_outputs/RUNS/Icequake/Icequake_exam
ple_20140629184210336.event"
with open(icequake_event_fname) as f:
    lines = f.readlines()
for line in lines:
    print(line)
```

```
DT,COA,X,Y,Z,Gaussian_X,Gaussian_Y,Gaussian_Z,Gaussian_ErrX,Gaussi
an_ErrY,Gaussian_ErrZ,Covariance_X,Covariance_Y,Covariance_Z,Covar
iance_ErrX,Covariance_ErrY,Covariance_ErrZ

2014-06-29 18:42:10.336,1.7339595976510833,-17.222986628219758,64.
33023353261483,509.0,-17.222204313905074,64.33008195752763,517.139
7962042175,4.869102842007661,6.528671375764944,3.3814495136040805,
-17.22153468322658,64.33009261928694,502.62342969754235,91.2981010
791098,74.24957087823364,88.3461566409338
```

In [23]:

```
# Show the .stn file, containing station time picks:
icequake_stn_fname = "SeisLoc_outputs/RUNS/Icequake/Icequake_exampl
e_20140629184210336.stn"
with open(icequake_stn_fname) as f:
    lines = f.readlines()
for line in lines:
    print(line)
```

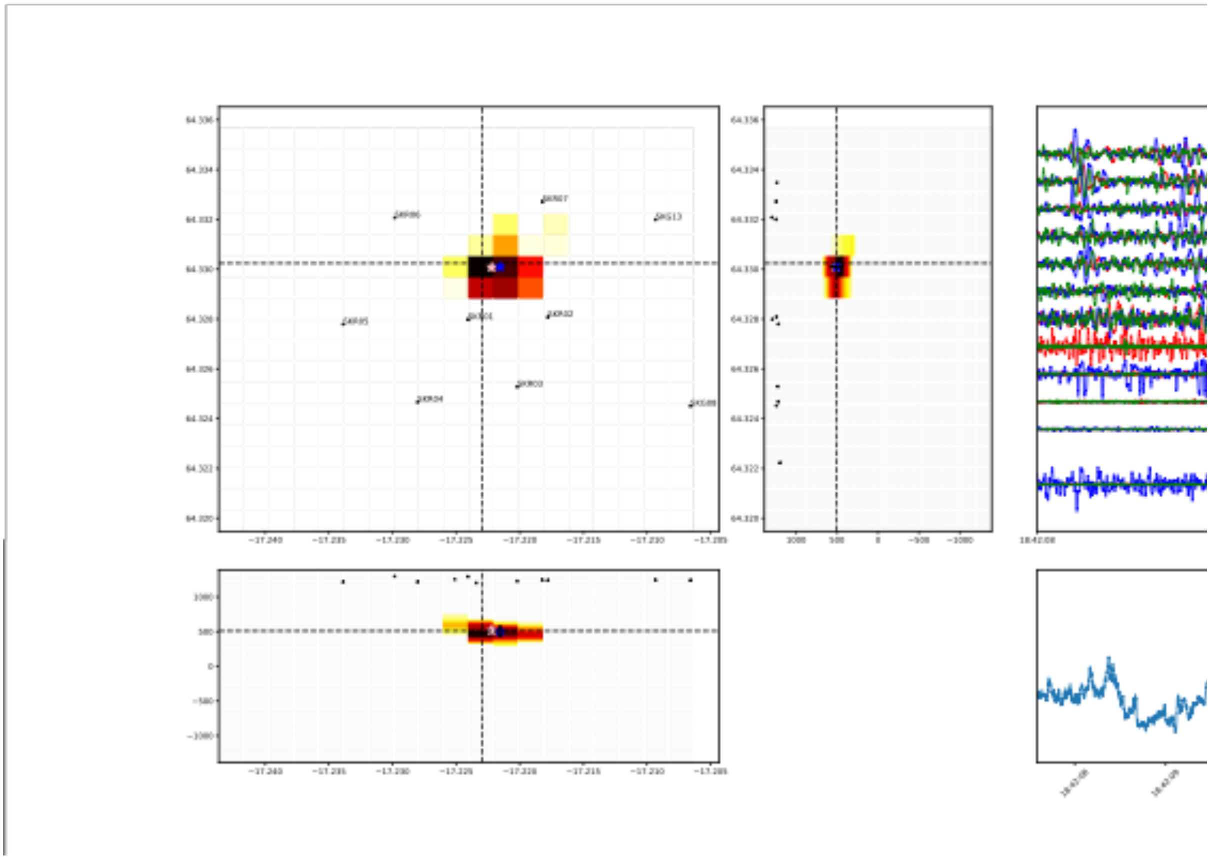
```
Name,Phase,ModelledTime,PickTime,PickError

SKR01,P,2014-06-29 18:42:10.563666,-1.0,-1.0

SKR01,S,2014-06-29 18:42:10.786861,-1.0,-1.0

SKR02,P,2014-06-29 18:42:10.559948,-1.0,-1.0

SKR02,S,2014-06-29 18:42:10.779498,-1.0,-1.0

SKR03,P,2014-06-29 18:42:10.588904,-1.0,-1.0

SKR03,S,2014-06-29 18:42:10.836842,-1.0,-1.0

SKR04,P,2014-06-29 18:42:10.604473,-1.0,-1.0

SKR04,S,2014-06-29 18:42:10.867673,-1.0,-1.0

SKR05,P,2014-06-29 18:42:10.590989,-1.0,-1.0

SKR05,S,2014-06-29 18:42:10.840970,-1.0,-1.0

SKR06,P,2014-06-29 18:42:10.578686,-1.0,-1.0

SKR06,S,2014-06-29 18:42:10.816605,-1.0,-1.0
```

```
SKR07,P,2014-06-29 18:42:10.562988,-1.0,-1.0

SKR07,S,2014-06-29 18:42:10.785517,-1.0,-1.0

SKG08,P,2014-06-29 18:42:10.681839,-1.0,-1.0

SKG08,S,2014-06-29 18:42:11.020885,-1.0,-1.0

SKG09,P,2014-06-29 18:42:10.748490,-1.0,-1.0

SKG09,S,2014-06-29 18:42:11.152879,-1.0,-1.0

SKG10,P,2014-06-29 18:42:10.764317,-1.0,-1.0

SKG10,S,2014-06-29 18:42:11.184223,-1.0,-1.0

SKG11,P,2014-06-29 18:42:10.740690,-1.0,-1.0

SKG11,S,2014-06-29 18:42:11.137432,-1.0,-1.0

SKG12,P,2014-06-29 18:42:10.725075,-1.0,-1.0

SKG12,S,2014-06-29 18:42:11.106508,-1.0,-1.0

SKG13,P,2014-06-29 18:42:10.614625,-1.0,-1.0

SKG13,S,2014-06-29 18:42:10.887777,-1.0,-1.0
```

In [31]:

```
# Show the coalescence pdf file, containing event origin time and location:
icequake_coal_image_fname = "SeisLoc_outputs/RUNS/Icequake/Icequake_example_20140629184210336_EventLocationError.pdf"
from IPython.display import IFrame # For plotting pdf
IFrame(icequake_coal_image_fname, width=800, height=400) # Plot pdf
```

Out[31]:



References:

Hudson, T.S., Smith, J., Brisbourne, A.M., and White R.S. (2019). Automated detection of basal icequakes and discrimination from surface crevassing. Annals of Glaciology, 79