Cross-Correlation Tutorial - SoCal Stations

May 5, 2020

1 Overview of Cross-Correlation Processing

1.1 Steps:

1.2 1. Download Data (python scripts)

```
Currently I download daily miniseed files for each station and channel
```

1.3 2. Pre-process data (python scripts)

```
Unpack mseed data, demean/detrend/taper/merge/bandpass,
remove instrument response, and rotate.
Output SAC files
```

1.4 3. Temporal & spectral normalization & Cross-Correlate (csh, C++, and fortran scripts)

1.5 4. Stack Cross-Correlations

1.6 5. Rotate

1.7 6. Symmetric

Note that each of these steps is detailed below, but please contact me (eliza.m.berg@gmail.com) with any questions you may have.

2 File Structure

2.1 My directory and file structure are organized (mainly) according to Codes vs. Data

2.2 Codes Directory (SoCalTutorial_2020):

2.2.1 stationlst.txt (From IRIS, main station text file)

2.2.2 Modules

Major code packages here that are referrenced elsewhere

Python: StationListSetup.py (creates station lists and datetime lists referrenced throughout download & unpack scripts)

Csh: do_whiten_xcorr.csh (perform temporal & spectral normalization, and cross-correlate) – called by Data directory > run_whiten_xcorr.csh

Executables: filter4, whiten_phamp, justCOR (referenced within do_whiten_xcorr.csh)

Directories:

FILTER4

```
*Main script to remove earthquake signals* May need to increase declared array lengths if have
```

WHITEN_PHAM

```
*Main script to whiten* Can change array lengths of signal if have lower sampling rate, may ne
```

JUSTCOR *Main script to cross-correlate*

MUST CHANGE (in justCOR.c) shdamp2.delta=0.25; and recompile

2.2.3 To recompile:

Type in terminal (within JUSTCOR_25 dir):

make clean

Followed by in terminal:

make

according to sampling rate (here 4Hz)

2.2.4 Scripts

Python scripts used to download and unpack data can be found here

Download mseed files: Download_SoCal_Test.py

Pre-Process mseed files (convert to SAC): Scripts/PreProcessing_SoCalbyMonth.py

2.2.5 Notebooks

Notebooks used to document each step found here

2.3 Data Directory (Jan_Mar_TA_Download)

2.3.1 Year (2015) of data

Months of data

Days of data, each day containing mseed files (not required) ft*SAC files (required). Will generate rest of files upon running whitening & cross-correlation scripts

COR_ZEN > directory containing cross-correlations (sorted by source station directory)

XML

2.3.2 Station lists

Needed Station Lists:

1. **SoCal_Output_YEAR_Mon_vfinal.csv (needed in Cross-Correlating)**

Example Contents of: SoCal_Output_2015_Jan_vfinal.csv

StationOutName Latitude Longitude Elevation

ARV 35.126900 -118.830090 258

FMP 33.712640 -118.293810 89

2. **Station List for Pre-Processing - Datadir/DownloadOutput_SoCal.csv**

Can skip if not using preprocessing python script

Generated in Downloading process, but can easily be made elsewhere

Contains (with header):

Database|Network|Station|Location|Channel|Latitude|Longitude|Elevation

Not-Needed Station Lists:

1. List of missing data: Missing_SoCal_YEAR_Mon.txt
2. 1comp only station list: SoCalStations_YEAR_Mon_vertonly.txt
3. 3comp station list: SoCalStations_YEAR_Mon_3comp.txt

2.3.3 Codes

Several codes are located here, including:

1. run_whiten_xcorr.csh (call whiten and cross-correlation scripts)
2. run_stack_rotate_symm.csh (calls stack, rotate, and make symmetric)

3 Download Data

First, I determine stations of interest from the IRIS builder and download the station list as a TEXT file (lower left under "Download Stations")

Example: http://ds.iris.edu/gmap/#network=CI&station=ADO,ARV,FMP&planet=earth

I upload the station text file as **"stationlist.txt" in dir SoCalTutorial_2020**

Next, I edit the variables in "Download_SoCal_Test.py" in dir SoCalTutorial_2020/Scripts (lines 26-53)

Finally, I run **Download_SoCal_Test.py** (formatted for python3). Directories are automatically created and mseed & XML files are uploaded (see Jan_Mar_TA_Download dir).

Note that if some files are not downloaded, often it's because of a server disconnect issue and re-running the script will grab files that it could not obtain previously (at least for the SCEDC server).

Input/Variables:

1. **Directory structure**

scriptdir='/uufs/chpc.utah.edu/common/home/u1015716/SoCalTutorial_2020/Scripts'

datadir='/uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download'

moduledir='/uufs/chpc.utah.edu/common/home/u1015716/SoCalTutorial_2020/Modules'

2. **Station list from IRIS**

stalist='/uufs/chpc.utah.edu/common/home/u1015716/SoCalTutorial_2020/stationlist.txt'

3. **Start time and endtime**

starttimemo=1 #Jan

starttimeyr=2015

endtime={"year":2015,"month":4,"day":1} #set to run thru end of March

4. **Dimensions of Lon/lat box to look for stations within when doing mass_downloader**

minlat=33.7; maxlat=35.2

minlon=-118.9; maxlon=-117.3

5. **Channel Types**

chantypes=['BH[ZNE]','HH[ZNE]']

Output/Variables:

1. **Output download info (stations and filenames); referrenced in pre-processing (NOT cross-corr scripts)**

downoutput='DownloadOutput_SoCal.csv'

DownloadOutput_SoCal.csv Contains:

Database|Network|Station|Location|Channel|Latitude|Longitude|Elevation

SCEDC|CI|ADO||BHZ, BHN, BHE|34.550460|-117.433910|908

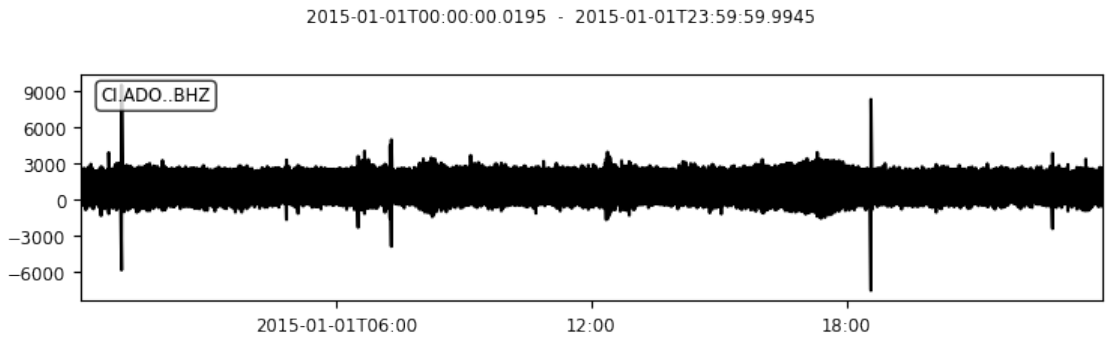
SCEDC|CI|ARV||BHZ, BHN, BHE|35.126900|-118.830090|258

SCEDC|CI|FMP||BHZ, BHN, BHE|33.712640|-118.293810|89

2. **Output individual file info (for user information, not referrenced later)**

downoutputfiles='DownloadOutput_SoCal_ALLFiles.csv'

[20]:
```
# Quickly plot example of downloaded miniseed file


from obspy import read
import obspy


datadir='/uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/
 ↪Tutorials/Jan_Mar_TA_Download'
datayear='2015'
datamon='Jan'
datamn='1' #month number
dataday='1'

datasta='ADO'
datanw='CI'
datachan='BHZ'

#ADO.CI.BHE.1_0_0.mseed
datastream=read(datadir+'/'+datayear+'/'+datamon+'/
 ↪'+datayear+'_'+datamn+'_'+dataday+'_0_0_0/'
     +datasta+'.'+datanw+'.'+datachan+'.'+dataday+'_0_0.mseed')

datastream.plot()
```



4 Pre-Processing

The main goal of pre-processing is to convert mseed files into daily SAC files with filled header info (applying some processing along the way). A station list is also generated in this script (referenced in cross-correlation scripts), as well as a list of 3component stations and 1component only stations.

Note that this can be done in a variety of ways, but I have used a python script here. Edits (other than variable settings, L27-99) should not be required.

Python3 script: **CodesDir(SoCalTutorial_2020)/Scripts/PreProcessing_SoCalbyMonth.py**

Inputs/Variables (L27-99):

1. Year & month [Inputs] for example, run (for January 2015 data) via: **python PrProcessing_SoCalbyMonth.py 2015 1**
2. **Datadir** (where 2015/Jan/2015_1_1_0_0_0/*mseed is located) & **scriptdir** (where this script is) [Variables]
3. Station List [Variable] - Datadir/DownloadOutput_SoCal.csv Generated in Downloading process, but can easily be made elsewhere Contains (with header): Database|Network|Station|Location|Channel|Latitude|Longitude|Elevation
4. Final sampling rate of output SAC files (variable) newsamprate=4 #new sampling rate to use, in Hz

Outputs:

1. **SAC files** for each channel of each station of each day, formatted: **YEAR/Mon/YEAR_Mo_Dy_0_0_0/ft_STA.LH?.SAC** where Mon=Name of month, Mo=number of month, and ?=Channel (Z,N,E)
2. **List of final station info (used in xcorr scripts)**

outputstainfo=datadir+'/SoCal_Output_'+str(runyear)+'_'+mostr+'_vfinal.csv' Example Contents of: SoCal_Output_2015_Jan_vfinal.csv StationOutName Latitude Longitude Elevation ARV 35.126900 -118.830090 258 FMP 33.712640 -118.293810 89

The following are general information for the user, and not used later on in the cross-correlation, etc codes

3. List of missing data

outputmissing=datadir+'/Missing_SoCal_'+str(yr)+'_'+mostr+'.txt'

4. 1comp only station list

one_complist=datadir+'/SoCalStations_'+str(runyear)+'_'+mostr+'_vertonly.txt'

5. 3comp station list

three_complist=datadir+'/SoCalStations_'+str(runyear)+'_'+mostr+'_3comp.txt'

Pre-Processing Steps include (for each station, described by 3 mseed files (3channels)):

1. **Create output station file list**

Check for duplicate names, create new names if necessary, and output info

2. **Demean, detrend, apply 1% taper** and merge traces for each mseed file (each channel),

combine channels into single stream (traces of new stream correspond to each channel)

3. **Bandpass** - Nyquist (of final sampling rate) to 100s
4. **Trim** data to be a full day (86400 seconds), so all final files will contain the same number of points
5. **Interpolate** data to preferred sampling rate (here I use 4Hz)
6. **Remove instrument response** (python converts from station units into m/s; requires XML files though) (I also apply a bandpass here with 5% taper on the corners to prevent amplifying noise during the deconvolution)
7. **Convert data from m/s to nm/s** (python format to SAC format)
8. **Insert azimuthal data to ZNE, format cmpaz and cmpinc of headers according to channel**

NOTE: you can **instead** rotate to ZNE (see zne=rotate2zne(..)code) and insert the azimuthal and inclination as noted in the tr.stats.sac['cmpaz']=0 (or 90, or 0) into the SAC headers. This is important to do this way IF there is an issue with inclination, as the ROTATE script will not account for inclination error (just azimuthal error) when converting to radial and transverse.

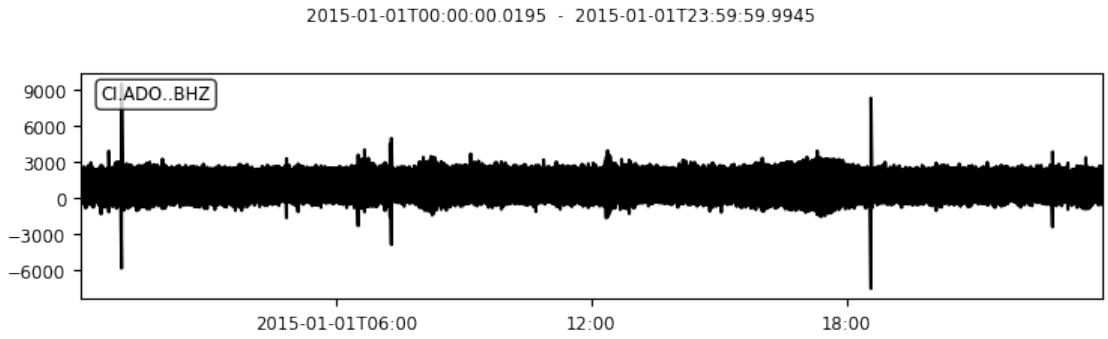
9. **Output SAC files** for each channel of each station of each day, formatted: **YEAR/Mon/YEAR_Mo_Dy_0_0_0/ft_STA.LH?.SAC**

where Mon=Name of month, Mo=number of month, and ?=Channel (Z,N,E)

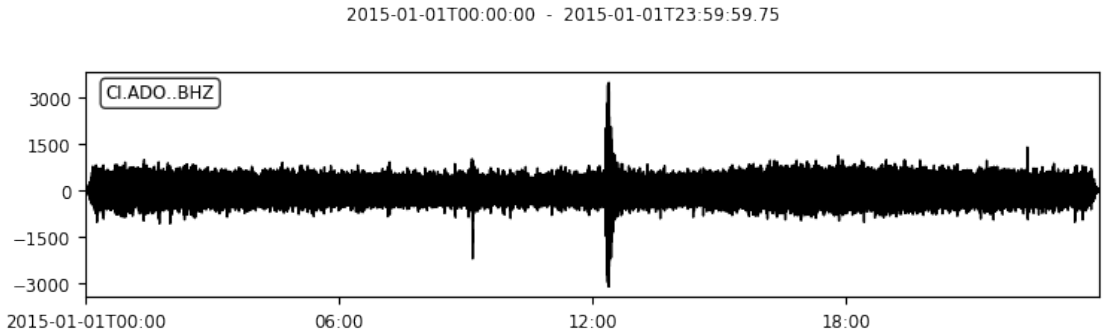
[21]:
```
# Quickly plot example of downloaded miniseed file, and resulting SAC file␣
 ↪created after pre-processing

datadir='/uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/
 ↪Tutorials/Jan_Mar_TA_Download'
datayear='2015'
datamon='Jan'
datamn='1' #month number
dataday='1'

datasta='ADO'
datanw='CI'
datachan='BHZ'

#ADO.CI.BHE.1_0_0.mseed
```

```
mseedf=datadir+'/'+datayear+'/'+datamon+'/
 ↪'+datayear+'_'+datamn+'_'+dataday+'_0_0_0/' \
    +datasta+'.'+datanw+'.'+datachan+'.'+dataday+'_0_0.mseed'

print('Example mseed file: ',mseedf)
datastream=read(mseedf)

datastream.plot()

#ft_ADO.LHZ.SAC
sacf=datadir+'/'+datayear+'/'+datamon+'/
 ↪'+datayear+'_'+datamn+'_'+dataday+'_0_0_0/' \
    +'ft_'+datasta+'.'+'LH'+datachan[-1]+'.SAC'

sacdatastream=read(sacf)
print('Example sac file: ',sacf)


sacdatastream.plot()
```

```
Example mseed file:  /uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data
_2015/Tutorials/Jan_Mar_TA_Download/2015/Jan/2015_1_1_0_0_0/ADO.CI.BHZ.1_0_0.mse
ed
```




```
Example sac file:  /uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2
015/Tutorials/Jan_Mar_TA_Download/2015/Jan/2015_1_1_0_0_0/ft_ADO.LHZ.SAC
```



---

5 Whiten & Cross-Correlate Data

5.1 do_whiten_xcorr.csh (called by DataDir/run_whiten_xcorr.csh):

`script calls:`

5.1.1 filter4 (from Modules/filter4)

```
This is the main script to remove earthquakes
This is in current dir's dir: FILTER4
Changed Makefile:
LDLIBS = -L/uufs/chpc.utah.edu/common/home/u1015716/PROG_TOOLS/lib -lfftw3
```

5.1.2 whiten_phamp

```
This is the main script to whiten
Changed Makefile:
LDLIBS =   -L/uufs/chpc.utah.edu/common/home/u1015716/PROG_TOOLS/lib -lfftw3

Changed include line in driver_c.c
Note: driver_c.c has LH lines: sprintf(nameZ,"ft_%s.LHZ.SAC",name);
```

5.1.3 justCOR

```
Edited include line of justCor.c and
Changed Makefile:
LDLIBS = -L/uufs/chpc.utah.edu/common/home/u1015716/PROG_TOOLS/lib -lfftw3
```

Also, **MUST CHANGE (in justCOR.c) shdamp2.delta=0.25; and recompile according to sampling rate (here 4Hz)**

5.2 To recompile:

Type in terminal (within JUSTCOR_25 dir): make clean

Followed by in terminal: make

Copy executable to above dir: cp justcor ../.

5.3 Inputs/Variables you may want to change:

1. **Directory Structure (Datadir/run_whiten_xcorr.csh)**

set mdir="/uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download"

2. **Station Information (Datadir/run_whiten_xcorr.csh)**

If you have created your own station file (see format below), can quickly change the name in this script

Note that the station.lst used by the whiten and cross-correlate executables requires: Sta Lon Lat Elev

*If you make your own station.lst file, you may also want to change the following lines in **run_whiten_xcorr.csh :***

```
set stafile=$mdir'/SoCal_Output_'$year'_'$month'_vfinal.csv'

sed 1d $stafile | awk '{print $1,$3,$2,$4}' > station.lst
```

Example Contents of: SoCal_Output_2015_Jan_vfinal.csv

StationOutName Latitude Longitude Elevation

ARV 35.126900 -118.830090 258

FMP 33.712640 -118.293810 89

Thus, the sed line in run_whiten_xcorr.csh re-formats into a station file with Sta Lon Lat Elev

3. **Path to do_whiten_xcorr.csh (Datadir/run_whiten_xcorr.csh)**

Modify path in lines:

#Temporal normalization, whiten, xcorr

```
csh /uufs/chpc.utah.edu/common/home/u1015716/SoCalTutorial_2020/Modules/do_whiten_xcorr.csh $y
```

4. **Whitening parameters (Modules/do_whiten_xcorr.csh)**

Need to change the low period limits (high frequency); changing the high period (low freq) does not add computational time so don't change those awk '{print "1000 800 0.55 0.53 1 1", $1}' stalst > param1.dat #4Hz

Change 0.55 and 0.53 according to sampling rate (go slightly above the nyquist)

5. **Delta in Cross-Correlation** (justCOR/justCOR.c) *Mentioned above*

shdamp2.delta=0.25;

6. **Cross-Correlation Npts and Path (Modules/do_whiten_xcorr.csh)**

   # path/justCOR npts SrcChan RecChan (Chan: 0=Z, 1=N, 2=E)

   # npts=SamplingRate*SecondsAfter0ofCrossCorr (2000 = 500s*4Hz)

/uufs/chpc.utah.edu/common/home/u1015716/SoCalTutorial_2020/Modules/justCOR 2000 0 0

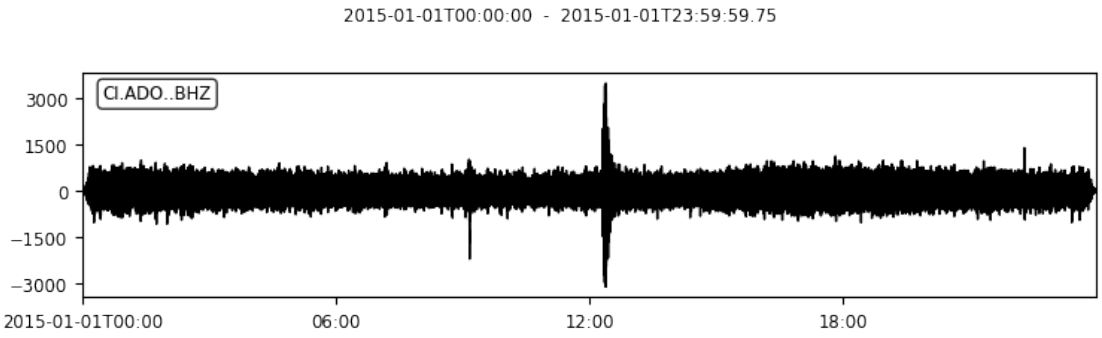
5.4 Run Via

cd Datadir csh run_whiten_xcorr.csh 2015 Jan

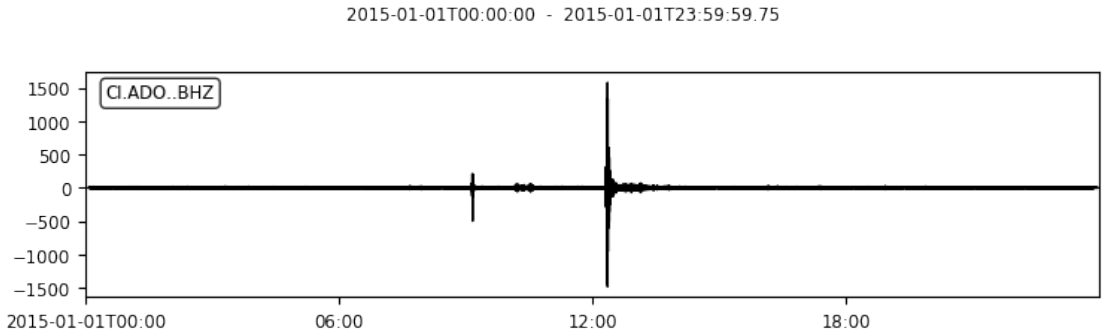
[16]:
```
# Quickly plot resulting cross-correlation

datadir='/uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/
 ↪Tutorials/Jan_Mar_TA_Download'
datayear='2015'
datamon='Jan'
datamn='1' #month number
dataday='1'

datasta='ADO'
datanw='CI'
datachan='BHZ'

src='ADO'
rec='FMP'
schan='Z'
rchan='Z'

#original SAC file
sacf=datadir+'/'+datayear+'/'+datamon+'/
 ↪'+datayear+'_'+datamn+'_'+dataday+'_0_0_0/' \
    +'ft_'+datasta+'.'+'LH'+datachan[-1]+'.SAC'

sacdatastream=read(sacf)

#EQft_ADO.LHZ.SAC (earthquake to remove)
eqf=datadir+'/'+datayear+'/'+datamon+'/
 ↪'+datayear+'_'+datamn+'_'+dataday+'_0_0_0/' \
     +'EQft_'+datasta+'.'+'LH'+datachan[-1]+'.SAC'
eqsacdatastream=read(eqf)

#whitened file
wtf=datadir+'/'+datayear+'/'+datamon+'/
 ↪'+datayear+'_'+datamn+'_'+dataday+'_0_0_0/' \
     +'wtZENft_'+datasta+'.'+'LH'+datachan[-1]+'.SAC'
wtsacdatastream=read(wtf)
```

```
print('Example SAC file: ',sacf)
sacdatastream.plot()

print('Example Earthquake file (removes eq from SAC file): ',eqf)
eqsacdatastream.plot()

print('Example Whiten file: ',wtf)
wtsacdatastream.plot()

print('\n\n\n') #add some space

#cross-correlation file
xcf=datadir+'/'+datayear+'/'+datamon+'/COR_ZEN/' \
     +src+'/COR_'+src+'_'+rec+'.SAC_'+schan+rchan
xcorrdatastream=read(xcf)

print('Example Cross-Correlation file: ',xcf)
xcorrdatastream.plot()
```

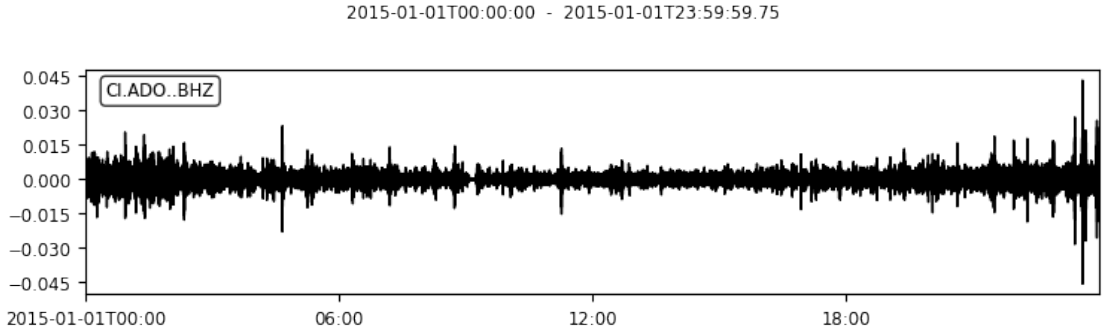
Example SAC file: /uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/2015/Jan/2015_1_1_0_0_0/ft_ADO.LHZ.SAC



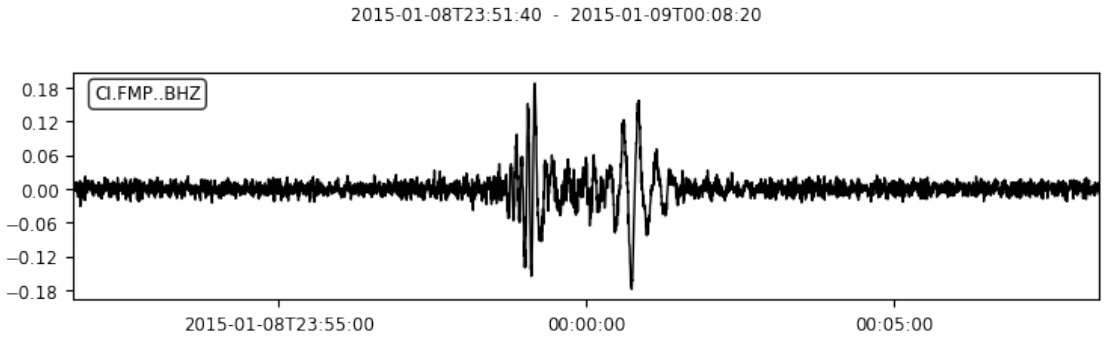
Example Earthquake file (removes eq from SAC file): /uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/2015/Jan/2015_1_1_0_0_0/EQft_ADO.LHZ.SAC



Example Whiten file: /uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/2015/Jan/2015_1_1_0_0_0/wtZENft_ADO.LHZ.SAC



Example Cross-Correlation file: /uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/2015/Jan/COR_ZEN/ADO/COR_ADO_FMP.SAC_ZZ



---

6 Note: Following are Contained in Datadir/run_stack_rotate_symm.csh

However, a step-by-step guide of each step within the run_stack_rotate_symm.csh script is included below

6.1 Run via:

```
cd Datadir
```

csh run_stack_rotate_symm.csh /uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/NewProcessing 2015

---

7 Stack

Note that this references Modules/lfstack (executable created by C-script in Modules/STACK)

usage:lfstack_from_iris station.lst path y1 y2

In Cross-Correlations headers, (should automatically be in the cross-correlation files if set correctly in the ft_Station.LH?.SAC files)

Needs USER8: number of days in stack

Needs User9, CMPAZ, Az, Baz

7.1 Run via:

1. **cd Datadir/Year (e.g., Jan_Mar_TA_Download/2015)**
2. **awk '{print \$1,\$2,\$3}' Jan/station.lst > station_sort.lst**

Create Station list containing Sta, Lon, Lat (assumes January contains same info as other months)

3. **/uufs/chpc.utah.edu/common/home/u1015716/SoCalTutorial_2020/Modules/ lfstack_Jan_Mar station_sort.lst /uufs/chpc.utah.edu/common/home/flin-group3/ emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/ 2015 2015 > output_stack.txt**

Run stacking script, will create STACK_ZNE dir

*Note that to run stack through the whole year, do (instead of step 3 above):

/uufs/chpc.utah.edu/common/home/u1015716/SoCalTutorial_2020/Modules/lfstack station_sort.lst /uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/ Tutorials/Jan_Mar_TA_Download/ 2015 2015 > output_stack.txt

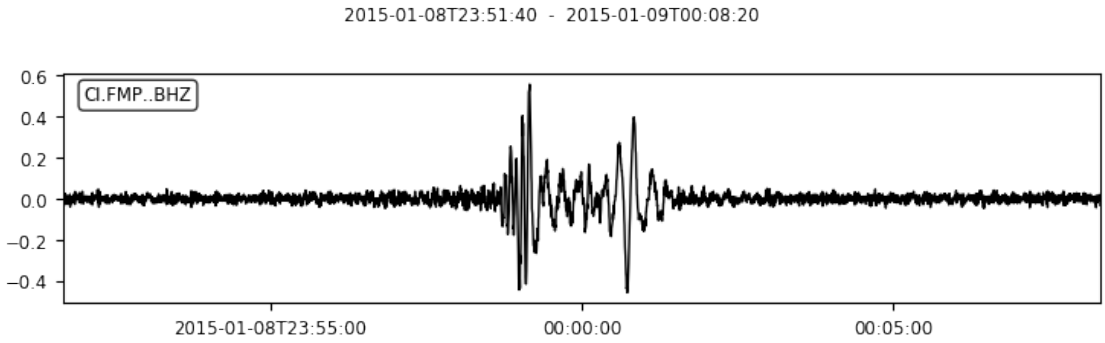
7.2 Output:

STACK_ZNE dir, containing stacked cross-correlations (formatted similarly to individual Mon/COR_ZEN)

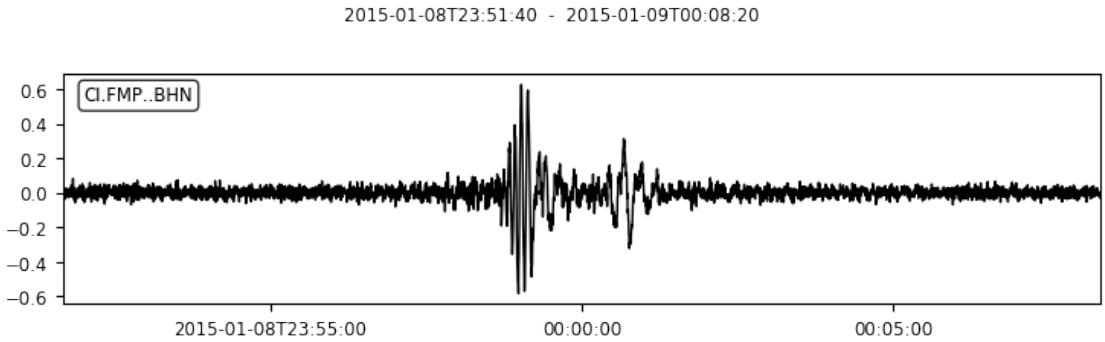
[17]:
```
# Plot a stacked SAC file
# Quickly plot resulting cross-correlation

datadir='/uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/
 ↪Tutorials/Jan_Mar_TA_Download'
datayear='2015'

src='ADO'
rec='FMP'
schan='Z'

rchan='Z'
rchan2='N'
rchan3='E'

#cross-correlation file
stackxcf=datadir+'/'+datayear+'/STACK_ZNE/' \
     +src+'/COR_'+src+'_'+rec+'.SAC_'+schan+rchan
stackxcorrdatastream=read(stackxcf)

print('Example Stack Cross-Correlation file: ',stackxcf)
stackxcorrdatastream.plot()



stackxcf2=datadir+'/'+datayear+'/STACK_ZNE/' \
     +src+'/COR_'+src+'_'+rec+'.SAC_'+schan+rchan2
stackxcorrdatastream2=read(stackxcf2)

print('Another example Stack Cross-Correlation file: ',stackxcf2)
stackxcorrdatastream2.plot()



stackxcf3=datadir+'/'+datayear+'/STACK_ZNE/' \
     +src+'/COR_'+src+'_'+rec+'.SAC_'+schan+rchan3
stackxcorrdatastream3=read(stackxcf3)

print('Another example Stack Cross-Correlation file: ',stackxcf3)
```

```
stackxcorrdatastream3.plot()
```

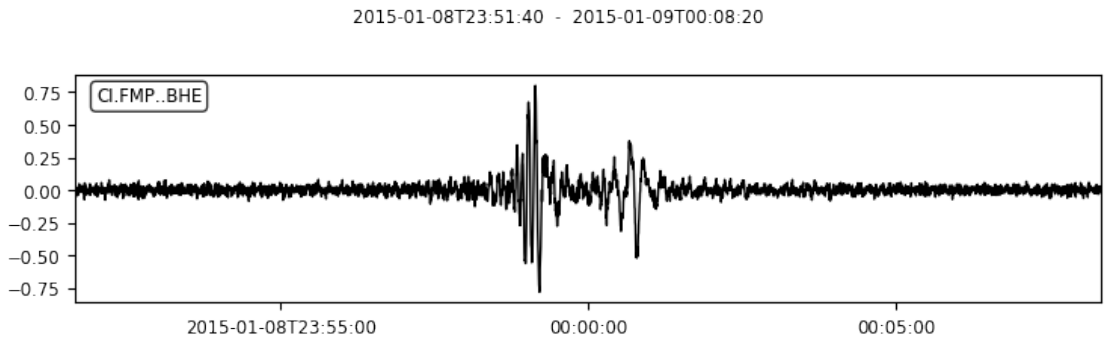
Example Stack Cross-Correlation file: /uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/2015/STACK_ZNE/ADO/COR_ADO_FMP.SAC_ZZ



Another example Stack Cross-Correlation file: /uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/2015/STACK_ZNE/ADO/COR_ADO_FMP.SAC_ZN



Another example Stack Cross-Correlation file: /uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/2015/STACK_ZNE/ADO/COR_ADO_FMP.SAC_ZE



8 Rotate

Note that this references Modules/lfrotate (executable created by C-script in Modules/ROTATE)

This script will rotate and apply azimuthal corrections for source-receiver pairs, if provided in the cross-correlation headers *This will automatically be included by the cross-correlation script if in the individual ft*SAC files*

To run lfrotate:

```
lfrotate stalist path
```

Note stalist should be 1-col file with station names

8.1 Run via:

1. **cd Datadir/Year/STACK_ZNE (eg., cd Jan_Mar_TA_Download/2015/STACK_ZNE)**
2. **awk '{print \$1}' ../station_sort.lst > sta.txt**

Create a station list with 1 column of station names

3. **/uufs/chpc.utah.edu/common/home/u1015716/SoCalTutorial_2020/Modules/lfrotate sta.txt ./**

Run rotate script, will create radial and transverse component information

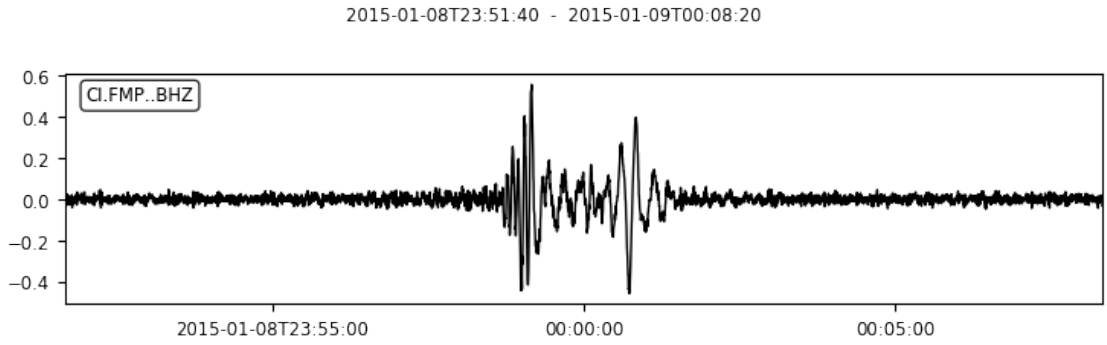
8.2 Output:

Transverse and Radial components (RR, RT, RZ, TR, TT, TZ, ZR, ZT)

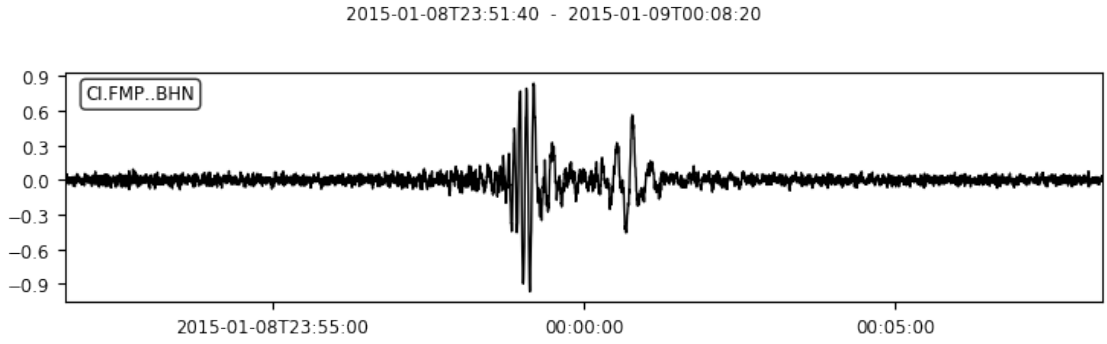
[18]:
```
# Plot rotated stacked SAC files

datadir='/uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/
 ↪Tutorials/Jan_Mar_TA_Download'
datayear='2015'
```

```
src='ADO'
rec='FMP'
schan='Z'

rchan='Z'
rchan2='R'
rchan3='T'

#cross-correlation file
stackxcf=datadir+'/'+datayear+'/STACK_ZNE/' \
     +src+'/COR_'+src+'_'+rec+'.SAC_'+schan+rchan
stackxcorrdatastream=read(stackxcf)

print('Example Stack Cross-Correlation file: ',stackxcf)
stackxcorrdatastream.plot()



stackxcf2=datadir+'/'+datayear+'/STACK_ZNE/' \
     +src+'/COR_'+src+'_'+rec+'.SAC_ro_'+schan+rchan2
stackxcorrdatastream2=read(stackxcf2)

print('Example Rotated Stack Cross-Correlation file: ',stackxcf2)
stackxcorrdatastream2.plot()



stackxcf3=datadir+'/'+datayear+'/STACK_ZNE/' \
     +src+'/COR_'+src+'_'+rec+'.SAC_ro_'+schan+rchan3
stackxcorrdatastream3=read(stackxcf3)

print('Example Rotated Stack Cross-Correlation file: ',stackxcf3)
stackxcorrdatastream3.plot()
```

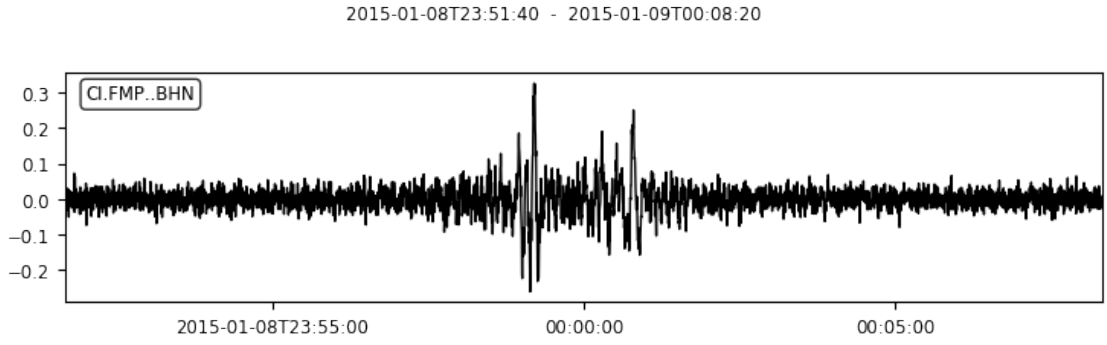
```
Example Stack Cross-Correlation file:  /uufs/chpc.utah.edu/common/home/flin-grou
p3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/2015/STACK_ZNE/ADO/COR_ADO
_FMP.SAC_ZZ
```



Example Rotated Stack Cross-Correlation file:
/uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/2015/STACK_ZNE/ADO/COR_ADO_FMP.SAC_ro_ZR



Example Rotated Stack Cross-Correlation file:
/uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/2015/STACK_ZNE/ADO/COR_ADO_FMP.SAC_ro_ZT



9 Symmetric

Note that this references Modules/SYMM/symm.sh

9.1 Edits:

1. Change the assigned output length (seconds) of the symmetric files, if you want (currently set to -300s to 300s) **Line51 (symm.sh): cut -300 300**

9.2 Run via:

/bin/sh symm.sh path stationlst (similar to lfrotate, stationlist should be 1-col file with station names)

1. Go to STACK_ZNE dir

   cd /uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/ Tutorials/Jan_Mar_TA_Download/2015/STACK_ZNE

2. Run symm.sh

   sh /uufs/chpc.utah.edu/common/home/u1015716/SoCalTutorial_2020/Modules/SYMM/symm.sh ./ sta.txt

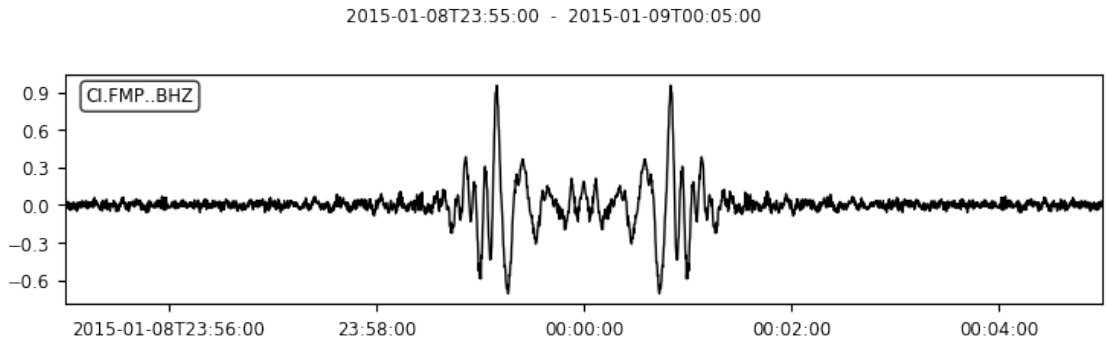
9.3 Output:

Symmetric of all station combinations (vertical, radial and transverse combinations)

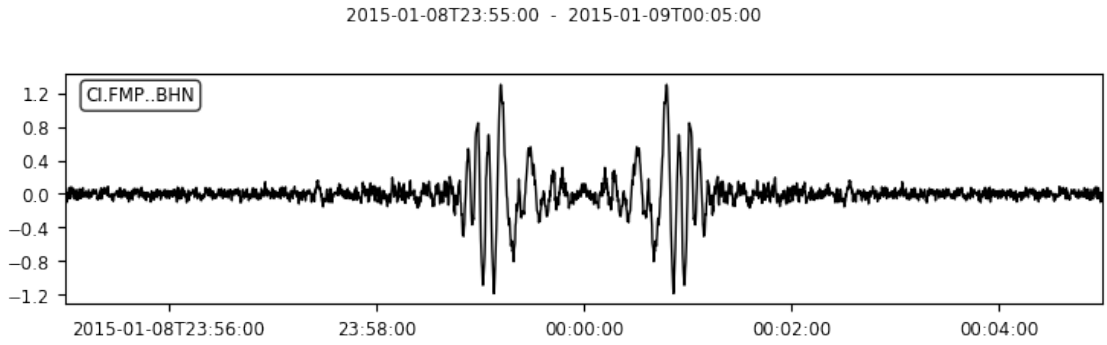
[19]:
```
# Plot rotated stacked SAC files

datadir='/uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/
 ↪Tutorials/Jan_Mar_TA_Download'
datayear='2015'

src='ADO'
rec='FMP'
schan='Z'

rchan='Z'
rchan2='R'
rchan3='T'

#cross-correlation file (ZZ)
stackxcf=datadir+'/'+datayear+'/STACK_ZNE/' \
     +src+'/COR_'+src+'_'+rec+'.SAC_'+schan+rchan+'_s'

#cross-correlation file (not-Z Z)
#stackxcf=datadir+'/'+datayear+'/STACK_ZNE/' \
```

```
#     +src+'/COR_'+src+'_'+rec+'.SAC_ro_'+schan+rchan+'_s'

stackxcorrdatastream=read(stackxcf)

print('Example Stack, Symmetric Cross-Correlation file: ',stackxcf)
stackxcorrdatastream.plot()



stackxcf2=datadir+'/'+datayear+'/STACK_ZNE/' \
     +src+'/COR_'+src+'_'+rec+'.SAC_ro_'+schan+rchan2+'_s'
stackxcorrdatastream2=read(stackxcf2)

print('Example Symmetric, Rotated Stack Cross-Correlation file: ',stackxcf2)
stackxcorrdatastream2.plot()



stackxcf3=datadir+'/'+datayear+'/STACK_ZNE/' \
     +src+'/COR_'+src+'_'+rec+'.SAC_ro_'+schan+rchan3+'_s'
stackxcorrdatastream3=read(stackxcf3)

print('Example Symmetric, Rotated Stack Cross-Correlation file: ',stackxcf3)
stackxcorrdatastream3.plot()
```

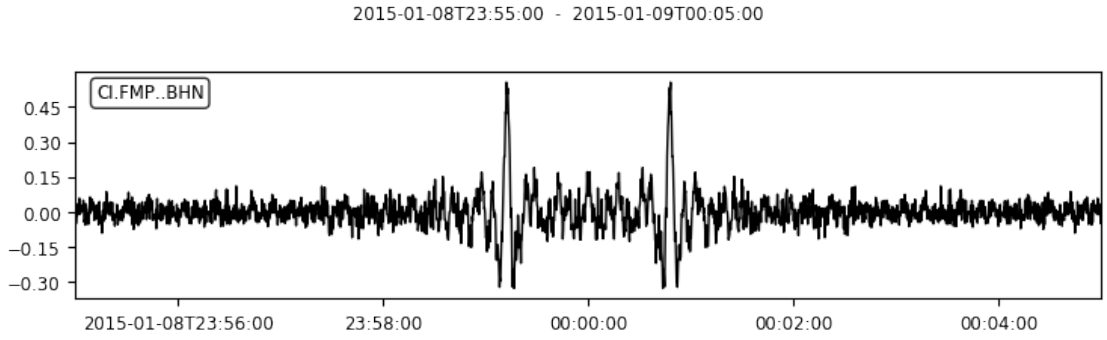
Example Stack, Symmetric Cross-Correlation file:
/uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/2015/STACK_ZNE/ADO/COR_ADO_FMP.SAC_ZZ_s



Example Symmetric, Rotated Stack Cross-Correlation file:
/uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/2015/STACK_ZNE/ADO/COR_ADO_FMP.SAC_ro_ZR_s



Example Symmetric, Rotated Stack Cross-Correlation file:
/uufs/chpc.utah.edu/common/home/flin-group3/emberg/USC_data_2015/Tutorials/Jan_Mar_TA_Download/2015/STACK_ZNE/ADO/COR_ADO_FMP.SAC_ro_ZT_s



[ ]: