

Stack_Example

August 31, 2015

0.1 Stacking Waveforms Examples

The following notebook contains examples for using the stack.py toolbox for stacking raw and band-pass filtered seismic waveforms. Currently the script can only operate with MSED formats, but additional support for other formats such as: SAC, SEED and SUDS will likely be added in the future. The user specifies the input path to the raw waveform. This waveform should have more than one trace, e.g. BHZ, BHE and BHN in one. A Stack object is created with this input, and the output is specified with one of the containing functions.

Each trace in the stream is stacked in a specified way. Three options are currently available: Linear Stack, Phase Stack and Phase-Weighted Stack.

```
In [1]: from tools.stack import Stack
        from obspy import read
        %matplotlib inline
```

The Stack object requires the MSED path as the only necessary input. Other options include 'filter_waveform' which is default to False, and 'band_lims' which is defaulted to [1,10]. So if you wish to band-pass filter the waveform before stacking, set 'filter_waveform' to true and specify which frequency range you wish to filter; 'band_lims' contains [min. frequency, max. frequency] for the band-pass filter.

```
In [2]: # here is a list of all of the functions and variables that the Stack class contains
        help(Stack)
```

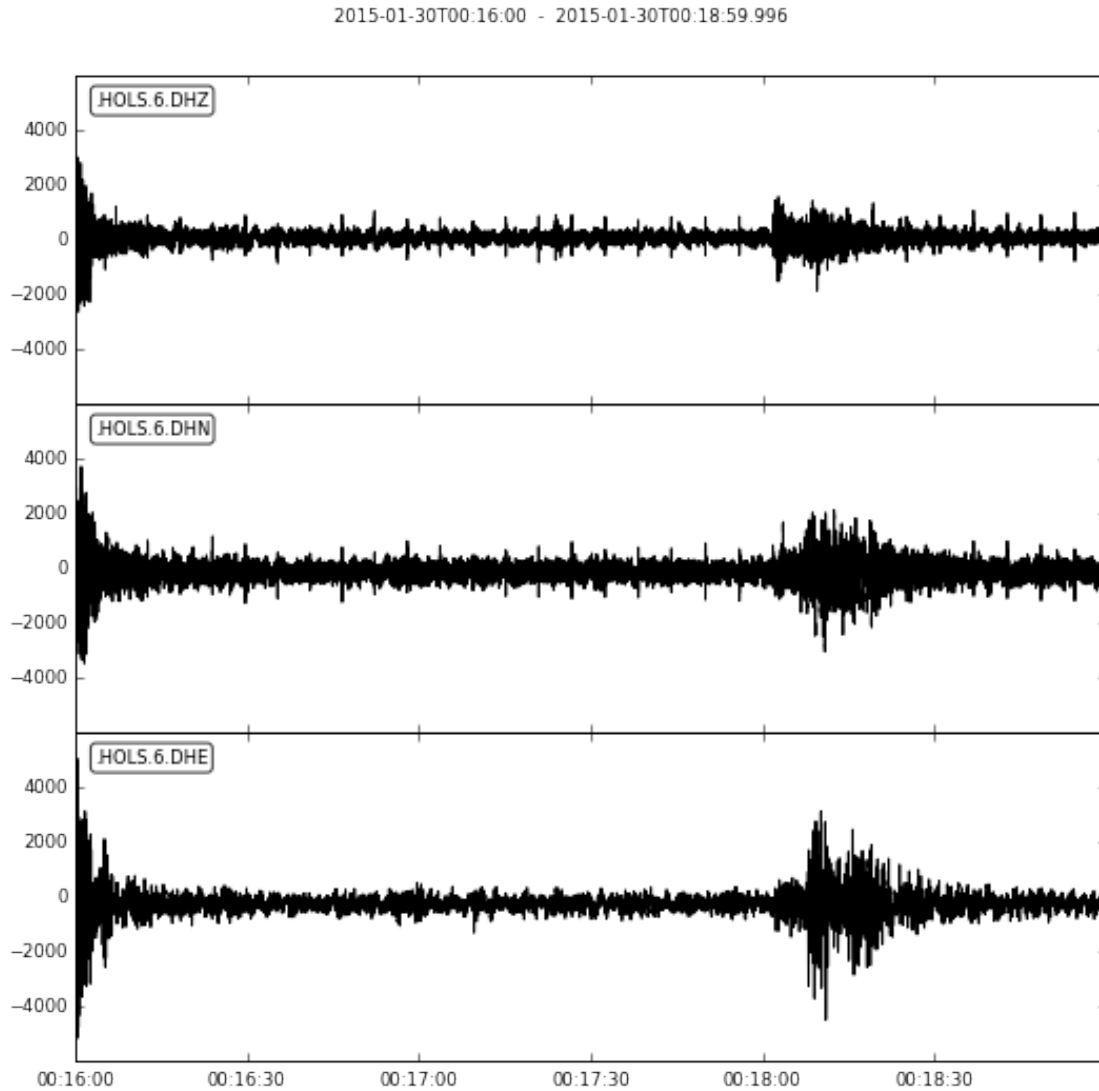
Help on class Stack in module tools.stack:

```
class Stack
|   The following class contains necessary functions to perform seismic
|   waveform stacking. The stacking improves Signal-to-Noise ratio (SNR)
|   and allows for better picks when doing event location.
|
|   Methods defined here:
|
|   __init__(self, st_path, filter_waveform=False, band_lims=[1, 10])
|
|   datetime_list(self, st=None)
|       Function that outputs a list of N datetime objects between the
|       start and end times of a given stream. N is dictated by the length
|       of the st[0].data list.
|
|   import_stream(self, st_path=None)
|       Function that uses the read function from obspy.core to import t
|       the given Stream object from miniseed.
|
|   lin_stack(self, st=None, norm=True)
```

```

|     The following function takes an obspy stream object and outputs
|     the linear stack of all traces within the stream. This is only useful
|     if all traces are various channels from the same station, OR are from
|     the same cross-correlation station pairs for ambient studies.
|
| phase_stack(self, st=None, norm=True)
|     The following function takes an obspy stream object and outputs
|     the phase stack of all traces within the stream. This is only useful
|     if all traces are various channels from the same station, OR are from
|     the same cross-correlation station pairs for ambient studies.
|
| plot_lin_stack(self, LS=None, show=True, save=False)
|     Function to plot the linear stack of two or more traces within
|     the input Stream object for the class: Stack. Show is the default,
|     if save is True, then show is automatically set to False
|
| plot_phase_stack(self, PS=None, show=True, save=False)
|     Function to plot the phase stack of two or more traces within
|     the input Stream object for the class: Stack. Show is the default,
|     if save is True, then show is automatically set to False
|
| plot_pw_stack(self, PWS=None, show=True, save=False)
|     Function to plot the phase-weighted stack of two or more traces within
|     the input Stream object for the class: Stack. Show is the default,
|     if save is True, then show is automatically set to False
|
| pw_stack(self, st=None, LS=None, PS=None, norm=True, sharp_v=2)
|     The following function takes an obspy stream object and outputs
|     the phase-weighted stack of all traces within the stream.
|     This is only useful if all traces are various channels from the
|     same station, OR are from the same cross-correlation station pairs
|     for ambient studies. Default to the second power. Even powers work.
|
In [3]: # set the path to the desired waveform, the example HOLS.mseed is provided.
        example_path = 'tools/examples/HOLS.mseed'
        # plot the original waveform to show what form it's in.
        st = read(example_path); st.plot()

```



To create a list of the linearly stacked waveforms, run `STACK.lin_stack` and the list is stored in `STACK.LS`

```
In [4]: # create the stack object
        STACK = Stack(example_path, filter_waveform=True, band_lims=[1,10])

        STACK.lin_stack()
        print 'List of Linearly Stacked Seismic Waveforms: ', STACK.LS
```

List of Linearly Stacked Seismic Waveforms: `[-0.04349118 -0.05880078 -0.07462341 ..., 0.00222808 0.00019075]`

To create a list of the phase stacked waveforms, run `STACK.phase_stack` and the list is stored in `STACK.PS` using the method from Schimmel and Paulssen (1997).

```
In [5]: STACK.phase_stack()
        print 'List of Phase Stacked Seismic Waveforms: ', STACK.PS
```

List of Phase Stacked Seismic Waveforms: `[2.04107800e-17 3.21125447e-03 2.78757337e-02 ..., 3.33333330e-01 3.33333333e-01]`

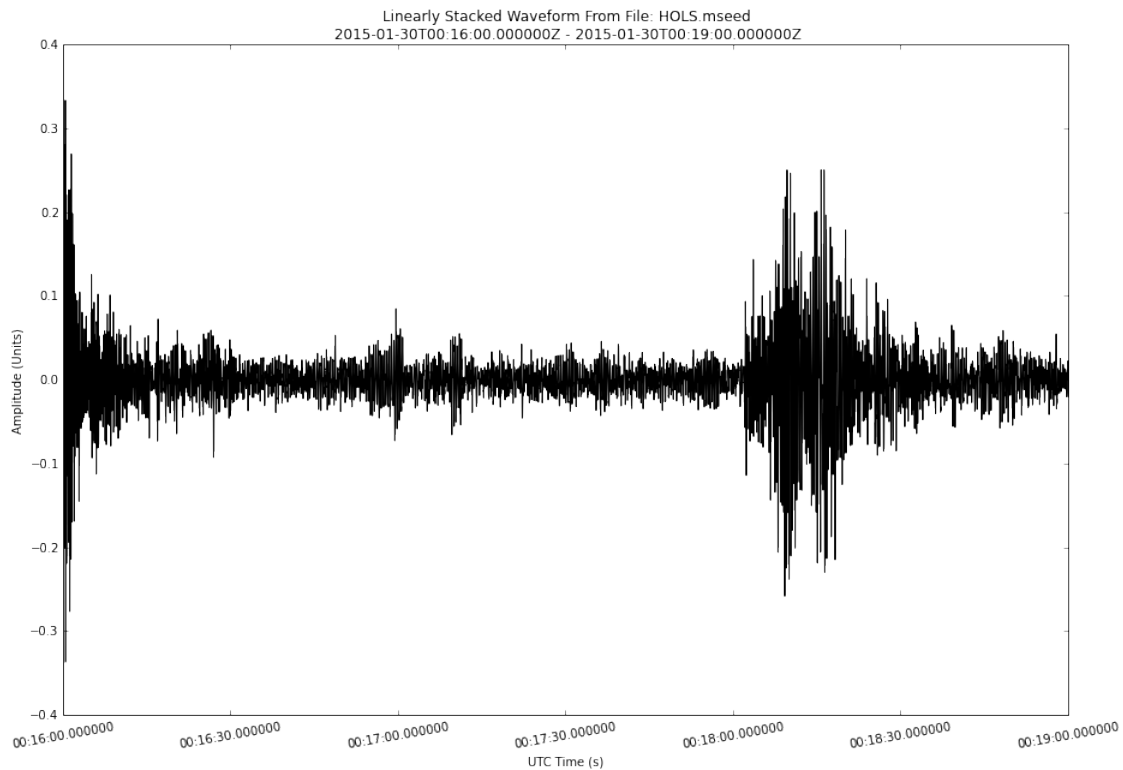
To create a list of the phase-weighted stacked waveforms, run `STACK.pw_stack` and the list is stored in `STACK.PWS` using the method from Schimmel and Paulssen (1997).

```
In [6]: STACK.pw_stack()
        print 'List of Phase-Weighted Stacked Seismic Waveforms: ', STACK.PWS
```

```
List of Phase-Weighted Stacked Seismic Waveforms: [ -3.18359273e-06 -1.04574116e-05 -6.98780471e-04
 1.17912270e-03  2.51060506e-04]
```

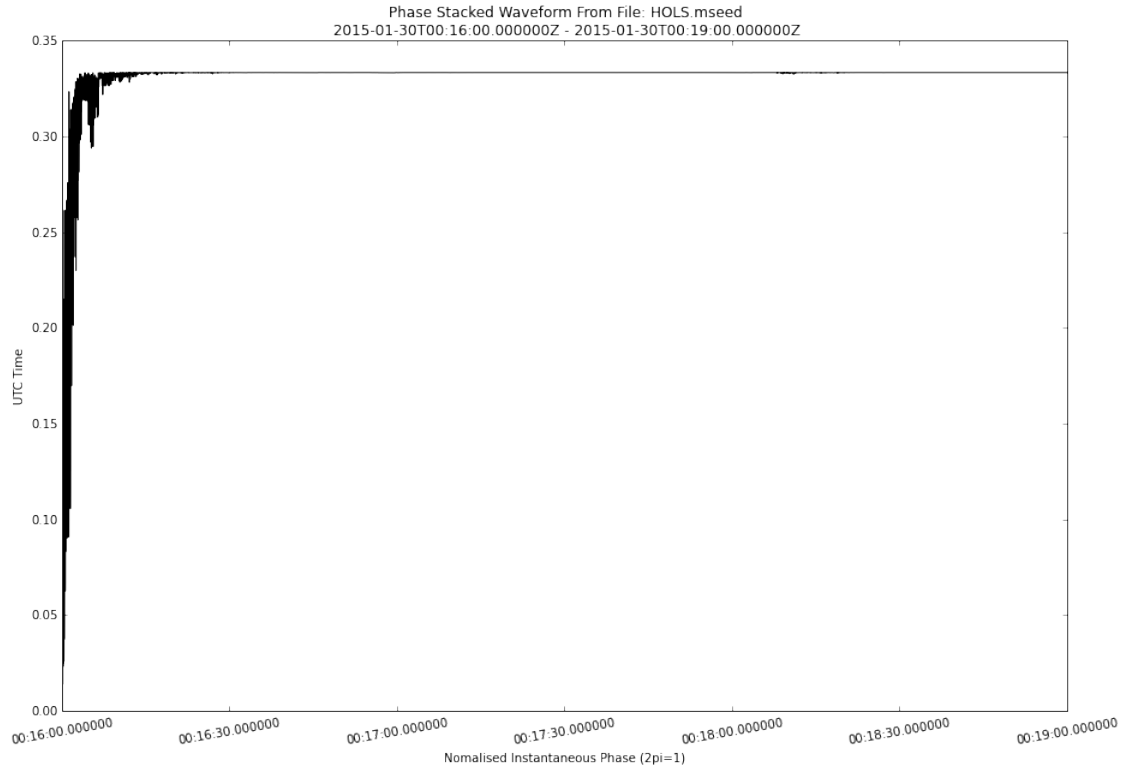
To create a plot of the linearly stacked waveforms, run `STACK.plot_lin_stack`. Options are to save or show the waveforms, as well as to plot from the initialised stack, or another `STACK.LS` that has been created previously. If `save=True`, the waveform will be saved to the same location as the example, but changed the named to `?.lin_stack.jpg` where `?` is the basename without the extension of the original MSED input file.

```
In [7]: STACK.plot_lin_stack(LS=None, show=True, save=False)
```



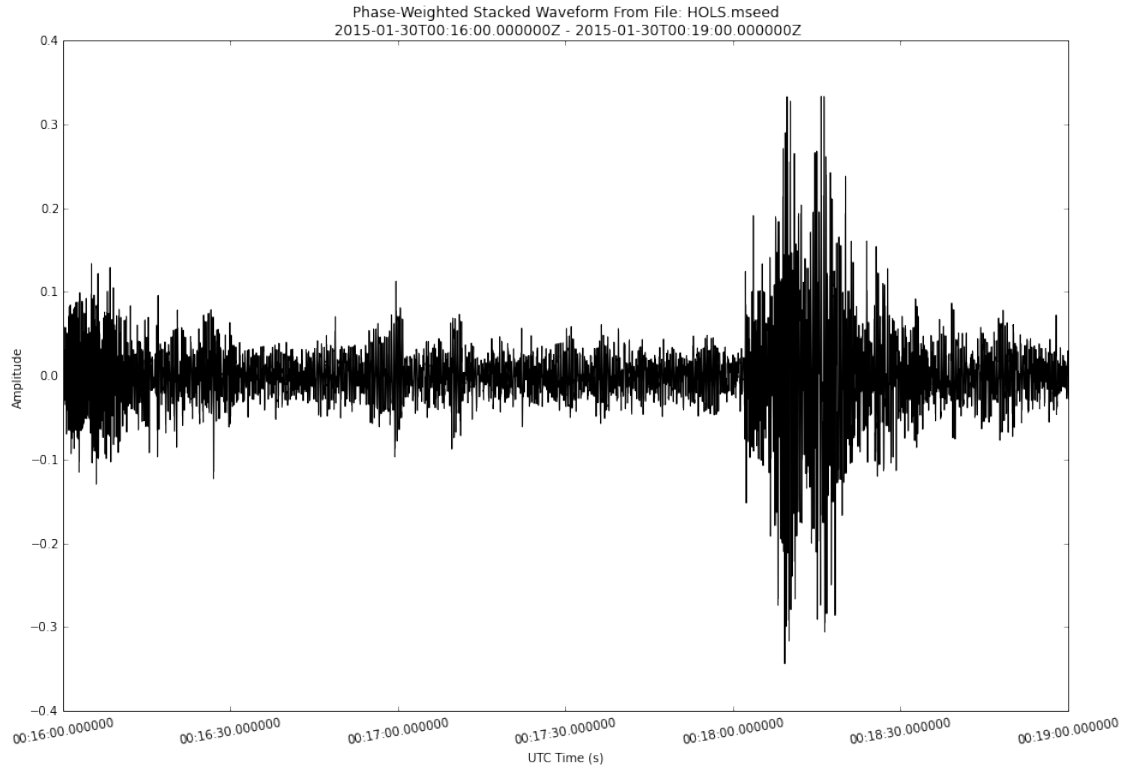
To create a plot of the phase stacked waveforms, run `STACK.plot_phase_stack`. Options are to save or show the waveforms, as well as to plot from the initialised stack, or another `STACK.PS` that has been created previously. If `save=True`, the waveform will be saved to the same location as the example, but changed the named to `?.phase_stack.jpg` where `?` is the basename without the extension of the original MSED input file.

```
In [8]: STACK.plot_phase_stack(PS=None, show=True, save=False)
```



To create a plot of the phase-weighted stacked waveforms, run `STACK.plot_pw_stack`. Options are to save or show the waveforms, as well as to plot from the initialised stack, or another `STACK.PWS` that has been created previously. If `save=True`, the the waveform will be saved to the same location as the example, but changed the named to `?.pw_stack.jpg` where `?` is the basename without the extension of the original MSEED input file.

```
In [9]: STACK.plot_pw_stack(PWS=None, show=True, save=False)
```



0.2 References

Schimmel, M., & Paulssen, H. (1997). Noise reduction and detection of weak, coherent signals through phase-weighted stacks. *Geophysical Journal International*, 130(2), 497-505. doi:10.1111/j.1365-246x.1997.tb05664.x