

UTCDateTime

Documentation: [UTCDateTime](#)

```
In [6]: from obspy.core import UTCDateTime
t = UTCDateTime()
t1 = UTCDateTime("2014-03-14T10:15:00+01:00")
t2 = UTCDateTime(2012, 9, 7, 12, 15, 0)
t3 = UTCDateTime(1347020100.0)
print t.year, t.julday, t.timestamp, t.weekday
print t+86400
```

```
2014 69 1394457212.41 0
2014-03-11T13:13:32.412180Z
```

Načítání seismogramů

```
In [7]: from obspy.core import read
st = read('http://examples.obspy.org/RJOB_061005_072159.ehz.new')
print st
tr = st[0] # assign first and only trace to new variable
print tr
```

```
1 Trace(s) in Stream:
.RJOB..Z | 2005-10-06T07:21:59.849998Z - 2005-10-06T07:24:59.844998Z | 200.0 Hz, 36000 samples
.RJOB..Z | 2005-10-06T07:21:59.849998Z - 2005-10-06T07:24:59.844998Z | 200.0 Hz, 36000 samples
```

```
In [8]: print tr.stats
tr.stats.station
tr.stats.gse2.datatype
```

```
network:
  station: RJOB
location:
  channel: Z
starttime: 2005-10-06T07:21:59.849998Z
endtime: 2005-10-06T07:24:59.844998Z
sampling_rate: 200.0
  delta: 0.005
  npts: 36000
  calib: 0.0948999971151
  _format: GSE2
  gse2: AttribDict({'instype': 'CM6', 'datatype': 'CM6', 'hang': -1.0, 'auxid': 'RJOB', 'vang': -1.0, 'calper':
```

```
1.0})
```

```
Out[8]: 'CM6'
```

```
In [10]: print tr.data  
print tr.data[0:3]  
print len(tr)
```

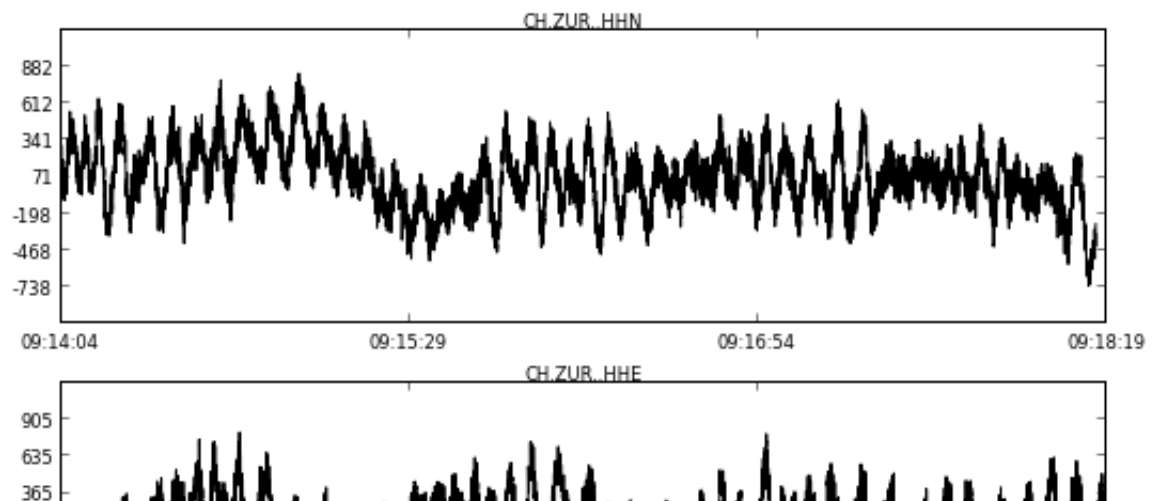
```
[-38  12  -4 ..., -14  -3  -9]  
[-38  12  -4]  
36000
```

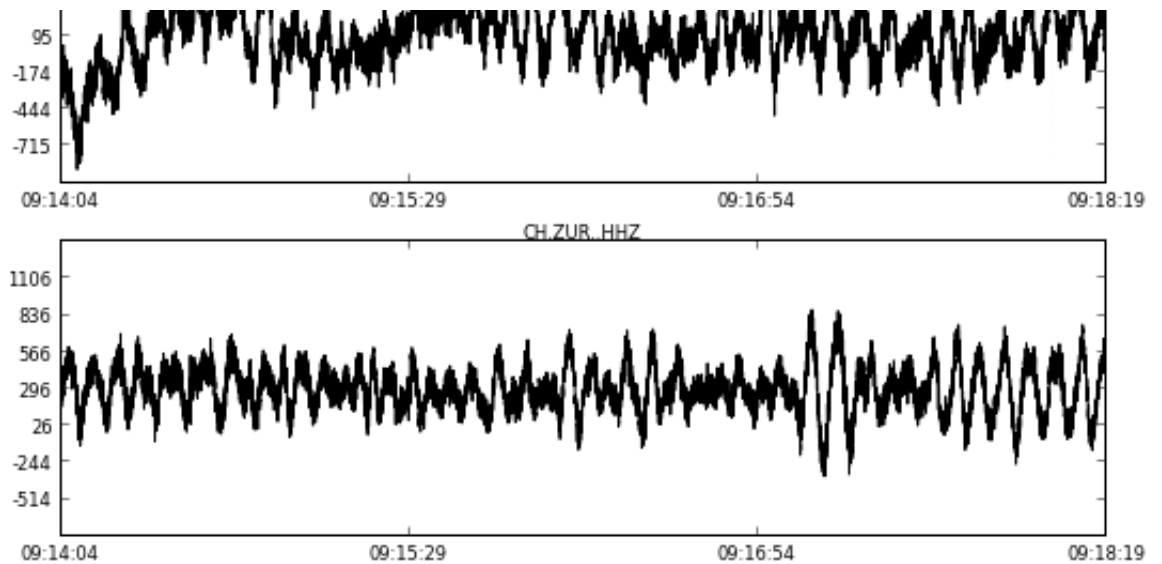
Arclink

```
In [1]: from obspy.core import UTCDateTime  
from obspy.arclink import Client  
  
t = UTCDateTime()  
try:  
    client = Client(host="arclink.ethz.ch", user="jiri.vackar@s  
    st = client.getWaveform('CH', 'ZUR', '', 'HH*', t-300, t)  
except:  
    print 'stahovani selhalo'  
else:  
    print st  
    st.plot()
```

```
3 Trace(s) in Stream:  
CH.ZUR..HHN | 2014-03-11T09:14:04.675067Z - 2014-03-  
11T09:18:16.333400Z | 120.0 Hz, 30200 samples  
CH.ZUR..HHE | 2014-03-11T09:14:04.675000Z - 2014-03-  
11T09:18:18.750000Z | 120.0 Hz, 30490 samples  
CH.ZUR..HHZ | 2014-03-11T09:14:04.675033Z - 2014-03-  
11T09:18:19.108367Z | 120.0 Hz, 30533 samples
```

```
2014-03-11T09:14:04Z - 2014-03-11T09:18:19Z
```





Filtrování

The following script shows how to filter a seismogram. The example uses a zero-phase-shift low-pass filter with a corner frequency of 1 Hz using 2 corners. This is done in two runs forward and backward, so we end up with 4 corners de facto.

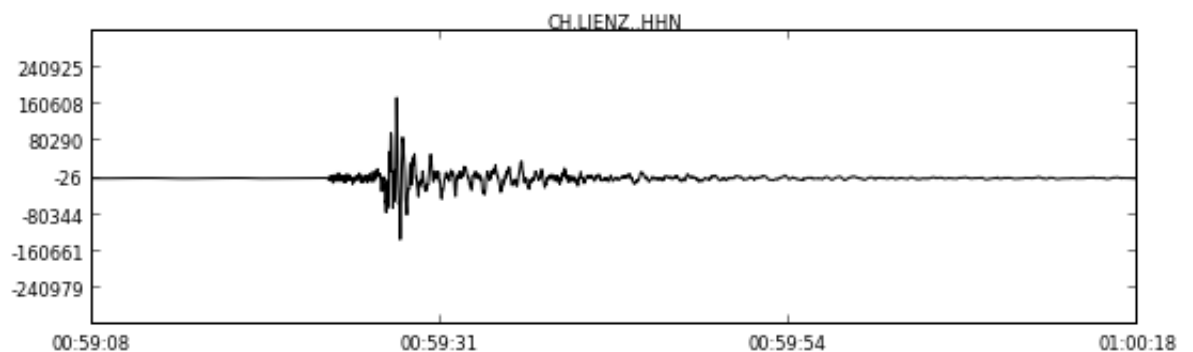
The available filters are:

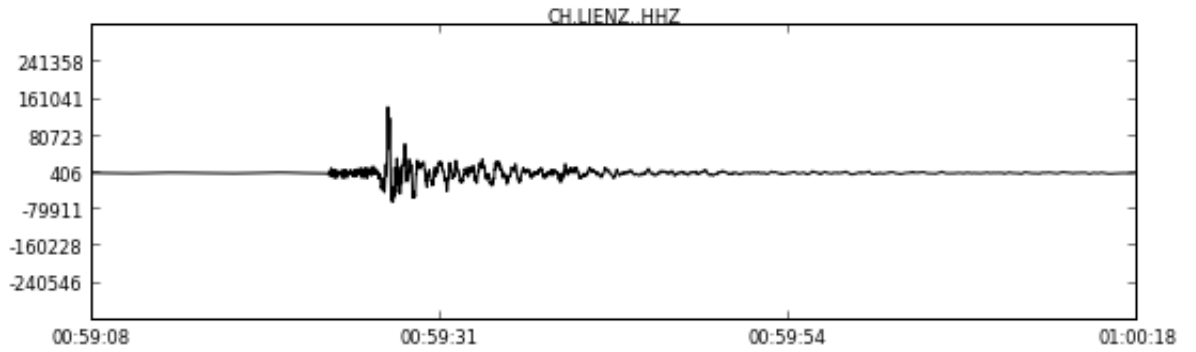
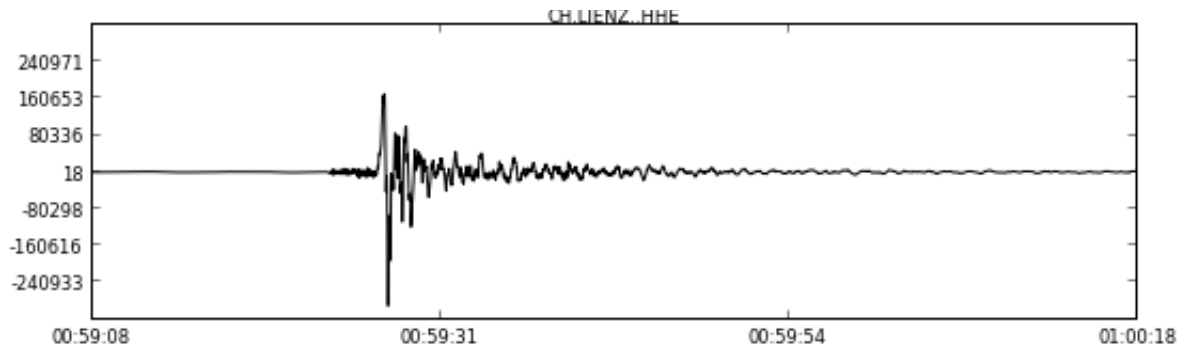
- bandpass
- bandstop
- lowpass
- highpass

Dokumentace: [obspy.core.stream.Stream.filter](#)

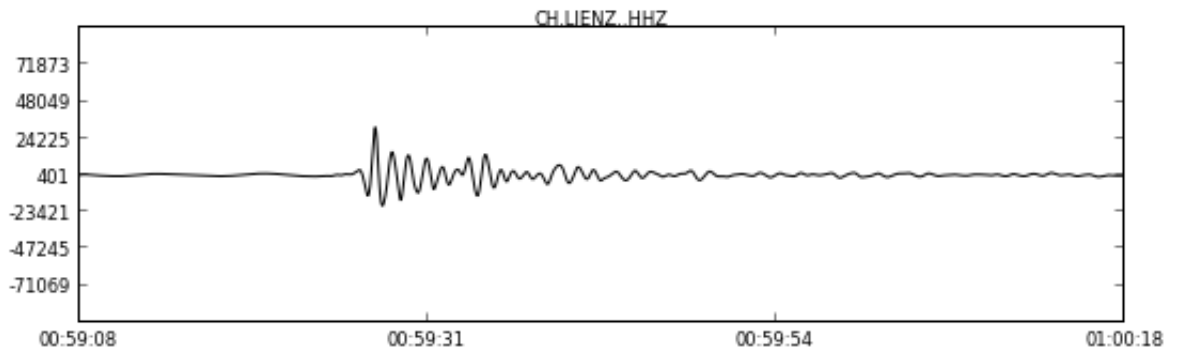
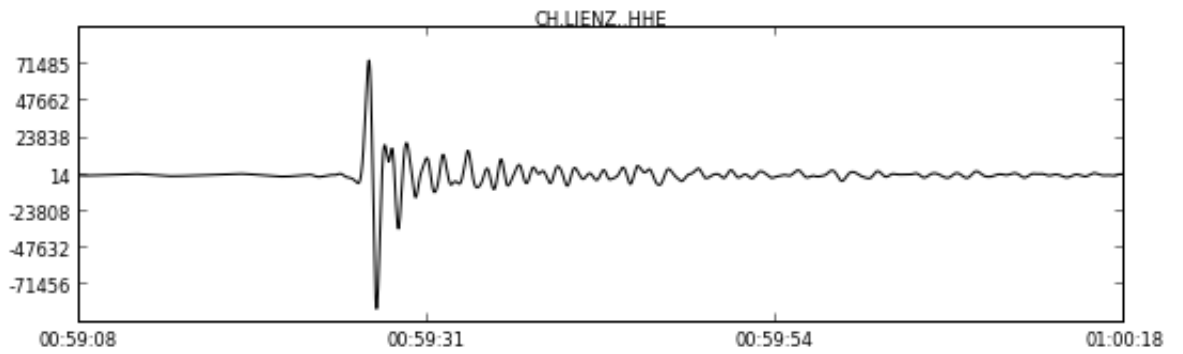
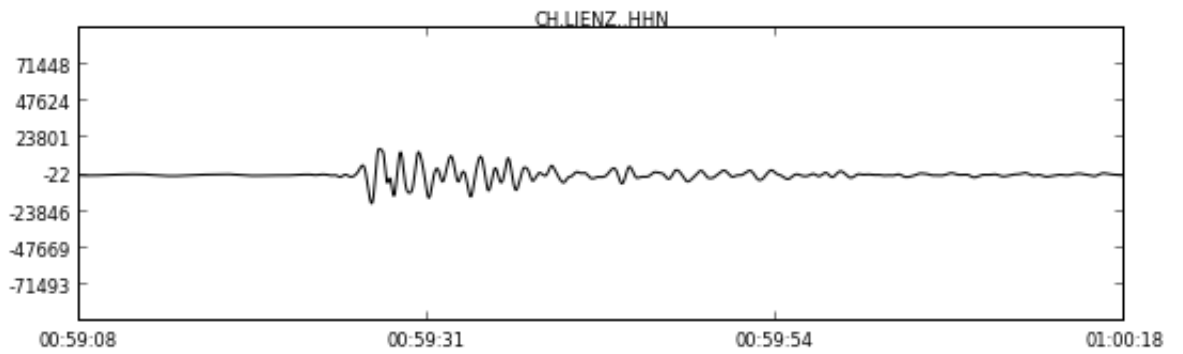
```
In [19]: t = UTCDateTime("2013-12-12 0:59:18")
st = client.getWaveform('CH', 'LIENZ', '', 'HH*', t-10, t+60)
st.plot()
st_filt = st.copy()
st_filt.filter('lowpass', freq=1.0, corners=2, zerophase=True)
st_filt.plot()
```

2013-12-12T00:59:08Z - 2013-12-12T01:00:18Z





2013-12-12T00:59:08Z - 2013-12-12T01:00:18Z



Obálky seismogramů a vykreslování pomocí matplotlib

Dokumentace: [Seismogram envelopes](#) [Matplotlib: plot](#)

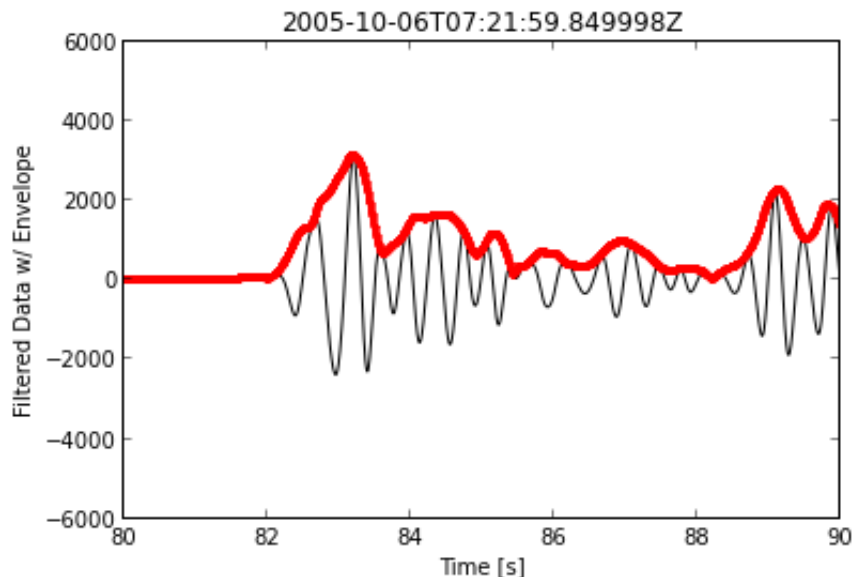
```
In [33]: import numpy as np
import matplotlib.pyplot as plt
from obspy.core import read
import obspy.signal

st = read("http://examples.obspy.org/RJOB_061005_072159.ehz.new")
npts = st[0].stats.npts
samprate = st[0].stats.sampling_rate

st.filter('bandpass', freqmin=1, freqmax=3, corners=2, zerophase=True)

# Envelope of filtered data
data_envelope = obspy.signal.filter.envelope(st_filt[0].data)

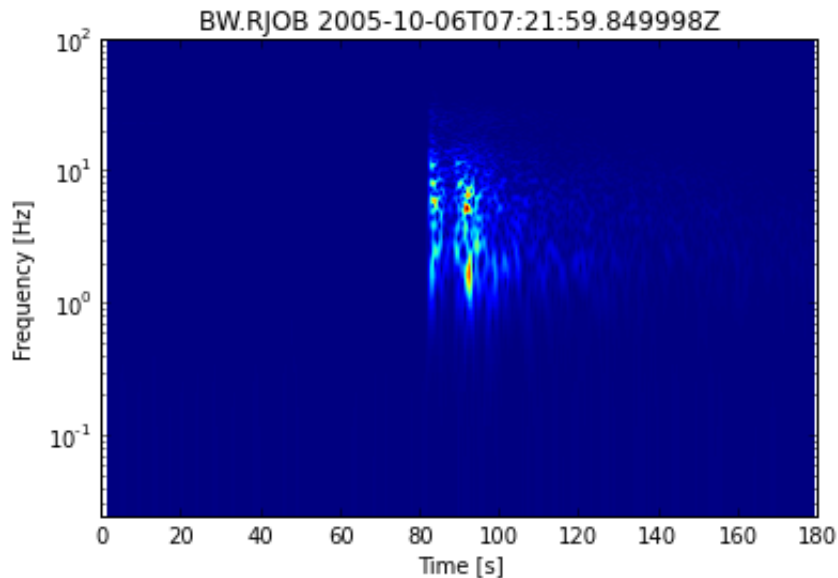
# The plotting, plain matplotlib
t = np.arange(0, npts / samprate, 1 / samprate)
plt.plot(t, st[0].data, 'k')
plt.plot(t, data_envelope, 'r.')
plt.title(st[0].stats.starttime)
plt.ylabel('Filtered Data w/ Envelope')
plt.xlabel('Time [s]')
plt.xlim(80, 90)
plt.show()
```



Spektrogram

```
In [34]: from obspy.core import read
```

```
st = read("http://examples.obspy.org/RJOB_061005_072159.ehz.new")
st.spectrogram(log=True, title='BW.RJOB ' + str(st[0].stats.station_id))
```



```
Out[34]: [None]
```

Nuly a póly, přenosová funkce přístroje

```
In [23]: from obspy.core import UTCDateTime
from obspy.arclink import Client
from obspy.signal import pazToFreqResp
import matplotlib.pyplot as plt

t = UTCDateTime("2013-12-12 10:34:21")
client = Client(host="arclink.ethz.ch", user="jiri.vackar@sed.eid.ethz.ch")
st = client.getWaveform('CH', 'VANNI', '', 'HHZ', t, t+10)
tr = st[0]
tr2 = tr.copy()

# stahneme nuly a poly pres ArcLink
paz = client.getPAZ('CH', 'VANNI', '', 'HHZ', t)

# nakreslime frekvencni odezvu
h, f = pazToFreqResp(paz['poles'], paz['zeros'], paz['gain'], 0.001)
plt.figure()
plt.loglog(f, abs(h))
plt.xlabel('Frequency [Hz]')
plt.ylabel('Amplitude')
plt.show()
```

```
# provedeme instrumentalni korekci
tr2.simulate(paz_remove=paz)

# vykreslime
t = np.arange(0, tr.stats.npts / tr.stats.sampling_rate, 1 / tr
plt.plot(t, tr.data, 'k')
plt2 = plt.twinx()
plt2.plot(t, tr2.data, 'r')
plt.show()
```

