1. Data Request

> firstly I used batch request
>
> 1. I search all the stations of GSN
>   then I get network code, CU, GT, IC, II, and IU.
> 2. Choose an event
>   I randomly choose an event from the USGS website. It is the M 5.1 - South Georgia Rise.
> 3. Data request
>   I use the BREQ_fast request to generate the format of data request email.
>   I only use the data of **II network**. Then I send an email to breq_fast@iris.washington.edu.
> 4. Data format converision (mseed >> sac)
>
> ```
> ls *.mseed | awk '{split($1,a,"."); print a[1] "." a[2], $1}' | awk '{printf -z0 "mseed2sac %s -m *%s.meta\n", $2,$1}' | sh
> ```
>
> 5. generate header file of SAC
>
> ```
> saclst kzdate kztime npts delta b e knetwk kstnm stla stlo stdp cmpaz cmpinc kcmpnm kstcmp f *.SAC > ../header.txt
> ```
>
> #### Unfortunately, I found that there is no **evdp, evla, evlo** in the header of SAC.
>
> Lastly, I chose another way to download waveform data by wilber3
>
> 1. change filename to net.sta.loc.chan.SAC
>
> ```
> ls *.SAC | awk '{split($1,a,"."); print $1, a[1]"."a[2]"."a[3]"."a[4]".SAC"}' | awk '{printf "mv %s %s\n",$1,$2}' | sh
> ```
>
> 2. synchronize the reference time
>
> ```
> sac>r *SAC
> sac>sync
> sac>wh
> ```
>
> 3. make header file of SAC
>
> ```
> saclst kzdate kztime npts delta b e kevnm evla evlo evdp gcarc dist az o khole knetwk kstnm stla stlo stdp cmpaz cmpinc kcmpnm kstcmp f *.SAC
> ```

2. Read Header and Data of SAC files using Obspy

In [9]:

```python
from obspy import read, UTCDateTime, read_inventory
st = read('../data/chile/*.SAC')
#take the first trace as an example
header = st[0].stats
data = st[0].data
print('------------------------------header of sac----------\n'
```

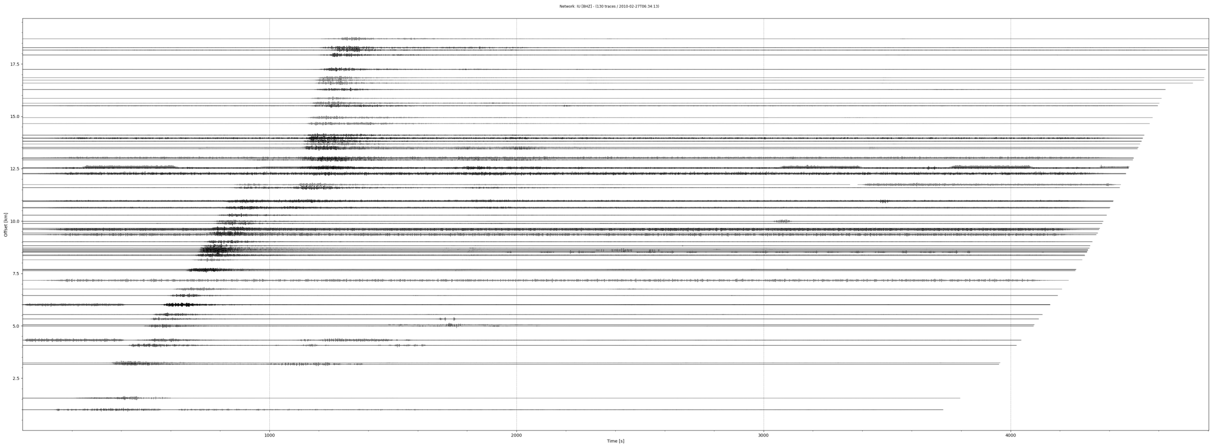
```
      ,header,
      '\n\n--------------------------data of sac-----------\n',data)
```

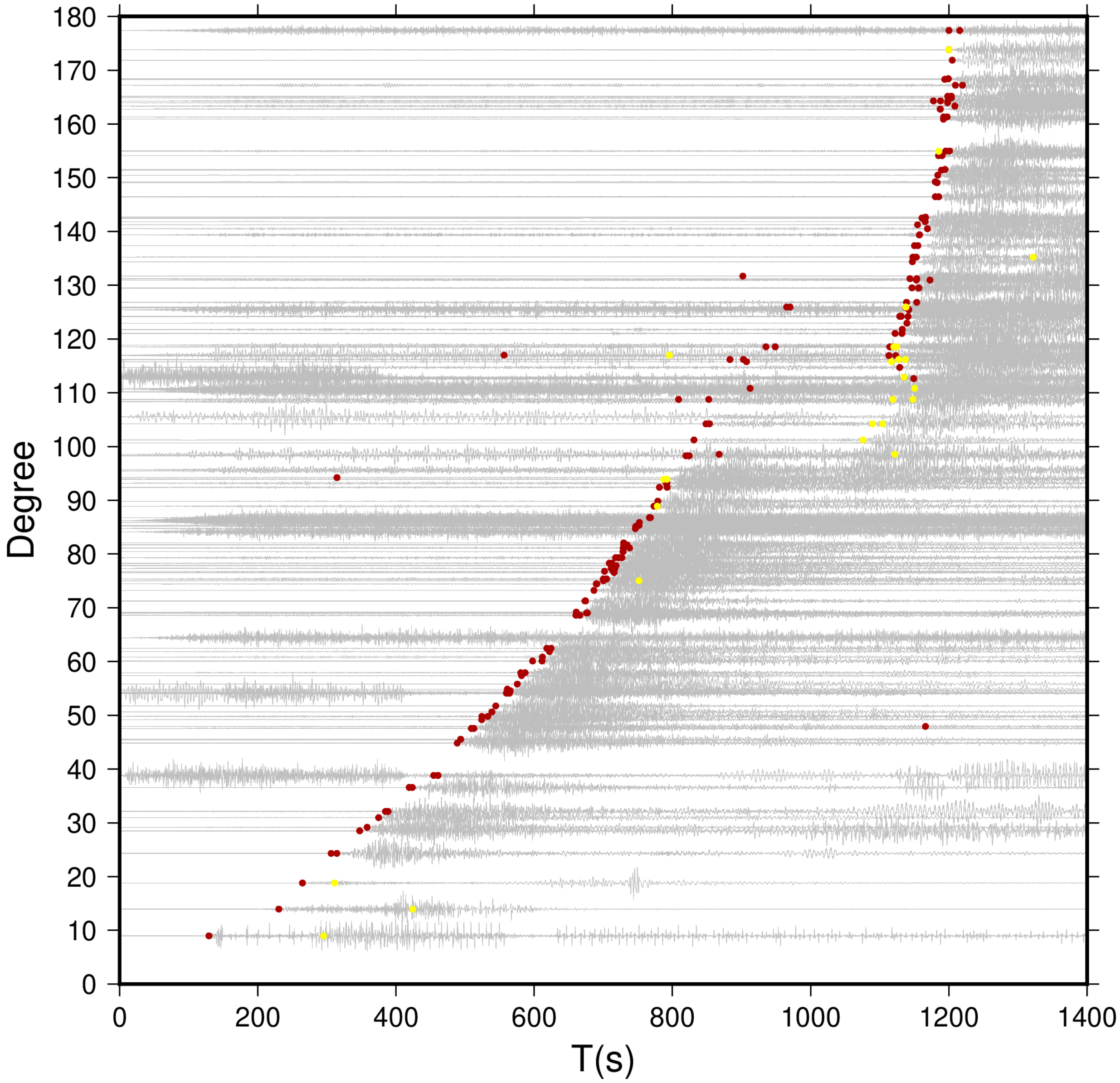
```
------------------------------header of sac----------
           network: AU
           station: MCQ
          location: 
           channel: BHE
         starttime: 2010-02-27T06:34:13.019500Z
           endtime: 2010-02-27T07:46:13.994500Z
     sampling_rate: 40.0
             delta: 0.025
              npts: 172840
             calib: 3.33982e+10
           _format: SAC
               sac: AttribDict({'delta': 0.025, 'depmin': -2.868547e-06, 'depma
x': 2.3147995e-06, 'scale': 3.33982e+10, 'b': 0.00050000002, 'e': 4320.9756,
'o': -0.0195, 'stla': -54.4986, 'stlo': 158.9561, 'stel': 14.0, 'stdp': 0.0,
'evla': -36.148499, 'evlo': -72.932701, 'evdp': 28.1, 'dist': 8806.7334, 'a
z': 207.84457, 'baz': 139.6021, 'gcarc': 79.301056, 'depmen': 4.1842047e-15,
'cmpaz': 90.0, 'cmpinc': 90.0, 'nzyear': 2010, 'nzjday': 58, 'nzhour': 6, 'nz
min': 34, 'nzsec': 13, 'nzmsec': 19, 'nvhdr': 6, 'npts': 172840, 'iftype': 1,
'idep': 6, 'leven': 1, 'kstnm': 'MCQ', 'kevnm': 'Near Coast Of Ce', 'kcmpnm':
'BHE', 'knetwk': 'AU', 'kinst': 'Streckei'})

----------------------------data of sac-----------
 [  1.23723590e-12   1.31756270e-12   1.42001179e-12 ...,   2.01126032e-10
   1.99127978e-10   1.96827415e-10]
```

3. Sort the traces in terms of distance and plot

In [18]:

```
from IPython.display import Image, display
z_comp = read('../data/chile_IU/*BHZ*SAC')
z_comp.detrend(); z_comp.taper(max_percentage=0.05)
z_comp.filter("bandpass", freqmin=1, freqmax=9, corners=2, zerophase=True)
for tr in z_comp:
    tr.stats.distance = tr.stats.sac['dist']
ref = (z_comp[0].stats.sac['nzyear'],
       z_comp[0].stats.sac['nzjday'],
       z_comp[0].stats.sac['nzhour'],
       z_comp[0].stats.sac['nzmin'],
       z_comp[0].stats.sac['nzsec'],
       z_comp[0].stats.sac['nzmsec'],
       z_comp[0].stats.sac['o'])
o_time = UTCDateTime(year=ref[0],
            julday=ref[1], hour=ref[2], minute=ref[3],
            second=ref[4], microsecond=ref[5]) + ref[-1]
z_comp.plot(type='section', orientation='horizontal', size=(4000,1600),
            reftime=o_time, outfile='../image/sort_obspy.png')
x = Image(filename='../image/sort_obspy.png')
y = Image(filename='../image/sort_sac.png')
display(x, y)
```





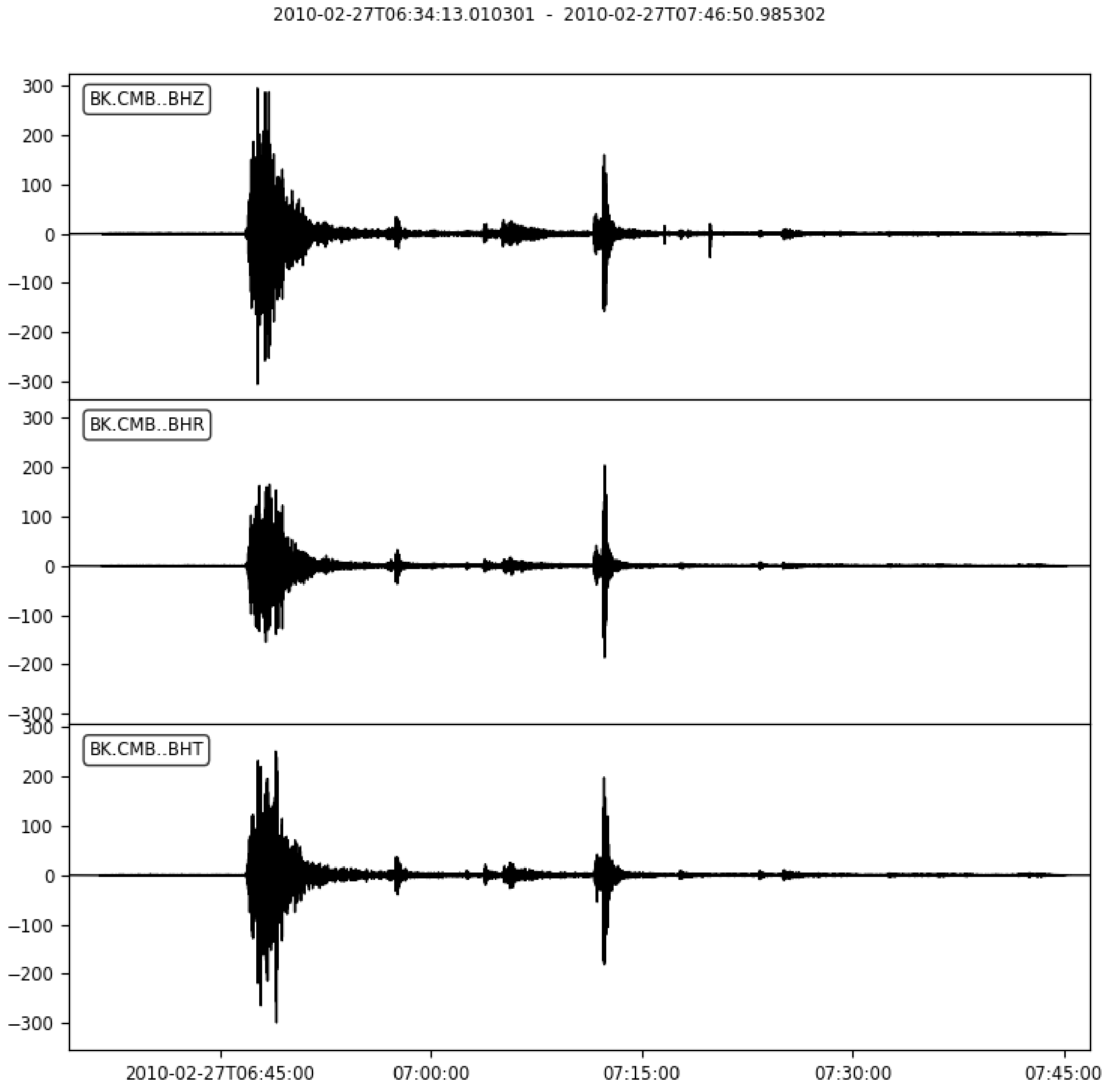
4. Remove Response and compare the rotation output

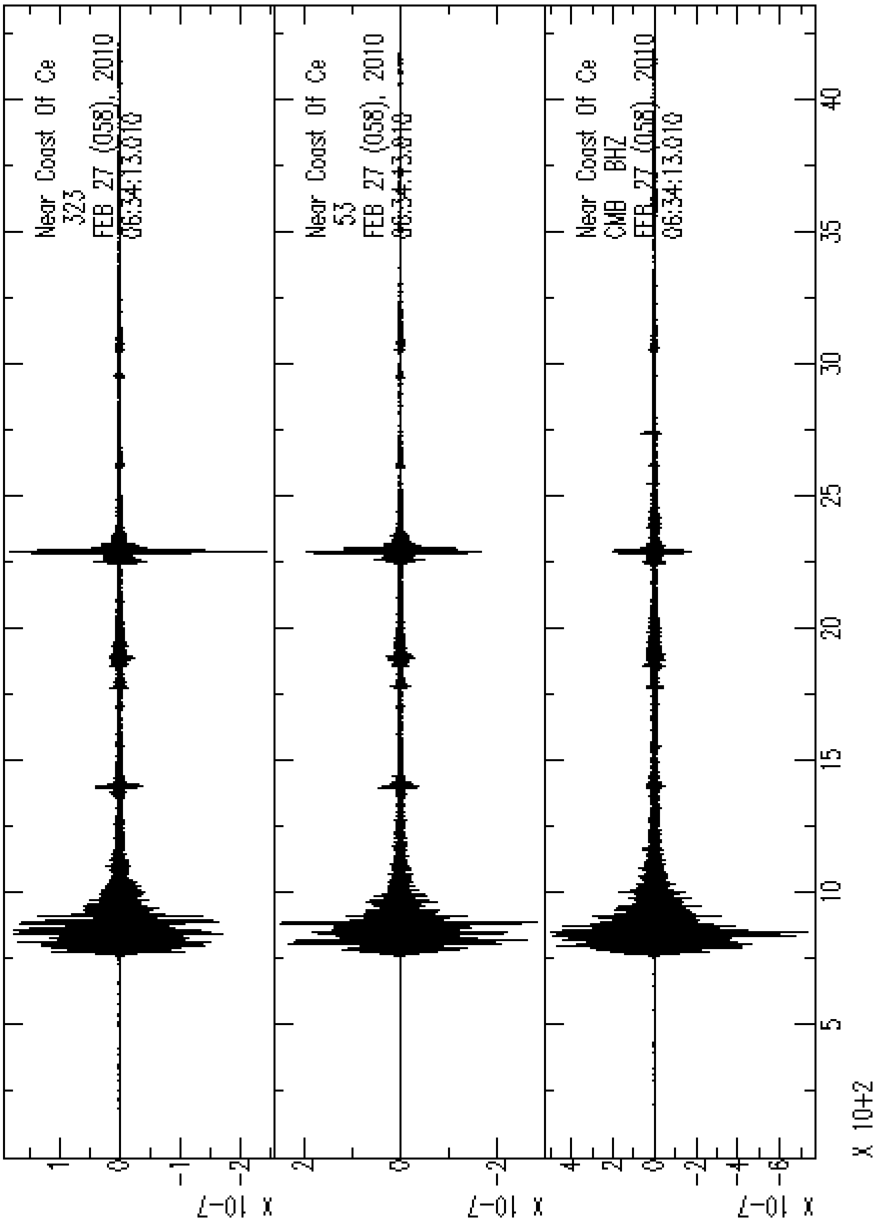
```
#all response files written to a common file
ls -d */ | awk '{printf cat %sSACPZ* >> SAC_PZs, $0}' | sh
rmean; rtr; taper
trans from pol s SAC.PZs to none freq 0.008 1 9 9.9
mul 1.0e9
```

```
#remove some recordings because no PZfiles attached
w over
```

In [25]:

```
rtz_st = read('../data/chile/BK.CMB..BH*.SAC')
#back_azimuth
baz = rtz_st[0].stats.sac['baz']
NE_st = rtz_st[:2]
NE_st.rotate(method="NE->RT", back_azimuth=baz)
rtz_st = NE_st + rtz_st[2]
#bandpass filter
rtz_st.detrend(); rtz_st.taper(max_percentage=0.05)
rtz_st.filter("bandpass", freqmin=1, freqmax=9, corners=2, zerophase=True)
rtz_st.plot(outfile='../image/rtz_obspy.png')
rtz_obspy = Image(filename='../image/rtz_obspy.png')
rtz_sac = Image(filename='../image/rtz_sac.png')
display(rtz_obspy, rtz_sac)
```





5. Pick more phases in 3-C

6. Trajectory of the Direct P wave

In [100…

```
import matplotlib.pyplot as plt
import numpy as np
rtz_st = read("../data/chile/II.ABPO.10.BH?.M.2010.058.063413.SAC")
plt.scatter(rtz_st[1][295:370], rtz_st[2][295:370], s=3, c='r')
plt.plot(rtz_st[1][295:370], rtz_st[2][295:370], c='b')
from scipy.optimize import leastsq
def fit(p,x):
    k,b=p
    return k*x+b
def error(p,x,y):
    return fit(p,x)-y
p0=[1,0]

para=leastsq(error,p0,args=(rtz_st[1][295:370],rtz_st[2][295:370]))
x = (np.arange(300)-100)*1e-13
k,b=para[0]
plt.plot(x,k*x+b,c='k')
```

```
plt.xlabel('Tangential Amplitude')
plt.ylabel('Vertical Amplitude')
plt.title('trajectory of direct P wave')
theta = np.rad2deg(np.arctan(k))
print("the slope angle is %.1f degree, it could be changed by the ratio of P/
plt.savefig('../image/trajectory.png', dpi=500)
```

```
the slope angle is 65.6 degree, it could be changed by the ratio of P/S ampli
tude
```