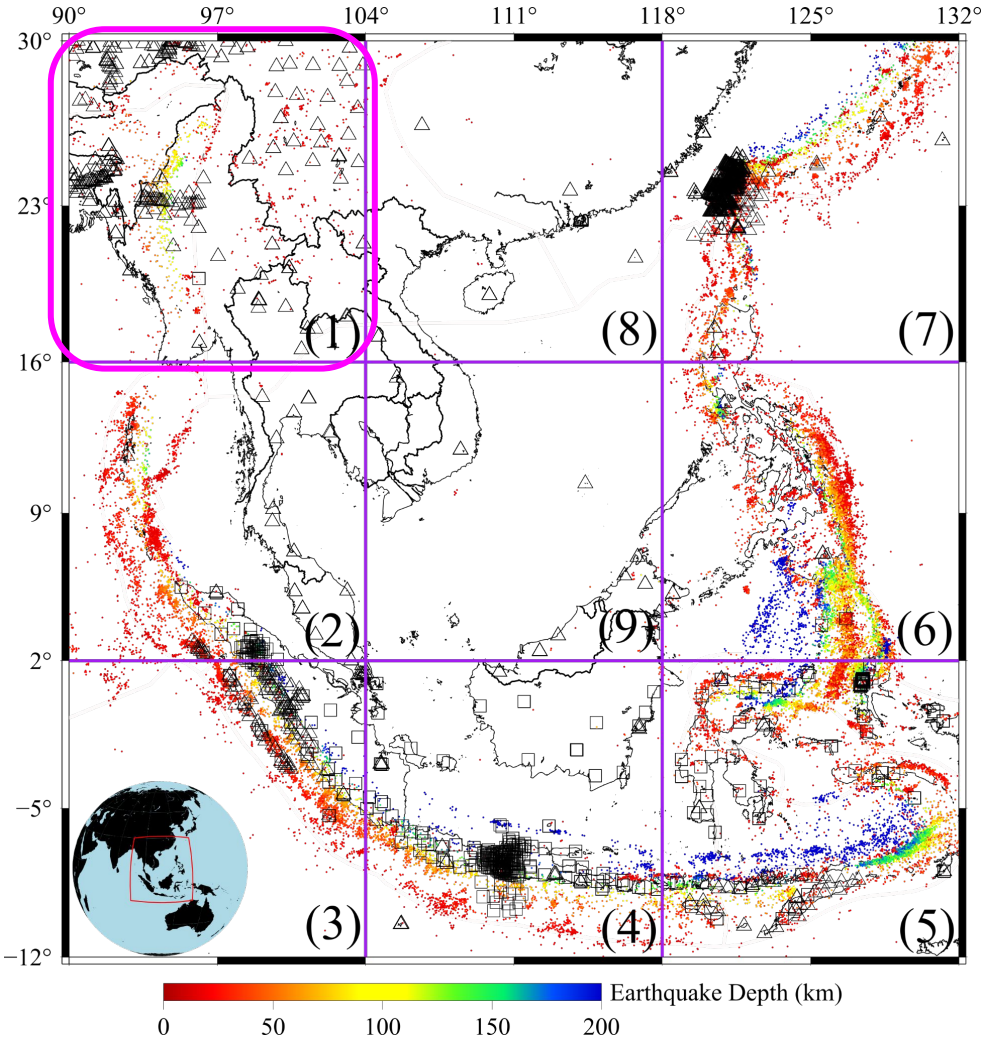
Prepare seismic data in southeast Asia

Tianjue Li, Shucheng Wu

2022 06.06



#SAPTARSHI ROY# 5

#SHASHWAT DROLIA# 4

#KUSH MUKESH KOTHARI# 3

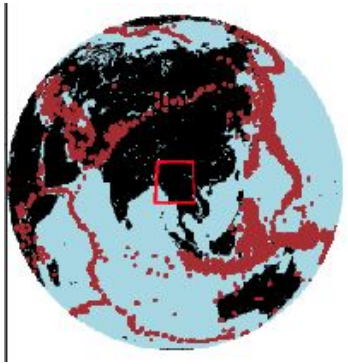
#SIDDHARTH RANJAN BAJPAYI# 6

#VISHESH MITTAL# 1

#RACCHIT JAIN# 2

#RISHIKESH S# (Take Examination) 7

#TUSHAR KHOKHAR# (Unwell) 8 & 9



Near Station Far Event (NSFE)
Stations within study region & events 25 to 95° away from study site.

```
# specify webservice
c_event = Client("USGS")

# define catalog duration
year1 = 2021
year2 = 2022
# define the study site center
lat = 23
lon = 97

## See details at https://docs.obspy.org/packages/autogen/obspy.clients.fdsn.client.Client.get_events.html
catalog = c_event.get_events(starttime=UTCDateTime(f"{year1}-01-01T00:00:00.000"), endtime=UTCDateTime(f"{year2}-01-01T00:00:00.000"),
    latitude=lat, longitude=lon, minradius=25, maxradius=95, minmagnitude=5.0)
```

0.get_events.ipynb
– prepare one-year earthquake catalog
Aim duration: 2000 - 2022.

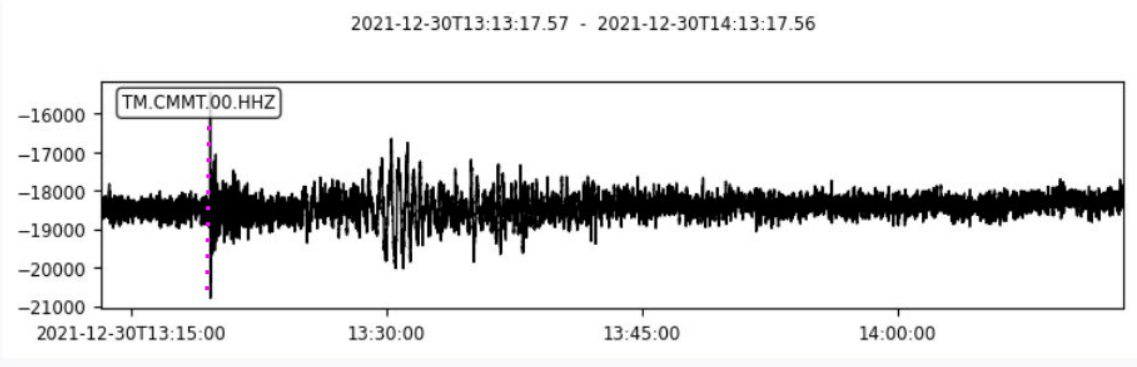
```
# define catalog duration
year1 = 2021
year2 = 2022

# define your study site
minlon, maxlon = 90, 104
minlat, maxlat = 16, 30

# define stations provider, after using IRIS, try GFZ
provider = "GFZ"
```

3.get_waveform.ipynb
– download one-year seismic data
Aim duration: 2000 - 2022.

```
jupyter nbconvert 3.get_waveform.ipynb --to python
```

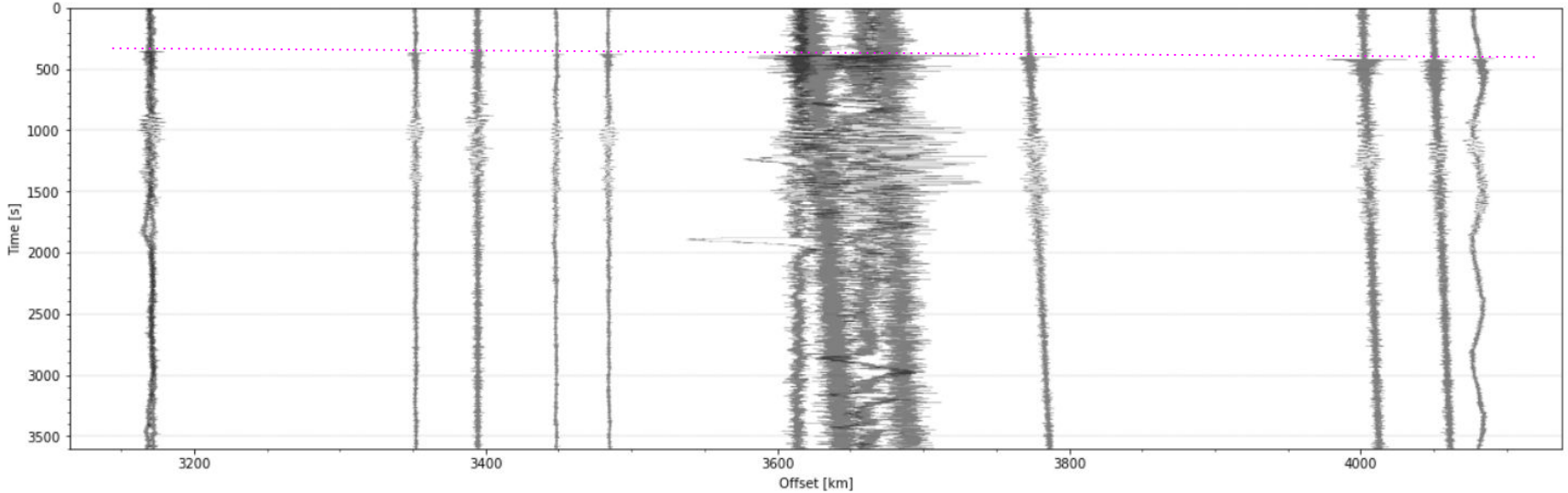
```
Run in terminal:
$ python 3.get_waveform.py
```

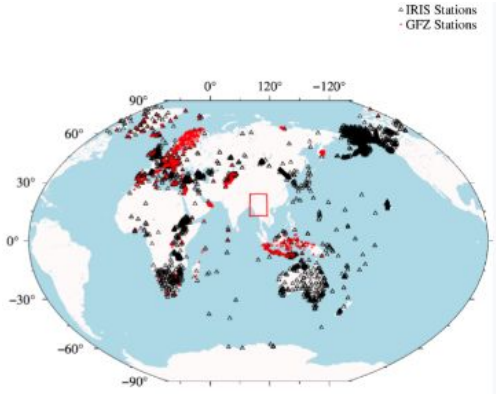
```
with open(catalog, "r") as csvfile:
    events = csv.reader(csvfile, delimiter=',')
    next(events, None)  # skip the headers
    for i,event in enumerate(events):
        if (i == 4):  ### only download nth event's waveforms for exercise, when normally download data, remove this "if" condition.
            origin_time = UTCDateTime(event[0])
            year = event[0][0:4]
            print(origin_time)
            event_fname = "".join(event[0].split("T")[0].split("-")) + "".join("".join("".join(event[0].split("T")[1].split("Z")).split(".")).split(":"))

        # Step 1: Data Selection
            # search for available stations within study site
            domain = RectangularDomain(minlatitude = minlat-0.5, maxlatitude = maxlat+0.5,
                                       minlongitude = minlon-0.5, maxlongitude = maxlon+0.5)

            restrictions = Restrictions(
            # Get data from event origin time to 1 hour after.
                starttime = origin_time,
                endtime = origin_time + 60*60,
```



Quickly view waveforms

First P





Near Event Far Station (NEFS)

Events within study region & stations 25 to 95° away from study site.

```
datafile = "./ISC_EHB_Catalog_1980-2018"

# define catalog duration
year1 = 2018
year2 = 2019

###define your study region here
minlon = 90
maxlon = 104
minlat = 16
maxlat = 30

df = pd.read_csv(f"{datafile}", sep=',', usecols=[3,4,5,6,7,12], header=10)
df.columns = ["date","time","lat","lon","dep","mag"]
dfnew = df[::-1]

f = open(f"Catalog_{year1}-{year2}","w")
for i,line in enumerate(zip(dfnew["date"],dfnew["time"],dfnew["lat"],dfnew["lon"],dfnew["dep"],dfnew["mag"])):
```

0.get_events.ipynb

– prepare one-year earthquake catalog

Aim duration: 2000 - 2019.

```
# define catalog duration
year1 = 2018
year2 = 2019

# define your study site
minlon, maxlon = 90, 104
minlat, maxlat = 16, 30
lonc = (minlon + maxlon) / 2
latc = (minlat + maxlat) / 2

# define stations provider, after using IRIS, try GFZ
provider = "GFZ"
```

3.get_waveform.ipynb

– download one-year seismic data

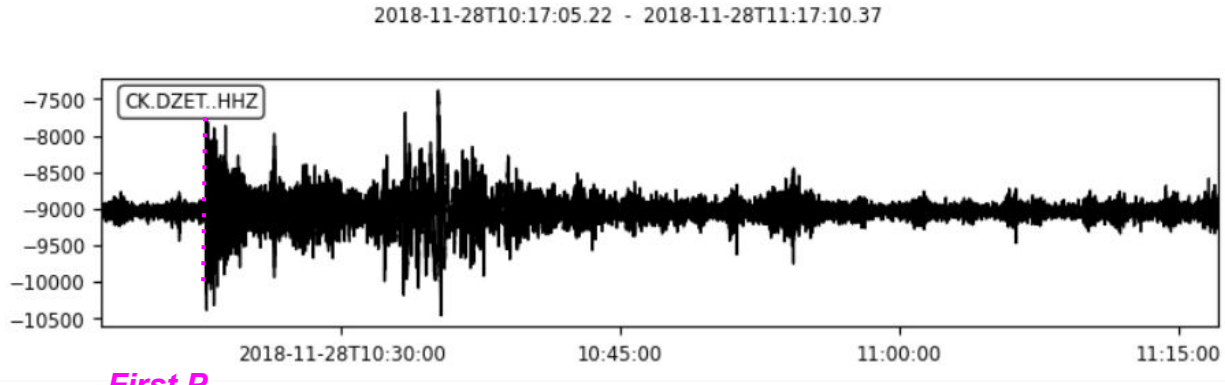
Aim duration: 2000 - 2019

```
jupyter nbconvert 3.get_waveform.ipynb --to python
```

Run in terminal:

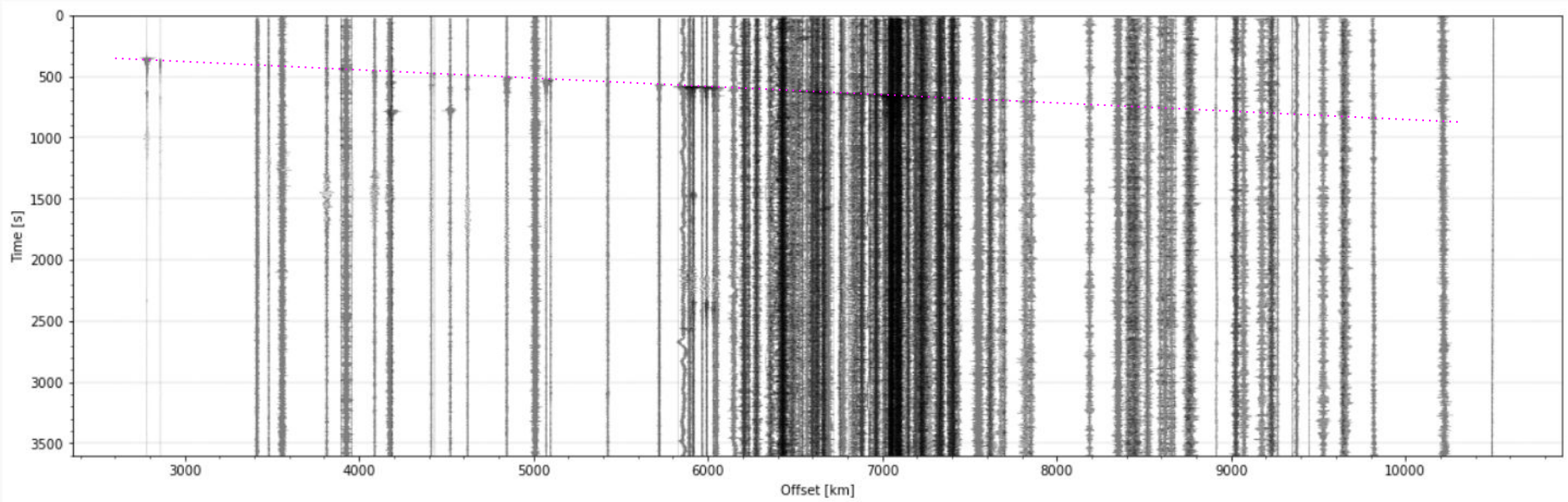
```
$ python 3.get_waveform.py
```

```
with open(catalog, "r") as evtfiles:
    for i,file in enumerate(evtfiles):
        event = file.split()
        if (i == 1):   ### only check kth event for excersice, when normally download data, remove this "if" condition.
            origin_time = UTCDateTime(event[0])
            year = event[0][0:4]
            event_fname = "".join(event[0].split("T")[0].split("-")) + "".join("".join("".join(event[0].split("T")[1].split("Z")).split(".")).split(":"))
            print(origin_time, year, event_fname)

        # Step 1: Data Selection
            # search for available stations globally
            domain = CircularDomain(latitude=latc, longitude=lonc, minradius=25.0, maxradius=95.0)
            restrictions = Restrictions(
            # Get data from event origin time to one hour after.
                starttime = origin_time,
                endtime = origin_time + 60*60,
```



Quickly view waveforms

First P



Data processing

Remove the instrument response

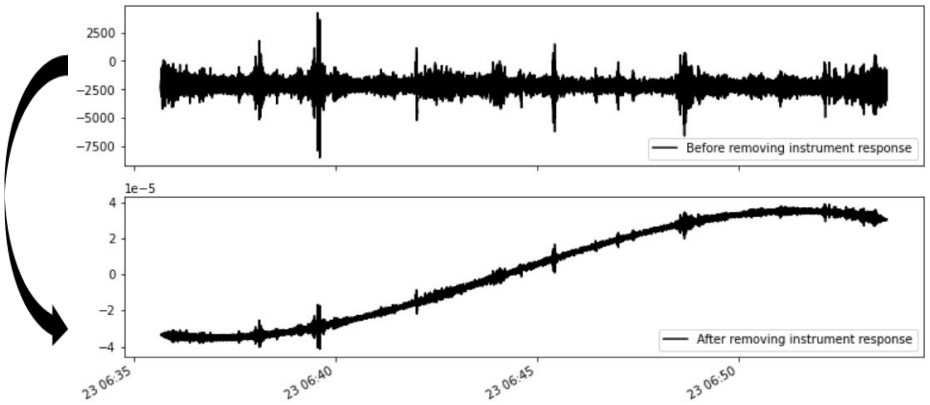
Resample and filter waveforms

Mark the P-wave traveltime on waveforms

Select waveforms according to signal-to-noise ratio

$$u(t) = x(t) * e(t) * q(t) * i(t)$$

where x(t) is the source time function, the "signal" the earthquake puts into the ground, e(t) and q(t) represent the effects of earth structure, and i(t) describes the instrument response of the seismometer.



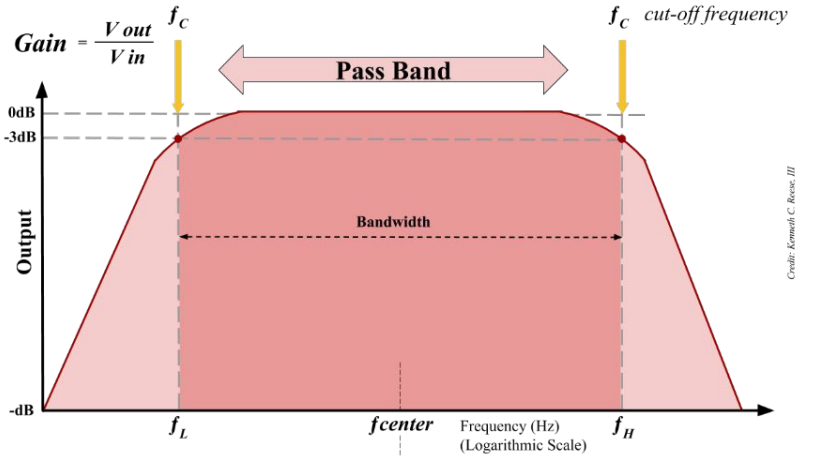
Data processing

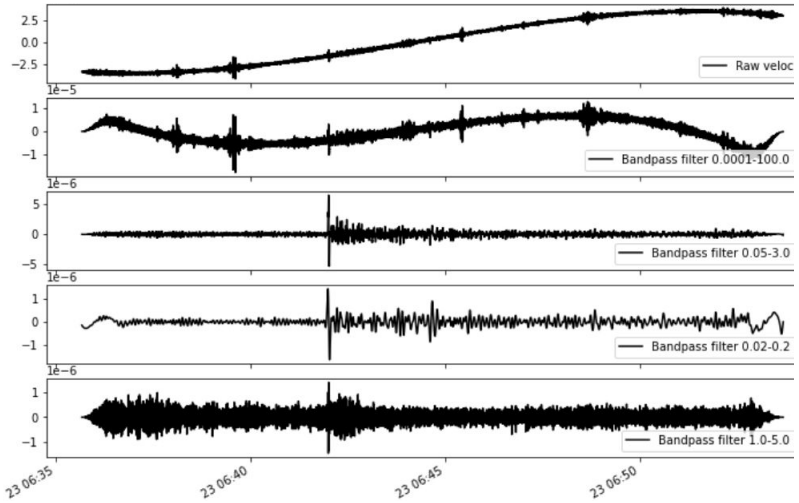
Remove the instrument response

Resample and filter waveforms

Mark the P-wave traveltime on waveforms

Select waveforms according to signal-to-noise ratio





Data processing

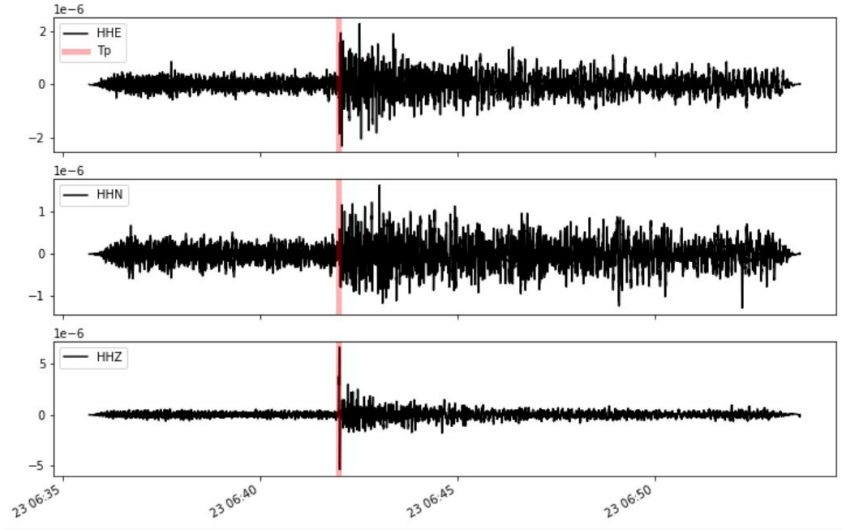
Remove the instrument response

Resample and filter waveforms

Mark the P-wave traveltime on waveforms

Select waveforms according to signal-to-noise ratio

Taup Tookit



Data processing

Remove the instrument response

Resample and filter waveforms

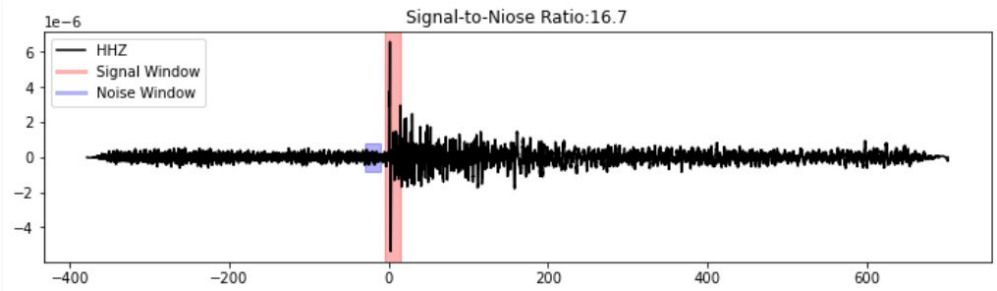
Mark the P-wave traveltime on waveforms

Select waveforms according to signal-to-noise ratio

$$SNR = \frac{P_{signal}}{P_{noise}}$$

Wanted component

Unwanted component



Next, download waveform if you haven't finished it;
During the time, retain high-quality waveforms using SNR.