

Behind and Beyond the Automation: Manual Mastery in Seismic Catalog Construction

speaker: Yijian Zhou

email: yijian.zhou@email.ucr.edu

2023-08-28 online tutorial



Outline

- Prepare your data
 - data download & cleaning
 - station distribution, data continuity
- Run your workflow
 - picking & association: window length, time residual, number of station ...
 - location: velocity model, distance weight, WDCT/C
- Check your result
 - number, location, time, magnitude
 - compare with a reference, check whether physically interpretable

Outline

- **Prepare your data**
 - data download & cleaning
 - station distribution, data continuity
- Run your workflow
 - picking & association: window length, time residual, number of station ...
 - location: velocity model, distance weight, WDCT/C
- Check your result
 - number, location, time, magnitude
 - compare with a reference, check whether physically interpretable

Obspy MassDownloader

obspy.clients.fdsn.mass_downloader

Mass Downloader for FDSN Compliant Web Services

This package contains functionality to query and integrate data from any number of [FDSN](#) formulate download requests in a way that is convenient for seismologists without having used by itself or as a library component integrated into a bigger project.

```
1  """ Download FDSN data with obspy MassDownloader
2  """
3  import os
4  from obspy import UTCDateTime
5  from obspy.clients.fdsn.mass_downloader import RectangularDomain, Restrictions, MassDownloader
6
7  # i/o paths
8  data_root = '/data/Example_data'
9  sta_dir = 'output/eg_stations'
10 # down params
11 providers = ["IRIS"]
12 chn_codes = ['HH*', 'EH*']
13 loc_codes = ['', "00", "01"]
14 num_workers = 10
15 start_date, end_date = UTCDateTime('20160101'), UTCDateTime('20170101')
16 lat_rng = [35, 38]
17 lon_rng = [111, 112]
```

https://github.com/YijianZhou/Seismic-Data-Preparation/blob/master/down_fdsn_mass.py

```
>>> from obspy.clients.fdsn.header import URL_MAPPINGS
>>> for key in sorted(URL_MAPPINGS.keys()):
...     print("{0:<11} {1}".format(key, URL_MAPPINGS[key]))
AUSPASS      http://auspass.edu.au
BGR           http://eida.bgr.de
EIDA         http://eida-federator.ethz.ch
EMSC         http://www.seismicportal.eu
ETH          http://eida.ethz.ch
GEOFON       http://geofon.gfz-potsdam.de
GEONET       http://service.geonet.org.nz
GFZ          http://geofon.gfz-potsdam.de
ICGC         http://ws.icgc.cat
IESDMC       http://batsws.earth.sinica.edu.tw
INGV         http://webservices.ingv.it
IPGP         http://ws.ipgp.fr
IRIS         http://service.iris.edu
IRISPH5      http://service.iris.edu
ISC          http://isc-mirror.iris.washington.edu
KNMI         http://rdsa.knmi.nl
KOERI        http://eida.koeri.boun.edu.tr
LMU          http://erde.geophysik.uni-muenchen.de
NCEDC        http://service.ncedc.org
NIEP         http://eida-sc3.infp.ro
NOA          http://eida.gein.noa.gr
ODC          http://www.orfeus-eu.org
ORFEUS       http://www.orfeus-eu.org
RASPIshake   https://fdsnws.raspberryshakedata.com
RESIF        http://ws.resif.fr
RESIFPH5     http://ph5ws.resif.fr
SCEDC        http://service.scedc.caltech.edu
TEXNET       http://rtserve.beg.utexas.edu
UIB-NORSAR   http://eida.geo.uib.no
USGS         http://earthquake.usgs.gov
USP          http://sismo.iag.usp.br
```

Check path structure

```
(base) [zhou20@kong11-linx MountCristo]$ ls continuous_waveforms_May | head
BK.OVRO.00.HHE__20200501T000000Z__20200502T000000Z.mseed
BK.OVRO.00.HHE__20200502T000000Z__20200503T000000Z.mseed
BK.OVRO.00.HHE__20200503T000000Z__20200504T000000Z.mseed
BK.OVRO.00.HHE__20200504T000000Z__20200505T000000Z.mseed
BK.OVRO.00.HHE__20200505T000000Z__20200506T000000Z.mseed
BK.OVRO.00.HHE__20200506T000000Z__20200507T000000Z.mseed
BK.OVRO.00.HHE__20200507T000000Z__20200508T000000Z.mseed
BK.OVRO.00.HHE__20200508T000000Z__20200509T000000Z.mseed
BK.OVRO.00.HHE__20200509T000000Z__20200510T000000Z.mseed
BK.OVRO.00.HHE__20200510T000000Z__20200511T000000Z.mseed
```

Check data

- One miniseed file may contains multiple traces!

```
>>> st=read('continuous_waveforms_May/NP.1679*')
>>> print(st)
336 Trace(s) in Stream:

NP.1679..HNE | 2020-05-08T14:54:39.080000Z - 2020-05-08T14:57:58.520000Z | 200.0 Hz, 39889 samples
...
(334 other traces)
...
NP.1679..HNZ | 2020-05-31T01:06:47.060000Z - 2020-05-31T01:10:29.995000Z | 200.0 Hz, 44588 samples

[Use "print(Stream.__str__(extended=True))" to print all Traces]
```

Merge traces and reorganize paths

```
# i/o paths
data_dir = '/data3/data/MountCristo/continuous_waveforms_August'
out_root = '/data3/data/MountCristo/continuous_data_clean'
st_paths = sorted(glob.glob(data_dir+'/*.mseed'))
fout = open('output/bad_samp-rate_08.csv', 'w')

for st_path in st_paths:
    print(st_path)
    fname = os.path.basename(st_path)
    date = fname.split('__')[1][0:8]
    out_dir = os.path.join(out_root, date)
    if not os.path.exists(out_dir): os.makedirs(out_dir)
    st = read(st_path)
    try: st.merge(fill_value=0)
    except:
        st.resample(100); st.merge(fill_value=0)
        fout.write(st_path+'\n')
    for tr in st:
        net, sta, chn = tr.stats.network, tr.stats.station, tr.stats.channel
        out_path = os.path.join(out_dir, '%s.%s.%s.%s.mseed'%(net, sta, date, chn))
        tr.write(out_path)
fout.close()
```

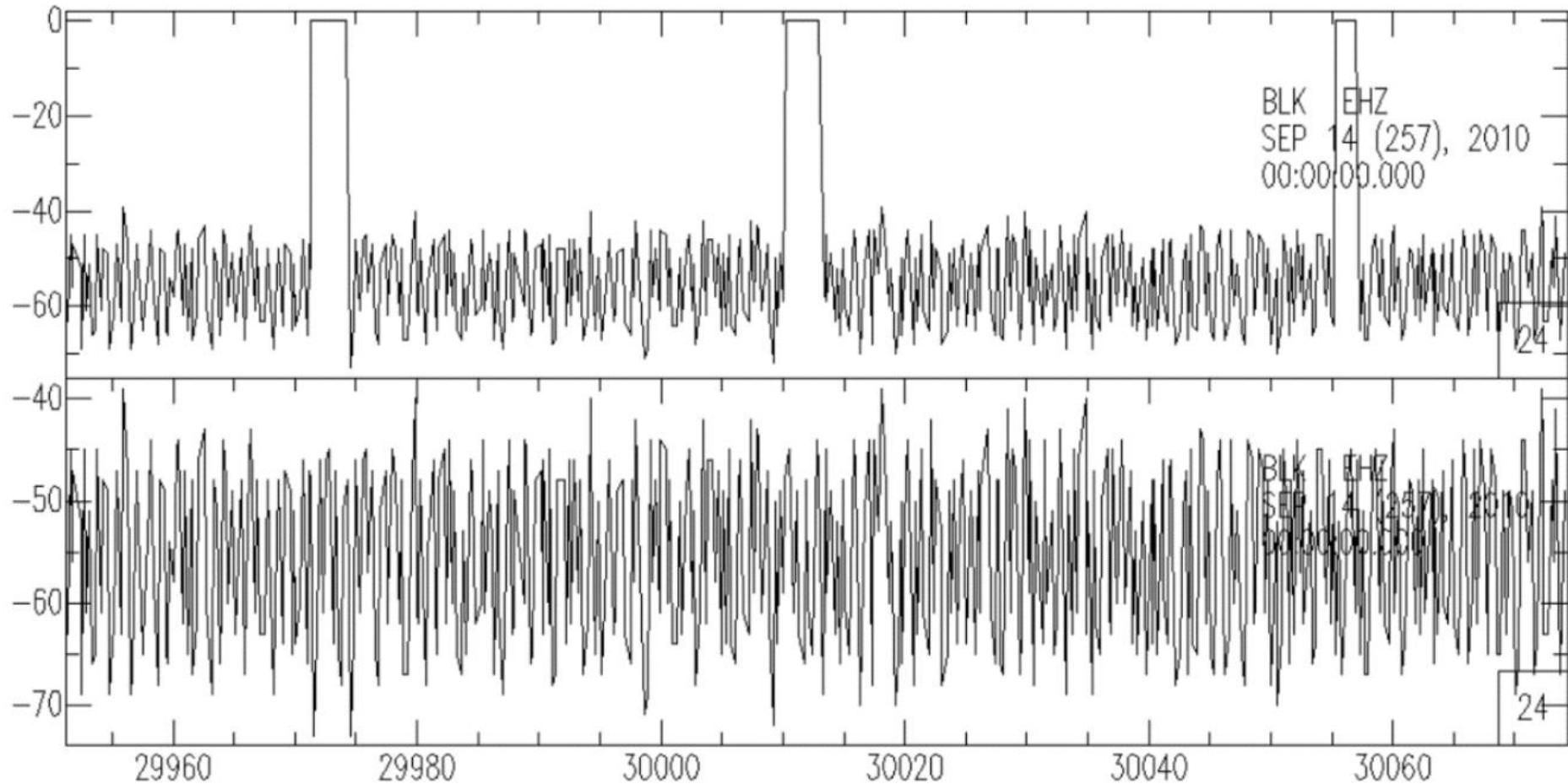
Merge traces and reorganize paths

```
(base) [zhou20@kong11-lnx MountCristo]$ ls continuous_data_clean/
20200501  20200514  20200527  20200609  20200622  20200705  20200718  20200731
20200502  20200515  20200528  20200610  20200623  20200706  20200719  20200801
20200503  20200516  20200529  20200611  20200624  20200707  20200720  20200802
20200504  20200517  20200530  20200612  20200625  20200708  20200721  20200803
20200505  20200518  20200531  20200613  20200626  20200709  20200722  20200804
20200506  20200519  20200601  20200614  20200627  20200710  20200723  20200805
20200507  20200520  20200602  20200615  20200628  20200711  20200724  20200806
20200508  20200521  20200603  20200616  20200629  20200712  20200725  20200807

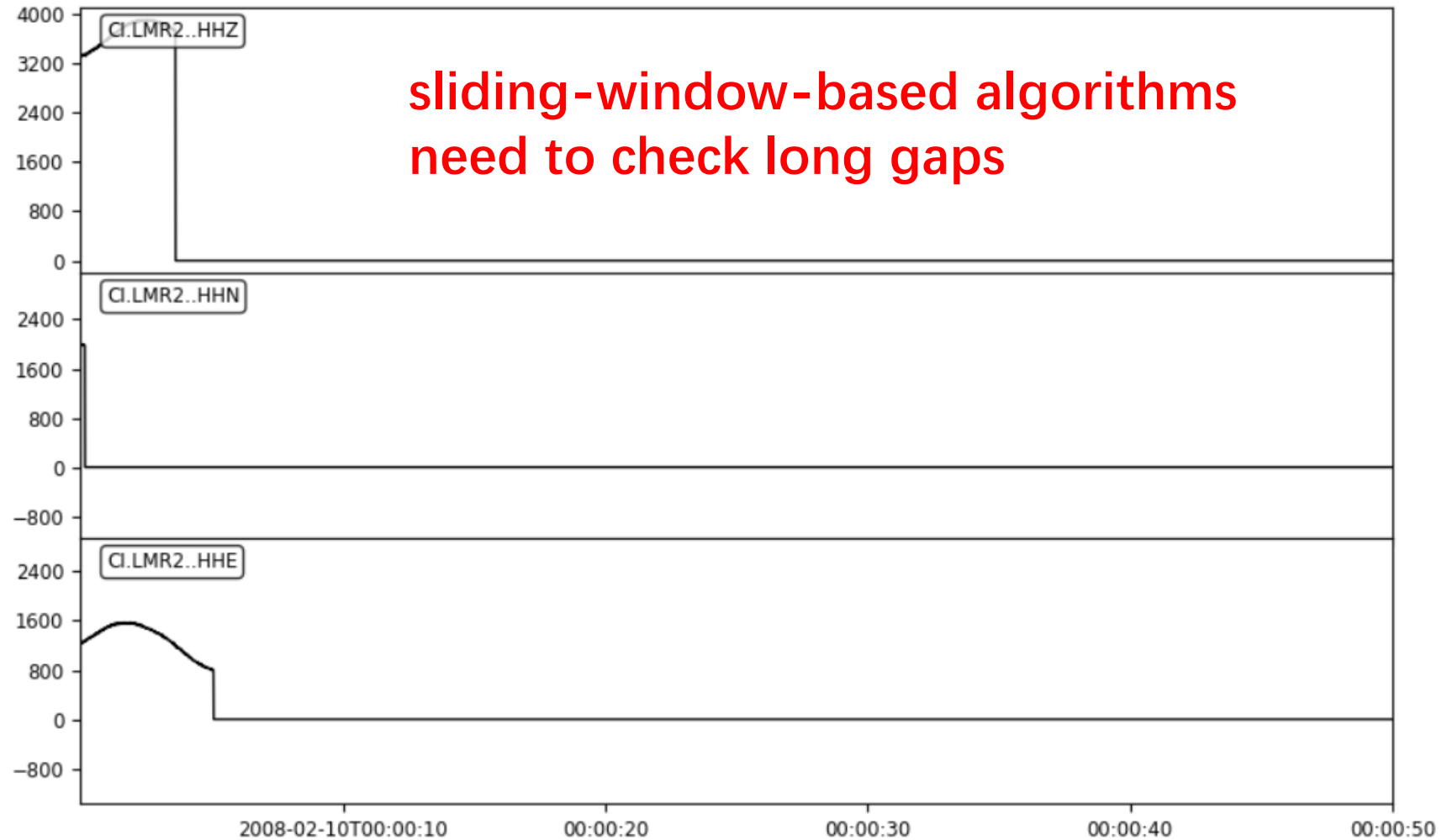
(base) [zhou20@kong11-lnx MountCristo]$ ls continuous_data_clean/20200501/
BK.OVR0.20200501.HHE.mseed  IM.NV31.20200501.BHZ.mseed  NN.GMN.20200501.HHN.mseed
BK.OVR0.20200501.HHN.mseed  NC.MBS1.20200501.HN1.mseed  NN.GMN.20200501.HHZ.mseed
BK.OVR0.20200501.HHZ.mseed  NC.MBS1.20200501.HN2.mseed  NN.KVN.20200501.HHE.mseed
CI.MLAC.20200501.HHE.mseed  NC.MCB.20200501.HHE.mseed  NN.KVN.20200501.HHN.mseed
CI.MLAC.20200501.HHN.mseed  NC.MCB.20200501.HHN.mseed  NN.KVN.20200501.HHZ.mseed
CI.MLAC.20200501.HHZ.mseed  NC.MCB.20200501.HHZ.mseed  NN.LCH.20200501.HHE.mseed
CI.TIN.20200501.HHE.mseed   NC.MCO.20200501.HNE.mseed  NN.LCH.20200501.HHN.mseed
CI.TIN.20200501.HHN.mseed   NC.MCO.20200501.HNN.mseed  NN.LCH.20200501.HHZ.mseed
CI.TIN.20200501.HHZ.mseed   NC.MCO.20200501.HNZ.mseed  NN.LHV.20200501.HHE.mseed
```

Fit input
data as
required by
the software,
instead of
modifying
the source
code!

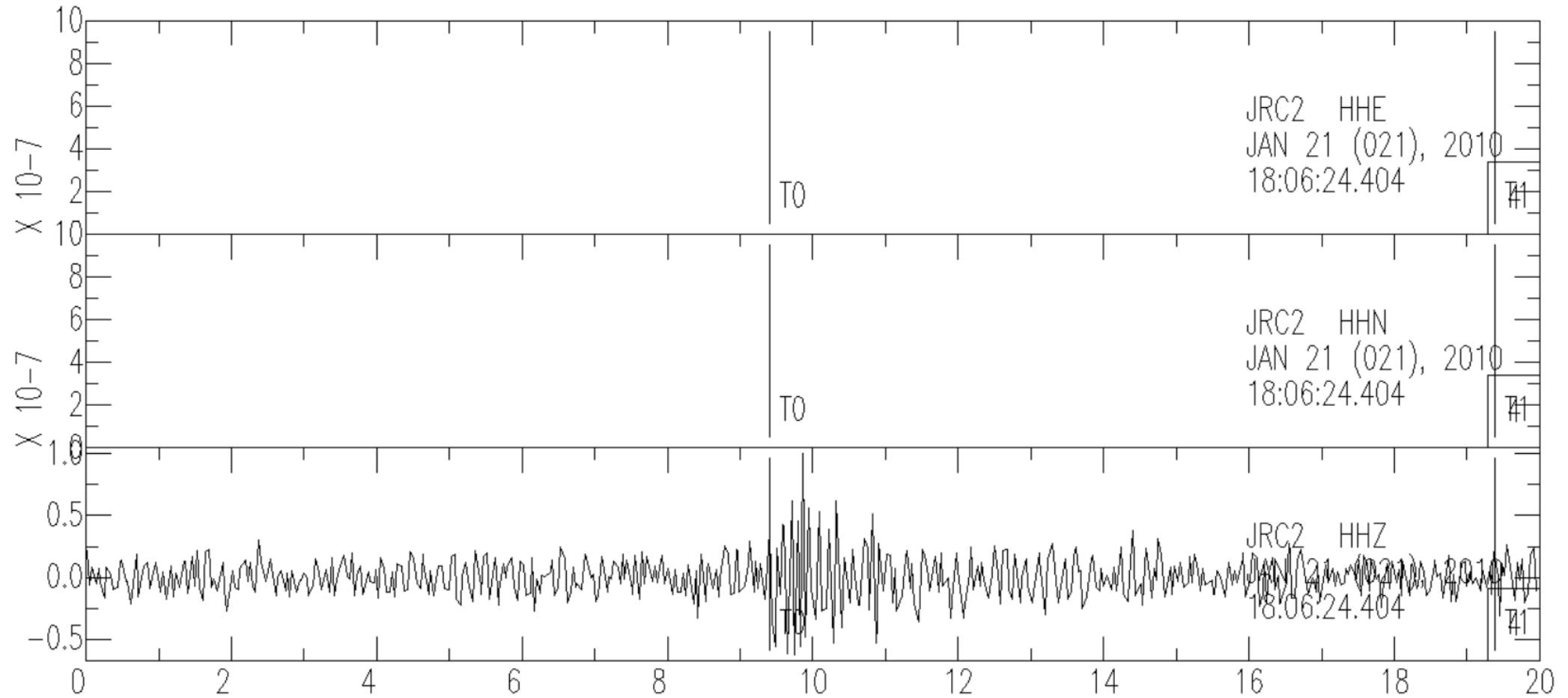
Data gap: short gap



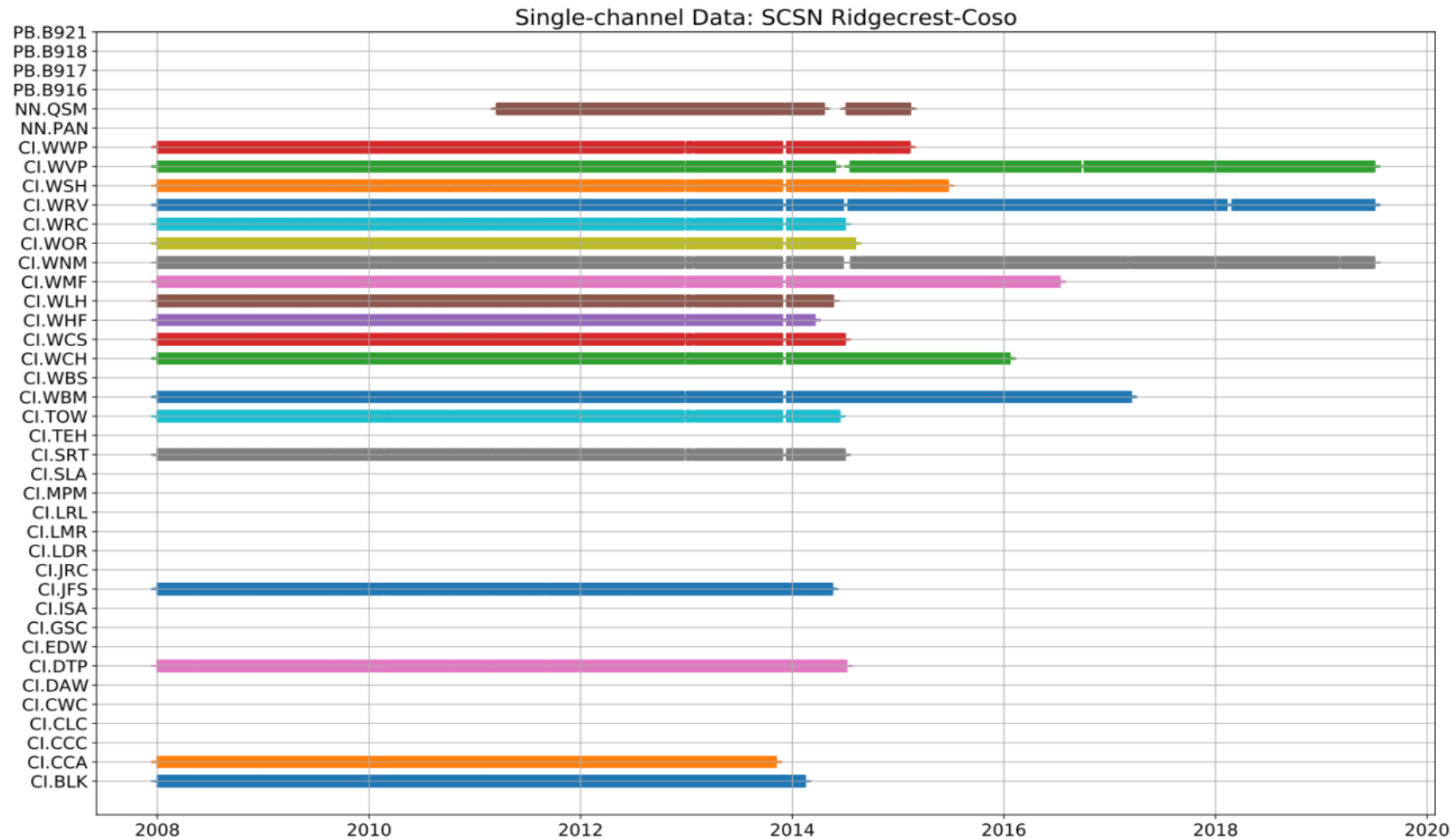
Data gap: long gap



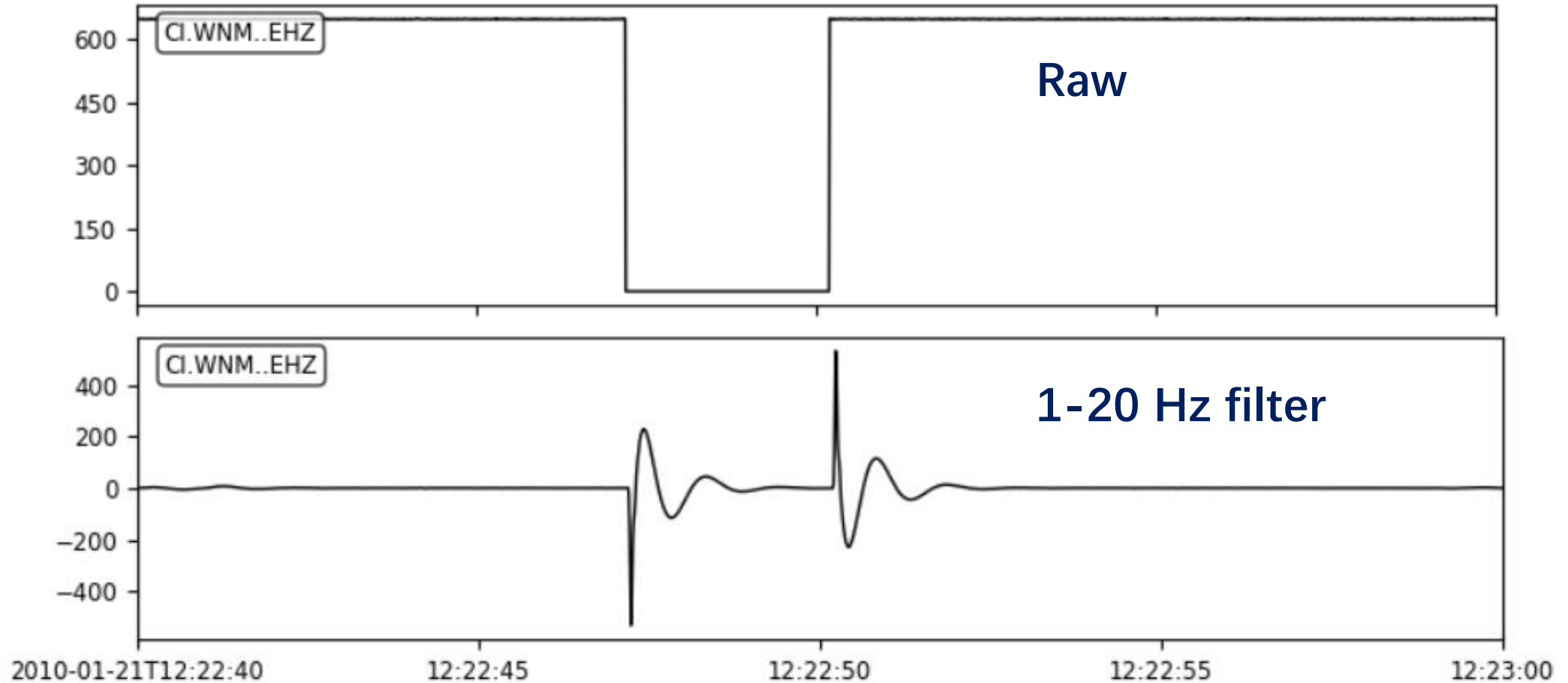
Missed channel & single-channel



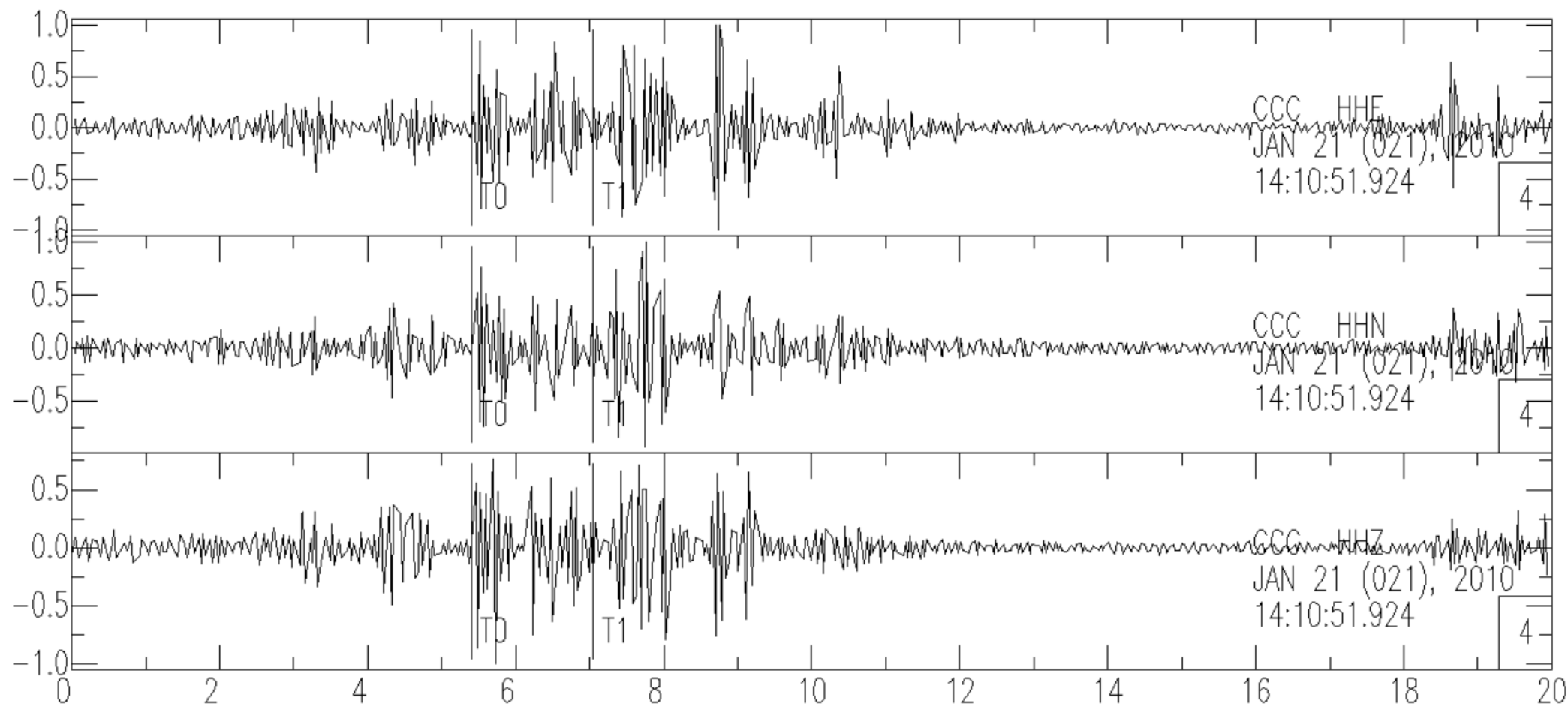
Missed channel & single-channel



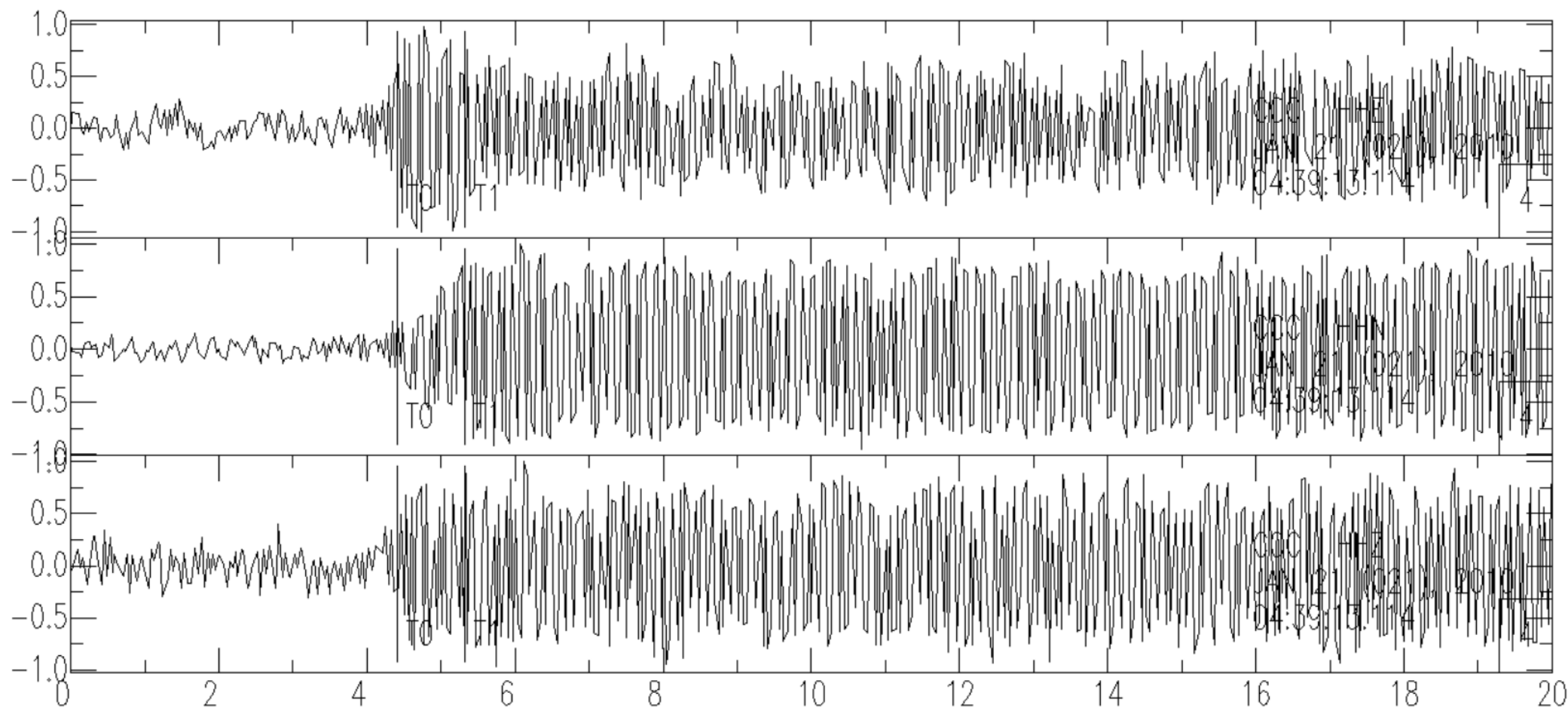
Data segmentation → glitches



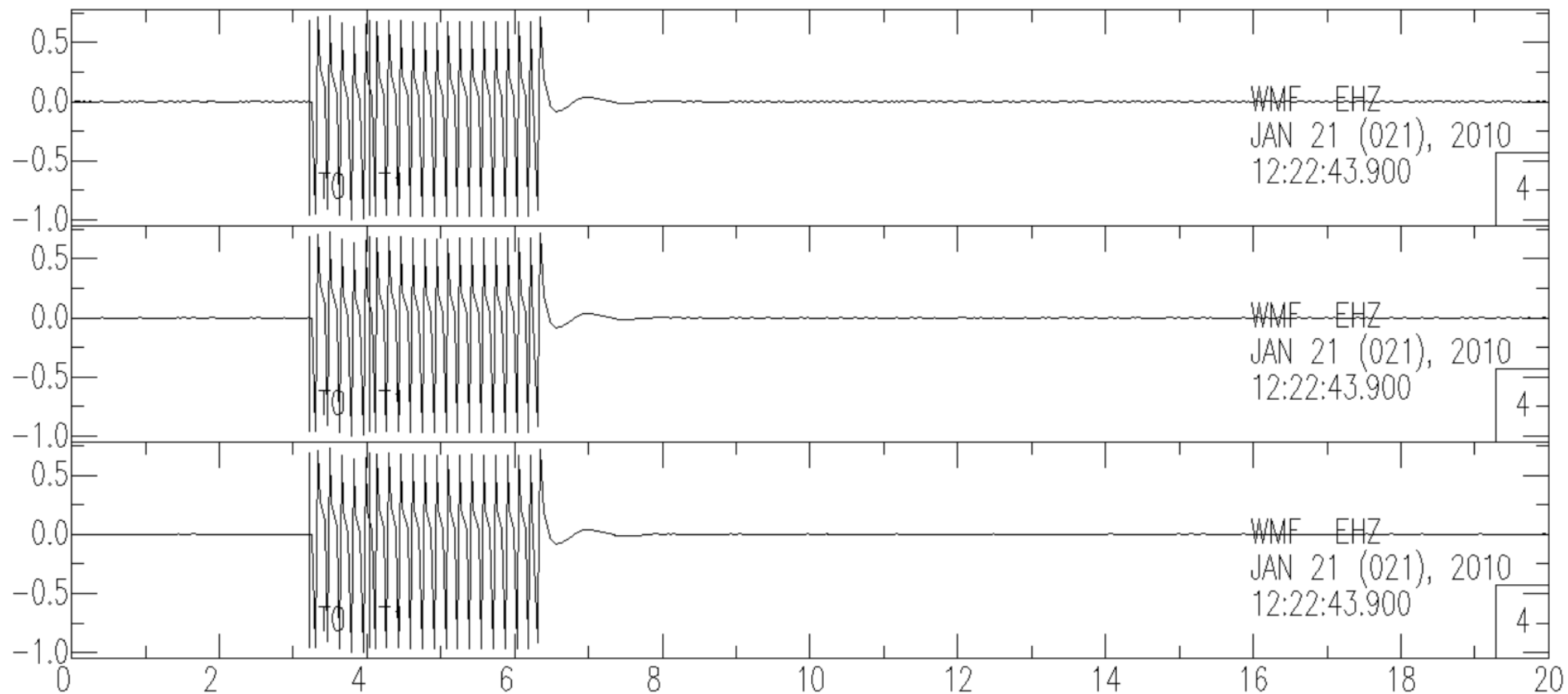
Data glitches



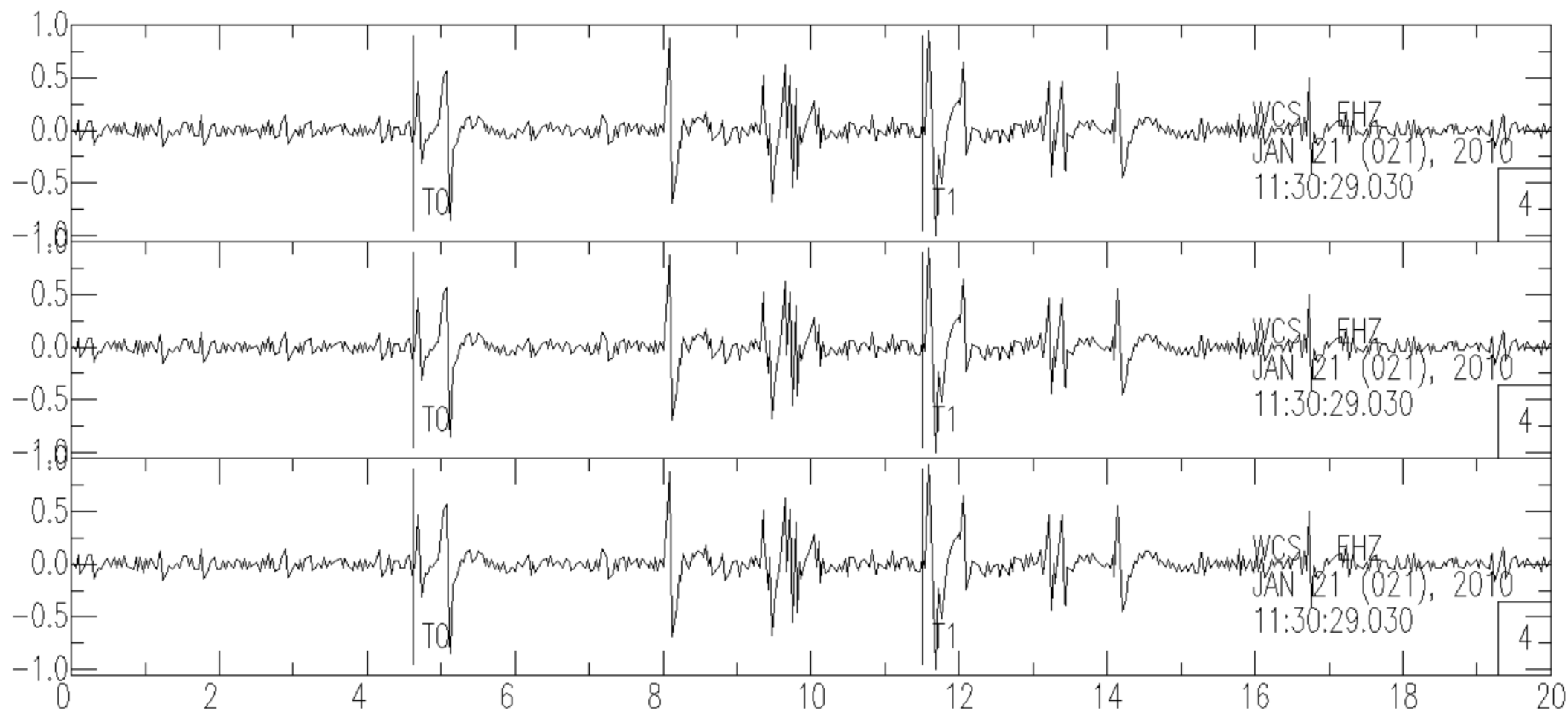
Data glitches



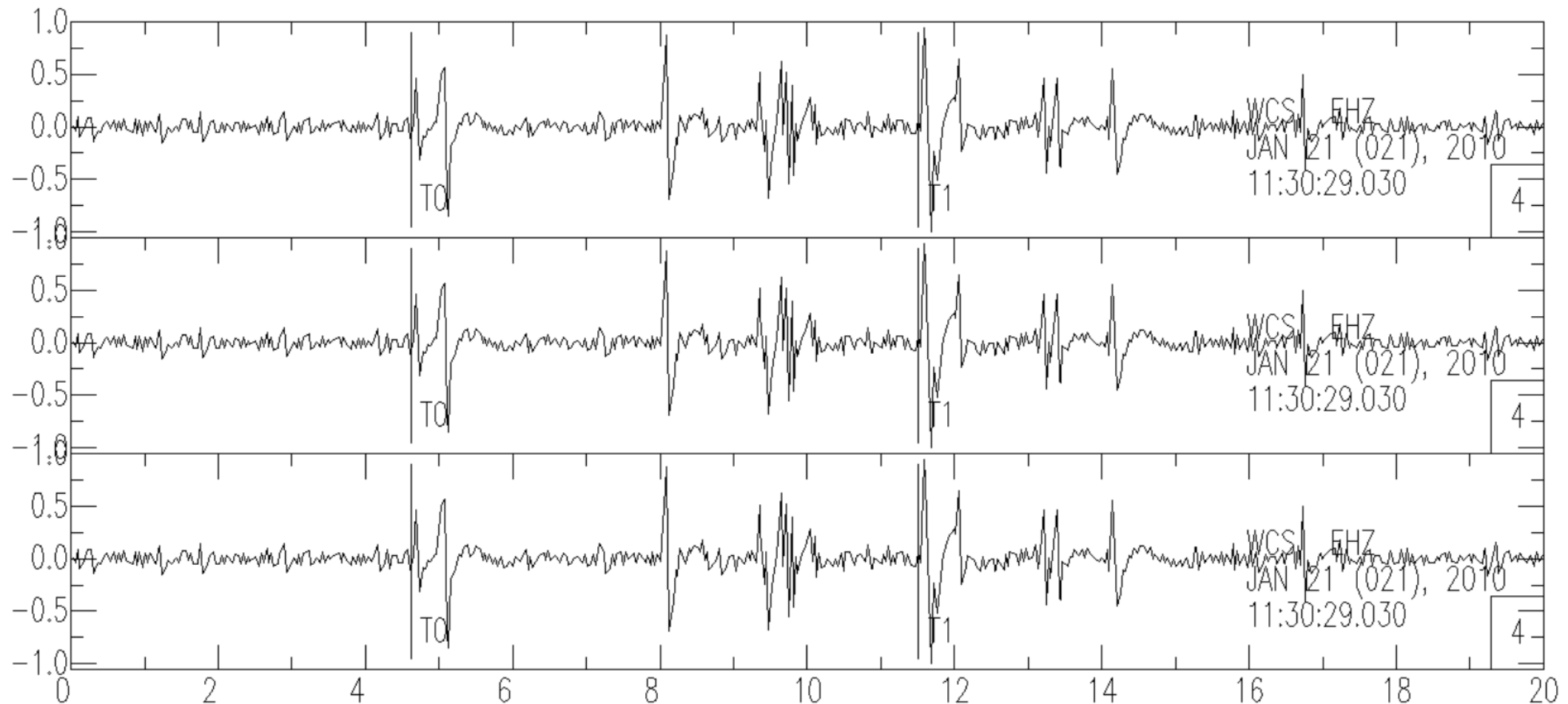
Data glitches

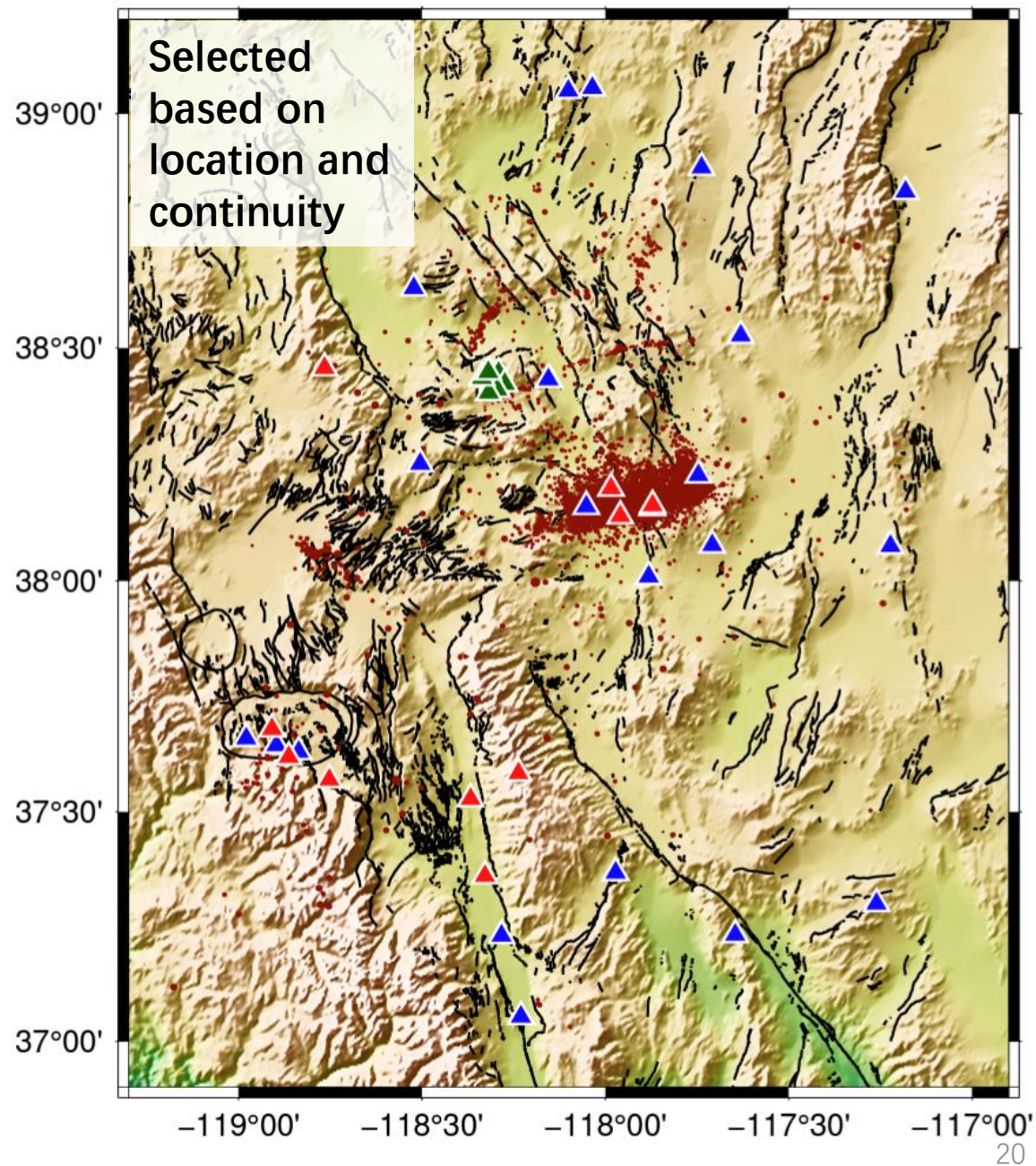
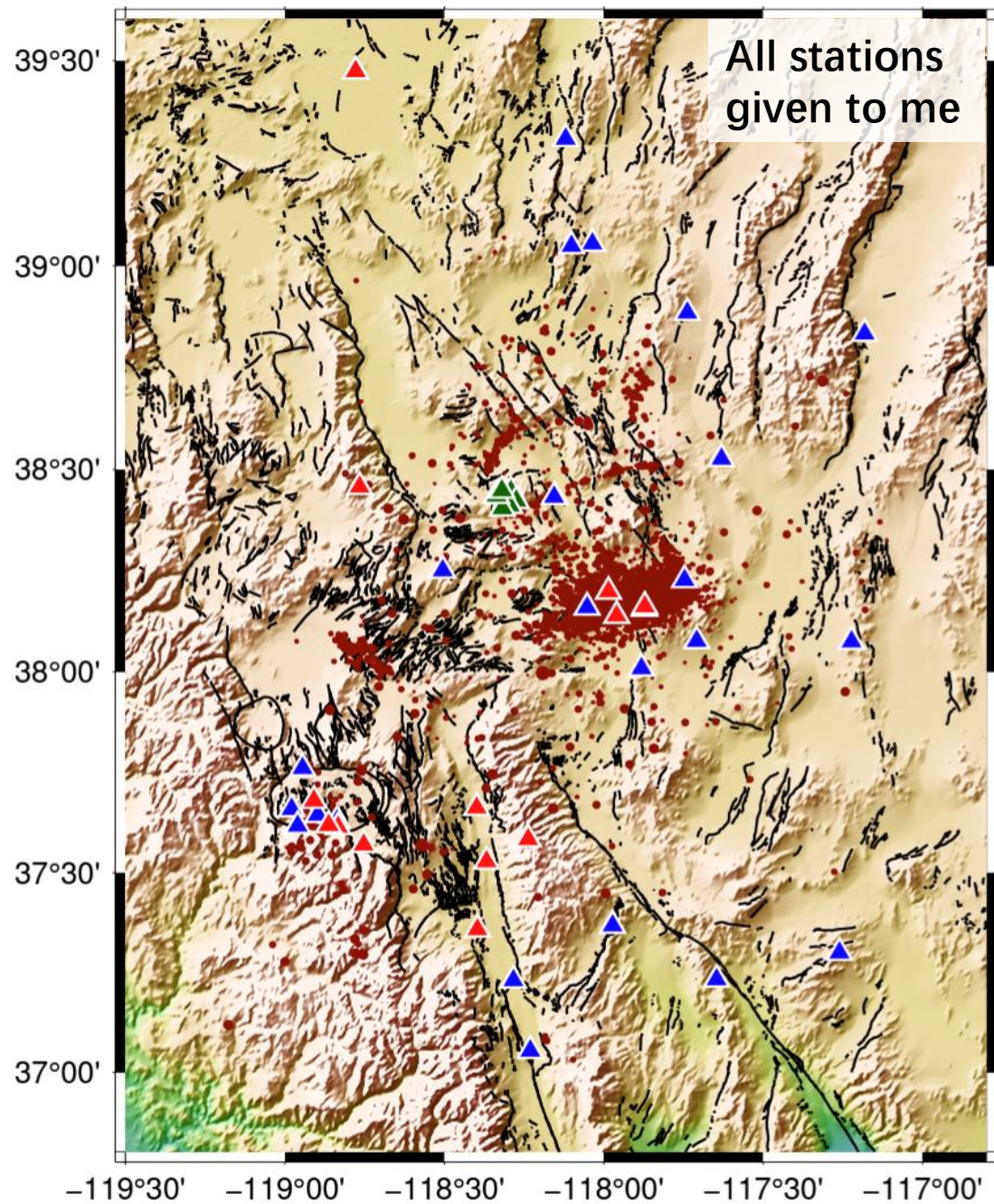


Data glitches



Data glitches should be identified by AI





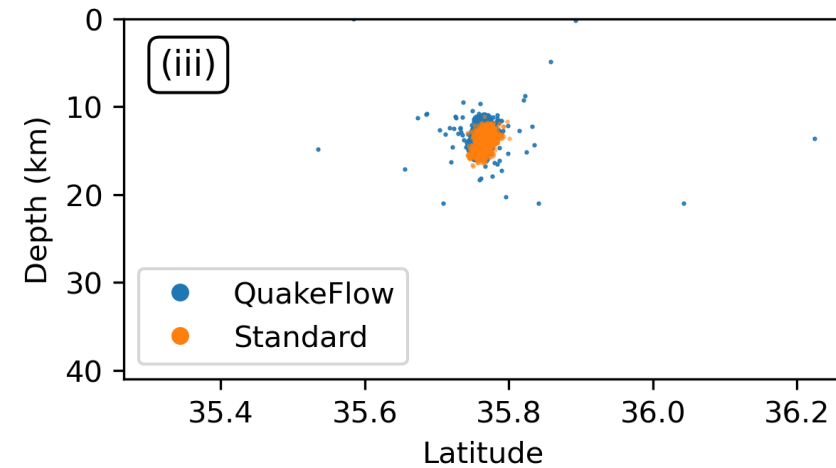
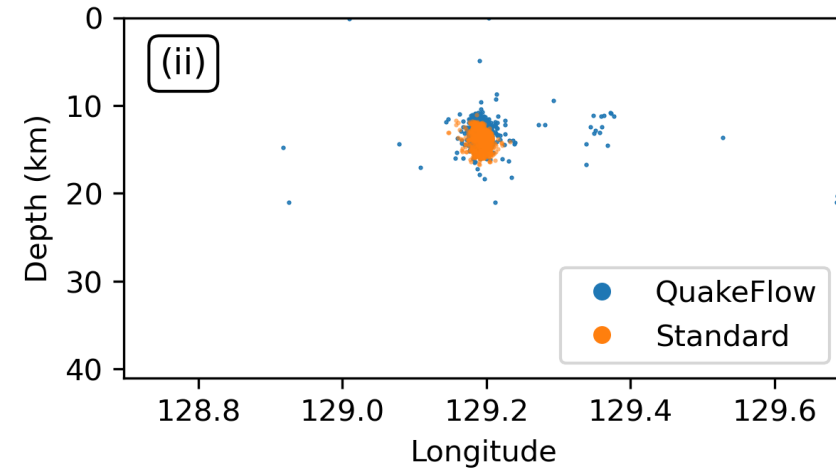
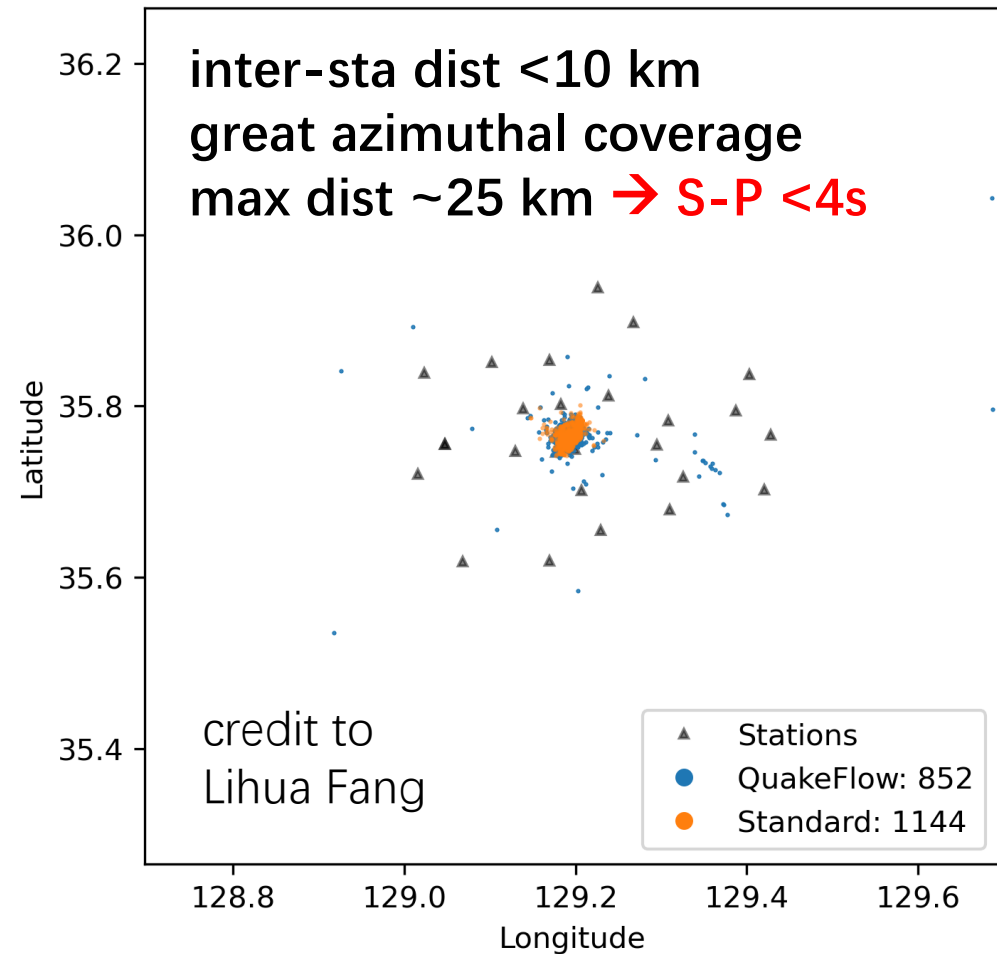
Outline

- Prepare your data
 - data download & cleaning
 - station distribution, data continuity
- **Run your workflow**
 - **picking & association: window length, time residual, number of station ...**
 - **location: velocity model, distance weight, WDCT/C**
- Check your result
 - number, location, time, magnitude
 - compare with a reference, check whether physically interpretable

Expected behavior of an USER

- **Users** need to
 - know the algorithm → whether it's suitable for you
 - know the meaning of params → the default params do not always work
 - fit the inputs as required, do NOT modify the source code
- **Developers** need to
 - make the algorithm general enough
 - make the input parameters intuitive, stable, and simple
 - balance the *flexibility* (less requirements on the input and more confusing for the users) and *stability*

Station distribution



PAL config: Picking

```
# 1. picker params
self.win_sta      = [0.8,0.4,1.] # win for STA: det, p, s
self.win_lta      = [6.,2., 2.] # win for LTA: det, p, s
self.win_kurt     = [5.,1.]     # win for kurtosis: long & short
self.trig_thres   = 12.         # threshold to trig picker (by energy)
self.p_win        = [.5,1.]     # search win for P
self.s_win        = 10.         # search win for S
self.pca_win      = 1.          # win_len for PCA filter
self.pca_range    = [0.,2.]     # time range to apply PCA filter
self.fd_thres     = 2.5         # min value of dominant frequency
self.snr_ratio_thres = 10       # max value of SNR ratio after peak rm
self.amp_ratio_thres = [10,2]   # max value of amp ratio for P/P_tail & S
self.amp_win      = [1.,4.]     # time win to get S amplitude
self.det_gap      = 5.          # time gap between detections
self.to_prep      = True        # whether to preprocess the raw data
self.freq_band    = [1,20]      # frequency band
```


PAL config: Picking

```
# 1. picker params
self.win_sta      = [0.8,0.4,1.]
self.win_lta      = [6. ,2., 2.]
self.win_kurt     = [5.,1.]
self.trig_thres   = 12.
self.p_win        = [.5,1.]
self.s_win        = 10.
self.pca_win      = 1.
self.pca_range    = [0.,2.]
self.fd_thres     = 2.5
self.snr_ratio_thres = 10
self.amp_ratio_thres = [10,2]
self.amp_win      = [1.,4.]
self.det_gap      = 5.
self.to_prep      = True
self.freq_band    = [1,20]
```

Suitable for most cases

```
# win for STA: det, p, s
# win for LTA: det, p, s
# win for kurtosis: long & short
# threshold to trig picker (by energy)
# search win for P
# search win for S
# win_len for PCA filter
# time range to apply PCA filter
# min value of dominant frequency
# max value of SNR ratio after peak rm
# max value of amp ratio for P/P_tail & S
# time win to get S amplitude
# time gap between detections
# whether to preprocess the raw data
# frequency band
```

PAL config: Picking

```
# 1. picker params
self.win_sta      = [0.8,0.4,1.]
self.win_lta      = [6. ,2., 2.]
self.win_kurt     = [5.,1.]
self.trig_thres   = 12.
self.p_win        = [.5,1.]
self.s_win       = 10.
self.pca_win      = 1.
self.pca_range    = [0.,2.]
self.fd_thres     = 2.5
self.snr_ratio_thres = 10
self.amp_ratio_thres = [10,2]
self.amp_win     = [1.,4.]
self.det_gap      = 5.
self.to_prep      = True
self.freq_band    = [1,20]
```

Set according to the station distribution

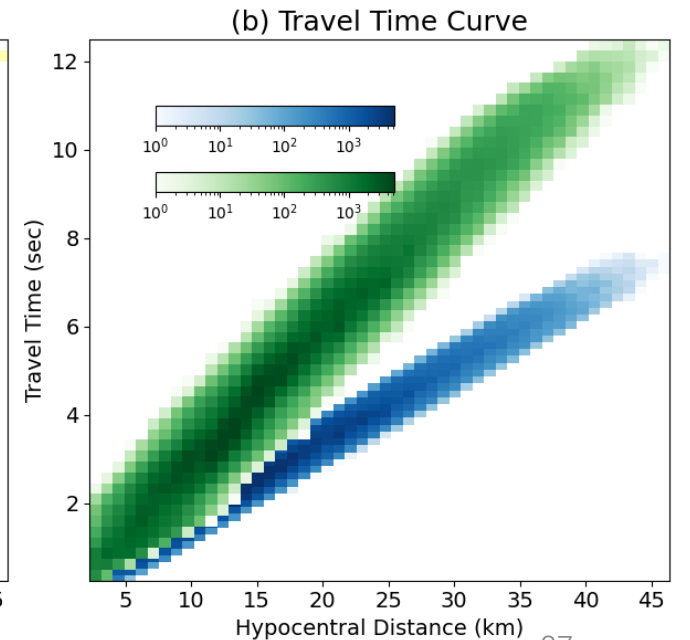
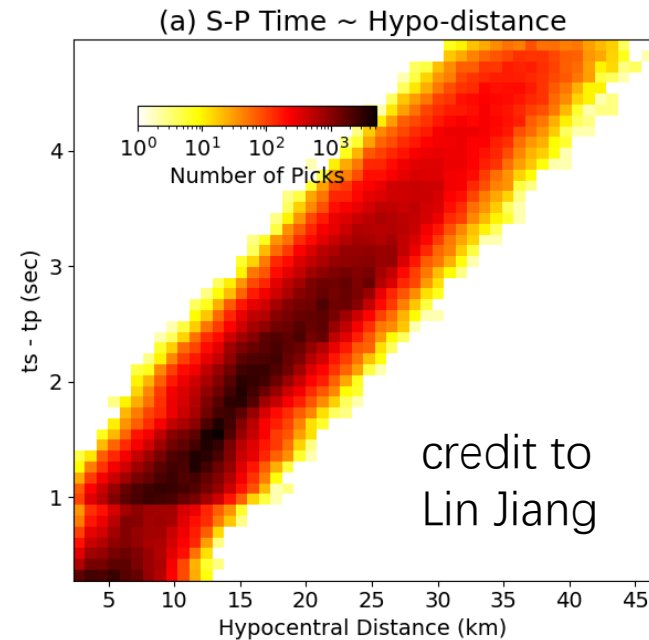
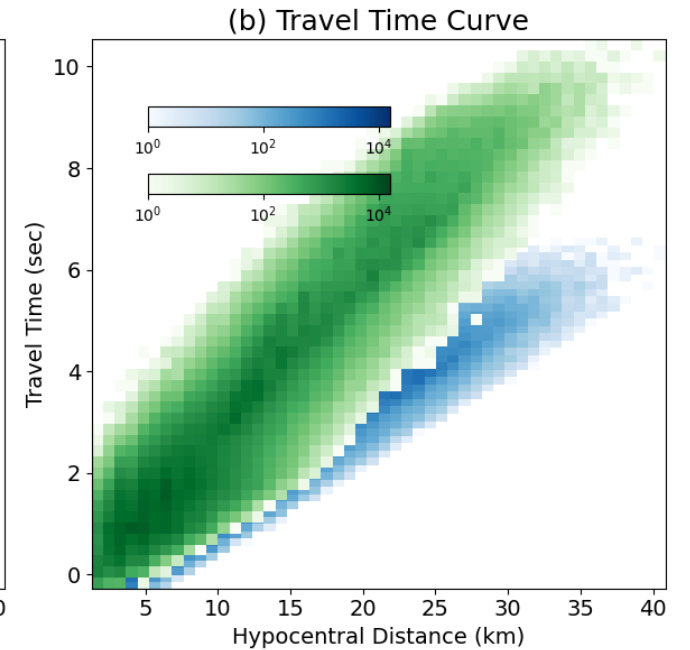
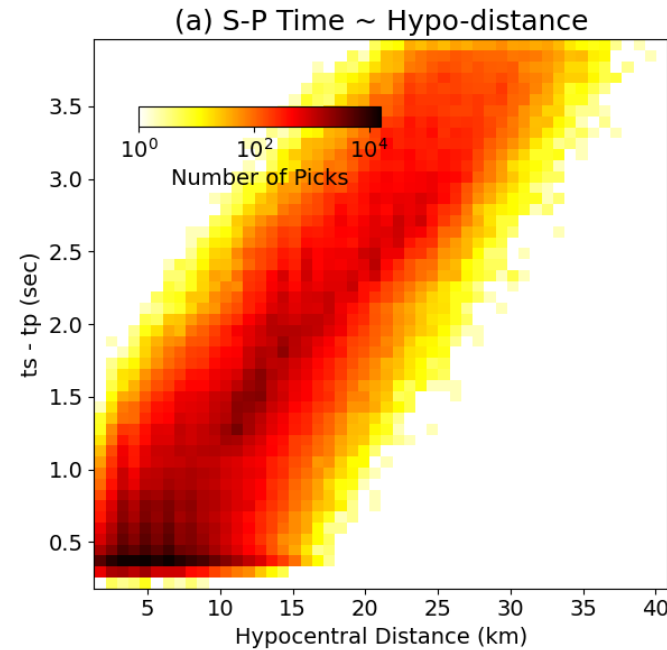
```
# win for STA: det, p, s
# win for LTA: det, p, s
# win for kurtosis: long & short
# threshold to trig picker (by energy)
# search win for P
# search win for S
# win_len for PCA filter
# time range to apply PCA filter
# min value of dominant frequency
# max value of SNR ratio after peak rm
# max value of amp ratio for P/P_tail & S
# time win to get S amplitude
# time gap between detections
# whether to preprocess the raw data
# frequency band
```

One special case: dense & large-N network

```
self.win_sta = [0.8, 0.3, 1.]  
self.win_lta = [6., 2., 2.]  
self.win_kurt = [4., 1.]
```

```
self.win_sta = [0.5, 0.3, 0.8]  
self.win_lta = [4., 1.5, 1.6]  
self.win_kurt = [4., 0.8]
```

Usually you don't need to tune picking params, because typical inter-station distance >5 km



PAL config: Association

```
# 2. assoc params
self.min_sta      = 4          # min num of sta to assoc
self.ot_dev       = 2.         # max time deviation for ot assoc
self.max_res      = 1.5       # max P res for loc assoc
self.max_drop     = 1          # max num of drop of each pick
self.xy_margin    = 0.1        # xy (lateral) range inferred from sta loc
self.xy_grid      = 0.02       # xy (lateral) grid size (in degree)
self.z_grids      = range(2,20,3) # z (dep) grids (in km)
self.vp           = 5.9        # averaged P velocity
```

PAL config: Association

```
# 2. assoc params
```

```
self.min_sta = 4
```

```
self.ot_dev = 2.
```

```
self.max_res = 1.5
```

```
self.max_drop = 1
```

```
self.xy_margin = 0.1
```

```
self.xy_grid = 0.02
```

```
self.z_grids = range(2,20,3) # z (dep) grids (in km)
```

```
self.vp = 5.9
```

Suitable for most cases

```
# min num of sta to assoc
```

```
# max time deviation for ot assoc
```

```
# max P res for loc assoc
```

```
# max num of drop of each pick
```

```
# xy (lateral) range inferred from sta loc
```

```
# xy (lateral) grid size (in degree)
```

```
# z (dep) grids (in km)
```

```
# averaged P velocity
```

PAL config: Association

```
# 2. assoc params
```

```
self.min_sta = 4
```

```
self.ot_dev = 2.
```

```
self.max_res = 1.5
```

```
self.max_drop = 1
```

```
self.xy_margin = 0.1
```

```
self.xy_grid = 0.02
```

```
self.z_grids = range(2,20,3) # z (dep) grids (in km)
```

```
self.vp = 5.9
```

Set according to the station distribution

```
# min num of sta to assoc
```

```
# max time deviation for ot assoc
```

```
# max P res for loc assoc
```

```
# max num of drop of each pick
```

```
# xy (lateral) range inferred from sta loc
```

```
# xy (lateral) grid size (in degree)
```

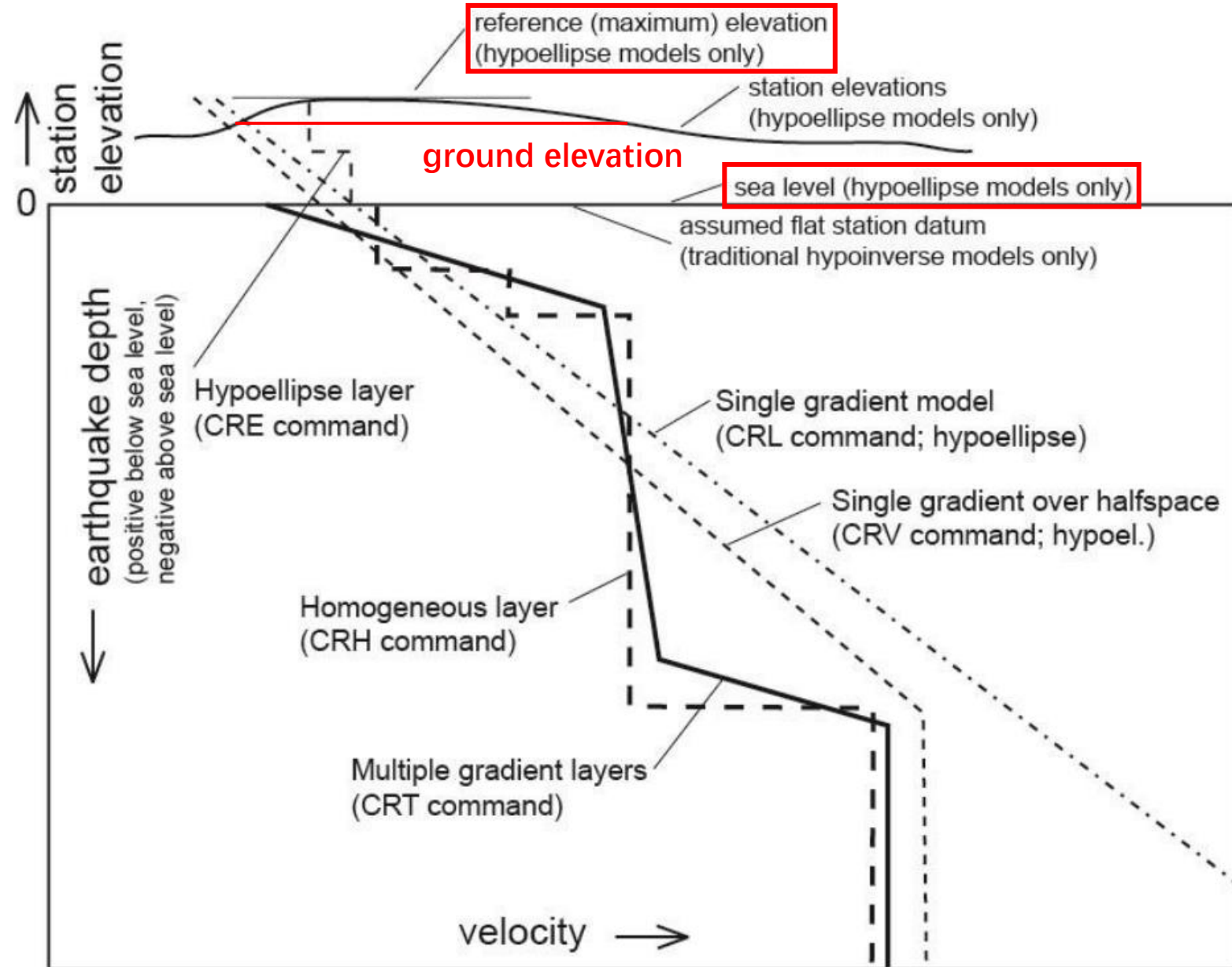
```
# z (dep) grids (in km)
```

```
# averaged P velocity
```

Run hypolnverse

- Station file
 - same as PAL, just copy into input/
- Velocity model
 - in CRE format, which supports station elevation
 - set *ref_ele* & *grd_ele*, make necessary correction
- Location parameters
 - weighting by distance
 - weighting by residual

Crustal model types



Klein, 2014


```
# geo ref
self.lat_code = 'N'
self.lon_code = 'W'
self.ref_ele = 2.5    # ref ele for CRE mod (max sta ele)
self.grd_ele = 1.5    # typical station elevation
# loc params
self.num_workers = 10
self.ztr_rng = np.arange(0,20,1)
self.p_wht = 0    # weight code index
self.s_wht = 1
self.rms_wht = '4 0.3 1 3'
self.dist_init = '1 60 1 2'
self.dist_wht = '4 40 1 3'
self.wht_code = '1 0.6 0.3 0.2'
self.pmod = 'input/velo_p_eg.cre'
self.smod = [None, 'input/velo_s_eg.cre'][0]
self.pos = 1.73    # provide smod or pos
```

Write CRE velocity model

HK Model with V_P/V_S Ratio of 1.73		
Depth to Top of Layer (km)	CRE Interfaces	P -Velocity (km/sec)
0.0	0.0	5.5
5.5	6.5	6.3
16.0	17.0	6.7
32.0	33.0	7.8

```

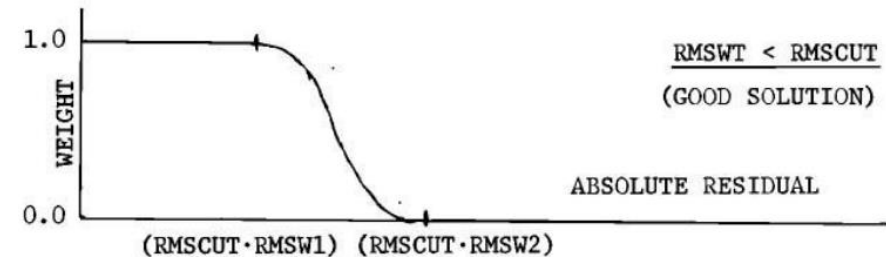
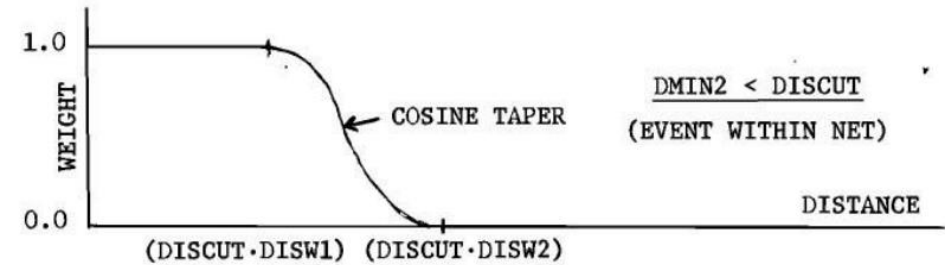
# geo ref
self.lat_code = 'N'
self.lon_code = 'W'
self.ref_ele = 2.5 # ref ele for CRE mod (max sta ele)
self.grd_ele = 1.5 # typical station elevation
# loc params
self.num_workers = 10
self.ztr_rng = np.arange(0,20,1)
self.p_wht = 0 # weight code index
self.s_wht = 1
self.rms_wht = '4 0.3 1 3'
self.dist_init = '1 60 1 2'
self.dist_wht = '4 40 1 3'
self.wht_code = '1 0.6 0.3 0.2'
self.pmod = 'input/velo_p_eg.cre'
self.smod = [None, 'input/velo_s_eg.cre'][0]
self.pos = 1.73 # provide smod or pos

```

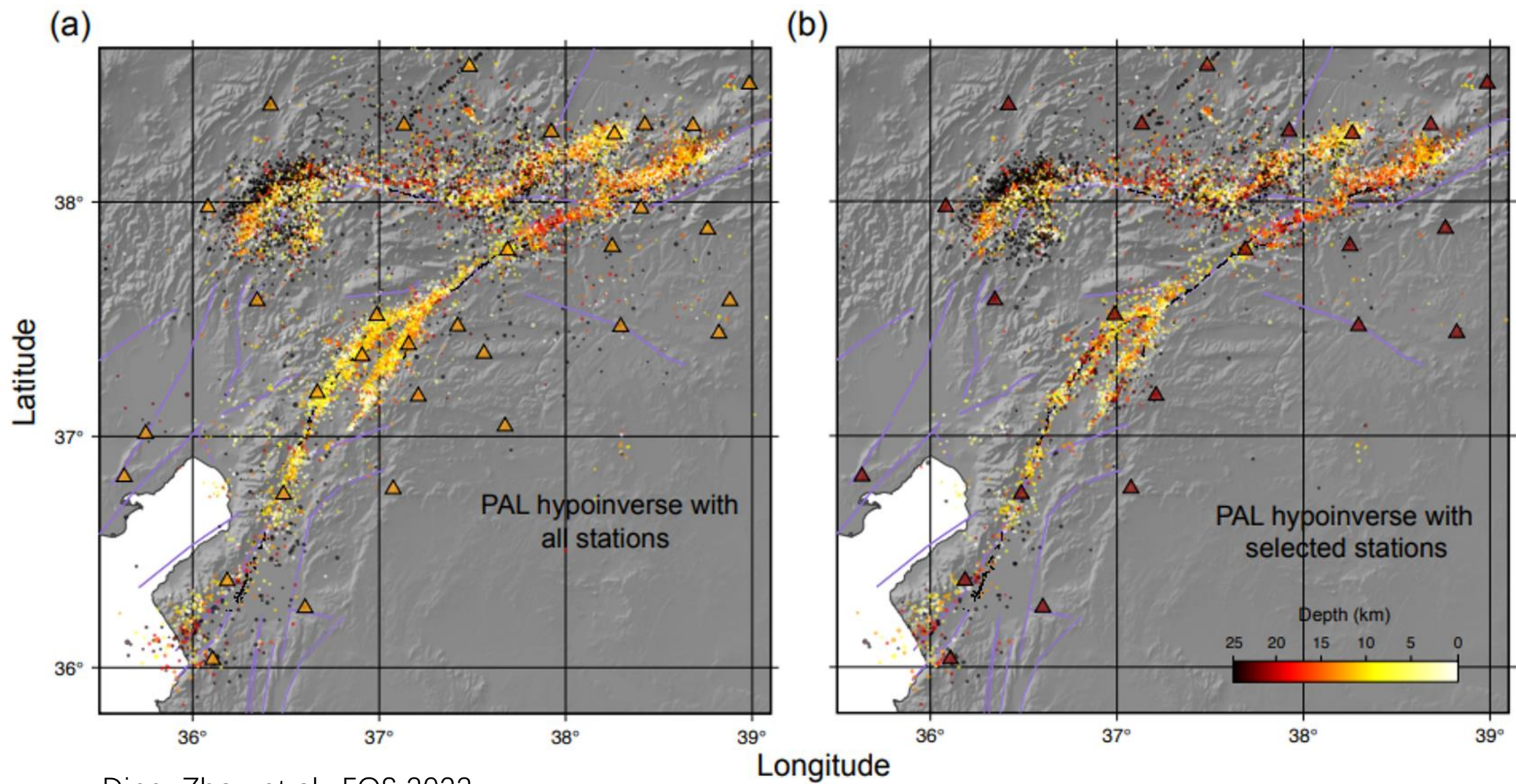
- initial distance weight for lateral location: use all stations <120 km
- use near-source stations (<40 km) to constrain depth in later iterations

Weighting scheme

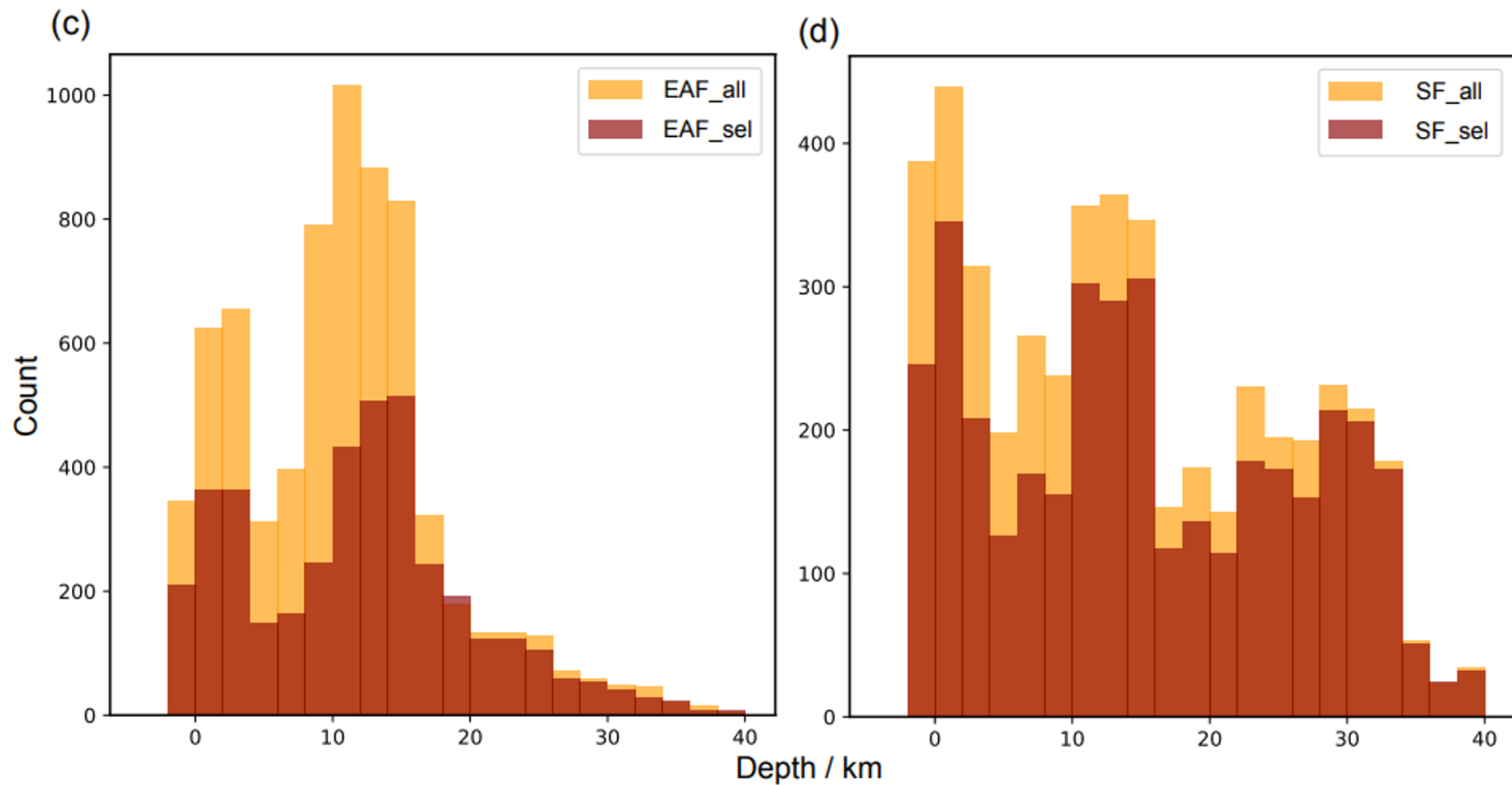
- Distance weighting
 - $< \text{min_dist}$: full weight
 - $> \text{max_dist}$: zero weight
 - $\text{min_dist} \sim \text{max_dist}$: cos taper
- Residual weighting
 - $< \text{min_res}$: full weight
 - $> \text{max_res}$: zero weight
 - $\text{min_res} \sim \text{max_res}$: cos taper



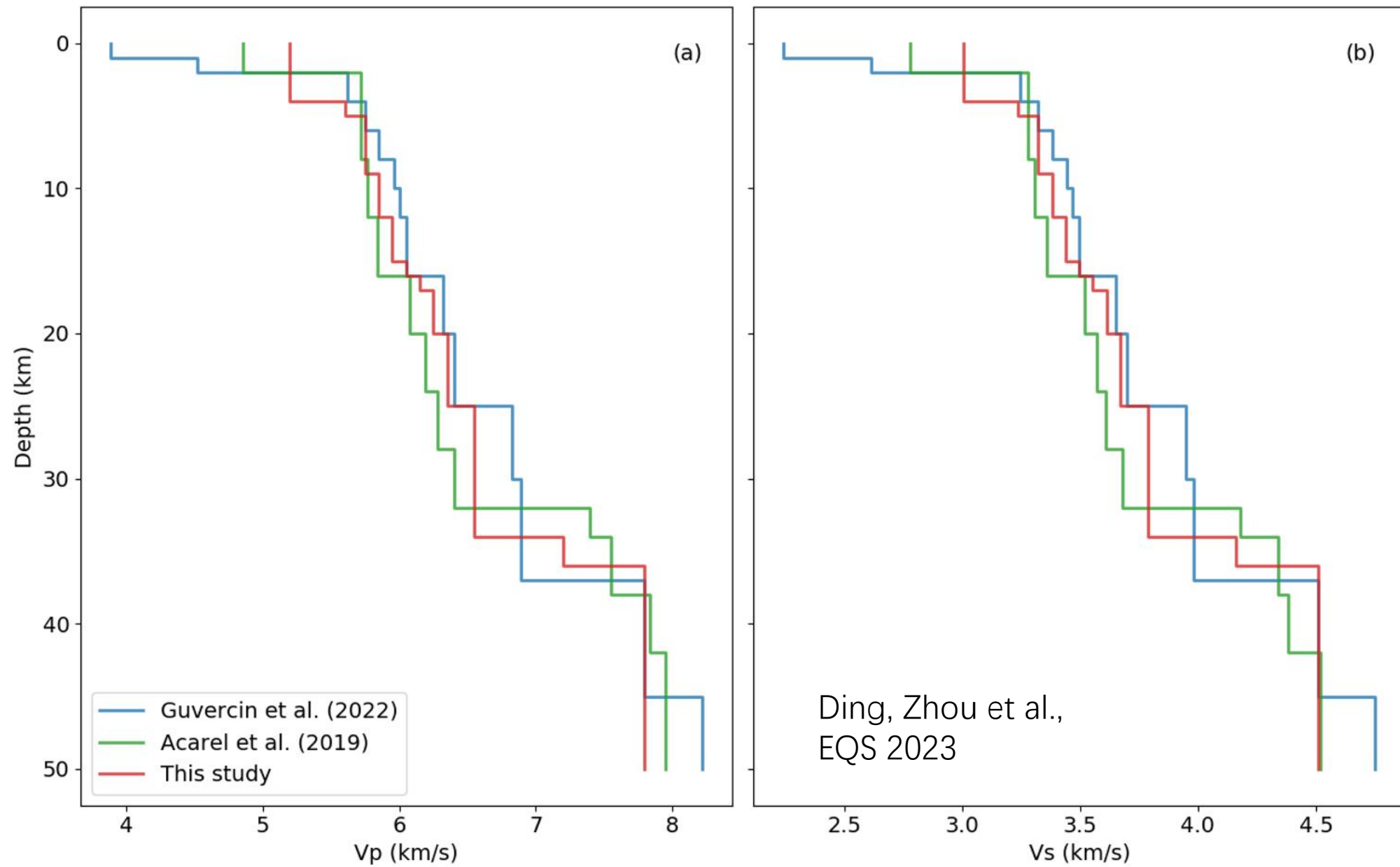
Klein, 2014

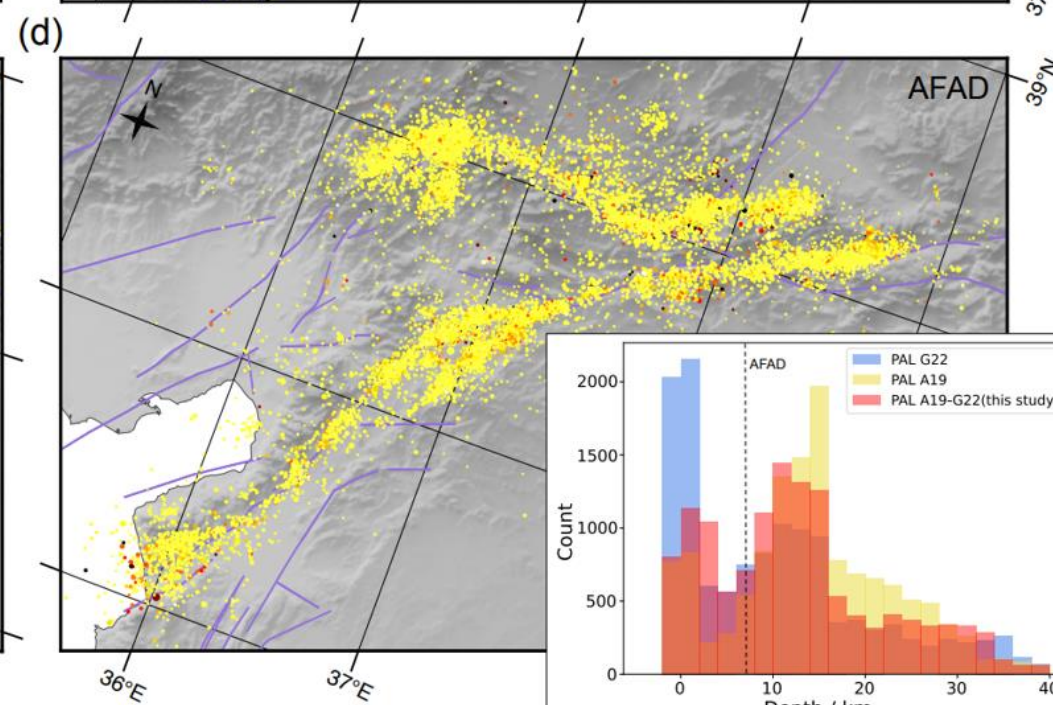
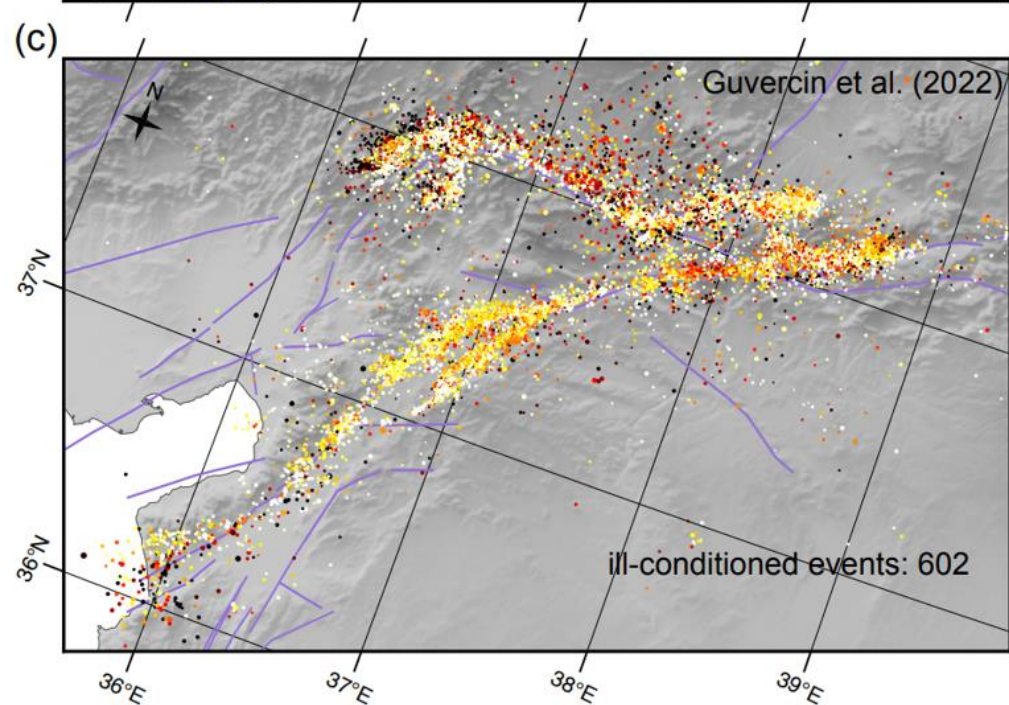
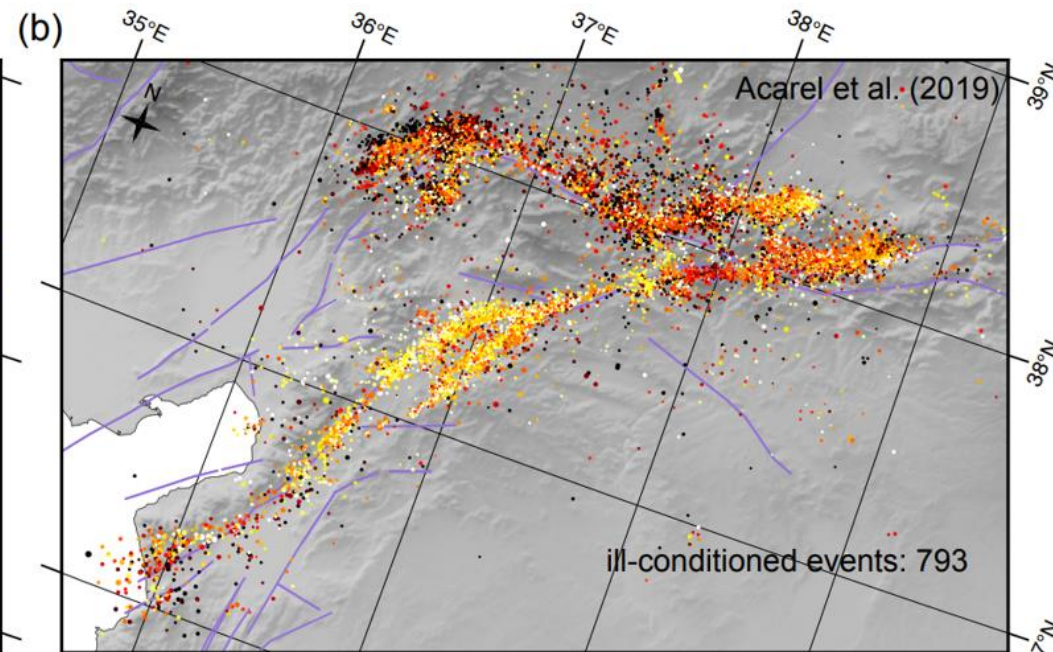
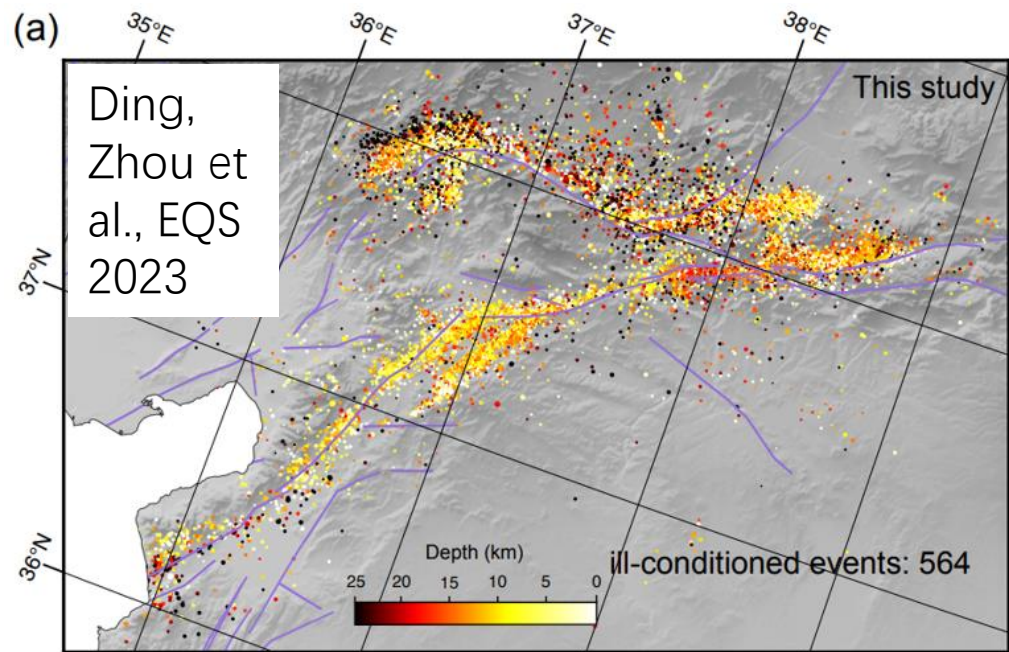


Ding, Zhou et al., EQS 2023



Ding, Zhou et al., EQS 2023





Tuning velocity model

- Find available models and check their consistency
 - best: from active source (direct observation)
 - good: ambient noise tomography (independent result)
 - else: Velest, TomoDD ... (similar data & coupled with location)
- Avoid sudden jump between layers
 - if so, you will see depth concentration
- See if too many “airquakes” appear
 - if so, make shallow layers higher velocity
- A good velocity model usually gives small number (<10%) of bad-located events (needs trial and error, sadly)

Run hypoDD

- Station file
- Phase file (*_full.pha*)
 - event line: ot, lat, lon, dep, mag, *evid*
 - phase line: net.sta, tp, ts
- Location parameters
 - ph2dt.inp
 - hypoDD.inp



<https://github.com/YijianZhou/Hypo-Interface-Py>

hypoDD -- A Program to Compute Double-Difference Hypocenter Locations

(*hypoDD* version 1.0 - 03/2001)

by

Felix Waldhauser

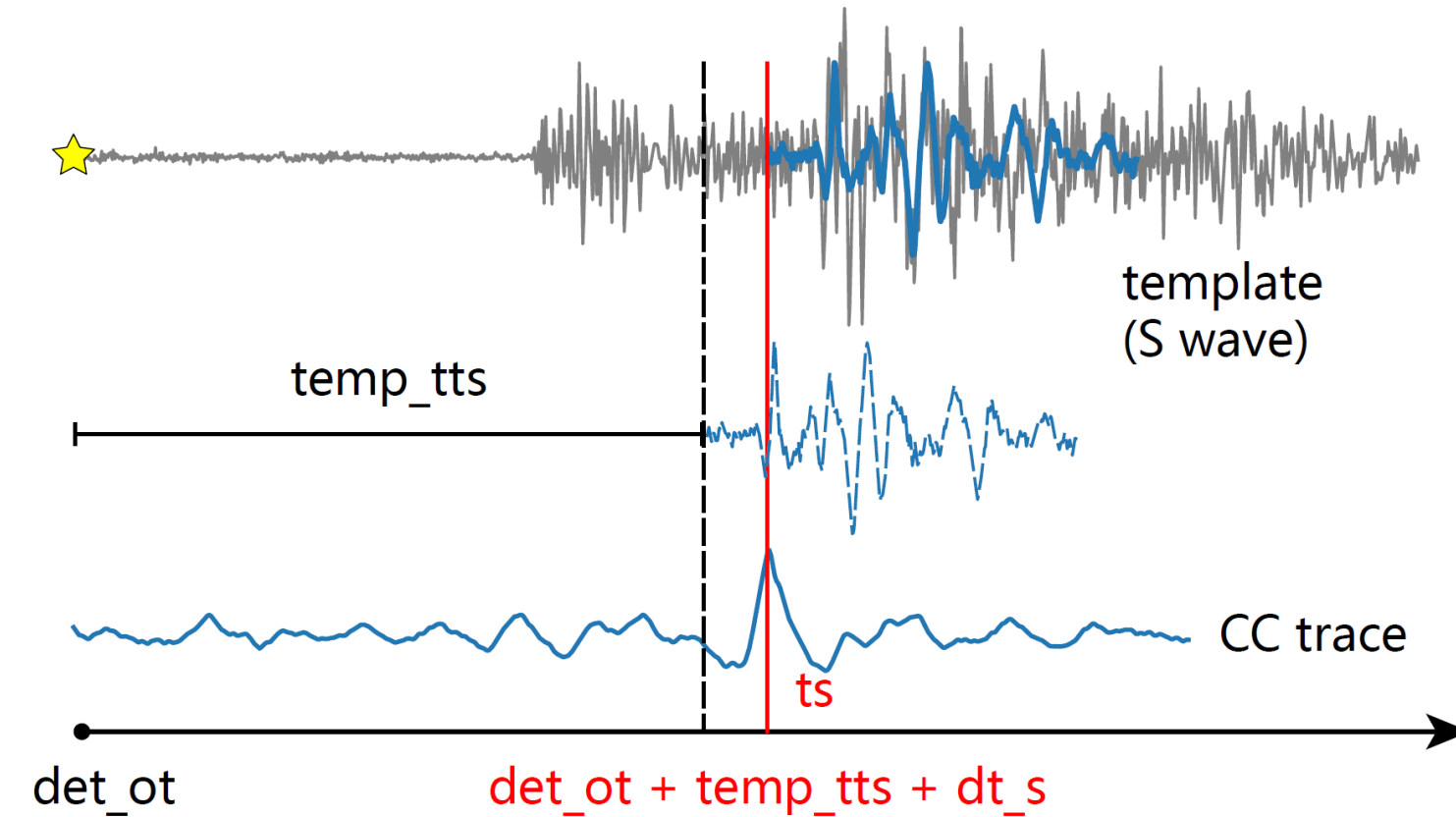
U.S. Geol. Survey
345 Middlefield Rd, MS977
Menlo Park, CA 94025
felix@andreas.wr.usgs.gov

**Please read
this document!**

Run hypoDD

```
# 2. ph2dt_cc
# 2.1 event linkage (initial calc & further selection)
self.cc_thres = [0.3, 0.3] ..... # CC thres for event pair
self.loc_dev_thres = [3, 3] ..... # km, maximum x-y location separation
self.dep_dev_thres = [4, 4] ..... # km, maximum depth separation
self.dist_thres = [100, 80] ..... # km, max epicentral dist
self.dt_thres = [[0.6, 1.], [0.5, 0.8]] ..... # sec, max P & S dtime
self.num_sta_thres = [4, 4] ..... # min sta_num for one event pair
self.max_sta = 15 ..... # max sta_num for one event pair
self.max_nbr = 200 ..... # max number of neighbor event
self.temp_mag = 0 ..... # min mag for templates
self.temp_sta = 4 ..... # min sta_num for templates
# 2.2 data preprocess
self.freq_band = [2., 20.]
self.samp_rate = 100
self.chn_p = [2] ..... # chn for P picking
self.chn_s = [0, 1] ..... # chn for S picking
self.win_temp_p = [0.5, 2.]
self.win_temp_s = [0.2, 3.8] ..... # pre-post phase arrival
self.win_event = [5, 20] ..... # event data cutting, just long enough
```

CC measurement of differential travel time



template event - $temp$
newly detected event - det

0. Detect on det_ot
on stacked CC trace
1. Pick ts by CC
 $tts = ts - det_ot$
2. Travel time difference
 $dt_s = tts - temp_tts$

Note 1. error in det_ot does not matters.

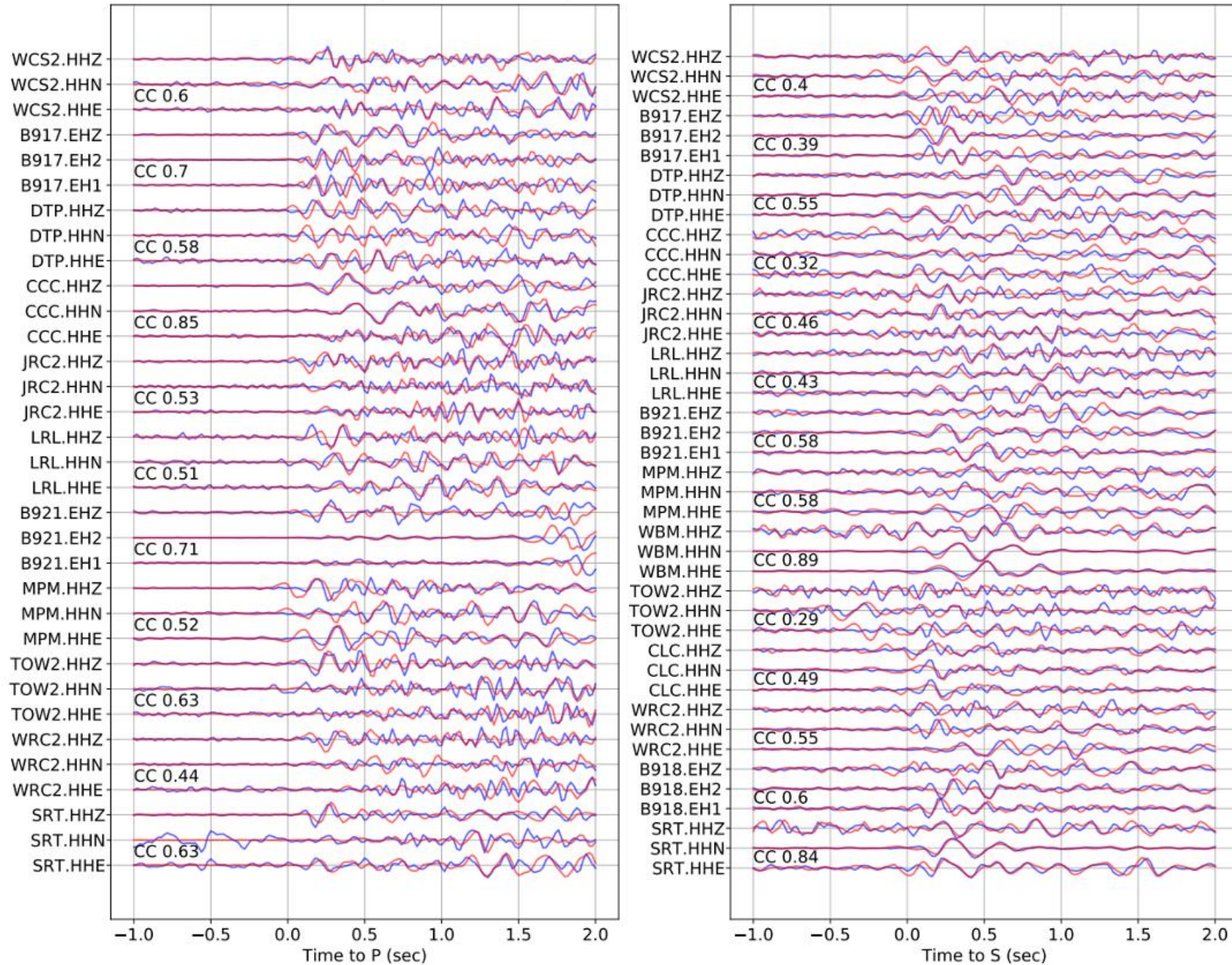
For multi-stations:

$[dt_s] = [ts - det_ot] - temp_tts$,
where det_ot is to be relocated

Note 2. picking error does not effect the dt measurement.

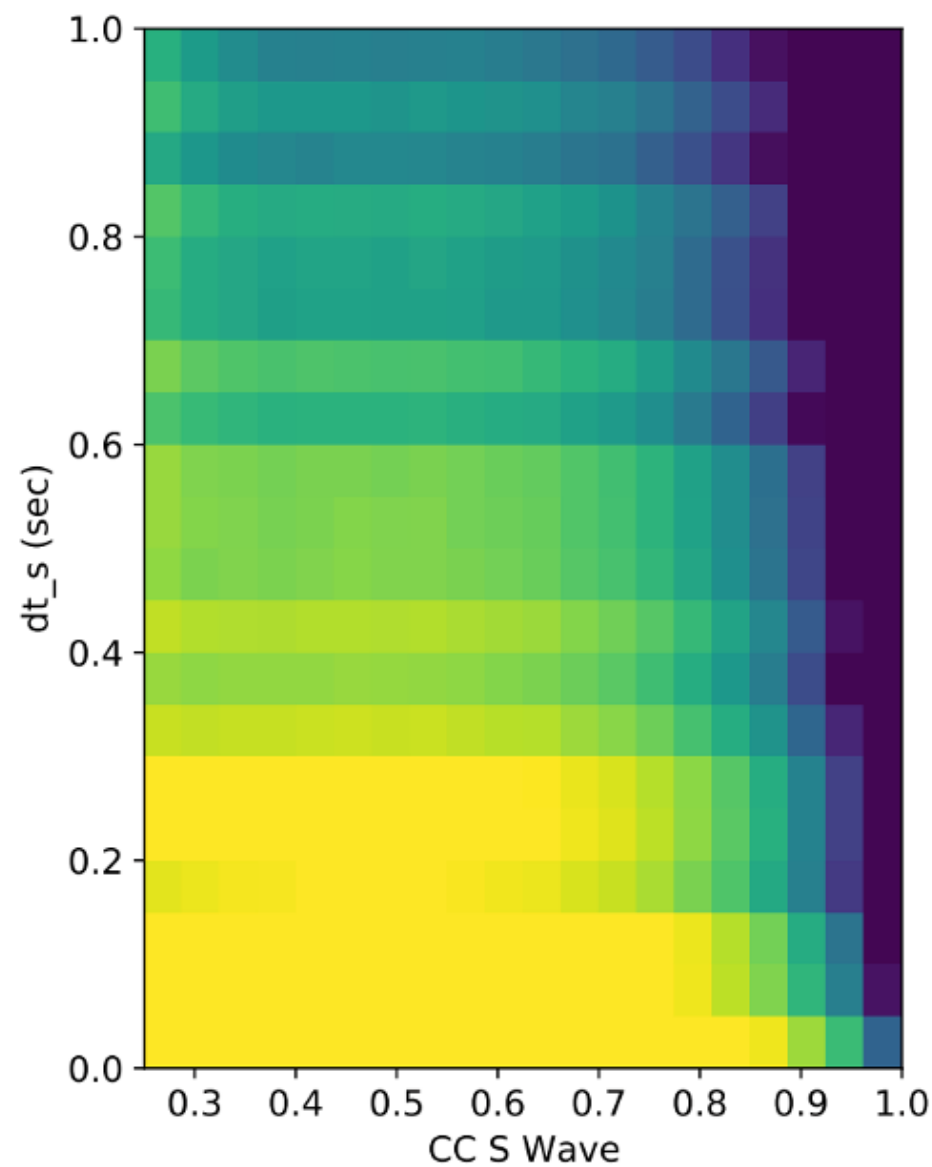
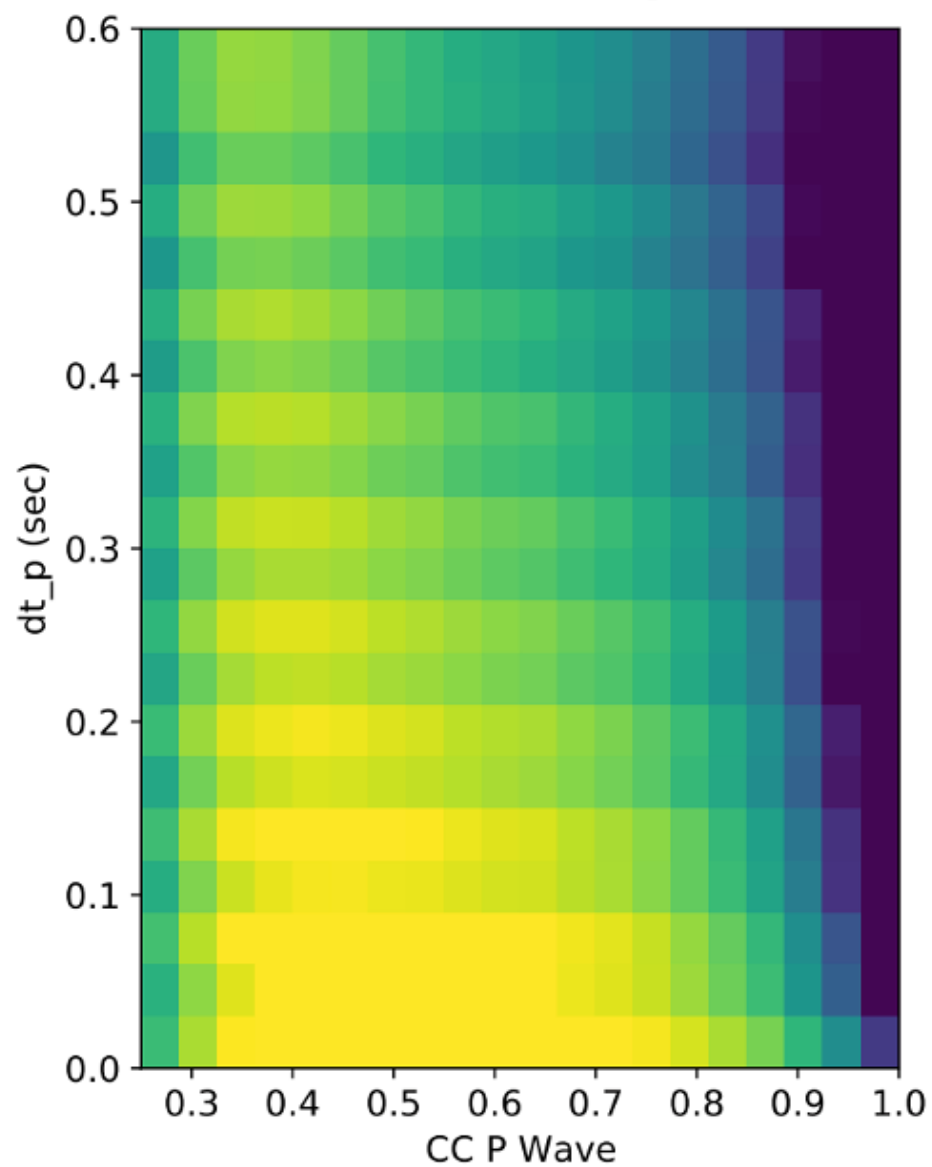
ts and $temp_tts$ share the same error

Example Waveform Alignment (MESS dt.cc): evid 11 to 0



Zhou et al.,
SRL 2021

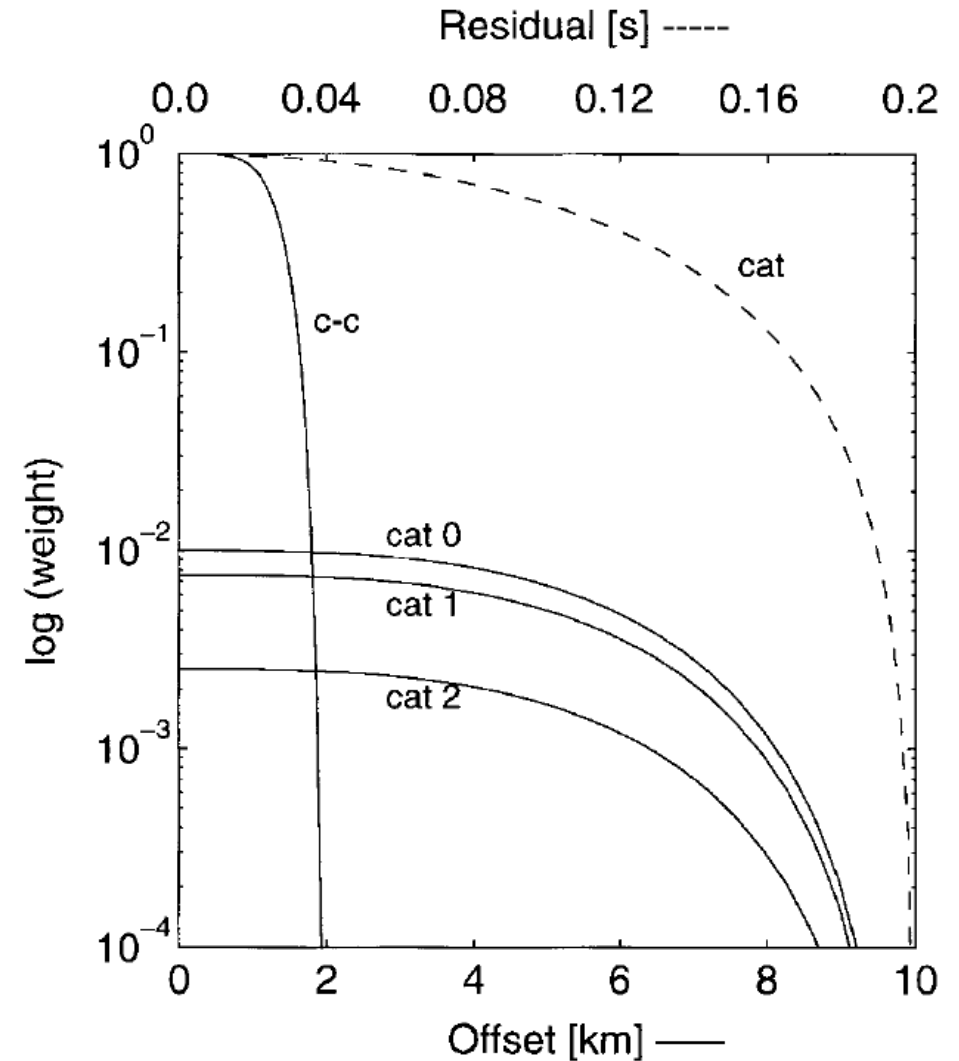
Ridgecrest MESS: Differential Time - CC



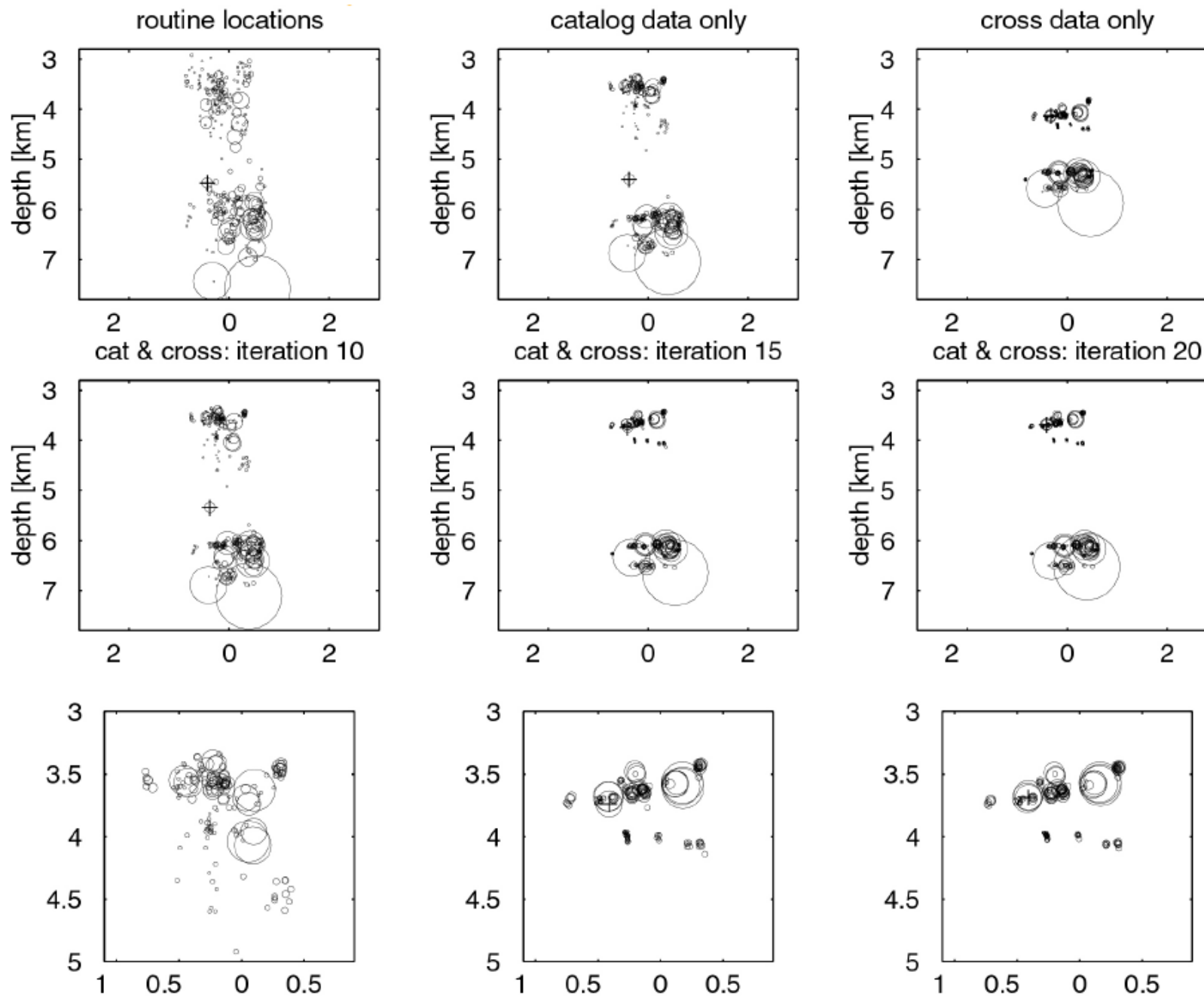
Zhou et al.,
SRL 2021

Weighting scheme

- Combination of *dt.ct* & *dt.cc*
- catalog-based dtime: *dt.ct*
 - wider spread
 - low resolution
- CC-based dtime: *dt.cc*
 - small interevent distance
 - high resolution
 - need correlated waveform



Waldhauser &
Ellsworth, BSSA 2000



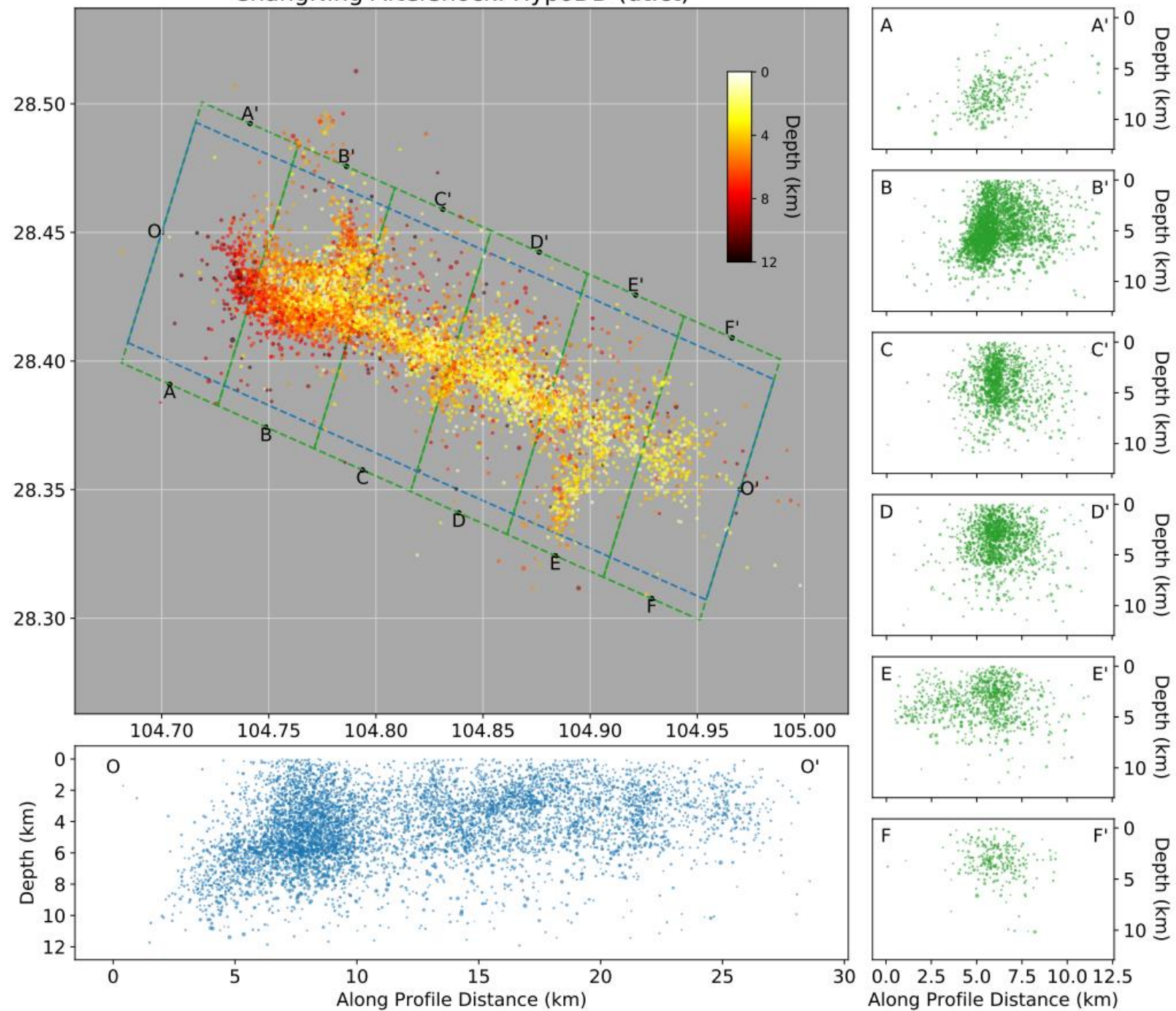
Waldhauser,
2001

Weighting scheme

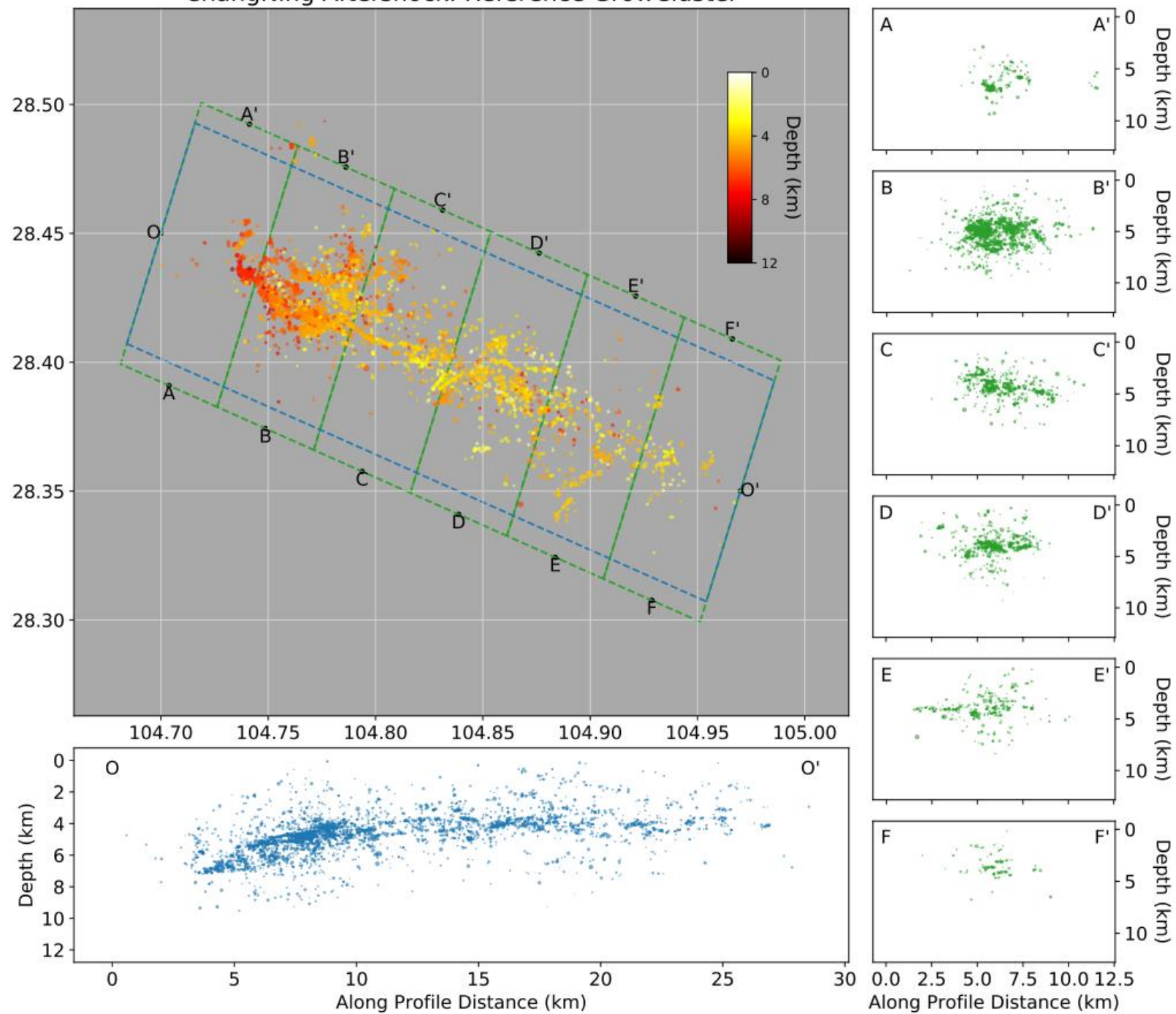
	Cross correlation data				Catalog data			
Iterations	A priori, P-wave WTCCP	A priori, S-wave WTCCS	Misfit weight (residual cutoff, factor times SD) WRCC	Dist. Weight (seperation in km) WDCC	A priori, P-wave WTCTP	A priori, S-wave WTCTS	Misfit weight (residual cutoff, factor times SD) WRCT	Dist. Weigh (seperation in km) WDCT
1-5					1.0	0.5	-9	-9
6-10					1.0	0.5	6	4
11-15	1.0	0.5	-9	2				
16-20	1.0	0.5	6	2				
21-25	1.0	0.5	6	0.5				

Waldhauser, 2001

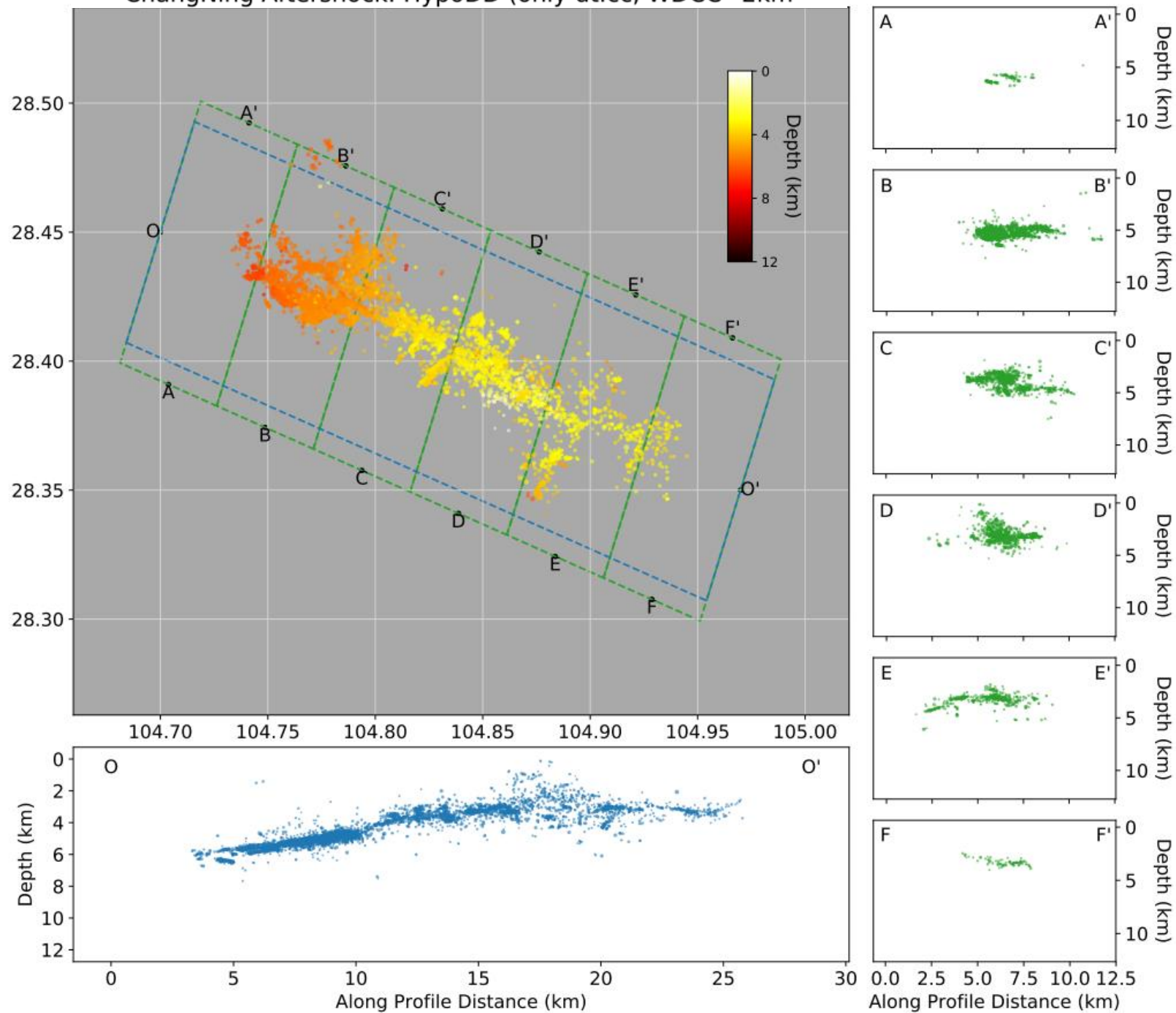
ChangNing Aftershock: HypoDD (dt.ct)



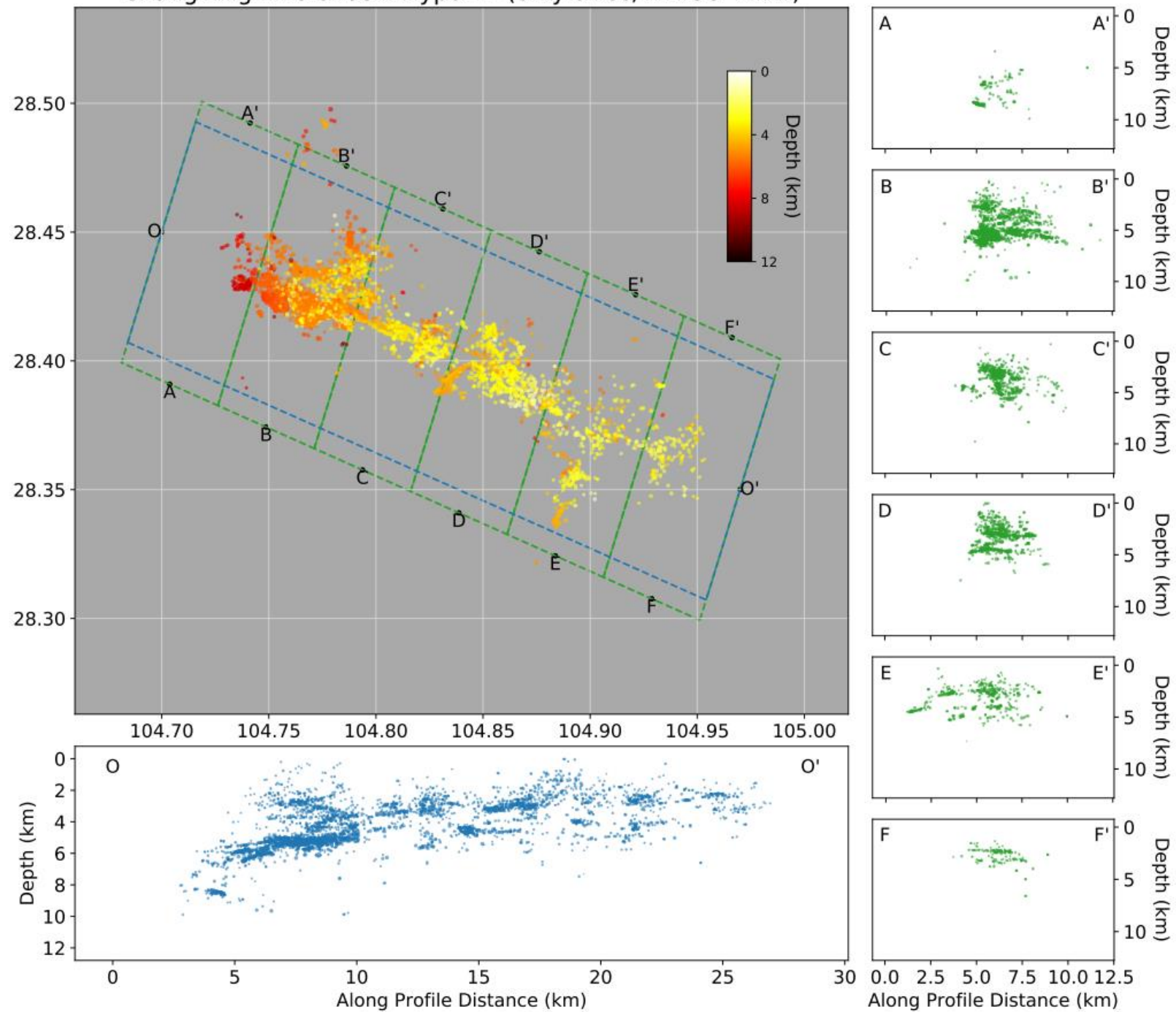
ChangNing Aftershock: Reference GrowCluster



ChangNing Aftershock: HypoDD (only dt.cc, WDC=2km)



ChangNing Aftershock: HypoDD (only dt.cc, WDCC=1km)



Understand hypoDD screen output

```
Reading data ... Mon Dec 28 15:12:00 2020
# events = 8201
# stations < maxdist = 70
# catalog P dtimes = 282623
# catalog S dtimes = 273463
# dtimes total = 556086
# events after dtype match = 7611
# stations = 68

no clustering performed.

RELOCATION OF CLUSTER: 1 Mon Dec 28 15:12:02 2020
-----
Initial trial sources = 7611
```

IT	EV	CT	RMSCT		RMSST	DX	DY	DZ	DT	OS	AQ	CND
	%	%	ms	%	ms	m	m	m	ms	m		
1	100	100	177	-23.2	0	389	424	639	92	0	1	147
2	1	100	100	177	-0.0	861	389	423	638	92	64	147
3	100	99	159	-10.1	861	116	145	299	34	64	16	147
4	2	99	99	159	-0.2	778	116	144	288	34	156	147
5	99	98	154	-2.8	778	64	79	183	18	156	17	142
6	3	99	98	154	-0.3	877	64	79	171	18	214	142
7	99	97	152	-1.3	877	43	54	137	12	214	14	140
8	4	98	97	152	-0.2	694	43	54	126	12	252	140

Reading data ... Mon Dec 28 15:27:43 2020

events = 7479

stations < maxdist = 70

cross corr P dtimes = 535767 (no OTC for 0 event pairs)

cross corr S dtimes = 515115 (no OTC for 0 event pairs)

catalog P dtimes = 202292

catalog S dtimes = 198327

dtimes total = 1451501

events after dtime match = 7084

stations = 68

no clustering performed.

RELOCATION OF CLUSTER: 1 Mon Dec 28 15:27:52 2020

Initial trial sources = 7084

IT	EV	CT	CC	RMSCT		RMSCC		RMSST	DX	DY	DZ	DT	OS	AQ	CND	
	%	%	%	ms	%	ms	%	ms	m	m	m	ms	m			
1	100	100	100	150	-27.7	240	-7.8	0	128	126	278	113	0	4	380	
2	1	100	100	100	150	-0.0	240	-0.0	1302	128	124	258	113	45	0	379
3		97	96	85	134	-10.9	231	-3.7	1302	116	113	246	31	45	4	361
4	2	97	96	85	133	-0.9	224	-2.9	645	116	113	244	31	43	0	357
5		95	91	75	126	-4.9	209	-6.7	645	67	68	168	17	43	2	327
6	3	95	91	75	125	-1.0	204	-2.7	1041	68	68	166	17	52	0	330
7	4	94	88	69	122	-2.4	199	-2.5	600	48	49	131	13	65	0	306
8	5	93	84	65	155	27.3	161	-18.9	625	239	249	397	56	86	0	334
9		92	78	54	151	-3.0	85	-47.5	625	126	134	262	32	86	4	309
10	6	91	78	50	146	-3.1	64	-23.9	576	121	128	247	31	84	0	311
11		91	76	43	136	-6.5	38	-41.7	576	69	69	152	18	84	1	303
12	7	91	76	40	134	-1.6	29	-23.3	565	68	68	146	18	85	0	304
13	8	90	75	36	130	-3.4	21	-27.4	571	40	41	90	11	85	0	290
14	9	90	74	34	127	-2.2	17	-17.7	572	29	28	64	7	87	0	280
15		89	73	33	125	-1.3	15	-10.0	572	22	21	49	6	87	1	268
16	10	89	73	33	125	-0.4	15	-2.3	557	22	21	48	6	89	0	271

Window is important in CC

- P wave: relatively clear arrival
- S wave: large amplitude, more stable
- Test on 2021 Shimian sequence
 - $p_win = [0.5, 2.]$ & $[0., 2.] \rightarrow \sim 24\% |\Delta dt_p| > 0.05s$
 - $p_chn = [Z]$ & $[E, N, Z] \rightarrow \sim 16\% |\Delta dt_p| > 0.05s$
 - as a comparison, S wave in win & chn test all $< 4\%$
- Conclusion
 - p_win starts before P; use Z-channel for dt_p measurement
 - $s_win > p_win$; dt_s should have higher weights

Outline

- Prepare your data
 - data download & cleaning
 - station distribution, data continuity
- Run your workflow
 - picking & association: window length, time residual, number of station ...
 - location: velocity model, distance weight, WDCT/C
- **Check your result**
 - **number, location, time, magnitude**
 - **compare with a reference, check whether physically interpretable**



Third Fire

08-15 01:31:37

师兄我地震数量只有6万个，但是分网格分成多大都会有increase MEV in hy/ph .inc，哪怕分成40,40都有这个🤔，hypodd.inc用的您给的参数，ph2dt.inc用的他自带的参数



周一剑

08-15 18:37:14

背景介绍一下



Third Fire

08-15 18:49:13

秭归三峡地区，主要断裂是下方的NNW仙女山断裂和NWW九畹溪断裂



周一剑

08-15 18:52:47

为什么有这么多地震？



Third Fire

08-15 18:55:53

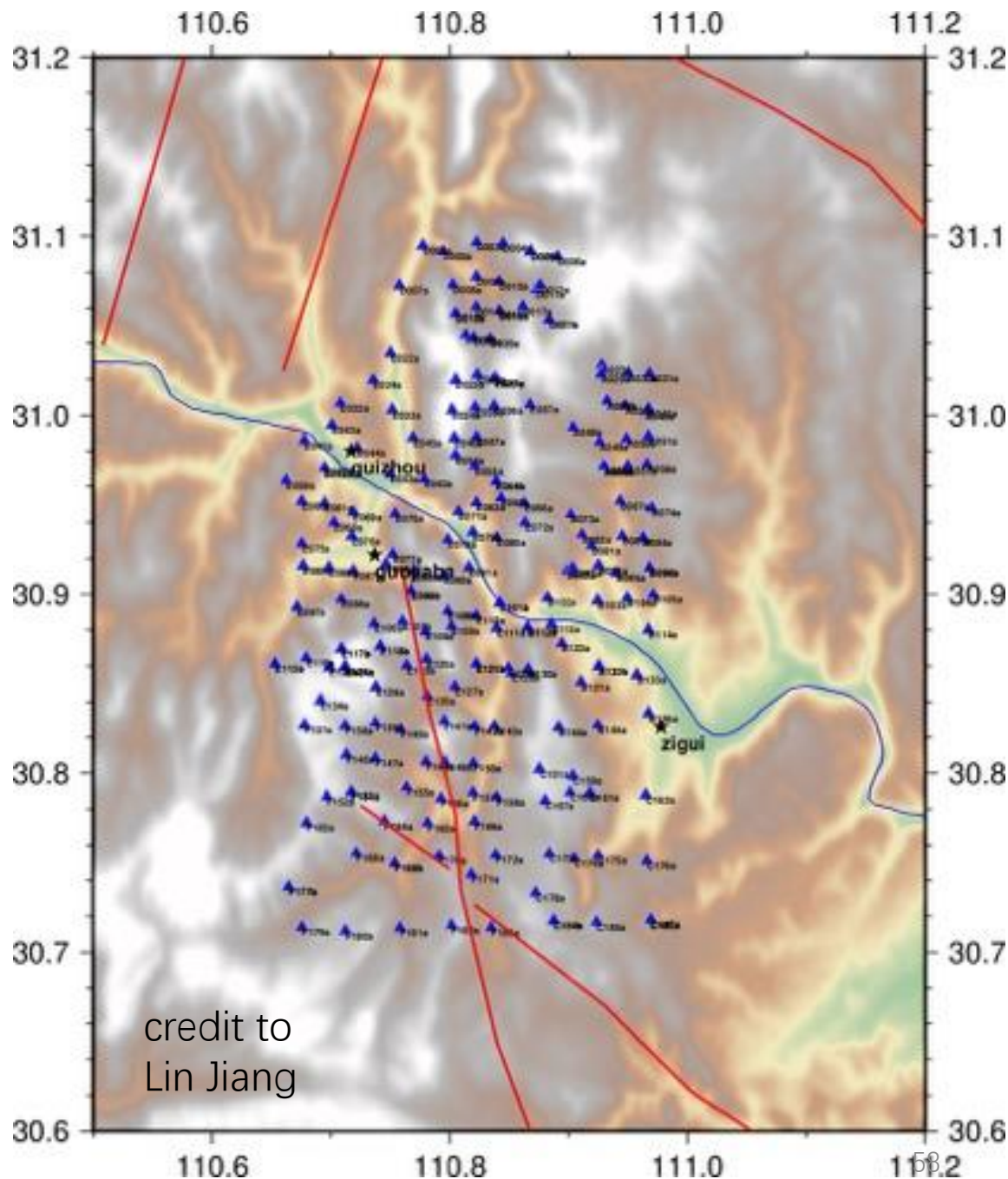
两年半有六万个小震微震我感觉还挺正常吧🤔



周一剑

08-15 18:56:11

不正常



Check the numbers during (re)location

- HypoInverse
 - number of bad-located events (*should be <10%*)
 - *calculate average location error for paper writing
- HypoDD
 - number of dropped events (*should be <10-15% if dt.ct, <20-25% if dt.cc*)
 - in the screen output, no * appears, and DX/Y/Z/T should always decrease
 - *calculate average relocation error for paper writing
- **Empirically, location uncertainties mainly comes from **velocity model** and **distance weighting**. Different location results may have similar mathematical location error**

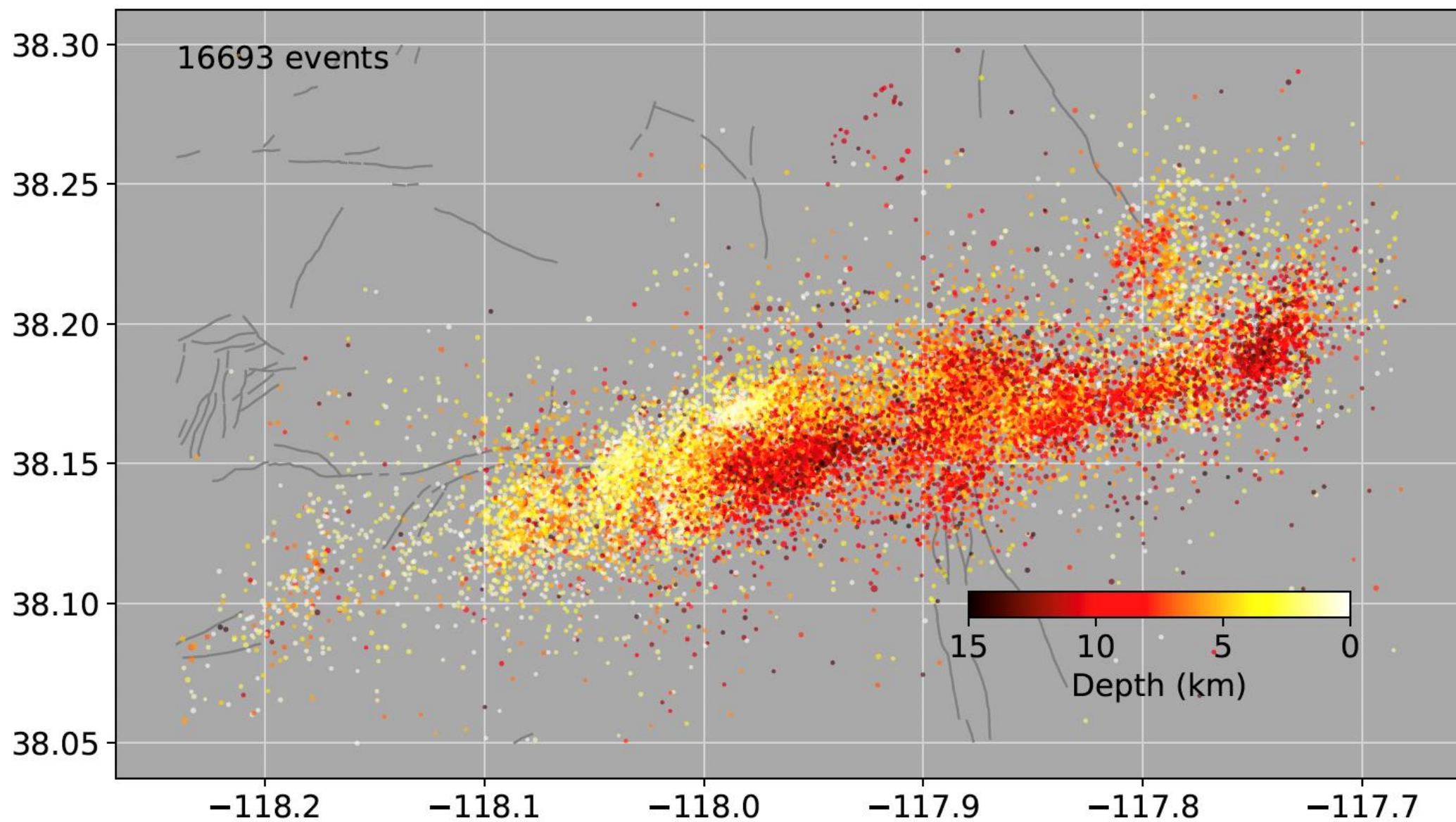
Check the ratio of associated picks

Pick		Association		Location	
Algorithm	Count (totals; ps-pairs)	Algorithm	Events / Picks	Algorithm	Bad; Drop
PAL	920,040; 460,020	PAL	13,585 / 81,513	hypoINV; hypoDD	419; 1056
PhaseNet	2,401,677; 1,019,414	PAL	36666 / 249,851	same as above	1445; 3925
		GaMMA	24,824 / 141,630 (ps-pair)		1848; 5710
PhaseNet (highpass)	2,915,602; 1,225,835	PAL	40,662 / 270,879	same as above	1795; 4772
		GaMMA	26,109 / 148,213		2217; 6407

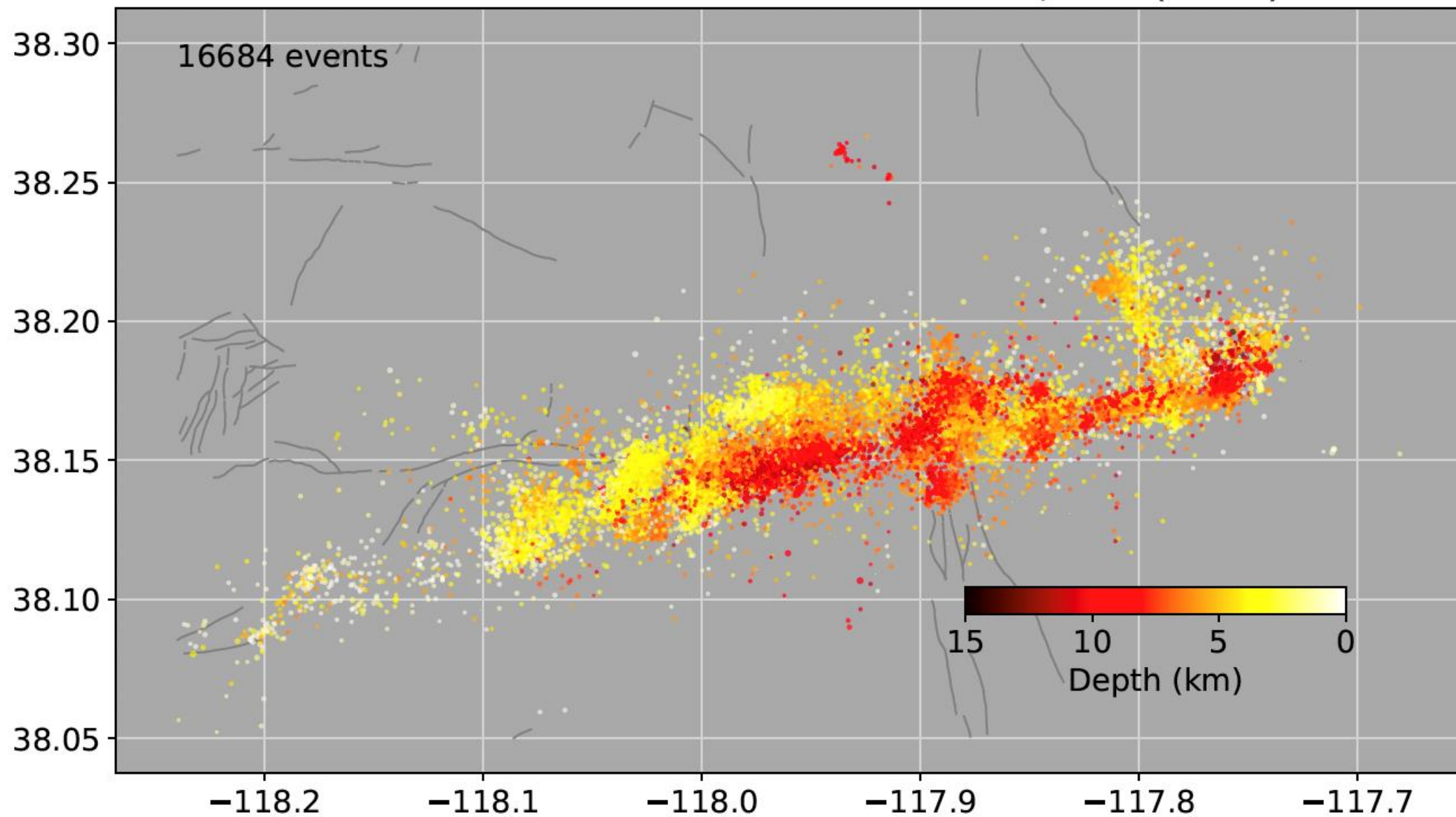
credit to Hongyang Ding

- Usually >80% PhaseNet picks should come in P&S pairs
- If >10% P&S pairs associated, data quality should be okay, and event detection should be accurate enough

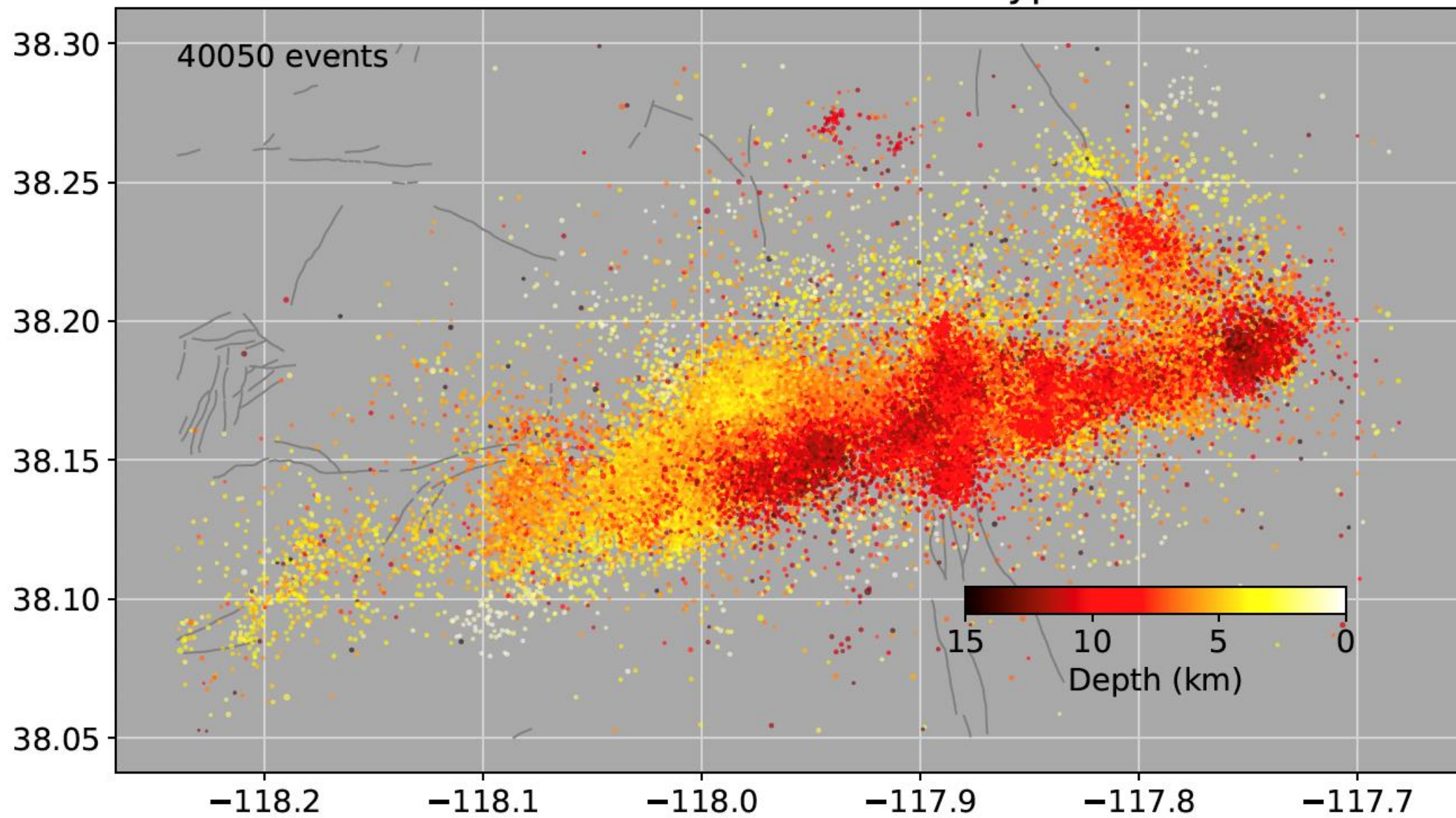
Mount Cristo 2020 Swarm: ANSS ComCat



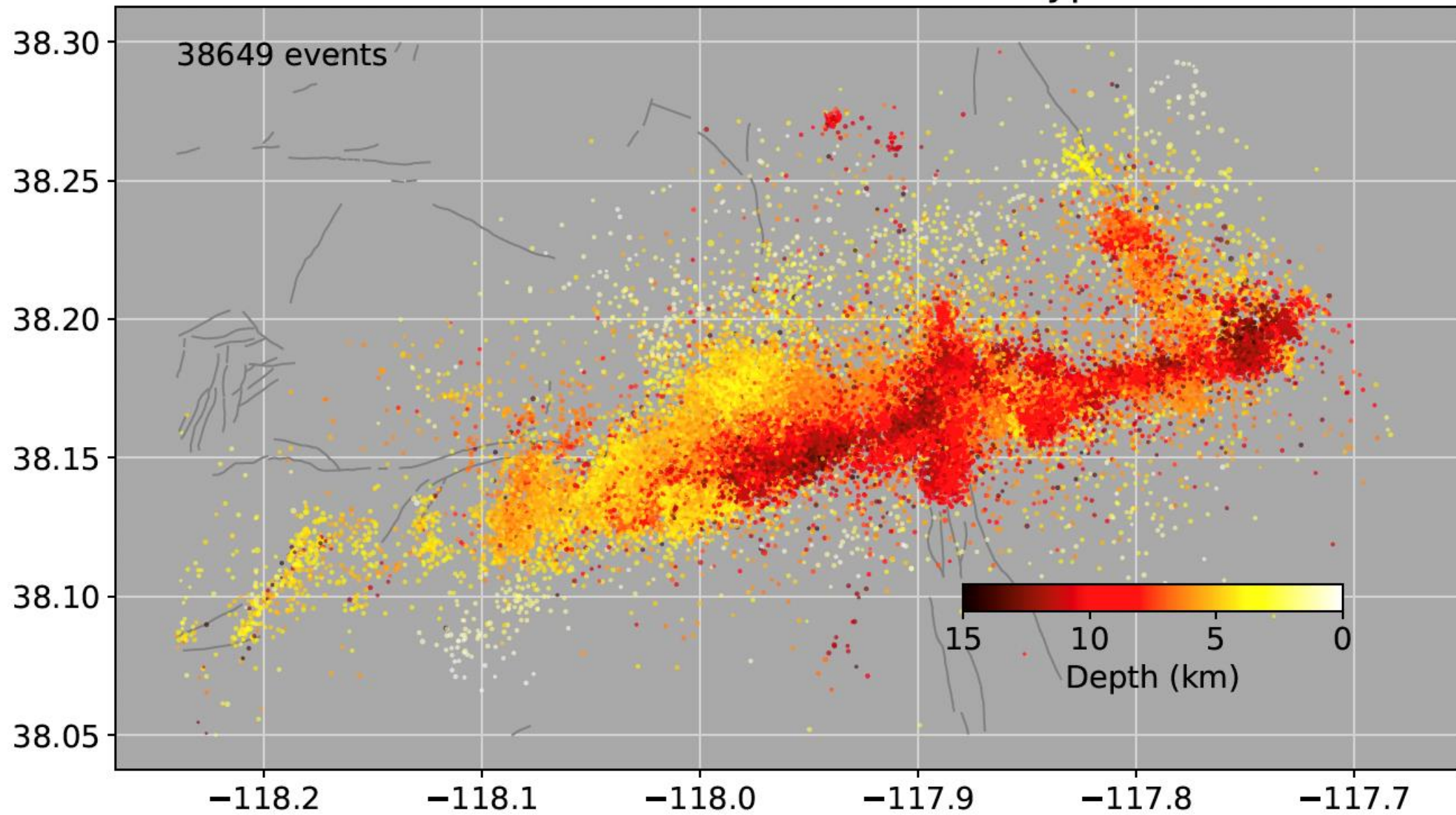
Mount Cristo 2020 Swarm: Rhul et al., SRL (2020)



Mount Cristo 2020 Swarm: PAL HypoInverse



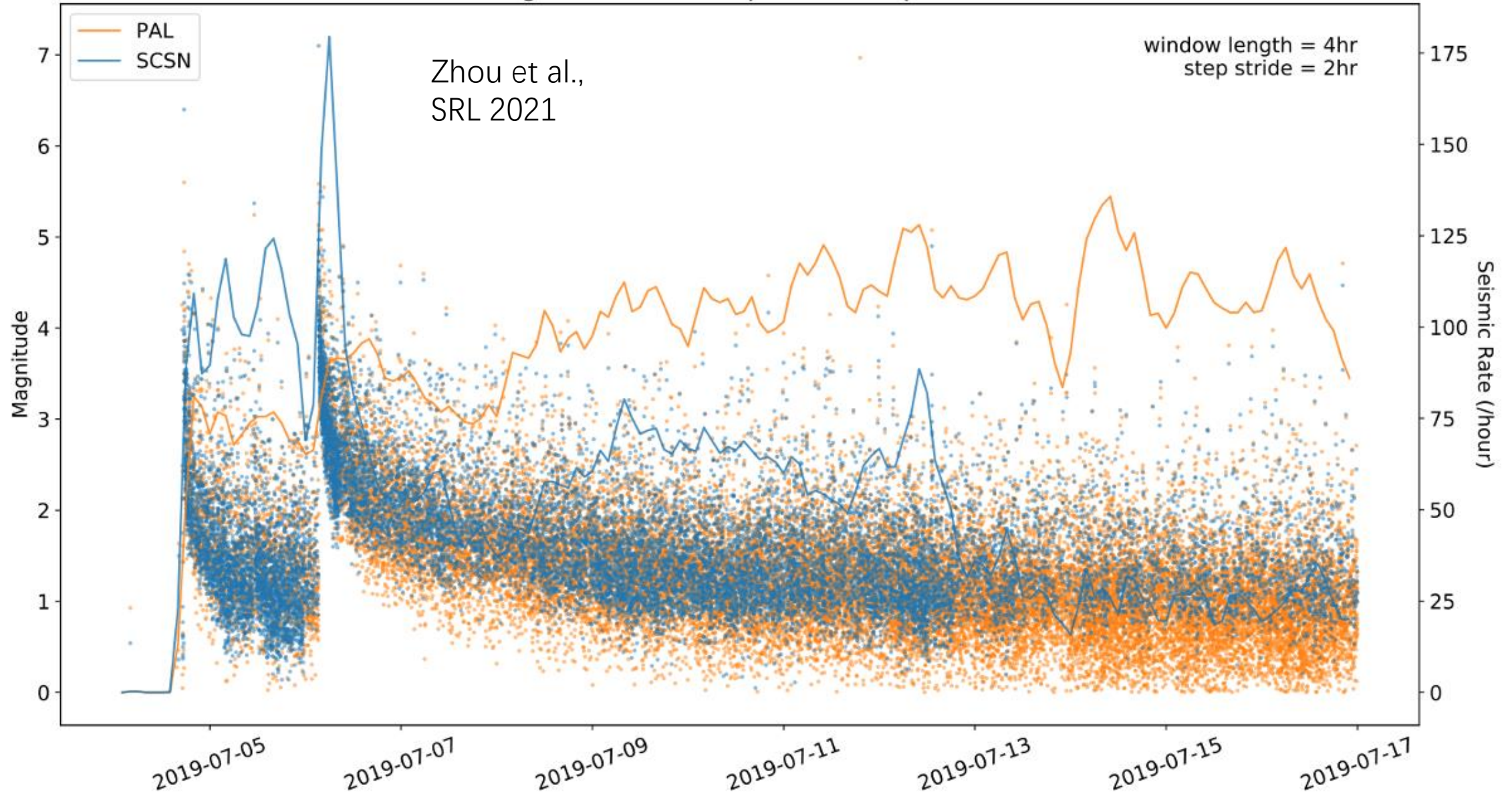
Mount Cristo 2020 Swarm: PAL HypoDD

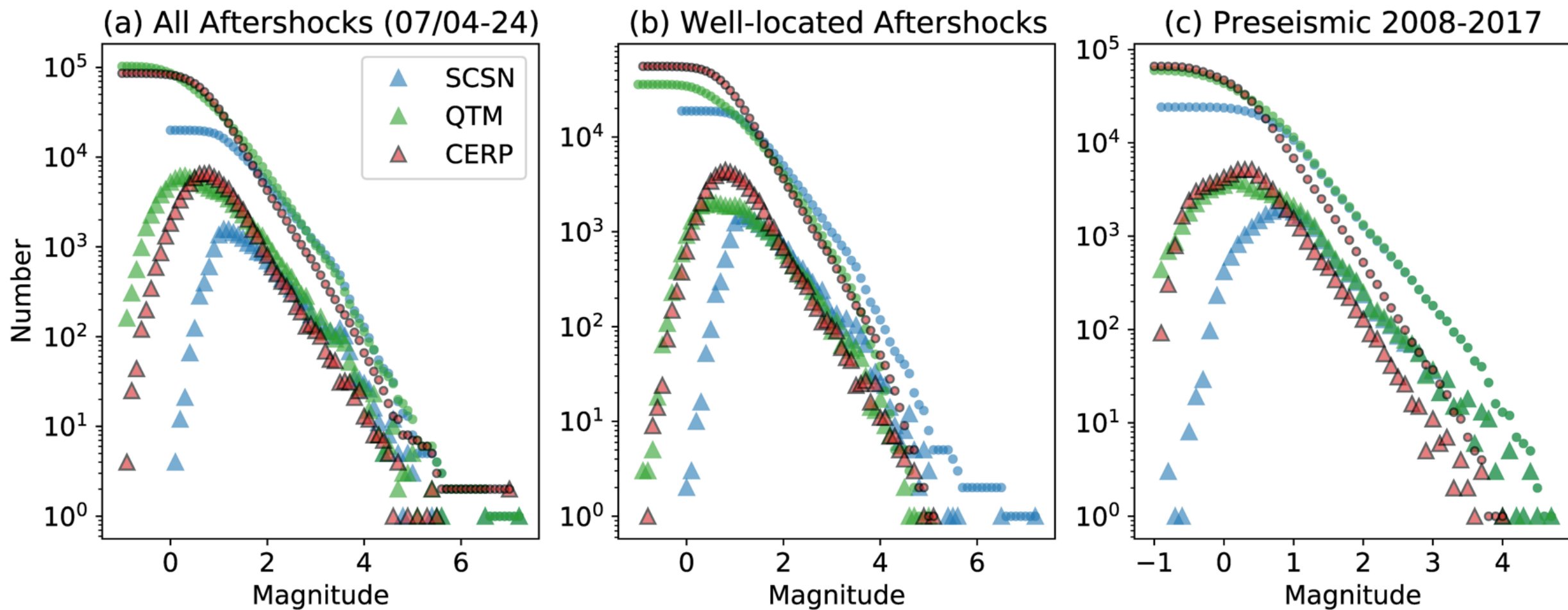


Interpretation of seismicity distribution

- Alignment with **fault traces** & **surface rupture** (if available)
- If it reveals unmapped fault, can **topography** give a hint?
- Continental strike-slip faults are usually near vertical, but not always (you do see cases where seismicity distributes “*off fault*”)
- → use **focal mechanism solution** to support your interpretation
- Remove possible artifacts before interpreting depth distribution
- Draw surface traces first, then make **fault-normal** profiles:
measure fault dip on a **1:1-scale plot**!

Magnitude-Time Sequence Comparison





Summary

- Prepare your data
 - data download & cleaning
 - station distribution, data continuity
- Run your workflow
 - picking & association: window length, time residual, number of station ...
 - location: velocity model, distance weight, WDCT/C
- Check your result
 - number, location, time, magnitude
 - compare with a reference, check whether physically interpretable