pipeline script

```
44  """ MOGABA pipeline script """
45  Run_CSFit = True
46  Auto_Flag = True
47  SaveCSLog = True
48  SaveCSFit = True
49  SavePSLog = True
50  LR_Swap   = False
51
52
53  antenna = 'tn'.upper()
54  station = 'K%s'%(antenna)
55  nw, nr  = 5*2, 5000
56  Polnum  = 0
```

Toggle Options

- Run_CSFit : cross-scan fitting
- Auto_Flag : flag bad scan(s) automatically
- SaveCSLog : save cross-scan log (Tsys, tau, el)
- SaveCSFit : save cross-scan fitting results
- SavePSLog : same with ‘SaveCSLog’ but for position-switching
- LR_Swap : interchange LCP-RCP
- nw, nr : number of walkers & repeats for MCMC in cross-scan fitting
- Polnum : selecteing polarization (for KQ data, 0=K&Q // 1=K // 2=Q)

pipeline script

```
58  flag_scan1 = [
59  # flagging-channels for Polnum==1
60  # 'scannum bad_channel(s)'
61  '123456 1 2 3 4'
62  ]
63
64  flag_scan2 = [
65  # flagging-channels for Polnum==2
66  # 'scannum bad_channel(s)'
67  '123456 1 2 3 4'
68  ]
69
70  Unpols = ['JUPITER', 'MARS', 'VENUS', 'SATURN']
71  Aref   = 'CRAB'
```

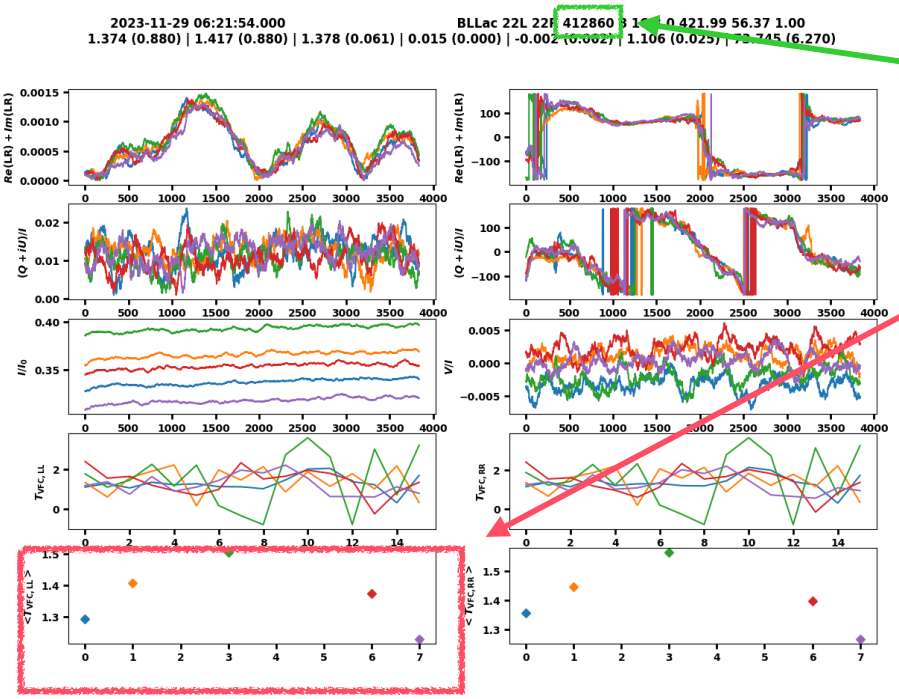
- **flag_scan (1&2) // For position-switching**

  : selecting bad scans for each source scan
  : see the Figures in next slide for an example

- **Unpols : selecting candidates for unpol. source (do not need to change)**

- **Aref : Polarization angle reference source**
  : if 'CRAB' was not observed, then it automatically find 'CRAB2' or 'CRAB1'
  : if you want to use '3C286', you need to change

pipeline script



- flag_scan (1&2) // For position-switching

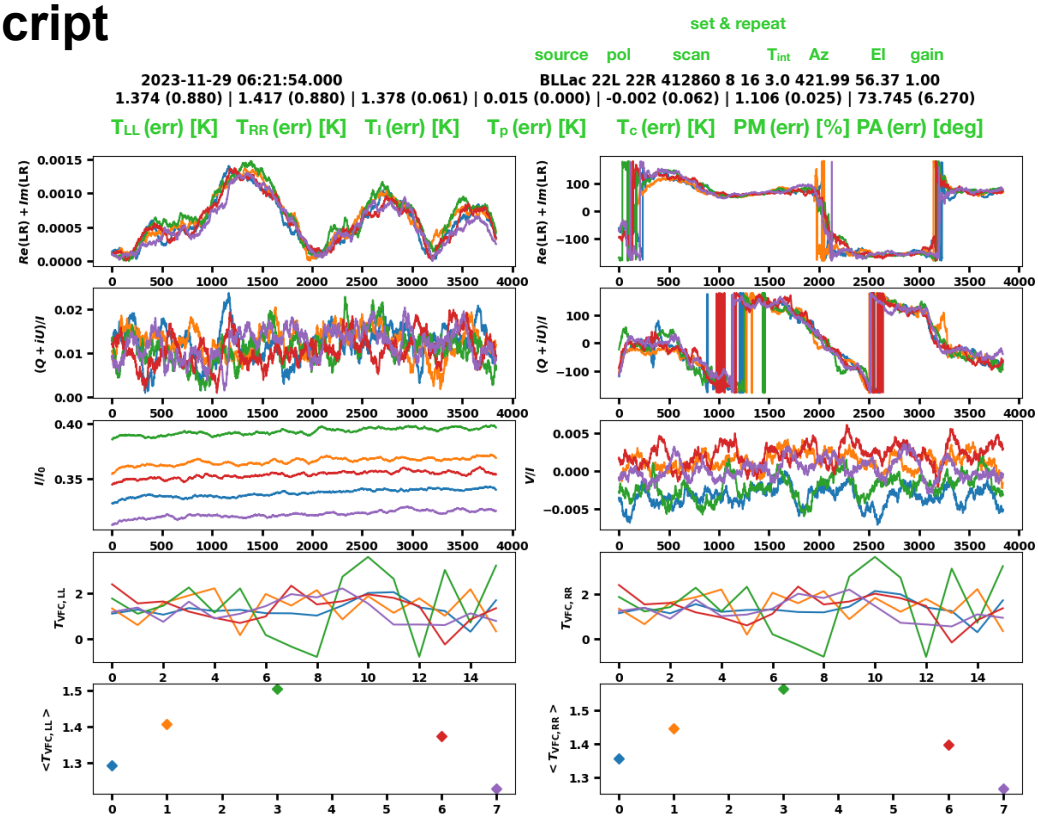
  : selecting bad scans for each source scan
  : see the Figures in next slide for an example

```
flag_scan1 = [
# flagging-channels for Polnum==1
# 'scannum bad_channel(s)'
'123456 1 2 3 4'
]

flag_scan2 = [
# flagging-channels for Polnum==2
# 'scannum bad_channel(s)'
'123456 1 2 3 4'
]

Unpols = ['JUPITER', 'MARS', 'VENUS', 'SATURN']
Aref   = 'CRAB'
```

pipeline script



pipeline script

```
path_       = os.getcwd()
path_p      = path_.replace('TRON Dropbox/Jeong hw', 'Dropbox') + '/FITS/'
path_c      = "~/Dropbox%s"%(path_p.split('Dropbox')[1])
path_dir    = "%s../"%(path_p)
path_cslog = "%s../data_cs/"%(path_p)
path_pslog = path_p

files = ['MOGABA_POLAMI23B_KQ_1_KTN.sdd',
         'MOGABA_K_11_KTN.sdd',
         'POLIDV_KQ_1_K%s.sdd'%(station)
]
pipe_log = 'mogaba_pipelog_%s_SE.log'%(antenna)


flag_file =[    # if you have a file to skip, attach here
'MOGABA_KQ_3_KTN.sdd',
'MOGABA_POLAMI_KQ_17_KUS.sdd'
]
```

- path_ : path where ‘mogaba_pipe_run.py’ in
- path_p & path_c : path to fits file
  (two paths should be same)
  (note that ‘path_c’ is for class/python)
- path_cs/pslog : path for saving log plots
- files : file name(s) of fits file(s)
- pipe_log : name for log_file of data processing
  (just for looking which process the code is doing)
- flag_file : bootstrapping for bas files
  (can’t open for some reasons)
  (or occurs an error)

pipeline script

```
237         for Npn in pol_range:
238             pipe_pos.polnum = Npn
239             pipe_pos.freq   = pipe_pos.freqs[Npn-1]
240             if np.logical_and(not pipe_pos.unpols_n, '3C84' in sour_lst):
241                 pipe_pos.unpols_n = ['3C84']
242             for Nunp, unpol in enumerate(pipe_pos.unpols_n):
243                 pipe_pos.unpol = unpol
244                 if     Auto_Flag:pipe_pos.autoflag  = Auto_Flag
245                 if not Auto_Flag:pipe_pos.bad_chans = flag_scans[Npn]
246                 if str(pipe_pos.freq)=='129' : pipe_pos.lr_swap=True
247                 writelog(path_dir, pipe_log, "Run Position-Switching (Unpol:%s
248                 pipe_pos.run_pos()
```

If there are no planet scans for unpol. source,
and if there is ‘3C84’ scan instead,
‘3C84’ will be used as unpol. source

‘LR_Swap’ is forced at 129 GHz
If you don’t want to use this, make it ‘False’