Phase / TEC screens WG

...

Leah, Jonny, HaoYang, James

Objectives for the week

1. Create phase / amplitude screens from H-ATLAS pointings, using LBCS calibrators independently calibrated as delay calibrators

   1a. Remove some directions, re-create screens, see if screens match removed directions

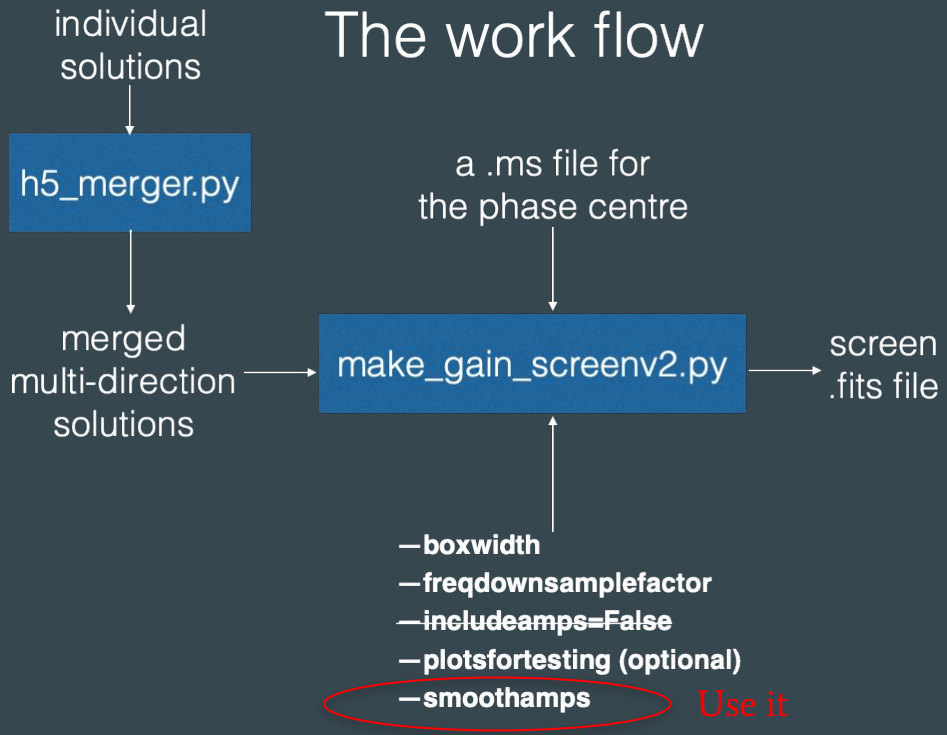
   [NOTE: these cannot be used for imaging, as they are in circular pol]

2. More generally, use screens to start testing intermediate resolution (~2”) imaging

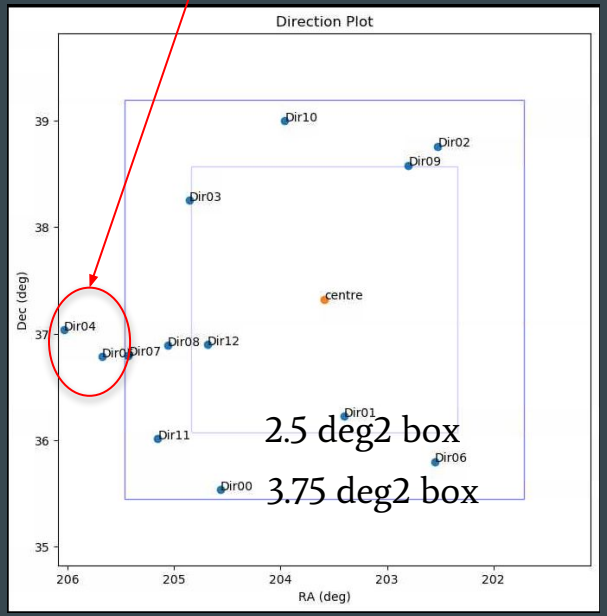
   2a. Pick a delay calibrator, apply to other directions, self-cal and use solutions to create TEC screens that can be applied for imaging

1. Create phase / amplitude screens from H-ATLAS pointings

The work flow



The –boxwidth is 2.5 x 2.5 deg^2 by default, make sure that all your directions are covered within this box.



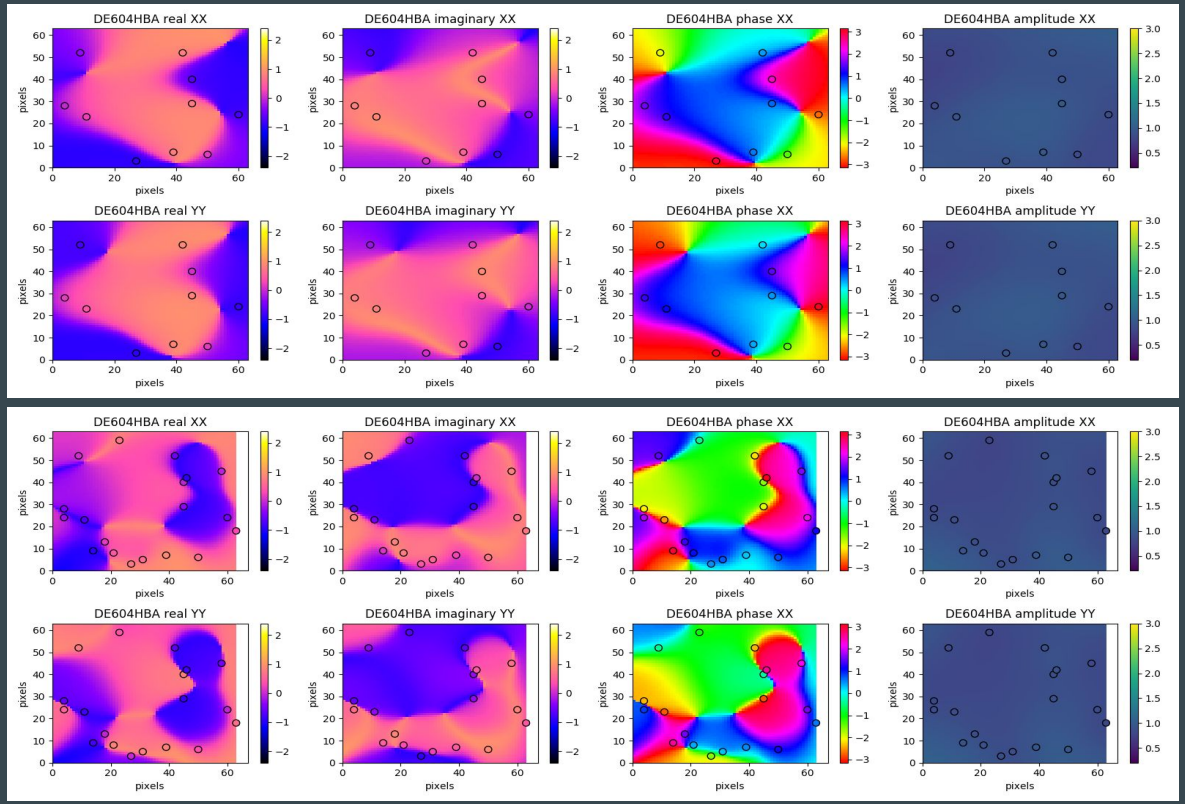
Script: make_screen_gainv2.py @Reinout, h5_merger.py @ Jurjen, Plot_dirandscreen.py @ Haoyang (Zoe)
https://github.com/zoeye859/lb_scripts/blob/main/Plot_dirandscreen.py

1a. Comparing removed directions with screens

Created screens for 2 H-ATLAS pointings with all directions and then just 10 - Amplitude smoothing means solutions look largely similar, lose a lot of phase info

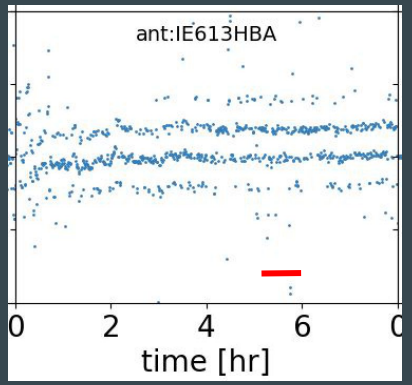
Created a script to go from screen + coordinate -> h5_parm

Can use losoto to compare solutions between screens and results from standard pipeline process

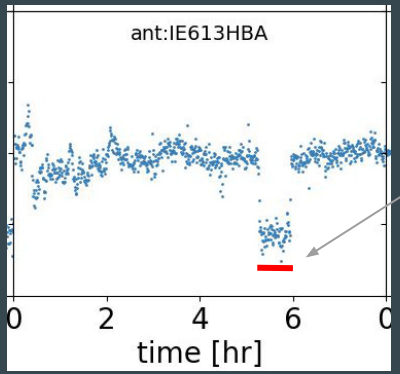


2a. Apply delay-cal and self-cal other directions to get TEC

Done, but need to ‘de-jump’ TEC solutions



Before de-jump



After de-jump

With current implementation of de-jumping, noisy or quick-changing data causes large offset

(linked with comparison of adjacent points)

Working on implementing outlier rejection to get rid of points more than 1 jump away

- Current plans are to compare values to a rolling median as moving across the observation, then clip
- May need to work with log values due to small deviations from dTEC of zero

Takeaways & Future work

- Getting more used to making / working with screens
- Working on improving handling of screens (better visualisation of directions!)
- Framework for extracting screen values at specific point
- Working on improving de-jumping

- Improve de-jumping
- Implement linear screens and test imaging