Draft version January 4, 2017
Preprint typeset using LaTeX style emulateapj v. 12/16/11

RUNNING THE PAPER POWER SPECTRUM PIPELINE

Matthew Kolopanis

Here I present an overview of the PAPER power spectrum pipeline used in creating the A15 and K17 power spectra. This memo aims to clarify the multiple steps of the pipeline including including the use of the sigloss calculations in power spectrum estimation

1. PAPER POWER SPECTRUM PIPELINE

The psa64 pipeline relies on four files found in capo/pspec_pipeline/:

1. the bash script: sigloss_bash_v2.sh
2. the main pspec and sigloss script: sigloss_sim.py
3. the boot strapping script: boots_to_pspec.py
4. Upper limit estimator: sigloss_limits.py

1.1. *sigloss_bash_v2*

The bash script runs the other three parts of the power spectrum pipeline and requires some parameters be set to properly estimate the power spectrum.

Parameter	Description
PREFIX	path pointing to data folder
EVEN_FILES	path and glob of even files
ODD_FILES	path and glob of odd files
WD	the PWD command
noise	boolean used to make noise simulations of pspec
boot	number of bootstraps
chans	list of channel ranges in capo chan option format (e.g. 95_115)
rmbls	list of baselines to remove in i,j pairs (e.g. 0_44)
t_eff	the effective integration time, in number of time bins
bl_length	baseline length in meters
SEP	Separation type (e.g. 0,1)

here's what a sample bash configuration will look like:

Listing 1 sigloss_bash_v2.sh

```
PREFIX=’../../../../exp_vs_inttime/lstbin_psa64_data_optimal’
EVEN_FILES=${PREFIX}’/even/sep0,1/lst*242.[3456]*.uvGAL’
ODD_FILES=${PREFIX}’/odd/sep0,1/lst*243.[3456]*.uvGAL’
WD=$PWD #get the sworking directory
noise=’’
boot=60
chans=’95_115’
rmbls=’15_16,0_26,0_44,16_62,3_10,3_25’
t_eff=69
bl_length=30
SEP=’0,1’
```

The remainder of the bash script executes the other files with the configurations set above. One additional line in the script is used to configure the injections used in the signal loss interactions:

Changing the logspace beginning, end and number of points will affect the number and range of injections.

Listing 2 sigloss_bash_v2.sh

```
    for inject in `python -c "import numpy; print ' '.join(map(str , numpy.logspace(-2,3,30)))"` ; do
```

Parameter	Description
LST_STATS	Boolean to collected metadata from lst binning
DELAY	Boolean to take delay transform
NGPS	Number of groups used in bootstrapping
cov_reg_level	scaling factor of identity to add to covariance before inversion

1.2. *sigloss_sim*

The main power spectrum and signal loss estimation script. This script has a few parameters still hard-coded in unfortunately.

other parameters previously set with this script are now passed in via the command line option parser.

An example of the configuration inside sigloss_sim.py looks like:

Listing 3 sigloss_sim.py

```
random.seed(0)
POL = opts.pol  # 'I'
LST_STATS = False
DELAY = False
NGPS = 5
INJECT_SIG = opts.inject_sig
PLOT = opts.plot
 cov_reg_level = 1e-10
```

1.3. *boot_to_pspec*

Boots to pspec takes output bootstrapped files for each injection level and performs the second round of bootstraps. It used the t_eff parameter to construct a waterfall with only the effective number of free parameters from bootstraps of all injections. Each time in the new waterfall is chosen randomly from a bootstrap file. This process is completed 100 times, then averaged and the standard deviation is taken to estimate the power spectrum and its uncertainties.

This script also computes the folded power spectrum, resulting in the only positive k-values for use in computing Δ^2.

1.4. *sigloss_limits*

Sigloss limits uses the outputs of boots to pspec to compute the signal loss corrected power spectrum evaluating the integral to find the probability each injected power spectrum value is greater than the 1-sigma upper limits of the data power spectrum in each k-bin.