Draft version January 3, 2017
Preprint typeset using LaTeX style emulateapj v. 12/16/11

RUNNING THE PSA64 POWER SPECTRUM PIPELINE

Matthew Kolopanis

Here I present an overview of the PAPER power spectrum pipeline used in creating the A15 and K16 power spectra. This memo aims to clarify the multiple steps of the pipeline including: setting up your configuration file, selecting data, and running signal loss scripts. I will conclude with a brief discussion on using the psa128 pipeline.

1. PSA64 POWER SPECTRUM PIPELINE

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

1. the configuration file: pspec_psa128.cfg
2. the bash power script: mk_psa128_pspec.sh

1.1. *Configuration File*

The configuration file is used to define key parameters used in the making of the power spectrum. Table 1 provides a list of all parameters, their descriptions and some examples.

Inside of the configuration file, either the EVEN_GLOB and ODD_GLOB keywords or the LST keyword should be set. The globs should be use to manually pick data files to use, these keywords take preference over the lst_select script. Inside your configuration file, all parameters must be given in the form:

```
export PARAMETER='STRING OF PARAMETER VALUE'
```

Parameter	Description
PREFIX	Title of output directory, figures and save files (npz) Recommend adding the Date for easy logging e.g Mar30_optimal_frf
pols	The polarization you wish to analyze options like I, XX, YY, XY
seps	the antenna separations to be used sep0,1 sep1,1 sep-1,1 etc your data directory should have these as subdirectories the pipeline looks for these directories to collect data
chans	channel ranges to be analyzed can be singe channel range: 95_115 or list of ranges: 30_50 51_71 78_98 103_123 127_147
ANTS	Antennas to analyze in pspec can be set to any AIPY compliant antenna value use 'cross' as default
LST	The RA (LST) of nights you wish to analyze expects input to be a range: .4_9
NBOOT	Number of bootstraps to perform
FILEAPPELLATION	file suffix e.g. uvGAL uvG uvGL uvGLS uvGLAS etc
EVEN_GLOB	A bash glob of even files e.g. lst.*242.[3456]*
ODD_GLOB	A bash glob of odd files e.g. lst.*243.[3456]*
covs	signal loss correction factor must have same length as chans keyword e.g '1.02 1.2 1.30 1.37 1.24' no input ('') defaults to 1.0 for all chans
noise	amplitude of noise to inject must have same length as chan or null for no noise
FILTER_NOISE	Boolean True to fringe rate filter injected noise
NOISE_ONLY	Boolean True to replace data with noise when injecting
USE_pI	Boolean True to use pI in pspec_cov_v???.py
SCRIPTSDIR	Path to directory where scripts are located
cal	the name of the cal file to use must be in your path or your python sys.path
PWD	pwd command given to script for logging
EVEN_DATAPATH	path to even nights
ODD_DATAPATH	path to odd nights
	Window function to multiply data by
WINDOW	e.g. Blackman-Harris, Kaiser3 current default is none
PLOT	Boolean to plot to X-window True or False
COV	Boolean to turn On and Off covariance True or False
BOOT	Boolean to turn on and off bootstrapping True or False
KPKPLOT	Boolean to turn on and off plotting True or False

TABLE 1
List of Parameters used in configuration file for use with mk_psa128_pspec.sh

A sample config file would look like this:

```
#this file configures a pspec run
# run with mk_pspec.sh <this file>


export PREFIX='Apr22_optimal_frf_test_noise_only_31Jy'

#chans=`python -c "print ' '.join(['%d_%d'%(i,i+39) for i in range(10,150,1)])"`
export pols='I'
export seps='sep0,1 sep1,1 sep-1,1'
#export chans='30_50 51_71 78_98 95_115 103_123 127_147'
export chans='95_115'
export ANTS='cross'
#export chans='95_115'
#export RA="1:01_9:00"
export NBOOT=100
export FILEAPPELLATION='uvGAL'

## use EVEN_GLOB and ODD_GLOB to manually select data
## script will use manaul glob over lst_select
export EVEN_GLOB='lst.*242.[3456]*'
export ODD_GLOB='lst.*243.[3456]*'
export LST="-.1_8.75"

#signal loss correction factor
#export covs='1.20 1.19 1.23 1.22 1.24 1.28'
#export covs='1.62 1.35 1.30 1.30 1.28 1.35'
export covs='1.26'

##amplitude of injected noise in Jansky
export noise='30'
##Fringe Rate Filter noise before injection
export FILTER_NOISE=True
#instead of injecting noise, override data with noise
export NOISE_ONLY=True

#replace pC with pI in pspec_cov_boot_v???.py
export USE_pI=False

#DATAPATH=fringe_hor_v006
export SCRIPTSDIR=~/src/capo/pspec_pipeline
export cal='psa6240_v003'
export PWD=`pwd`
export EVEN_DATAPATH="${PWD}/lstbin_psa64_ali_optimal/even"
export ODD_DATAPATH="${PWD}/lstbin_psa64_ali_optimal/odd"
export WINDOW='none'

#to separately run scripts
export PLOT=False #to plot things in COV and BOOT scripts
export COV=True
export BOOT=True
export KPKPLOT=True
```

1.2. *Running the Bash Script*

Now we have our configuration file set, we need to get ready to run the power spectrum. With how the sample config file is set, we should compute the power spectrum in the directory which contains the data folder (lstbin_psa64_ali_nofrf). Once in the correct directory, we can run the bash script with the command:

```
user@folio /path/to/script/mk_psa128_pspec.sh /path/to/config/pspec_psa128.cfg
```

if we are running this directly from the capo/pspec_pipeline directory it will look like:

```
user@folio /path/to/capo/pspec_pipeline/mk_psa128_pspec.sh /path/to/capo/pspec_pipeline/pspec_psa128
    .cfg
```

2. SIGNAL LOSS

The signal loss calculation also relies on two main files:

1. the configuration file: sigloss_psa64.cfg
2. the bash power script: make_sigloss.sh

Parameter	Description
PREFIX	Title of your output director, figures and save files (npz) Recommend adding the Date in your prefix for easy logging e.g Mar30_sigloss_optimal_frf
PSPEC	This is the name of the power spectrum we computed before probably the same as PREFIX from your pspec_psa128.cfg file
PLOT	N/A
COV	N/A
BOOT	N/A
KPKPLOT	N/A

TABLE 2
List of Parameters used in signal loss configuration file for use with make_sigloss.sh

The signal loss configuration file has a few differences outlined in Table 2
A sample signal loss config file looks like:

```
#this file configures a pspec run
# run with mk_pspec.sh <this file>


export PREFIX='Mar30_sigloss_optimal_frf'
export PSPEC='Mar30_optimal_frf'

#chans=`python -c "print ' '.join(['%d_%d'%(i,i+39) for i in range(10,150,1)])"`
export pols='I'
#export seps='sep0,1 sep-1,1 sep1,1'
export seps='sep0,1'
export chans='30_50 51_71 78_98 95_115 103_123 127_147'
ANTS='cross'
#export chans='95_115'
#export RA="1:01_9:00"
export NBOOT=40
export FILEAPPELLATION='uvGAL'

## use EVEN_GLOB and ODD_GLOB to manually select data
## script will use manaul glob over lst_select
export EVEN_GLOB='lst.*242.[3456]*'
export ODD_GLOB='lst.*243.[3456]*'
export LST="-.1_8.75"

#DATAPATH=fringe_hor_v006
export SCRIPTSDIR=~/capo/pspec_pipeline
export cal='psa6240_v003'
export PWD=`pwd`
export EVEN_DATAPATH="${PWD}/lstbin_psa64_ali_nofrf/even"
export ODD_DATAPATH="${PWD}/lstbin_psa64_ali_nofrf/odd"
export WINDOW='none'
```

Running signal loss is very similar to the power spectrum:

```
user@folio /path/to/script/make_sigloss.sh /path/to/config/sigloss_psa128.cfg
```

if we are running this directly from the capo/pspec_pipeline directory it will look like:

```
user@folio /path/to/capo/pspec_pipeline/make_sigloss.sh /path/to/capo/pspec_pipeline/sigloss_psa128.
    cfg
```

The signal loss script will output the maximum signal loss correction factors it calculates inside of the log file for each channel/polarization combination. These correction factors can be put back into the 'covs' keyword of the power spectrum configuration file to correct the output power spectrum.