SPAM: Source Peeling and Atmospheric Modeling

SPAM is a Python-based extension to AIPS (Greisen 2003), aimed at reducing high-resolution, low-frequency radio interferometric observations in a very efficient, systematic and reproducible way. Special features in SPAM, like direction-dependent ionospheric calibration and image-plane ripple suppression, will help to make high-quality sub-GHz images. SPAM is a Python module, including some C-code optimalizations, that uses the Python-to-AIPS interface ParselTongue (Kettenis et al. 2006), which itself is based on ObitTalk (Cotton 2008). ParselTongue provides access to AIPS tasks, data files (images & visibilities) and tables. SPAM also uses several standard Python libraries like scipy, pylab, matplotlib, and numpy. Data reductions are captured in well-tested Python scripts that executes AIPS tasks directly (mostly during initial data reduction steps), calls high-level functions that make multiple AIPS or ParselTongue calls, and require few manual operations. SPAM now also includes a fully automated pipeline for reducing legacy GMRT observations at 150, 235, 325 and 610 MHz. Some users have also successfully applied it to legacy GMRT 1.4 GHz observations.

News

- 21-May-2017: Moved homepage including SPAM webpage to new domain.
- 06-Jan-2016: Update of webpage. Release of SPAM pipeline for GMRT. Full AIPS 31DEC13 binary install (LNX64 only), additional catalogs and more now available as single tarball.
- 15-Jan-2016: Obit now available as binary install (LNX64 only). Added simple description how to run the pipeline.

Download and install

Here are instructions on how to download and install SPAM on your Linux 64-bit system.

Here are instructions on how to run the SPAM pipeline (UNDER CONSTRUCTION).

Starting up SPAM

Setting up SPAM in Linux bash is done through:

```
. <spam installation path>/setup.sh
```

Setting up SPAM in Linux (t)csh is done through:

```
source <spam installation path>/setup.csh
```

SPAM uses AIPS in such a way that it allows creation of an isolated, self-consistent work environment per project. To run SPAM, first create a project directory:

```
mkdir -p <new project directory>
cd <new project directory>
```

Then start ParselTongue:

```
start_parseltongue.sh . <AIPS userid>
```

The AIPS userid can be any number from 11 to 9999. I usually stick with 11

In the project directory, the start_parseltongue.sh script has set up a local AIPS work environment with 35 AIPS disks (work01-35) and a few other directories (fits, datfil, prtfil, runfil, da00, etc.). Most of this is quite specific to AIPS.

If all went well, the final output will have been

```
Welcome to ParselTongue 2.3
>>>
```

Next, we import SPAM as a python module

```
from spam import *
```

If this returns without errors it is likely that the installation has gone well.

Running the SPAM pipeline

Older materials

SPAM tutorial for the 3gc3 workshop “The Elephants In The Room”, Port Alfred, South Africa, on Feb. 19, 2013.
SPAM example script for reducing GMRT 330 MHz data.

00017629