# A User Friendly Interface for the ALMA Pipeline Heuristics

Heiko Hafok[1], Dirk Muders[1], Lindsey Davis[2], George Kosugi[3], Takeshi Nakazato[3],
[1]Max-Planck Institut für Radioastronomie, Bonn, Germany
[2] National Radio Astronomy Observatory, Socorro, USA
[3] National Astronomical Observatory Japan, Tokyo, Japan

## Introduction

The ALMA (Atacama Large Millimeter Array) Pipeline Heuristics System (PHS) is designed to automatically reduce data taken in standard observing modes and produce standard data products.

The PHS is implemented as a set of Python recipes consisting of a list of high-level reduction stages or tasks, for processing single dish and interferometry data sets and combinations thereof. These scripts run in the CASA (Common Astronomy Software Applications) data reduction environment. The PHS must be available to pipeline core developers, testers, and maintainers, to the pipeline commissioning team, staff at the ALMA regional centers, and to end users who may wish to execute parts of the PHS on their desktops resources permitting. To support this diverse user group the user interface must ensure ease-of-use, yet support complex processing setups for experts, e.g. parallelization of processing steps by source, time, frequency band, etc where appropriate.

The initial low-level PHS interfaces for single dish and interferometry were developed independently and in parallel to accommodate the different requirements in each area and to get some basic heuristics algorithms working quickly. Although both interfaces support reduction recipes they diverged in style and the degree to which they supported customization. In this poster we describe a new PHS user interface which will support both single dish and interferometry reductions in a more uniform way and facilitate pipeline customization.

This interface is targeted towards ALMA staff who will develop new and improve existing PHS recipes for the standard ALMA pipeline, and at astronomers who may be interested in developing their own reduction pipelines in CASA. The new user commands correspond to the existing PHS stages but are easier to execute interactively and tune. Technically, the user commands are implemented as CASA Python tasks, and look to the user exactly the same as core CASA tasks. Lower level Python commands which can be used to construct new tasks are also available for the expert user

## User Interface Requirements

- Must support rapid heuristics recipe development and deployment
- Must support both batch and manual mode operation from CASA
- Must support both single dish (SD) and Interferometry (IF) heuristics
- Must support control of input and output data sets, tables, images, logs
- Must support data set slicing, e.g. by spectral window, polarization
- Must support algorithm tuning where appropriate
- Must support task parallelization where appropriate
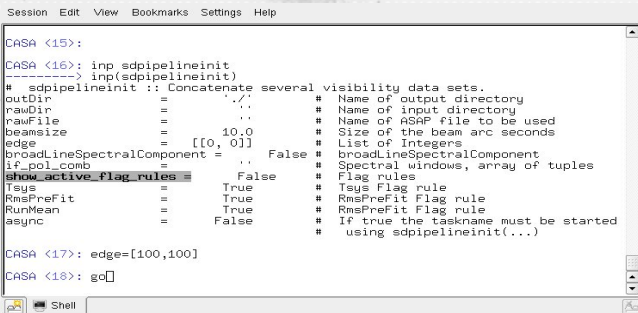- Must be easy and intuitive to use
- Must be familiar to CASA users



Fig. 2: Sample CASA task interface of the single dish heuristics startup task. Global heuristics parameters, e.g. beamsize or edge, are set at the top level. Task specific heuristics parameters, e.g. for baseline fitting, can be set be used to override the supplied defaults for the following tasks.
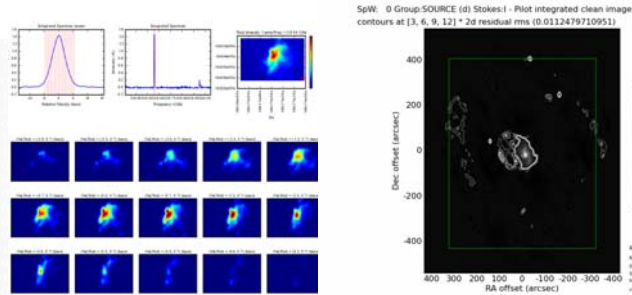


Fig. 1: Sample heuristics output: a) CO channel maps of M17 produced by the single dish heuristics recipe, and b) an X band mosaic image of Sagittarius A produced by the interferometry heuristics mosaic recipe.

## The Single Dish / Interferometry Heuristics Stages Interface

- Provides user friendly Python wrappers for the pipeline stages
- Provides a clean interface for interactive pipeline tuning and testing
- Provides a clean interface for creating automated pipeline recipes
- Supports the storage and recovery of intermediate stage results
- Supports control of stage input and output
- Supports control of plotting behavior
- Hides the pipeline stage infrastructure from the user
- Provides user friendly Python wrappers for the heuristics engines
- Reuses the existing and well tested heuristics engine classes
- Minimizes refactoring of the heuristics engines
- Enables control of the heuristics engines algorithm parameters
- Hides technical details of the engines from the user
- Supports the single script with many parameters reduction recipe model used by single dish heuristics
- Supports the lists of stages with run engine model used by interferometry heuristics

## The Single Dish / Interferometry Stages Task Interface

- The stages interface is wrapped with the CASA Python tasking interface as shown in Figure 2 and the sample single dish pipeline recipe shown below
- The Casa tasking interface combined with the stages interface provides the unified user interface architecture required to pipeline process ALMA data sets.

```
Sample Script for Single Dish analysis with the task interface in casapy:

inp sdpipelineinit
rawDir='/homes/hafok/PipelineWork/test'
rawFile='M16-CO43.ASAP'
rawDir='/homes/hafok/PipelineWork/rawdata.new'
outDir='/homes/hafok/PipelineWork/test/resultPipeline'
edge=[100,100]
broadComponent=True
beamsize= 10.0
go
sd_createPositionOverviewPlots()
sd_get_if_pol()
sd_groupData()
sd_remove1stOrderBaseline()
sd_accumulateSpectra()
sd_spectralLineDetect()
sd_spatialLineDetect()
sd_removeBaseline()
sd_flagData()
sd_regridData()
sd_plotResultSpectra()
sd_plotSparseSpectraMap()
sd_plotChannelMap()
sd_writeMSDataCube()
```

# Atacama Large Millimeter Array

## The Pipeline Heuristics Subsystem Group