

# ALMA Offline User Test Report

Alessandra Rossetti

INAF - Istituto di Radioastronomia - Bologna, Italy

## 1 Test definition

The purpose of this test was to evaluate the capability of CASA (Common Astronomy Software Application), a millimeter/radio data reduction, imaging and analysis package for astronomers, that can handle both interferometric (synthesis) and single dish data.

The test involved nine astronomers selected by their ARC's or Integrated Product Team (IPT) to represent the ALMA project in Japan (Masathoshi Ohishi and Shige Takakuwa [NAO]), Europe (Robert Lucas [IRAM], Silvia Leurini [ESO] and Alessandra Rossetti [INAF]), Chile (Paulo Cortes [U. de Chile]) North America (Chris Carilli, Tony Remijan [NRAO], Lewis Knee [NRC]). Testers were a mix of very experienced interferometrists or single-dish experts, a few relative novices and some with a more programmer-oriented experience. As such, they represented a broad, un-biased view of CASA development from an ALMA-user perspective.

The test had two main aims:

- testing Single Dish calibration and imaging;
- Synthesis regression testing.

### 1.1 Single Dish calibration and imaging

This was the **first** scientific test of single dish data reduction after incorporation of the ATNF Spectral Analysis Package (ASAP). Although the software is different, the look and the feel of the interface should be similar to the one developed for synthesis data reduction and imaging. The ASAP requirements have been translated into requirements appropriate for ALMA, but not all ASAP requirements have been implemented yet. Therefore, the main focus of the test was on functionality: testers had to evaluate the Command-Line Interface (CLI) for single dish processing imported from the ASAP package and to decide whether the available functions are adequate for single dish data analysis. The CLI is still tool-based, even if some prototypical tasks have been developed to illustrate how the synthesis and the single dish interface can be unified (for a description of the difference between tools and tasks see section 2.1). Testers had to provide recommendations on task design, inputs, defaults, additional tasks needed and had to comment on the documentation (in-line and Cookbook).

For a detailed list of the datasets and regression scripts, see <http://projectoffice.aips2.nrao.edu/ALMA2007.03/ALMA2007.03.html>

## 1.2 Synthesis Regression and Benchmark

The CASA software was tested for single baseline reduction and analysis in April 2006. Since then most Glish<sup>1</sup> capabilities have been imported into the CASA framework so that all datasets previously reduced in AIPS++/Glish could be reduced in CASA with Python scripts. The AIPS++ tool structure has been converted to more user-friendly tasks (similar to AIPS or MIRIAD) and one of the aims of the test was the evaluation of the CLI task implementation. Task-based CASA regression scripts developed for datasets used in previous tests were provided to the testers in order to verify that they worked as intended and produced scientifically valid results. The scripts covered most automated procedures needed to reduce and image synthesis data. Testers had to change the scripts to try several data reduction procedures and they had to exercise non-automated user interface: visualization, flagging, tools to evaluate calibration and imaging quality, analysis. They also had to provide recommendation on task design, inputs, defaults, additional tasks needed and had to comment on the documentation (in-line and Cookbook).

In addition to basic data reduction, this test verified that working CASA scripts are available for all datasets provided during the test.

For a detailed list of the datasets and regression scripts, see <http://projectoffice.aips2.nrao.edu/ALMA2007.03/ALMA2007.03.html>

## 2 CASA software

### 2.1 Tools and tasks

CASA is a suite of astronomical data reduction tools and tasks that can be run via the IPython interface to Python. This allows an easy navigation within the package (shell access, namespace completion, auto-parenthesis, command history with searching, numbered input/output, macro system for re-execution of previous input, etc.)

Tools in CASA provide the full capability of the package. Originally, CASA consisted of a collection of tools, combined in toolkits. Then, the toolkit routines have been replaced by tasks, which are essentially IPython interface scripts to the tools, but with specific, limited access to them.

The idea for having tasks is that they are simpler to use, provide a more familiar interface, and are easier to learn for astronomers (above all for those who are familiar with other software packages for the reduction of interferometric data, such as AIPS, Miriad and CLIC).

While running CASA, the user interacts with tasks, either indirectly by providing parameters to a task, or directly by running a task. Each task has a well-defined purpose, and a number of inputs which can be set by the user. Tasks are a layer on top of the tools - when a task is running, a series of tools in a toolkit are running. Several tasks are available and they are

---

<sup>1</sup>Glish is both a programming language and an environment for data acquisition and analysis and is part of the AIPS++ project of the NRAO.

grouped into *functionalities* (see Table 1): Import/Export, Information, Editing, Display/Plot, Calibration, Imaging, Modelling and Utility.

Table 1: Available Tasks

Import/Export	Information	Editing	Display/Plot
importvla	listhistory	flagautocorr	plotants
importasdm	listobs	lagdata	plotcal
importfits		flagxy	plotxy
importuvfits			viewer
exportfits			
exportuvfits			
Calibration	Imaging	Modelling	Utility
accum	clean	contsub	help task
bandpass (B)	feather	setjy	help par.parameter
blcal (M, MF)	ft	uvmodelfit	taskhelp
correct	invert		tasklist
gaincal (G)	makemask		browsetable
fluxscale	mosaic		split
fringecal (K)			restore
clearcal			
pointcal			
smooth			

## 2.2 Data

Interferometric data are stored into a Measurements Set (MS), which is a generalized description of data from any interferometric or single dish telescope. The MS consists of a main table containing the visibility data and associated sub-tables containing secondary information such as antennas position, scan number, spectral window number, subarray number, polarization identification, etc..

## 3 Test results

Testers worked on assigned datasets from the VLA, PdBI, BIMA and GBT. Some testers reduced and/or imaged VLA, DRAO and BIMA data that they brought from their home institutes.

All testers reported that the regression scripts worked well and produced results comparable to the published ones. Testers modified the input of the regression scripts, flagged data interactively, used the viewer, plotters and table browser extensively to evaluate results and

used different calibration and imaging strategies. This allowed them to provide recommendation on improvements needed for data calibration, visualization and analysis.

They found that the basic functionalities needed to process interferometric data worked well and produced scientifically valid results at speeds comparable to existing packages.

Single dish calibration, gridding and imaging procedures worked well too, but it was necessary to improve the functionality needed to visualize and analyze spectral line data. Furthermore, the tool-based interface was difficult to work with and should be converted in a task-based one. For both single dish and synthesis data processing, the primary emphasis for future development should be placed on improving visualization capabilities, making the software more robust and provide user-friendly feedback.

### 3.1 Impression of the software

In the post-test wrap-up discussion some considerations and comments on the software have been made.

- Parameters sometimes were not consistent within tasks: e.g. there were inputs with the same meaning but with different names.
- The software was easy to navigate but error messages were obscure and programmer-oriented. Sometimes errors gave the impression the system crashed. The user feedback from the entire CASA package should be reviewed and reworked to be more user-oriented. This issue was considered at a very high priority.
- Testers had different points of view concerning global parameters: they were considered to be not useful by those who liked to work in the scripting interface, but very useful by those who worked from the manual command-line interface. In conclusion, it would be highly desirable to be able to switch global parameters ON/OFF.
- Testers liked the task-based command-line interface but not all the toolkit routines had been replaced by tasks. All tools should be converted into the task interface, and the tools themselves should be available only for developers and advanced users who want to develop their own tasks.
- The syntax of the single dish tool-based interface was considered to be inadequate and it should be converted to tasks and made to look like the synthesis interface.
- Some tasks did not work properly in all their functions, e.g. *flagdata* should be able to undo flagging but this option didn't work.
- The functionalities were still not complete for both synthesis and single dish data reduction, above all for data visualization and analysis. It would be useful to be able to plot the visibilities of a single baseline, to average data in frequency and/or time, to combine spectral windows, to select data by intent (e.g. all calibrator sources). The

software should be able to make multiple Gaussian fits to one or more line profiles and give the results of the fit in an output file. The display of the data on the user screen was slow. In the meantime the software should provide a *sanity check* to the user and give a warning if a process is expected to take a long time.

- The calibration solutions could not be displayed in the desired number of ways and this has been considered to be a limitation by several testers: in particular, it would be useful to plot the calibration solutions themselves (maybe automatically as in CLIC) and to plot solutions for the spline and polynomial solutions for which the confidence is better established by visual examination of the quality of the fit. Thus the solution fits need to be displayed along with the actual data being fitted.
- The table browser was very unstable. It crashed repeatedly in several different ways.
- The image functionality worked like IMAGR in AIPS. For some testers it would be nice to have an imaging and deconvolution set-up like Miriad as well, with three separate tasks to invert, clean and restore an image. Larger tasks should be separated into smaller ones to allow the user more control of the reduction process.
- Testers liked the viewer but found it limiting in some respects. It should be possible to set display parameters from the command-line interface, for example in order to set a zooming rectangle. The viewer statistics should be able to provide the integrated flux of a region drawn interactively by the user over an arbitrary region. It should be able to plot a wedge with the intensity scale and the synthesized beam of the map. The viewer allows one to plot in a graphical window the spectral profile of the source at a single point on the image. In order to identify channels not contaminated by line emission, it should be possible to display an average spectrum of a user-defined region on a cube using the Spectral Profile option. Multiple spectral line graphical windows should be allowed. It has been suggested to have the same interface between the plot tasks and the viewer.
- The software was sensitive to extra space and capitalization. The extra space prevented the use of “cut-and-paste” to enter parameters into the command-line interface.
- Sometimes the system crashed due to several bugs, for example if you run the *split* task on an already existing output file or the *plotxy* task on a measurement set which does not exist, or when in *flagxy* you select an antenna that has no data. It should have a quiet abort and better feedback about what it is doing.

## 3.2 Impression of documentation

### 3.2.1 Cookbook

Testers generally liked the organization and the general level of the Cookbook, but it needs further work. Some introductory material should be provided at the beginning of each chapter and the level should be consistent throughout all the chapters. There were numerous mistakes

in the examples, which apparently reflected the last changes to the task interface that had been made before the test.

In the post-test wrap-up discussion testers arrived at some basic conclusions:

- The Cookbook should be task-oriented, and the explanation of the tool use should be in a separate document, a User Reference Manual, for developers and advanced users. This should be mentioned in the Cookbook introduction.
- A flow chart showing the steps needed to reduce typical synthesis data would be useful in the first chapter.
- In the main text the Cookbook should give the simplest cases, listing the tasks needed at each step of the data reduction process. A more detailed series of examples should be provided in an appendix. The Cookbook should also give a more complete description of the task inputs and of how they might be used in a task.
- Several testers suggested that a more extensive introductory background should be provided (e.g. an explanation of the different deconvolution methods that are available and of the different weights of the clean) and that it would be useful to cite references for more detailed information.
- A tutorial would be a helpful addition to the Cookbook. It could be web-based or a well-annotated script and dataset linked to the Cookbook examples.

### 3.2.2 In-line help

While the in-line help for the synthesis package was much better than that for the single dish package, both need additional work to make them useful and complete for the user. It also still contained several factual errors. During the post-test discussion Tony Remijan volunteered to help rewrite the in-line help. The following considerations were made:

- the in-line help should provide a brief introduction of a task's purpose;
- it should give more information on parameters and how to use them, explaining at least the type of variable expected (very important in Python) and the units;
- long tasks (clean, mosaic) should have a longer help (e.g.  $\sim 2$  pages);
- the in-line help should be consistent with the Cookbook;
- the in-line help should have a hierarchical system (like the AIPS help/explain).
- the single dish in-line help needs to be reviewed and rewritten as it is developer-oriented and not user-oriented.

### 3.3 Impression of the test

Testers reported they obtained excellent help from the CASA staff and NRAO scientists helping out with the user support.

Seven testers reported that their overall testing experience was “good”, one gave a grade of “excellent” and one “fair”.

Most thought the test was well-designed in order to test the software, to give suggestions on the implementation and to evaluate documentation. However, several suggested that the script evaluations should have been given less emphasis and that more testing with other user data would have produced more realistic stress on the software. They commented that working with their own data they would have gone more deeply into the data reduction process, spending more time on a single data set in order to make a direct comparison between CASA and other software packages allowing them to really evaluate the scientific results. Indeed, five testers **did** use their own data to flag, calibrate and/or image with the CASA software. Several testers thought that it would be important to evaluate how the software treats data from other interferometers, e.g. SMA or CARMA. Also, adding non-expert users to future test-sessions was considered important, because it will not be as easy for them to find a “way around the system”.

## 4 Plans for the Beta Release

The CASA developers are preparing for a *Beta release* to a limited number of astronomers in September 2007. The 5<sup>th</sup> user test represented the final spot-check of the development to ensure that the software will be ready for the beta release and ALMA commissioning use. A clear plan has been laid out and coordinated with the ARCs to participate in the beta release and provide feedback to the developers. In the post-test wrap-up discussion, most testers agreed to continue working with the CASA software at their home institutes with their own data to provide a wider range of testing than can be done in just one week. For this purpose, at the beginning of April all testers will be allowed to install CASA on their own computers.

The priorities to prepare for the beta release of the CASA software in September are:

- CASA should be more stable. This should take priority over the implementation of the functionality;
- CASA should be easy to install;
- CASA should be very clear about what data format can be loaded and processed; for the beta release the software should support standard VLA data (and possibly PdBI data).
- there should be a complete and consistent set of documentation;
- CASA should give understandable user-oriented error messages (Python responses are not helpful);

- the command-line interface and any graphical user interfaces should be consistent across the package and they should be similar for the synthesis and single dish data reduction process;
- the single dish data reduction and analysis software should be given second priority for the beta release.

For a detailed report on the Offline User Test 5 see the Final Report:  
<http://projectoffice.aips2.nrao.edu/ALMA2007.03/ALMA2007.03.html>

A.R. – 2 April 2007