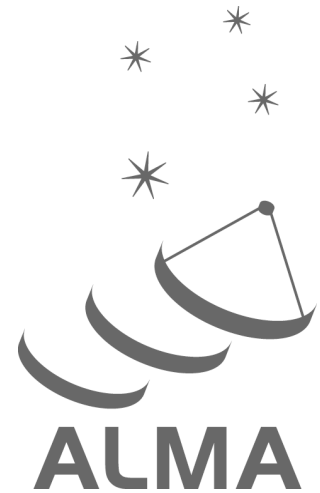


# ALMA Science Pipeline Guide: CASA 4.5.1

## Interferometry and Single-Dish Data



[www.almascience.org](http://www.almascience.org)

## User Support:

For further information or to comment on this document, please contact your regional Helpdesk through the ALMA User Portal at [www.almascience.org](http://www.almascience.org). Helpdesk tickets will be directed to the appropriate ALMA Regional Center at ESO, NAOJ or NRAO.

## Revision History:

---

Version	Date	Editors
1.0	January 2016	Pipeline Team

---

---

## Contributors

Add names and logo's for any additional contributions beyond the ALMA Executives



In publications, please refer to this document as:

**ALMA Pipeline Team, 2016, ALMA Science Pipeline Quickstart Guide, Version 1, ALMA**

---

# TABLE OF CONTENTS

<b>1</b>	<b>PURPOSE</b> .....	<b>3</b>
<b>1.1</b>	<b>NEW SINCE THE PREVIOUS VERSION OF THIS DOCUMENT</b> .....	<b>3</b>
<b>2</b>	<b>OBTAINING THE PIPELINE</b> .....	<b>4</b>
<b>3</b>	<b>PIPELINE-RELATED DOCUMENTATION</b> .....	<b>4</b>
<b>4</b>	<b>ALMA INTERFEROMETRY DATA</b> .....	<b>4</b>
<b>4.1</b>	<b>PRODUCING CALIBRATED MEASUREMENT SETS FROM ALMA PIPELINE DELIVERIES: SCRIPTFORPI.PY</b> .....	<b>4</b>
<b>4.2</b>	<b>THE PIPELINE INTERFEROMETRY CALIBRATION SCRIPT: CASA_PIPESCRIPT.PY</b> .....	<b>4</b>
4.2.1	Running casa_pipescript.py from ALMA Delivery Packages .....	5
4.2.2	Imaging after using casa_pipescript.py .....	5
<b>4.3</b>	<b>THE PIPELINE MODIFYING THE PIPELINE RUN</b> .....	<b>7</b>
4.3.1	Note on the Pipeline Mode .....	7
4.3.2	Removing ASDMs from the processing .....	7
4.3.3	Introducing additional flagging .....	7
4.3.4	Editing calibrator fluxes used by the Pipeline .....	7
4.3.5	Manipulating the Pipeline Context .....	7
<b>4.4</b>	<b>THE PIPELINE WEBLOG FOR INTERFEROMETRY DATA</b> .....	<b>8</b>
4.4.1	Navigation .....	8
4.4.2	Home Page .....	9
4.4.3	Observation Summary .....	9
4.4.4	By Topic .....	10
4.4.5	By Task .....	10
4.4.6	Selecting and filtering plot displays .....	11
4.4.7	Weblog Quality Assessment (QA) Scoring .....	12



<b>4.5</b>	<b>CASA COMMANDS LOG .....</b>	<b>15</b>
<b>5</b>	<b>ALMA TOTAL POWER (SINGLE DISH) DATA .....</b>	<b>15</b>
<b>5.1</b>	<b>THE SINGLE DISH PIPELINE CALIBRATION SCRIPT: CASA_PIPESCRIP.T.PY .....</b>	<b>15</b>
5.1.1	Running casa_pipescript.py from ALMA Delivery Packages .....	16
5.1.2	Manual Imaging after running casa_pipescript.py .....	17
<b>5.2</b>	<b>MODIFYING THE PIPELINE RUN .....</b>	<b>17</b>
5.2.1	Flagging Auto-correlation data.....	17
<b>5.3</b>	<b>WEBLOG QUALITY ASSESSMENT SCORING FOR SINGLE-DISH DATA .....</b>	<b>17</b>
<b>5.4</b>	<b>NAVIGATION OF THE WEBLOG FOR SINGLE-DISH DATA.....</b>	<b>19</b>
5.4.1	hsd_importdata .....	19
5.4.2	hsd_flagdata .....	19
5.4.3	hifa_tsyscal.....	19
5.4.4	hifa_tsysflag.....	19
5.4.5	hsd_mstoscantable.....	19
5.4.6	hsd_inspectdata .....	19
5.4.7	hsd_calsky .....	20
5.4.8	hsd_applycal.....	21
5.4.9	hsd_baseline .....	21
5.4.10	hsd_blflag.....	26
5.4.11	hsd_imaging .....	27



# 1 Purpose

The ALMA Science Pipeline is being developed for the automated processing of ALMA interferometry and single-dish data, and for the combination of data taken using multiple ALMA arrays. At present, the Pipeline is commissioned for the calibration of the majority of ALMA interferometry datasets and for the provision of diagnostic calibrator images. It is also commissioned for the end-to-end calibration and imaging of ALMA Single-Dish data acquired from Cycle 3 onwards. Data combination from multiple arrays and science target imaging for interferometric data are not yet commissioned. This document describes how to obtain the ALMA Pipeline, how to use it to calibrate ALMA interferometric and single-dish data and use of the Pipeline WebLog, the set of plots output by the Pipeline.

The Pipeline calibrates and images ALMA data automatically using Pipeline tasks. Pipeline tasks use CASA tasks wherever possible to perform the data reduction, and Pipeline tasks can be viewed and executed within CASA in exactly the same way as CASA tasks. For example, in a CASA version also containing the Pipeline, to view the inputs possible for Pipeline task `hifa_importdata` type: `inp hifa_importdata`. To see all the tasks available in CASA type `tasklist`.

The Pipeline is data-driven i.e. the characteristics of each dataset drives the calibration and imaging strategy (the **Pipeline Heuristics**). During the Pipeline run, information on which calibration tables should be used etc are stored in the **Pipeline Context**. Both the **Heuristics** and the **Context** are implemented as python classes.

In order to determine if the Pipeline was used in the processing of an ALMA dataset, please consult the README file in the data delivery package. Some projects may contain a mix of both manually and PL calibrated data.

## 1.1 New Since the Previous Version of This Document

- There is now a Chapter on Pipeline Processing in the ALMA Cycle 3 Technical Handbook
- The Pipeline WebLog has a new structure on the By Task sub-pages when multiple execution blocks have been processed
- New section on Manipulating the Pipeline Context
- New WebLog Quick Tips information
- New section on the CASA Commands Log written by the Pipeline
- New section on processing ALMA Single-Dish data using the Pipeline

## 2 Obtaining the Pipeline

The Cycle 3 Pipeline is released as part of CASA 4.5.1. While earlier versions of the Pipeline needed special CASA installations (for CASA 4.2.2 and 4.3.1), this is no longer necessary: the Pipeline is a standard part of CASA 4.5.1. CASA 4.5.1 is available, along with installation instructions, from the **Documents & Tools** Section of the **ALMA Science Portal**: (<http://www.almascience.org/>). If any issues are encountered with CASA 4.5.1 installation, please contact the ALMA Helpdesk via the link on the ALMA Science Portal. ALMA Cycle 3 data must be processed using CASA 4.5 or later.

## 3 Pipeline-related Documentation

The User documentation currently relating to Pipeline, available from the ALMA Science Portal, is:

- **ALMA Science Pipeline Quickstart Guide**: Overview of the ALMA Science Pipeline (this document)
- **ALMA Science Pipeline Reference Manual**: Description of individual Pipeline tasks
- **ALMA Technical Handbook**: Chapter 13 is devoted to Pipeline Processing
- **ALMA QA2 Data Products**: Description of files included in ALMA deliveries

## 4 ALMA Interferometry Data

### 4.1 Producing Calibrated Measurement Sets from ALMA Pipeline Deliveries: `scriptForPI.py`

If the Pipeline was used during the ALMA Quality Assurance Process, this will be noted in the **README** file of an ALMA delivery. In order to convert ALMA raw data into calibrated Measurement Sets, for example before re-imaging the data, then the script **`scriptForPI.py`** should be used. **`scriptForPI.py`** directly applies calibration and flagging tables determined during the ALMA Quality Assurance process to raw data to create calibrated measurement sets. The instructions for using this script will be in the **README** file of the ALMA delivery. **Using `scriptForPI.py` is the recommended and fastest method of obtaining calibrated ALMA data from the delivery. For deliveries that used the ALMA Pipeline, `scriptForPI.py` must be run from within a version of CASA that contains the Pipeline.**

**`scriptForPI.py`** uses the script **`casa_piperestorescript.py`**. **`casa_piperestorescript.py`** should not be executed directly/standalone at present. When data calibrated by the Pipeline are restored using **`scriptForPI.py`**, note that additional directories such as **`working`** and **`rawdata`** are created. These directories and their contents should be ignored.

### 4.2 The Pipeline Interferometry Calibration Script: `casa_pipescript.py`

To make any alterations to the calibration and flagging performed during the ALMA Quality Assurance Process, data have to be fully re-calibrated using the script **`casa_pipescript.py`** (instead of using **`scriptForPI.py`**). **`casa_pipescript.py`** is a python script for performing full data

re-calibration using the Pipeline (typical script shown in Figure 1). Note that the Pipeline works on completed Observing Unit Sets (defined in the ALMA Technical Handbook), such that multiple executions of the same Scheduling Block (execution blocks, ASDMs) can be calibrated using a single script to produce a calibrated Measurement Set per ASDM. Information on individual Pipeline tasks is in the **ALMA Science Pipeline Reference Manual**. The default processing recipe can change between different Pipeline releases.

### 4.2.1 Running `casa_pipescript.py` from ALMA Delivery Packages

To reproduce the Pipeline calibration performed as part of the ALMA Quality Assurance Process:

- Copy `casa_pipescript.py` from the **script** directory to the **raw** directory
- Copy `flux.csv` and `*flagtemplate.txt` and `antennapos.csv` (if present) from the **calibration** directory to the **raw** directory

In the **raw** directory:

- Make sure the naming of the raw ALMA data is consistent with those provided in the script (e.g. if the data ends in `.asdm.sdm` then move to names which do not have this suffix)
- Start the version of CASA containing Pipeline using `casapy --pipeline`, then type `execfile('casa_pipescript.py')`

Running the script will create:

- a calibrated MS for each ASDM in the same directory
- test images of the bandpass and phase calibrator (1 per spectral window in `*.image` format. To view a `*.image` file e.g. use `casaviewer image_file_name`)
- a `pipeline*/html` directory containing
  - the Pipeline WebLog - the html pages of plots from the calibration process. Access using e.g. `firefox index.html`
  - `casa_commands.log` –a list of the CASA tasks used by Pipeline tasks during calibration

**Note that to re-run the Pipeline multiple times, it is recommended to start each time from a clean directory containing only the raw data, `flux.csv`, `antennapos.csv` file (if present) and `*flagtemplate.txt` files and the `casa_pipescript.py` script.** If the `flux.csv` and `*flagtemplate.txt` files are not present in the directory, Pipeline will create new default versions of these files, which will not contain any edits made to them by ALMA staff during the Quality Assurance Process. Also note that it is advisable to have 8-10 GB RAM and 50-75 GB disk space per ALMA raw data file (ASDM) available to perform the Pipeline calibration. Please contact ALMA via the Helpdesk if assistance is needed with data reprocessing.

### 4.2.2 Imaging after using `casa_pipescript.py`

In order to use the imaging script (`scriptForImaging.py`) provided by the ALMA Observatory after using `casa_pipescript.py`, then the science spectral windows must be “split” out from the calibrated measurement sets and the measurement sets output with a `.split.cal` suffix.

```

__rethrow_casa_exceptions = True

h_init()

try:
    hifa_importdata(dbserve=False, vis=['uid__A002_X877e41_X452'],
session=['session_1'])

    hifa_flagdata(pipelinemode="automatic")

    hifa_fluxcalflag(pipelinemode="automatic")

    hif_rawflagchans(pipelinemode="automatic")

    hif_refant(pipelinemode="automatic")

    hifa_tsyscal(pipelinemode="automatic")

    hifa_tsysflag(pipelinemode="automatic")

    hifa_antpos(pipelinemode="automatic")

    hifa_wvrgcalflag(pipelinemode="automatic")

    hif_lowgainflag(pipelinemode="automatic")

    hif_gainflag(pipelinemode="automatic")

    hif_setjy(pipelinemode="automatic")

    hifa_bandpass(pipelinemode="automatic")

    hifa_spwphaseup(pipelinemode="automatic")

    hifa_gfluxscale(pipelinemode="automatic")

    hifa_timegaincal(pipelinemode="automatic")

    hif_applycal(pipelinemode="automatic")

    hif_makeimlist(intent='PHASE,BANDPASS,CHECK')

    hif_makeimages(tlimit=2.0)

    hif_exportdata(pipelinemode="automatic")

finally:
    h_save()

```

Figure 1: Example Pipeline calibration script casa\_pipescript.py



The science spectral windows are specified in the Pipeline Weblog (*Home > Observation Summary > Measurement Set Name > Spectral Setup*, in the ID column) or can be determined using the CASA task **listobs** e.g. `listobs('uid__A002_X89252c_X852.ms')` and results will be in the CASA logger. To perform the split, in CASA e.g.:

```
split('uid__A002_X89252c_X852.ms',outputvis='uid__A002_X89252c_X852.ms.split.cal',spw='17,19,21,23')
```

If the script named **scriptForFluxCalibration.py** is present in the script directory, this must also then be executed prior to running the imaging script.

## 4.3 The Pipeline Modifying the Pipeline Run

### 4.3.1 Note on the Pipeline Mode

In the Pipeline standard calibration script, the Pipeline Mode is defined as “automatic” for each task. In this mode, the task takes the default settings of Pipeline and only a limited number of parameters are exposed for editing by a user. Setting the pipeline mode to “interactive” will usually enable the values of a larger number of parameters to be changed. See the **ALMA Science Pipeline Reference Manual** for more details.

### 4.3.2 Removing ASDMs from the processing

ASDMs can be removed from the processing by editing the `vis=` and `session=` lists in `hifa_importdata` in `casa_pipescript.py`.

### 4.3.3 Introducing additional flagging

Additional manual flagging can be introduced to any Pipeline reduction by editing the **\*flagtemplate.txt** files. Examples of the syntax to use in editing these files are given at the top of the files. The flags are applied when **hifa\_flagdata** is run.

### 4.3.4 Editing calibrator fluxes used by the Pipeline

Pipeline currently reads quasar calibrator fluxes from the **flux.csv** file. If a regularly-monitored quasar has been used as the flux calibrator, the flux of this provided from the ALMA Quality Assurance process can be over-ridden by editing it in this file. Only values for the flux calibrator can be over-ridden, since values for the bandpass and phase calibrators are derived during data calibration

### 4.3.5 Manipulating the Pipeline Context

It is recommended to always run the Pipeline using python scripts. New Pipeline runs/scripts need to be initialised using **h\_init** in order to create an empty **Context**.

If it is wanted to halt a Pipeline run before the end of processing, the **Context** should be saved at that point using **h\_save**. In order to resume the run, use **h\_resume** to load the saved **Context** before resuming the run. See the **ALMA Science Pipeline Reference Manual** for more information.

If it is wanted to mainly use the Pipeline to calibrate a dataset but to e.g. insert a different bandpass table into the processing, the following procedure should be followed:

1. Run the pipeline until the end of the bandpass table creation task **hifa\_bandpass**
2. View the calibration tables that Pipeline will use with **hif\_show\_calstate**
3. Export the calibration tables Pipeline uses to a file on disk using **hif\_export\_calstate**
4. Edit the file to replace the name of the Pipeline-created bandpass table with the one it is wanted to use instead
5. Import the file back to the **Context** using **hif\_import\_calstate** and resume the processing

## 4.4 The Pipeline WebLog for Interferometry Data

The WebLog is a set of html pages that give a summary of how the calibration of ALMA data proceeded. The WebLog produced during the ALMA Quality Assurance Process will be in the **qa** directory of an ALMA delivery. To view the WebLog, untar and unzip the file using e.g. **tar zxvf \*weblog.tar.gz** . This will provide a **pipeline\*/html** directory containing the WebLog, which can be viewed using a web browser e.g. **firefox index.html**. The WebLog aims to provide both a quick overview of datasets and also give methods for exploring the calibration and flagging in detail. Therefore most calibration pages of the WebLog will first give a single “representative” view of that calibration (or flagging) step. Users can then choose whether to further select a detailed view of all the plots associated with that calibration step. These plots can easily be filtered by a combination of outlier, antenna and spectral window criteria. Calibration and flagging steps also undergo an automated scoring to give an “at a glance” indication of any trouble points. **Throughout the WebLog, links are denoted by text written in blue** and it is usually possible to click on thumbnail plots to enlarge them. Where histograms are displayed, in modern web browsers it is possible to draw boxes on multiple histograms to select the plots associated with those data points.

### 4.4.1 Navigation

To navigate the main pages of the WebLog, click on items given in the bar at the top of the WebLog home page. Also use the **Back** button provided at the upper right on some of the WebLog sub-pages. Avoid using “back/previous page” on your web browser (although this can work on modern browsers).

## 4.4.2 Home Page

The first page in the WebLog gives an overview of the observations (proposal code, data codes, PI, observation start and end time), a pipeline execution summary, and an observation summary. Clicking on the bar at the top of the home page enables navigation to **By Topic** or **By Task**.

**WebLog Quick Tips**

- Any text written in blue, including headings, is a link to further information
- To go straight to viewing calibrated science target plots, go to **By Task > hif\_apprycal** and scroll down to the bottom
- Histograms can have selector boxes drawn on them using the mouse
- Commands for re-creating some WebLog plots in CASA are now provided

## 4.4.3 Observation Summary

The Observation Summary table lists all the measurement sets included in the pipeline processing. Each measurement set is calibrated independently by the pipeline. The table provides a quick overview of the ALMA receiver band used, the number of antennas, the start/end date and time, the time spent on source, the array minimum and maximum baseline length, the rms baseline length and the size of that measurement set. To view the observational setup of each measurement set in more detail, click on the name of it to go to its overview page.

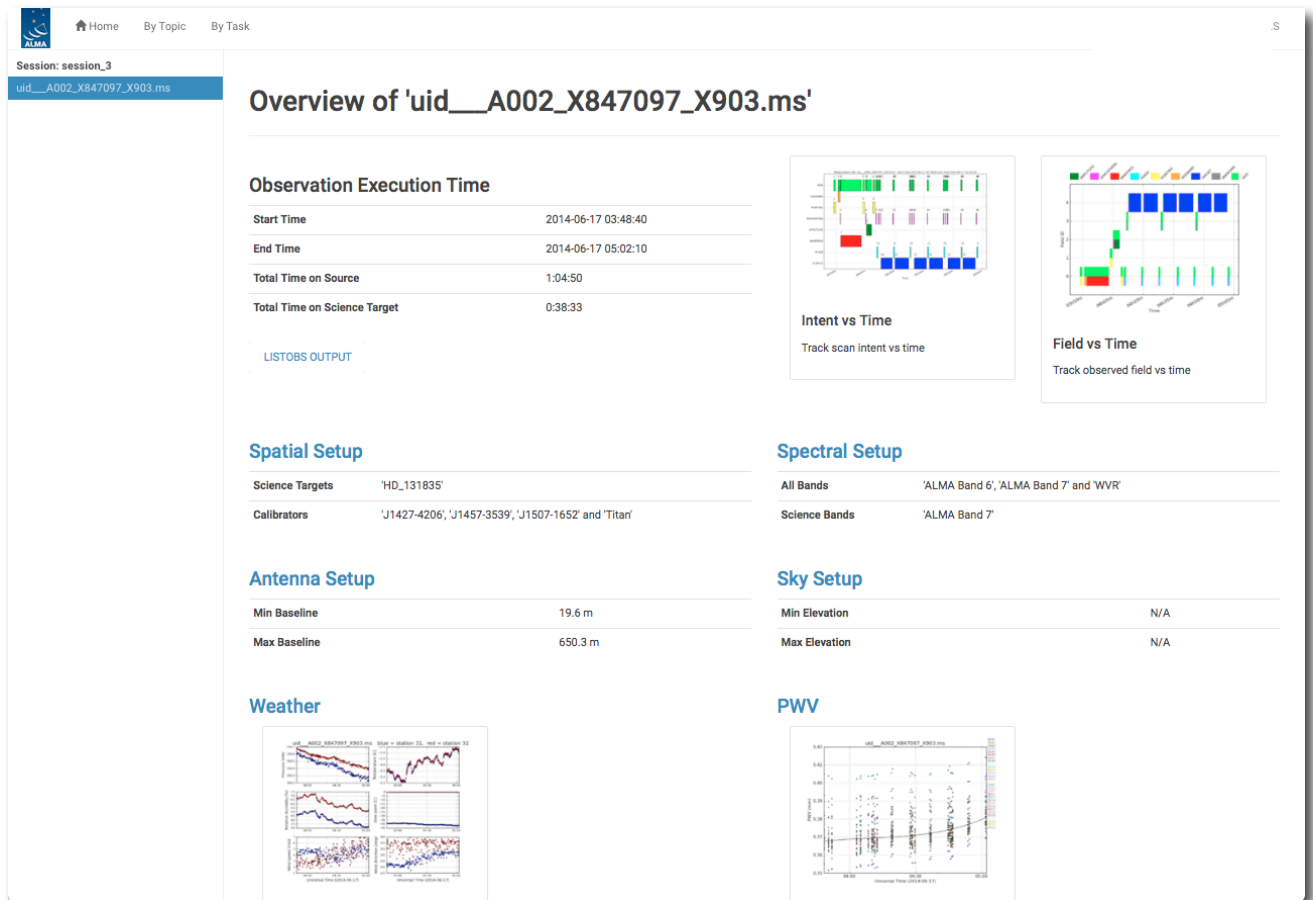
Measurement Set	Receivers	Num Antennas	Time (UTC)			Baseline Length			Size
			Start	End	On Source	Min	Max	RMS	
Observing Unit Set Status: uid://A001/X121/X4b0 Scheduling Block ID: uid://A001/X121/X4a8									
Session: session_3									
uid__A002_X847097_X903.ms	ALMA Band 7	31	2014-06-17 03:48:40	2014-06-17 05:02:10	0:38:33	19.6 m	650.3 m	276.4 m	36.4 GB

Figure 2: WebLog Home Page. The Navigation Bar is circled in red.

### 4.4.3.1 Measurement set overview pages

Each measurement set overview page has a number of tables: **Observation Execution Time**, **Spatial Setup (includes mosaic pointings)**, **Antenna Setup**, **Spectral Setup** and **Sky Setup (includes elevation vs. time plot)**. For more information on the tables titled in blue text, click on these links. There are additionally links to **Weather**, **PWV** and **Scans** information. Two thumbnail plots, which can be enlarged by clicking on them, show the observation structure

either as **Field Source Intent vs Time** or **Field Source ID vs Time**. To view the CASA listobs output from the observation click on **listobs output**.



**Figure 3: A Measurement Set Overview Page. Click on the table headings in blue to see more information about each.**

#### 4.4.4 By Topic

The **By Topic** page provides an overview of all **Warnings and Errors** triggered, a Quality Assessment overview in **Tasks by Topic** and **Flagging Summaries** for the processing.

#### 4.4.5 By Task

The **By Task Summary** page gives a list of the calibration and flagging steps performed on the dataset. It is not displayed per measurement set as the Pipeline performs each step on every measurement set sequentially before proceeding to the next step e.g. it will import and register all measurement sets with the Pipeline before proceeding to perform the ALMA deterministic flagging step on each measurement set. The name of each step on the By Task page is a link to more information. On the right hand side of the page, the bars and scores indicate how well the Pipeline processing of that stage went. All green bars should indicate a fairly problem-free dataset. Symbols next to the tasks, such as “?”, are given on the basis of the Pipeline QA Score.

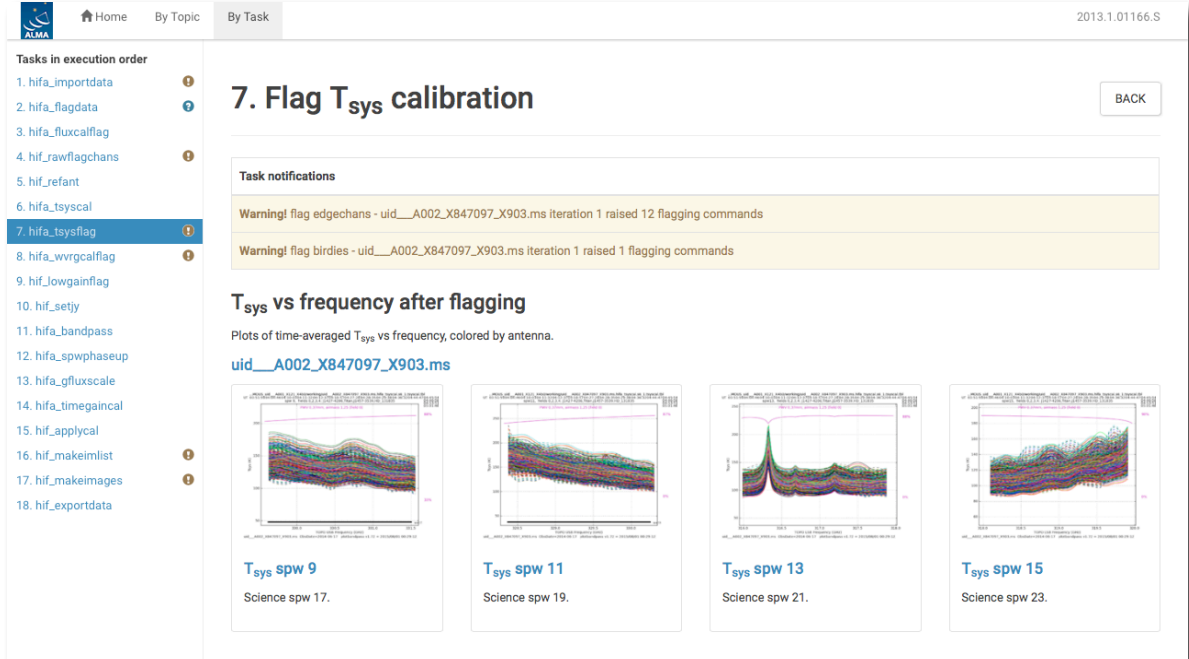
##### 4.4.5.1 Task sub-pages

The task sub-pages (accessed by clicking on e.g. hifa\_importdata) provide the outcome, or the representative outcome, of each Pipeline task executed. Additionally by clicking on the appropriate blue links on each sub-page, the **CASA Log**, **Pipeline QA**, **Input Parameters** and

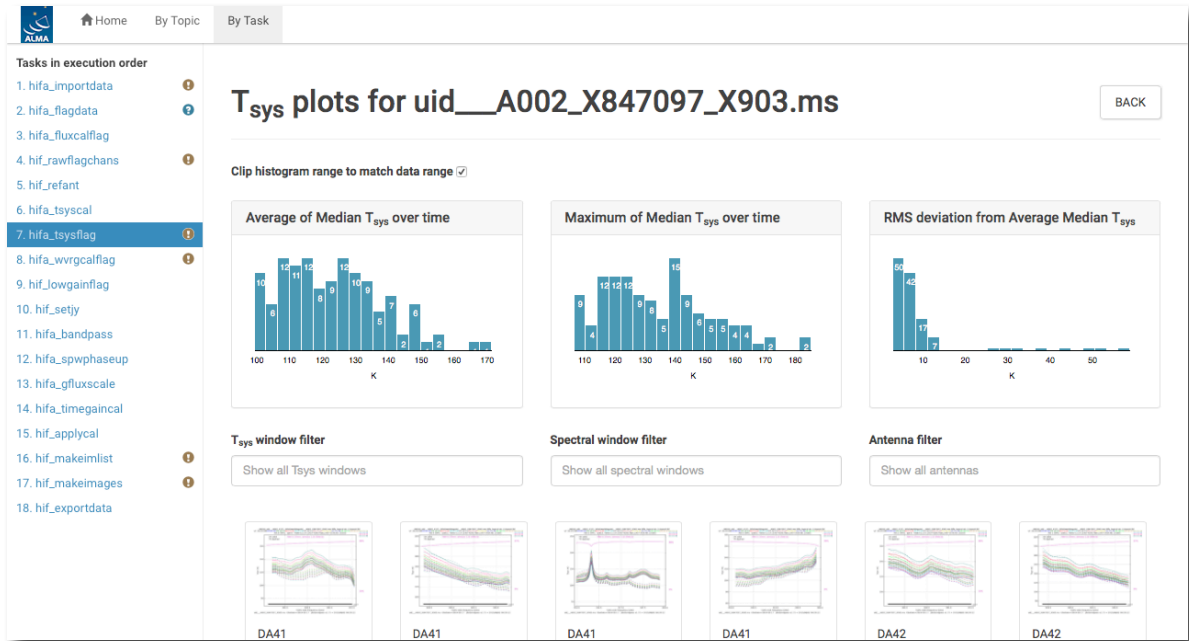
**Task Execution Times** can be accessed. Most sub-pages have further links in order to access a more detailed view of the outcome of each task. These links are often labelled by the measurement set name. For a fast assessment of results, go straight to the **By Task > hif\_applycal** page.

#### 4.4.6 Selecting and filtering plot displays

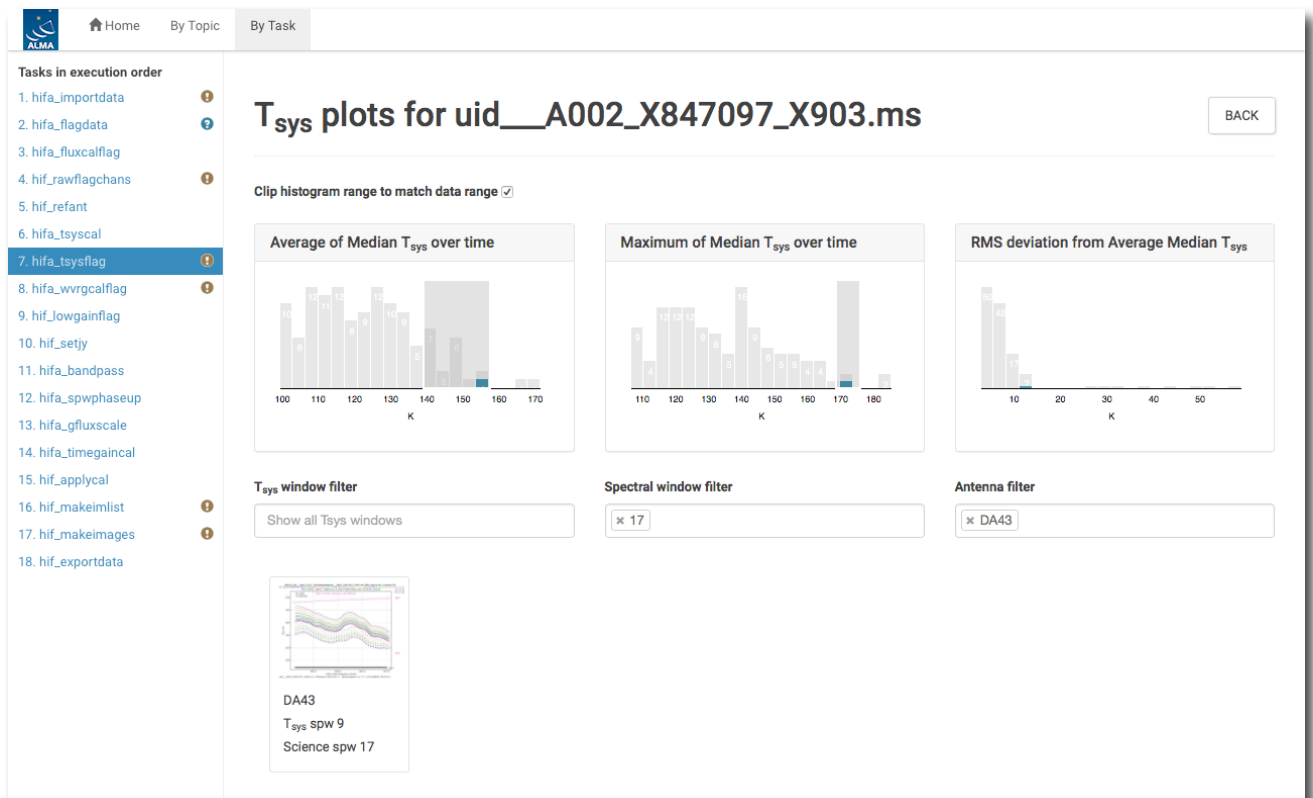
Using the **By Task > hifa\_tsysflag: Flag Tsys calibration pages** as an example.



**Figure 5: Flag Tsys Calibration Overview Page.** A representative view of the T<sub>sys</sub> for each spectral window is shown. To see the measurements for individual antenna/spectral windows, click on the measurement set name, circled here in red.



**Figure 6: The Tsys plots for all antennas. Above the plots are histograms indicating the distributions of metrics based on the median Tsys.**



**Figure 7: Filtering to the plot of interest by drawing grey boxes on two of the histograms using a mouse button, and adding additional spectral window and antenna filters. To clear the grey box filters on the histograms, click on white space in the histograms.**

#### 4.4.7 Weblog Quality Assessment (QA) Scoring

Pipeline tasks have scores associated with them in order to quantify the quality of the dataset and the calibration. The scores are between 0.0 and 1.0 and are colorized such that:

Score	Colour	Comment
0.90-1.00	Green	Standard/Good
0.66-0.90	Blue	Below standard
0.33-0.66	Yellow	Warning
0.00-0.33	Red	Error

Pipeline Task	Pipeline QA Scoring Metric	Score
<b>hifa_importdata</b>	Checking that the required calibrators are present	1.0 all present
		0.1 subtracted for missing bandpass or flux calibrator
		1.0 subtracted for missing phase calibrator or Tsys calibration
		0.5 subtracted for existing processing history
<b>hifa_flagdata</b>	Determining percentage of incremental flagging	<p>0 &lt; score &lt; 1 == 60% &lt; fraction flagged &lt; 5%" (for 'online', 'shadow', 'qa0', 'before' and 'applycal') where "0 &lt; score &lt; 1 == HIGH% &lt; fraction flagged &lt; LOW%" means</p> <ul style="list-style-type: none"> <li>• Score is 0 if flag fraction is &gt;= HIGH%</li> <li>• Score is 1 if flag fraction is &lt;= LOW%</li> </ul> <p>Score is linearly interpolated between 0 and 1 for fractions between HIGH% and LOW%</p>
<b>hifa_fluxcalflag</b>	Determining percentage of incremental flagging	0%-5% -> 1.0, 5%-50% -> 1.0...0.5, >50% -> 0.0
<b>hif_rawflagchans</b>	Determining percentage of data flagged due to deviant channels in rawdata	0% -> 1.0, 100% -> 0.0
<b>hif_refant</b>	Determining if a reference antenna could be found	1.0 if reference antenna could be determined, 0.0 otherwise
<b>hifa_tsysflag</b>	Determining percentage of incremental flagging	0%-5% -> 1.0, 5%-50% -> 1.0...0.5, >50% -> 0.0
<b>hifa_antpos</b>	Determining fraction of antennas without position corrections	1.0 if no corrections needed; 0.0 if all antennas corrected. Score is equal to the fraction of antennas with no offsets

<b>hifa_wvrgcalflag</b>	Checking phase RMS improvement	0.0 if RMS(before)/RMS(after) < 1, 0.5 ... 1.0 for ratios between 1 and 2, and 1.0 for ratios > 2
<b>hif_lowgainflag</b>	Determining percentage of incremental flagging	0%-5% -> 1.0, 5%-50% -> 1.0...0.5, >50% -> 0.0
<b>hif_gainflag</b>	Determining percentage of incremental flagging	Score 1: 0%-5% -> 1.0, 5%-50% -> 1.0...0.5, >50% -> 0.0  Score 2: if a flagging view could be created then 1.0, otherwise 0.0
<b>hifa_bandpass</b>	Judging phase and amplitude solution flatness per antenna, spectral window and polarization (interim measure, future scoring will be based on data with solutions applied)	two algorithms: Wiener entropy and derivative deviation, and signal-to-noise ratio (scores: Wiener entropy: error function with 1-sigma deviation of 0.001 from 1.0; derivative deviation: error function with 1-sigma deviation of 0.03 for the outlier fraction; signal-to-noise ratio: error function with 1-sigma deviation of 1.0 for the signal-to-noise ratio)
<b>hifa_spwphaseup</b>	Determining fraction of spectral windows without phase solutions transferred from other windows	Score is the fraction of spectral windows for which phase solutions are unmapped to expected number of spectral windows
<b>hifa_gfluxscale</b>	Determining SNR of fitted flux values	Fitted flux values with SNR < 5.0 are assigned a score of 0.0, SNR > 20.0 a score of 1.0, and a linearly scaled value in between
<b>hifa_timegaincal</b>	Determining X-Y / X2-X1 phase solution deviations	Standard deviation of X-Y phase difference converted to path length: 1.0 if lower than 4.25e-6 m, 0.0 if higher than 7.955e-2 m, with an exponential decrease in between. Standard deviation of X2-X1 phase differences of subsequent integrations converted to path length: 1.0 if lower than 3.08-e-5 m, 0.0 if higher than 2.24e-2 m, with an exponential decrease in between. NB: The high limits are currently dummies. Determining their realistic values is still under development.
<b>hifa_applycal</b>	Determining percentage of incremental flagging	0%-5% -> 1.0, 5%-50% -> 1.0...0.5, >50% -> 0.0
<b>hif_exportdata</b>	Determining Pipeline products have been exported	1.0 when files successfully exported



## 4.5 CASA Commands Log

The **CASA Commands Log** is written by the Pipeline and is provided as part of an ALMA delivery. This is not the same as the casapy log. The **CASA Commands Log** provides all of the CASA task commands (as opposed to Pipeline tasks) used by the Pipeline to process a dataset. While this log cannot be used to create a CASA reduction script that is equivalent to the Pipeline processing, it is very useful indeed for seeing what CASA commands and parameter settings in these commands were used. The log is commented to indicate which Pipeline stage the tasks were called from and why.

## 5 ALMA Total Power (Single Dish) Data

This section explains how to use the Pipeline for ALMA Total Power (TP) data and how to navigate the Pipeline WebLog.

For TP data, there is currently no script to restore raw ALMA data to calibrated measurement sets (MSes) by directly applying the Observatory-determined calibration tables. Instead, for ALMA TP data that have been calibrated and imaged by the Single Dish Pipeline, when **scriptForPI.py** is used, it will call **casa\_pipescript.py** to perform a full re-calibration of the data (see Section 5.1). Please note that processing may take a significant amount of time to re-calibrate the TP Data. To know how long it will take, see the “Execution Duration” which is shown on the top-page of the WebLog.

Note that the images included in a delivery package have a native frequency resolution and a cell size of one ninths of beam size, as the CASA Guides recommend. If you just want to change them to your preferable frequency resolution and cell size, we recommend you to import the delivered FITS cubes to CASA and regrid it using the CASA task **imregrid**. If you want to perform baseline subtraction using your preferable mask range rather than the pipeline-defined range, we recommend to do it on the images using the CASA task **imcontsub** or recommend to do it using the CASA task **tsdbaseline** during your own manual calibration (refer to the CASA Guides).

### 5.1 The Single Dish Pipeline Calibration Script: casa\_pipescript.py

The **scriptForPI.py** calls **casa\_pipescript.py**. A typical script for processing Cycle 3 data is shown in Figure 8.

**Figure 8: Example Single Dish Pipeline calibration script casa\_pipescript.py**

```
__rethrow_casa_exceptions=True

h_init()

hsd_importdata(vis = ['uid__A002_X877e41_X452'])

hsd_flagdata(pipelinemode='automatic')

hifa_tsyscal(pipelinemode='automatic')
```

```
hifa_tsysflag(pipelinemode='automatic')
hsd_mstoscantable(pipelinemode='automatic')
hsd_inspectdata(pipelinemode='automatic')
hsd_calsky(pipelinemode='automatic')
hsd_applycal(pipelinemode='automatic')
hsd_baseline(pipelinemode='automatic')
hsd_blflag(pipelinemode='automatic')
hsd_baseline(pipelinemode='automatic')
hsd_blflag(pipelinemode='automatic')
hsd_imaging(pipelinemode='automatic')
hsd_exportdata(pipelinemode='automatic')

h_save()
```

Although CASA 4.5.1 Pipeline is optimized to process Cycle 3 data, in case that the data of earlier Cycle than 3 is processed in Pipeline, users will be aware that there is one additional command, “**hsd\_simplexscale(pipelinemode='automatic')**” after the `hsd_applycal` command in `casa_pipescript.py`.

### 5.1.1 Running `casa_pipescript.py` from ALMA Delivery Packages

For TP data, the easiest way to reproduce the Pipeline calibration from ALMA Delivery Packages is to run **scriptForPI.py**. Alternatively the following steps can be performed manually but it is essentially the same.

To reproduce the Single Dish Pipeline calibration performed as part of the ALMA Quality Assurance Process:

- Copy **casa\_pipescript.py** from the **script** directory to the **raw** directory
- Copy **jyperk.csv** and **\*flagtemplate.txt** from calibration directory to the **raw** directory

In the raw directory:

- Make sure the naming of the raw ALMA data is consistent with those provided in the script (e.g. if the data ends in **.asdm.sdm** then move to names which do not have this suffix)
- Start the version of CASA containing Pipeline using **casapy --pipeline**, then type **execfile('casa\_pipescript.py')**

Running the script will create:

- A calibrated MS per antenna for each ASDM in the same directory
- A **pipeline-\*/html** directory containing
  - The Pipeline WebLog
  - The CASA Commands Log

### 5.1.2 Manual Imaging after running casa\_pipescript.py

After calibrating with `casa_pipescript.py`, it is possible to re-image using the CASA Single Dish task, **sdimaging**, with user-defined parameters. Single Dish Pipeline creates a calibrated MS per antenna for each ASDM, which are to be selected as **infile**s parameter in **sdimaging** command. For other parameters in **sdimaging**, refer to **script/\*casa\_commands.log** under the delivered package.

## 5.2 Modifying the Pipeline Run

### 5.2.1 Flagging Auto-correlation data

It is possible to insert manually (in a text file, *flagtemplate.txt*) additional flagging not identified by pipeline if necessary. Please pay attention that you need to add “&&\*” (which means cross- and auto-correlation) or “&&&” (which means auto-correlation) in “antenna” keyword with the name of the antenna to flag a bad antenna. You can find an example in Figure 9.



Figure 9: Weblog view of `hsd_flagdata` showing an example of `flagtemplate.txt`

## 5.3 WebLog Quality Assessment Scoring for Single-Dish Data

Pipeline Task	Pipeline QA Scoring Metric	Score
<b>hsd_importdata</b>	Checking that the required calibrators are present	1.0 ATMOSPHERE intents are present
		0.5 subtracted for existing processing history
		0.5 subtracted for existing model data
		1.0 one continuous observing session

		1.0 all source coordinate are present
<b>hsd_flagdata</b>	Determining percentage of incremental flagging	<p>0 &lt; score &lt; 1 == 60% &lt; fraction flagged &lt; 5%" (for 'online', 'shadow', 'qa0', 'before' and 'applycal')</p> <p>where "0 &lt; score &lt; 1 == HIGH% &lt; fraction flagged &lt; LOW%" means</p> <ul style="list-style-type: none"> <li>• Score is 0 if flag fraction &gt;= HIGH%</li> <li>• Score is 1 if flag fraction &lt;= LOW%</li> </ul> <p>Score is linearly interpolated between 0 and 1 for fractions between HIGH% and LOW%</p>
<b>hsd_baseline</b>	Checking that one or more than one emission line is detected by line-finder.	1.0 there is more than one emission line detected in at least one spw.
		0.0 No line is detected in all spw.
<b>hsd_bflag</b>	Determining percentage of incremental flagging	<p>1.0 &lt;5% flagged.</p> <p>1.0 – 0.5 5%-50% flagged</p> <p>0.0 &gt; 50% flagged</p>
<b>hsd_imaging</b>	Checking that all Kelvin-to-Jy conversion factors are provided	0.0 Missing Kelvin-to-Jy conversion factor for some data
<b>hsd_exportdata</b>	Checking that the required files are exported	1.0 pipeline processing request file is exported
		1.0 pipeline weblog file is exported
		1.0 pipeline script file is exported
		1.0 pipeline commands log file is exported
		1.0 No missing final flag version files
		1.0 No missing final apply commands files
		1.0 No missing caltables files

## 5.4 Navigation of the WebLog for Single-Dish data

This section describes navigation of the Task sub-pages for each Single Dish Pipeline task starting from “By Task” tab. General explanation for “Home”, “Observation Summary”, “By Topic” and “By Task” pages, refer to Section 4.4. For task description, refer to “ALMA Science Pipeline Reference Manual”.

### 5.4.1 hsd\_importdata

The weblog for hsd\_importdata task shows the summary of imported MSes, and also flux densities of calibrators (for pointing, sideband ratio, or WVR). Single Dish Pipeline never use the values imported here.

### 5.4.2 hsd\_flagdata

The weblog for hsd\_flagdata shows summary of percentage of flagged data per MS due to binary data and online flagging, manually inserted template, shadow, unwanted intents, and edge of channel. Note that the value in “Before Task” column corresponds to the percentage of flagged data by binary data flagging, which is applied when importing ASDMs (at hsd\_importdata).

### 5.4.3 hifa\_tsyscal

The weblog for hifa\_tsyscal task shows a list of pair of Tsys and science spectral windows to be used for Tsys (amplitude-scale) correction, and also show an original Tsys spectra per spectral window.

### 5.4.4 hifa\_tsysflag

The weblog for hifa\_tsysflag task shows flagged Tsys spectra per spectral window after heuristic flagging is applied.

### 5.4.5 hsd\_mstoscantable

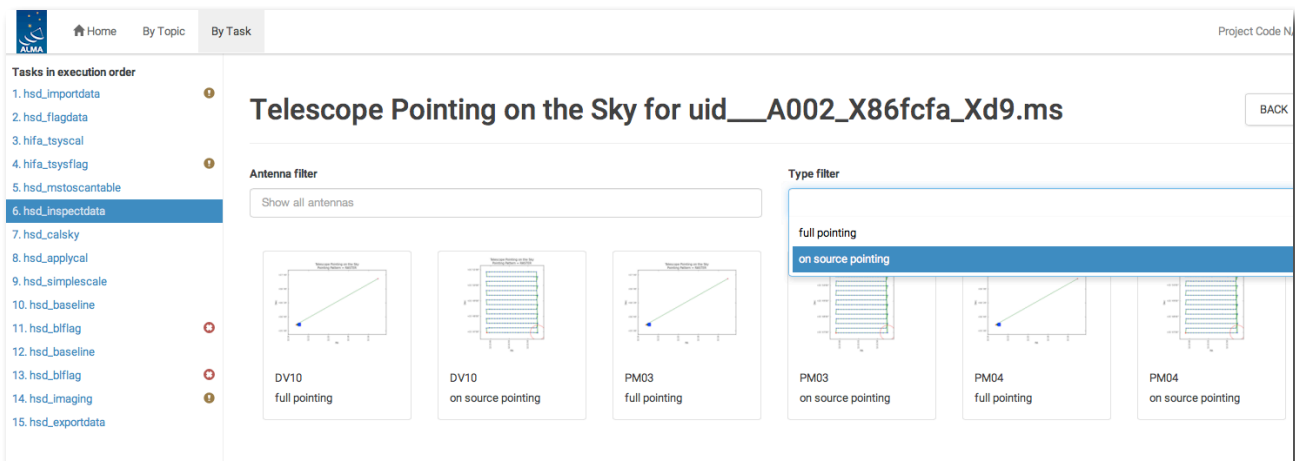
The weblog for hsd\_mstoscantable task shows a table of MS and its converted Scantables (with an suffix of “.asap”). Newly generated Tsys tables (per antenna per spectral window) at this stage are also shown with the list of their original Tsys table from hifa\_tsyscal.

### 5.4.6 hsd\_inspectdata

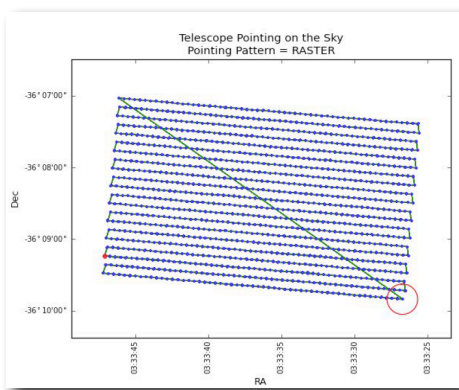
The weblog for hsd\_inspectdata shows plots of Telescope Pointing, Azimuth and Elevation, Weather, and WVR. Both flagged and unflagged data are displayed in different symbol and color: i.e., flagged data is always shown as gray color dots.

#### Note on WVR Plots

The plots show the power directly read from the WVR receiver vs time.



**Figure 10: the detailed page of Telescope Pointing for hsd\_inspectdata. Two types of plots appear in the hsd\_inspectdata weblog page for each antenna. One is for telescope pointing including ON and OFF source pointings, while the other is only for ON source pointings (also see Figure 11). These two types of plots can be filtered by choosing “full pointing” or “on source pointing”.**



**Figure 11: An example of “ON source pointing” plot. The red circle indicates the beam size of the antennas and its location is the starting position of the raster scan. The Red (small) dot indicates the last position of the raster. The green line is the motion of the antenna, and thus, the green line from the top left corner to the bottom right corner indicates that the antenna goes to the last scan and returns to the start position.**

### 5.4.7 hsd\_calsky

The weblog for hsd\_calsky shows the integrated OFF spectra per spectral window (Figure 12). At the top of the plots the coordinates of the OFF position are shown. The y-axis is the direct output from the correlator, which means the values are dominated by signals from both the atmosphere and receivers.

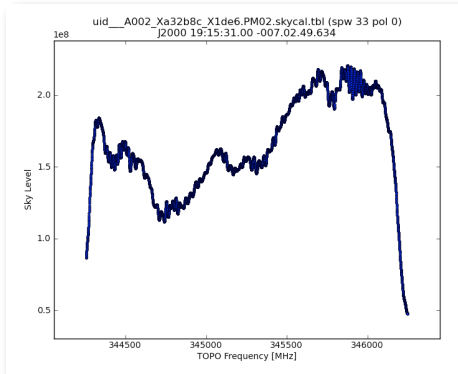


Figure 12: An example of OFF spectra

### 5.4.8 hsd\_applycal

It shows a list of the calibrated Scantables with a name of the applied Tsys and Sky tables per antenna and per spectral window.

### 5.4.9 hsd\_baseline

The weblog of hsd\_baseline shows the spectral maps (which sometimes we call “sparse profile map”) before and after baseline subtraction, and plots for line detection by clustering analysis. In the summary page, the spectral maps for **before** and **after** baseline subtraction (top and bottom lines in Figure 13, respectively). The detail page has several similar plots but for each EB, antenna, and polarization. Note that the \*representative\* spectral map for each spectral window in the summary page was randomly chosen out of all spectral maps in the detail pages.

#### Note on R.A. vs Dec. plots in Line Detection by Clustering Analysis

There are four different plots per spectral window, i.e. “clustering\_detection”, “clustering\_validation”, “clustering\_smoothing”, and “clustering\_final” (Figure 15). The number of plots in each figure is the same as that of candidate line components.

The line finding algorithm first integrates spectrum per XY-grid (per beam size) per three time-domain groups (i.e. per voxel), and searches for emission line candidates having a brightness above a certain threshold. The “cluster\_detection” plot (Figure 16) shows the grid cells having emission line exceeding the threshold. Then the algorithm calculates how many spectra with emission lines are in each voxel, and judges whether the grid cell possibly contains true emission lines. At this line detection validation step, the ratio of the number of spectra having detected emission lines (defined as “Nmember”) per grid cell and the number of total spectra belonging to the grid cell (“Nspectra”) is calculated. The ratio, Nmember/Nspectra, is shown in the “clustering\_validation” plot (see Figure 17). The grid cell is marked as 1) “Validated” if  $N_{\text{member}}/N_{\text{spectra}} > 0.5$  (Blue squares), 2) “Marginally validated” if  $N_{\text{member}}/N_{\text{spectra}} > 0.3$  (Cyan squares), and 3) “Questionable” if  $N_{\text{member}}/N_{\text{spectra}} > 0.2$  (Yellow squares).

After the validation step, the grid containing the Nmember/Nspectra ratio per grid cell is smoothed by a Gaussian-like grid function. This is to eliminate the isolated grid cells having emission line candidate, while to enhance the grid cells having emission line that continues to the neighboring grid cells. Figure 18 shows an example of “clustering\_smoothing”, displaying a

map of smoothed Nmember/Nspectra ratio.

Finally, the mask channel range is estimated per spectrum. In the validated area through validation and smoothing step (blue squares in Figure 18, and green squares in Figure 20), emission-line channels (mask) are calculated over the spatial domain, extrapolated per spectrum and applied to the baseline subtraction. The mask channel range is determined and used in baseline subtraction also in the green squares in Figure 20 (“clustering\_final”). The mask range is set as the same as the one of the spectra at the nearest edge of validated area: i.e. a spectra at a red circle in the straight line between the center of a blue square and the center of the nearest green square in Figure 19.

The determined line-width and line-center of the identified emissions are analyzed by clustering analysis (Figure 21). Then each combination of clusters is scored based on the cluster size on the “line center”-”line width” space. The scoring is empirically defined so that the score gets better (smaller) when [1] the cluster size is smaller, [2] the number of clusters is smaller, and [3] the number of outliers is fewer than those of other clusters. The users will know which number of clusters is more plausible by searching for the number of clusters with the lowest score. This plot is basically for developers.

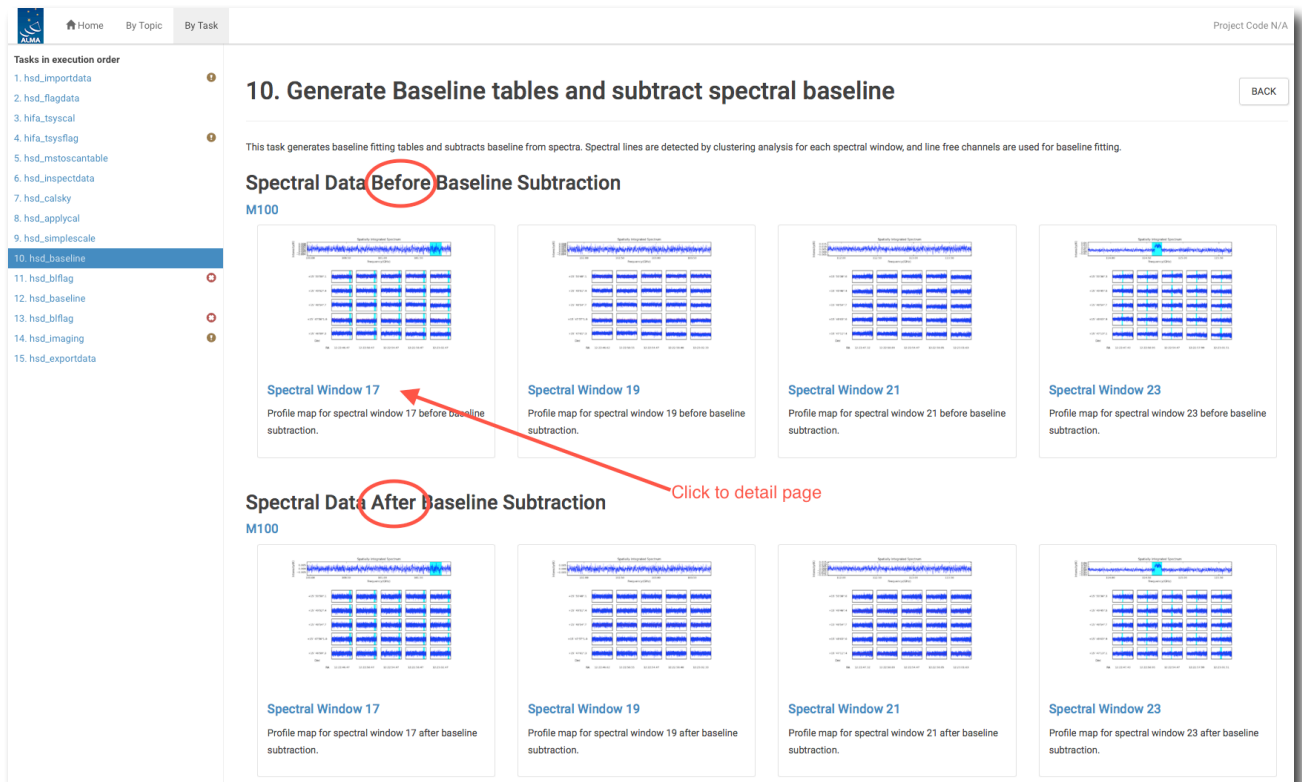
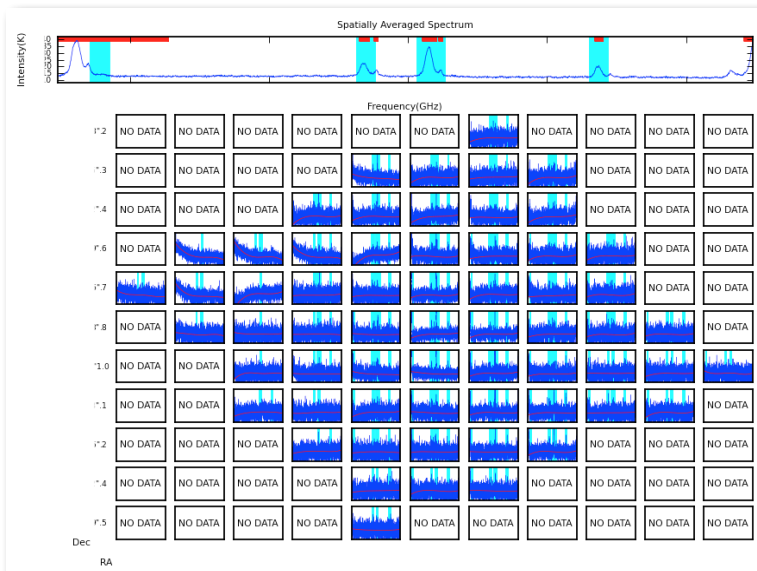


Figure 13: The summary page of hsd\_baseline.





**Figure 14: An example of a plot of averaged spectrum and spectral map. On the top of the spectral map, a spatially averaged spectrum per EB, antenna, spectral window and polarization is shown. The cyan filled regions indicate the mask channels containing identified emission line. The red thick bar indicates additionally masked channels to avoid from fitting unwanted features such as lines at the band edge and atmospheric absorption lines. On the bottom, each small panel shows \*one\* representative spectrum per grid cell. The red lines in each small panel indicate the fitted baseline in the plots of “Before Baseline Subtraction”, and the zero-level in the plots of “After Baseline Subtraction”, respectively.**

**Figure 15: Details page of results of clustering analysis (hsd\_baseline)**

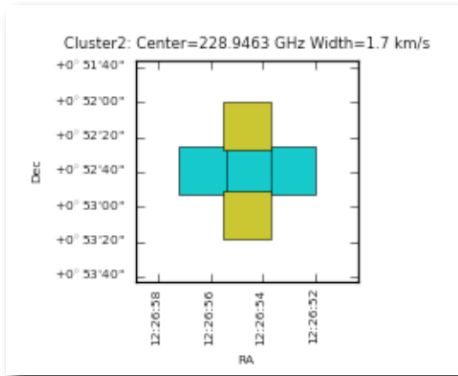


Figure 16: Example of “clustering\_detection” plot. The yellow grid cells show a region that there is one (pipeline-defined) time-domain group having detected emission lines. Cyan squares indicate a grid cell that there are more than one time-domain groups having detected emission lines.

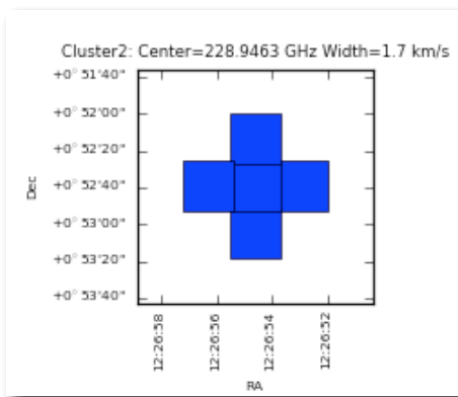


Figure 17: Example of “clustering\_validation”. Blue squares represent a grid cell with  $N_{\text{member}}/N_{\text{spectra}} > 0.5$  (Validated).

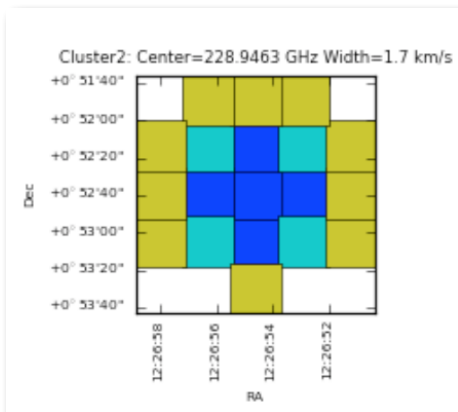


Figure 18: Example of “clustering\_smoothing”. Blue squares represent the grid cells with points exceeding a defined threshold, i.e., the grid cells having promising emission lines that have smoothly distributed to the neighboring grid cells. Cyan and yellow squares are the grid cells with points slightly below the threshold (Border), and lower than threshold (Questionable), respectively.

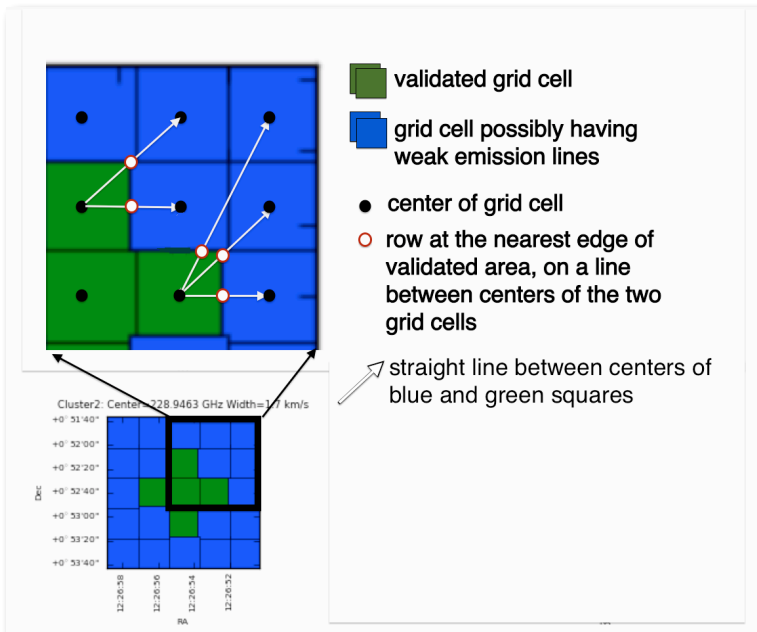


Figure 19: In the blue squares, the mask channel range is used of the one for the spectrum at the nearest edge of validated area (white-filled red circle). Green squares indicate final grid cells where the masked emission line channels are calculated and applied to the baseline subtraction. Blue squares represent final grid cells where the calculated line protection channels are applied to the baseline subtraction.

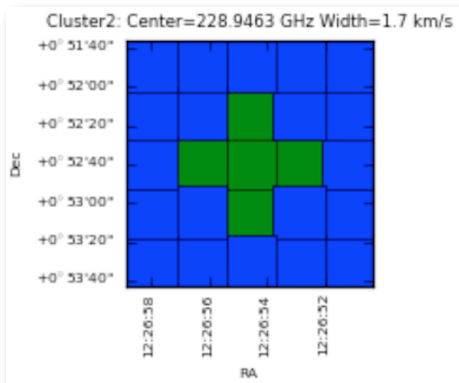


Figure 20: An example of "clustering\_final". Color indications are the same as Figure 19.

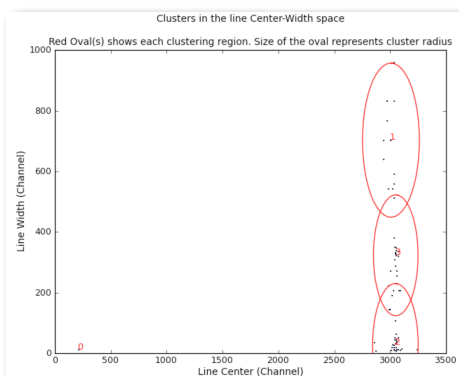
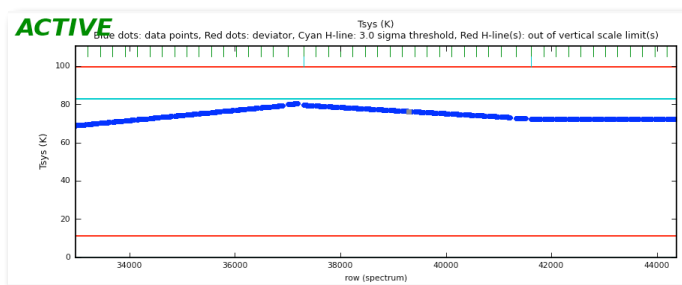


Figure 21: Example of line detection clustering in the line width vs. line center parameter space. The small dots indicate spectra containing identified emission line. The red ovals show each clustering region with a size of the cluster radius.

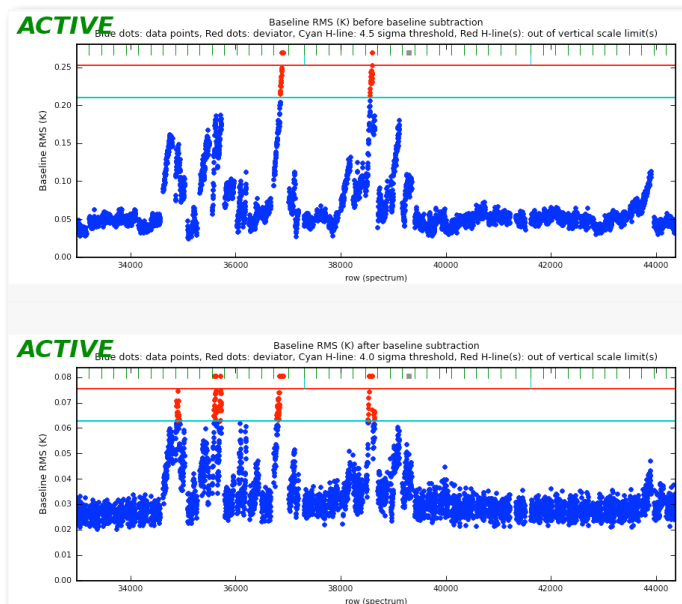
## 5.4.10 hsd\_blflag

The weblog for hsd\_blflag task shows summary of percentage of flagged data per MS per antenna per spw per polarization. Note that “Online” column represents the percentage of flagged data before the hsd\_blflag task. The percentage of flagged data by this task are shown in the “Baseline RMS”, “Running mean” and “Expected RMS” column.

There are seven figures by clicking the “details” page. They are plots of (1) Tsys (Figure 22), (2) baseline RMS before baseline subtraction, (3) baseline RMS after baseline subtraction (Figure 23), (4) deviation of moving average of baseline RMS before baseline subtraction, (5) deviation of moving average of baseline RMS after baseline subtraction, (6) baseline RMS before baseline subtraction with expected RMS, and (7) baseline RMS before baseline subtraction with expected RMS as a function of time series spectrum. For the detailed derivation of these parameters, refer to the ALMA Pipeline Reference Manual.



**Figure 22:** An example of Tsys plot. The blue, red and gray dots represent data points, outliers, and previously flagged data respectively. Cyan line is a threshold.



**Figure 23:** An example of baseline RMS plot, before (top panel) and after (bottom panel) the baseline subtraction. The symbols and lines are the same as in Figure 22.

## 5.4.11 hsd\_imaging

The summary page for hsd\_imaging shows a list of the applied Kelvin-to-Jy conversion factors, spectral maps (Figure 24), channel maps (Figure 25), baseline RMS map, and integrated intensity map.

### Note on Spectral Maps (Profile Map)

Three types of spectral maps are available in the weblog for hsd\_imaging (Figure 24): 1) a simplified profile map of the combined image per spectral window at the top, 2) a simplified profile map per antenna, and 3) a detailed profile map. To access the simplified profile map per antenna, click the corresponding “Spectral Window”. Each spectrum of (1 and 2 above) the simplified profile map corresponds to an averaged spectrum in area which corresponds to one-ninths of the image size (imsize), so that the total number of spectra in the profile map is 8 times 8 (If the number of pixels (along x- or y-axis) is less than eight, it shows all spectrum per pixel.) To see the detailed profile maps, click the icon with a symbol of polarization in the polarization column (see Figure 24). Each bin of the profile map is equivalent to spectrum per pixel, but with an interval of three cells. Due to the limitation of the available number of plots possible per page (max 5 times 5 plots in a page), the rest of the plots is displayed in other pages.

### Note on Channel Maps

In the detail page, the number of channel maps corresponds to the number of emission lines that have been identified by the clustering analysis.

### Note on Baseline RMS Map

The Baseline RMS Map is created using baseline RMS stored in the baseline tables. The baseline RMS is calculated by hsd\_baseline using emission free channels.

### Note on Integrated Intensity Map

The Integrated Intensity Map for each spectral window is generated using immoments task using all available channel range.

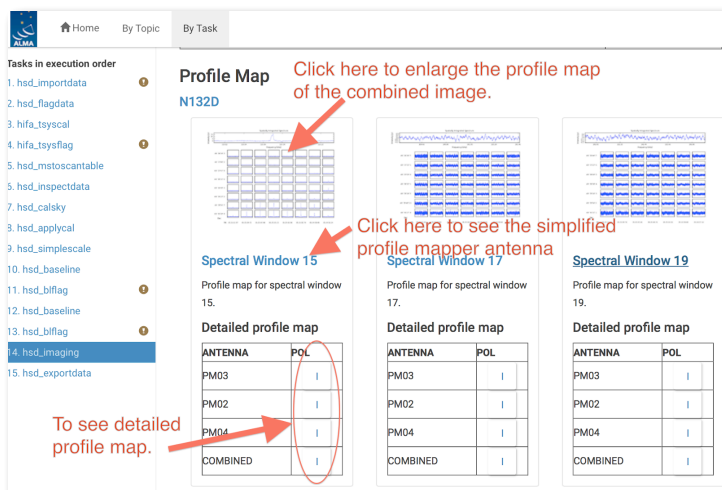


Figure 24: Summary page of hsd\_imaging

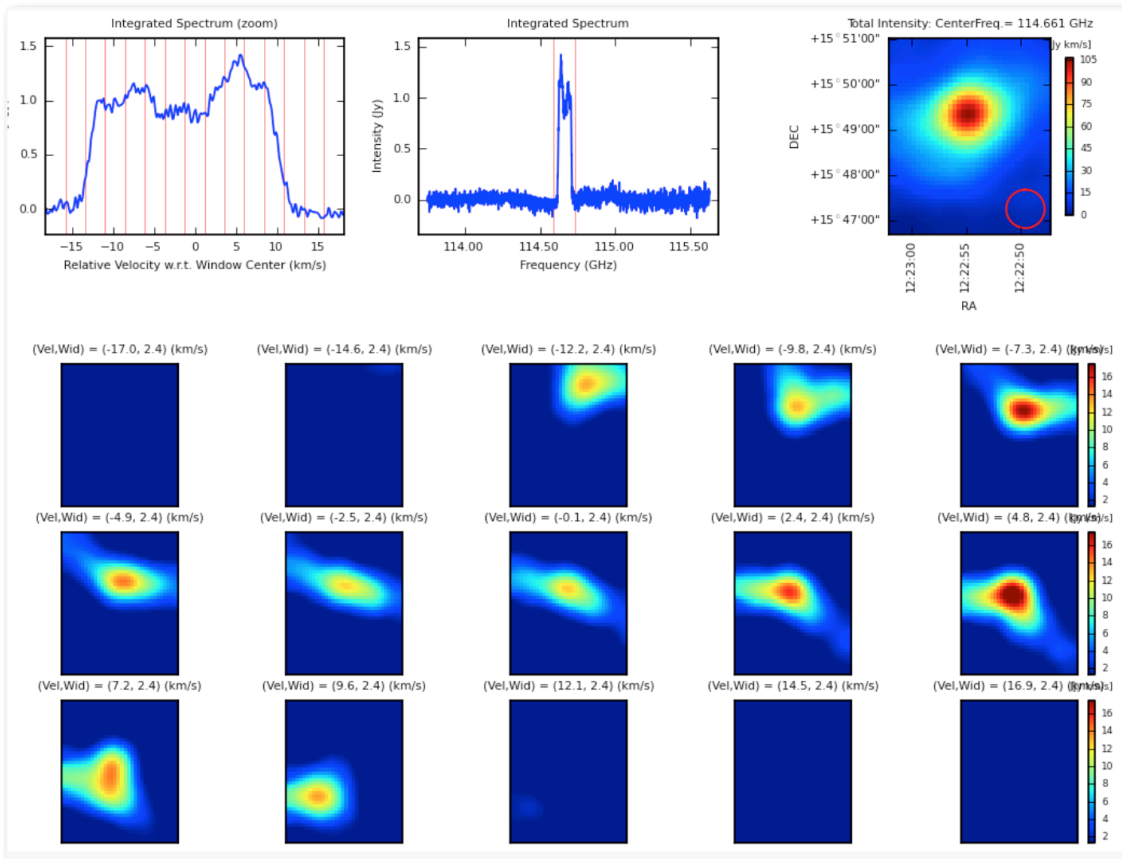


Figure 25: An example of channel map per detected line. *Top-middle*) the identified emission line with a determined linewidth (sandwiched by two red vertical lines), overplotted on the averaged flux spectrum (in Jy) as a function of frequency (in GHz). *Top-left*) the zoom-up view of the identified emission line, but with velocity axis. (Please be aware of that the horizontal axis of the top-left panel is converted from frequency of the top-middle panel to velocity.) The vertical axis is the averaged (over the entire map) flux in Jy and the horizontal axis is in units of km/s. The (center) velocity of 0 km/s corresponds to the center frequency of the emission line, while the velocity ranges equivalent to the masked region as identified emission line. The line velocity width is gridded into 15 bins, which are shown as red vertical lines. *Top-right*) the total integrated intensity map (in Jy/beam km/s) over the channel range of the determined line width. *Bottom*) the channel maps within the velocity range of the identified emission line. Each panel corresponds to each bin on the top-left panel.



The Atacama Large Millimeter/submillimeter Array (ALMA), an international astronomy facility, is a partnership of the European Organization for Astronomical Research in the Southern Hemisphere (ESO), the U.S. National Science Foundation (NSF) and the National Institutes of Natural Sciences (NINS) of Japan in cooperation with the Republic of Chile. ALMA is funded by ESO on behalf of its Member States, by NSF in cooperation with the National Research Council of Canada (NRC) and the National Science Council of Taiwan (NSC) and by NINS in cooperation with the Academia Sinica (AS) in Taiwan and the Korea Astronomy and Space Science Institute (KASI).

ALMA construction and operations are led by ESO on behalf of its Member States; by the National Radio Astronomy Observatory (NRAO), managed by Associated Universities, Inc. (AUI), on behalf of North America; and by the National Astronomical Observatory of Japan (NAOJ) on behalf of East Asia. The Joint ALMA Observatory (JAO) provides the unified leadership and management of the construction, commissioning and operation of ALMA.

