SC/20.3

Starlink Project
Starlink Cookbook 20.3

H. S. Thomas, Malcolm J. Currie, & H. A. L. Parsons

2021 June 22

The Heterodyne Data Reduction Cookbook 1.3

Abstract

This cookbook provides a short introduction to Starlink facilities, especially SMURF—the Sub-Millimetre User Reduction Facility—and KAPPA—the Kernel Application Package—for reducing, displaying, and analysing ACSIS data. In particular, this cookbook illustrates the various steps required to reduce the data; and gives an overview of the ORAC-DR pipeline which carries out many of these steps using a single command. Specialised pipeline recipes are discussed. The cookbook also introduces cube analysis with the GAIA display tool, and spectral analysis with SPLAT.

Contents

List of Figures

Acronyms

ACSIS	Auto-Correlation Spectrometer and Imaging System
ARD	ASCII Region Definition
CADC	Canadian Astronomy Data Centre
CSO	Caltech Submillimetre Observatory
FITS	Flexible Image Transport System
FWHM	Full-Width at Half-Maximum
GAIA	Graphical Astronomy and Image Analysis Tool
HDS	Hierarchical Data System
JCMT	James Clerk Maxwell Telescope
LSB	Lower sideband
MSB	Minimum Schedulable Block
NDF	Extensible *N*-Dimensional Data Format
HARP	Heterodyne Array Receiver Programme
QA	Quality Assurance
RxA	Receiver A
SAMP	Simple Application Messaging Protocol
SDF	Starlink Data File
S/N	Signal-to-Noise ratio
USB	Upper sideband
WCS	World Co-ordinate System
WVM	Water Vapour radioMeter

Chapter 1 Introduction

1.1 This cookbook

This cookbook is designed to instruct ACSIS users on the best ways to reduce and visualise their data using Starlink packages.

This guide covers the following:

- Chapter 1 – Computer resources and software you'll need before getting started.
- Chapter 2 – Starlink basics—tools for exploring your data.
- Chapter 3 – Heterodyne observing at the JCMT.
- Chapter 4 – Naming convention and examining raw ACSIS data.
- Chapter 5 – The ACSIS pipeline and recipes explained.
- Chapter 6 – How to run the pipeline.
- Chapter 7 – Regridding your data with makecube.
- Chapter 8 – Analysing your reduced cube.
- Chapter 9 – Advanced analysis techniques such as changing co-ordinate frames.
- Chapter 10 – Using GAIA to analyse your data.
- Chapter 11 – Plotting spectra using SPLAT.
- Chapter 12 – Getting your data from CADC.

Style convention
In this cookbook the following styles are followed:

- `%` represents the command-line prompt.
- Command-line examples are shown in `navy typewriter font.`
- Parameters, FITS keywords, parameters and their values are presented in `typewriter font`.
- The names of Starlink commands appear as in sans serif font.
- The names of commands, menu items, tabs, or buttons within graphical tools (such as GAIA or SPLAT) appear **like this**.
- The names of windows produced by graphical tools appear "`like this`".
- The names of attributes appear like StdOfRest.
- Useful tips or reminders are given in outlined boxes.

1.2 Before you start: computing resources

Before reducing heterodyne data you should consider whether you have sufficient computing resources for your data. That said, there can only be vague guidelines because resource usage depends on many factors, such as the spatial area and resolution, the number of observations being reduced together, observing and bandwidth modes (see Section 3.3), and the number of subbands. However, the main factor is the volume of data being processed concurrently.

The ORAC-DR [4, 14] pipeline can demand approaching 300 GB of peak storage for the largest (about a degree square) HARP maps, and at least 24 GB of memory during spectral-cube creation. For more-normal-sized maps of individual targets, say 20 square arcminutes from your observations, would only require about 10 GB of storage and most modern computers would have sufficient memory. The storage requirements can more than double if all intermediate files are retained for diagnostic purposes; intermediate files are normally removed at the end of each pass through a recipe. Reducing manually permits you to tidy files as you move along, but is very time consuming.

Reducing Nāmakanui data and HARP stares, small jiggle maps, or older RxA data is undemanding of resources.

1.3 Before you start: Starlink software

This manual utilises software from the Starlink package; SMURF [5], KAPPA [8], GAIA [10], ORAC-DR [4], CONVERT [6], CUPID [1], CCDPACK [11], and PICARD [12]. Starlink software must be installed on your system, and Starlink aliases and environment variables must be defined before attempting any ACSIS data reduction. You can download Starlink from the Starlink webpage.

Below is a brief summary of the packages used in this cookbook and how to initialise them. Note that all the example commands are shown within a UNIX shell.

1.4 Options for reducing your data

You have three options for processing your data:

(1) performing each step manually,
(2) write your own scripts, or
(3) running the automated pipeline.

The automated pipeline is recommended for new users of JCMT heterodyne data or those unfamiliar with the Starlink software. The pipeline approach works well if your project is suited to using one of the standard recipes, as are most projects. Running the pipeline is probably essential if you have a lot of data to process. To use the science pipeline, skip straight to Chapter 5.

Performing each step by hand allows more fine-tuned control of certain processing and analysis steps, although some fine-tuning is available in the pipeline via recipe parameters (see Section 6.2).

Once you have determined your optimal parameters you can pass them to the pipeline or a script. Chapter 7 and Chapter 8 discuss the manual approach.

While you have the option of running the pipeline yourself, you can find pipeline reduced files for individual and co-added observations by night in the JCMT Science Archive (JSA).

These have been reduced using the ORAC-DR pipeline with the recipe specified in the MSB or with the Legacy Survey recipe. Principal investigators (PIs) and co-investigators (co-Is) can access these data through the JSA before it becomes public by following the instructions in Section 12.

Package	**Description**	**Initialise**	**Help**
SMURF	The Sub-Millimetre User Reduction Facility (SMURF) contains makecube that will process raw ACSIS data into spectral cubes.	`% smurf`	`% smurfhelp` **SUN/258**
KAPPA	A general-purpose applications package with commands for processing, visualising, and manipulating NDFs.	`% kappa`	`% kaphelp` **SUN/95**
CONVERT	CONVERT allows the interchange of data files to and from NDF. Other formats include IRAF, FITS, and ASCII.	`% convert`	`% conhelp` **SUN/55**
CUPID	CUPID is a package of commands that allows the identification and analysis of clumps of emission within one-, two- or three-dimensional data arrays	`% cupid`	`% cupidhelp` **SUN/255**
ORAC-DR	The ORAC-DR Data Reduction Pipeline [4] is an automated reduction pipeline. ORAC-DR uses SMURF and KAPPA (along with other Starlink tools) to perform an automated reduction of the raw data following pre-defined recipes.	`% oracdr_acsis`	**SUN/230**
PICARD	PICARD uses a similar pipeline system as ORAC-DR but for the post-processing of reduced data. While mainly implemented for SCUBA-2 data, there are a few recipes that are heterodyne compatible. See Appendix A for more details and a description of the available recipes.	`% picard RECIPE <files>`	**SUN/265**

Tool	**Description**	**Open**	**Help**
GAIA	GAIA is an interactive image and data-cube display and analysis tool. It incorporates tools such as source detection, three-dimensional visualisation, clump visualisation, photometry, and the ability to query and overlay on-line or local catalogues.	`% gaia`	**SUN/214** **SC/17**
HDSTRACE	This tool lets you examine the contents of Starlink data files.	`% hdstrace <file>`	**SUN/102**
SPLAT	SPLAT is a graphical spectral-analysis tool. It can also interact with the Virtual Observatory.	`% splat`	**SUN/243**

Chapter 2
Good to Know: Starlink

2.1 File format

Starlink routines run on HDS files (Hierarchical Data System), which normally have a `.sdf` extension (Starlink Data File). HDS files can describe a multitude of formats. The most common HDS format you will encounter is the NDF (*N*-Dimensional Data Format)[7]. This is the standard file format for storing data that represent *N*-dimensional arrays of numbers, such as spectra, and images. The parameter files discussed in Section 2.3 are also HDS format.

Raw data retrieved from the JCMT Science Archive come in FITS format. For information on converting between FITS and NDF see Appendix B.

Tip

The `.sdf` extension of HDS filenames is not required when running most Starlink commands (the exception is PICARD).

2.2 Parameters

Parameters control the behaviour and data processed in Starlink applications. A parameter expects a value that is one of the following types: string, boolean, integer, or single- or double-precision floating point. You can either specify parameter values on the command line or in response to prompts. Most parameters have sensible defaults leaving you to concentrate on the main parameters. Parameter usage is described in a tutorial.

2.3 How to find the current parameter values

A directory called `adam` is created in your home-space by default when you run Starlink applications. In this directory you will find HDS files for each of the application that you have run. These files contain the parameters used and results returned (if appropriate) from the last time you ran a particular application.

You can specify a different location for the `adam` directory by setting the environment variable `ADAM_USER` to be the path to your alternative location. This is useful when running more than one reduction to avoid interference or access clashes.

To see the ADAM parameters, run HDSTRACE on any of the parameter files. For example, to see which parameters were used and the results from last time you ran stats, you can type the following from anywhere on your system:

```
% hdstrace ~/adam/stats
```

This will report the following

```
STATS  <STRUC>
   ADAM_DYNDEF    <DEFAULTS>      {structure}
   COMP           <_CHAR*132>     'DATA'
   ORDER          <_LOGICAL>      TRUE
   MAXIMUM        <_DOUBLE>       36.666870117188
   MAXWCS         <_CHAR*132>     '10.625136, -0.381122, -6.057133'
   MINWCS         <_CHAR*132>     '10.720136, -0.017795, -31.05713'
   MEAN           <_DOUBLE>       0.25890738014562
   MINIMUM        <_DOUBLE>       -2.7510244846344
   SIGMA          <_DOUBLE>       0.72429928153974
    ..               ..                  ..
    ..               ..                  ..
```

You can see stats was run on the data array with ordered statistics as well as the resulting values. Any of the parameters returned by running HDSTRACE on an ADAM file can be extracted using the command parget. In the example below, the mean value from the last instance of stats is printed to the screen.

```
% parget mean stats
```

Tip

Try not to exit from Starlink tasks by breaking in, since this may corrupt the parameter file (reported as an integrity check error). If this happens delete the parameter file. Enter `!!` at a prompt if possible to have a clean exit.

2.3.1 Extracting a value for scripting

parget is designed to make life easier when passing values between shell scripts. In the C-shell scripting example below, the median value from histat is assigned to the variable med. Note the use of the back quotes.

For more information on scripting our work see Appendix C.

```
set med = `parget median histat`
```

If the parameter comprises a vector of values these can be stored in a C-shell array. For other scripting languages such as Python, the alternative vector format produced by setting parameter VECTOR to TRUE may be more appropriate. Single elements of a parameter array may also be accessed using the array index in parentheses.

In addition to running hdstrace on the ADAM file, you can find a list of all parameter names that can be returned with parget in the KAPPA manual under 'Results Parameters' for the command in question. Note that these names may be different from the names returned in a terminal when running application.

2.4 How can I view the metadata?

There are two KAPPA tasks which are extremely useful for examining your metadata: fitslist and ndftrace, which can be used to view the FITS headers and properties, such as dimensions, of the data respectively. The third option is the stand-alone application HDSTRACE.

fitslist This lists the FITS header information for any NDF (raw or reduced). This extensive list includes dates & times, source name, observation type, band width, number of channels, receptor information, exposure time, start and end elevation and opacity. In the example below, just the object name is extracted.

```
% fitslist file | grep OBJECT
```

Likewise, if you know the name of the keyword you want to view, you can use the fitsval command instead, for instance

```
% fitsval file OBJECT
```

ndftrace ndftrace displays the attributes of the NDF data structure. This will tell you, for example, the units of the data, pixel bounds, dimensions, world co-ordinates (WCS), and axis assignations.

```
% ndftrace map fullframe fullwcs
```

An NDF can contain more than one set of world co-ordinates. The `fullwcs` parameter requests that all sets be listed rather than the currently chosen one.

hdstrace hdstrace lists the name, data type and values of an HDS (Hierarchical Data System) object. The following example shows the structure of a time-series cube, including the pixel origin of the data structure. To show just the first *x* lines of values for each parameter include the option `nlines=x` on the command line:

```
% hdstrace file nlines=3
```

Otherwise to see all the lines and information that is available in each extension use `nlines=all`. The example below displays all the system temperature values and pipes the result to 'less' to make viewing easier.

```
% hdstrace file.MORE.ACSIS.TSYS nlines=all | less
```

You can see it descends two levels (into `MORE` and `ACSIS`) to retrieve the information. Other information available at this level include receptors and receiver temperatures.

Full details of ndftrace and fitslist can be found in the KAPPA manual. Details on HDSTRACE can be found in the HDSTRACE manual.

2.5 What has already been done to the data?

If you are presented with a data file you may wish to see what commands have been run on it and, in the case of a co-added cube, which data went into it. Two KAPPA commands can help you with this:

hislist The KAPPA command hislist will return the history records of the NDF.

```
% hislist myfile brief
```

Including the `brief` option returns a simple list of the commands. Omitting this will include the text associated with each command and can be useful for finding out what parameters were used in each case. This works for all NDFs including those reduced by the pipeline, for these all the commands executed as part of the pipeline reduction script will be listed. The version of software used for each step is also reported.

provshow The KAPPA command provshow displays the details of the NDFs that were used in the creation of the given file. It includes both immediate parents and older ancestor NDFs.

```
% provshow file
```

2.6 How to examine, process or extract a subset of your data

For all Starlink commands you can specify a sub-section of your data on which to run an application. You do this by appending the bounds of the section to be processed the NDF name. This may be on the command line or in response to a prompt.

The example below runs stats on a sub-cube within your original cube. This sub-cube is defined by bounds given for each axis. The upper and lower bounds for each axis are separated by a colon, while the axes themselves are separated by a comma. Note that the use of quotes is necessary on a UNIX shell command line, but not in response to a prompt or in many other scripting languages.

```
% stats 'cube(10.5:10.0,0.0:0.25,-25.:25.)'
```

The bounds are given in the co-ordinates of the data (Galactic for Axes 1 and 2 and velocity for Axis 3). You can find the number and names of the axes along with the pixel bounds of your data file by running ndftrace. In the example above the ranges are 10.5° to 10° in Longitude (note the range goes from left to right), 0° to 0.25° in Latitude, and -25 to +25 km s^{-1} in velocity. To leave an axis untouched simply include the comma but do not specify any bounds.

Tip

Be careful to give world co-ordinate values in floating point. Integers are interpreted as pixel indices.

To define your bounds in FK5 co-ordinates use the following format.

```
% stats 'cube(22h18m:22h32m,05d:05d30m,)'
```

To write a section of your data into a new file (called `newcube.sdf` in the example below) use ndfcopy with bounds.

```
% ndfcopy 'cube(10.5:10.0,0.0:0.25,-25:25)' newcube title='"l=10.5 sub-section"'
```

Here the option `title` defines a title for the new cube which replaces the title derived from the original.

You can also define a region as a number of pixels from a given origin. The example below extracts a 25×25 pixels cube around the position $l = 10°$, $b = 0°$.

```
% ndfcopy 'cube(10.0~25:0.0~25,)' newcube
```

The extent of an NDF section can be specified as an arc-distance using WCS co-ordinates in the format of **"centre extent"**.

For instance, **'image(10:12:30 40am,-23:23:43 40am)'** will create a section of extent 40 arcminutes on both axes ("as" and "ad" can be used in place of "am", indicating arcseconds and degrees).

This only works if the NDF's current Frame is a SkyFrame (an error is reported otherwise).

Tip

To help identify your sub-region in pixel co-ordinates, open your file in GAIA and change the built-in co-ordinates to PIXEL via **Image-Analysis** > **Change coordinate** > **Built-in coordinates**.

2.7 How to get help

Command	Description	Usage
showme	If you know the name of the Starlink document you want to view, use showme. When run, it launches a new web page or tab displaying the hypertext version of the document.	`% showme sun95`

findme	findme searches Starlink documents for a keyword. When run, it launches a new web page or tab listing the results.	`% findme kappa`
		`% findme heterodyne`

docfind	docfind searches the internal list files for keywords. It then searches the document titles. The result is displayed using the UNIX more command.	`% docfind kappa`

Run routines with prompts	You can run any routine with the option `prompt` after the command. This will prompt for every parameter available. If you then want a further description of any parameter type `?` at the relevant prompt.	`% makemap prompt`
		`% REF - Ref. NDF /!/> ?`

Chapter 3 JCMT Heterodyne Observing

3.1 Spectral-line observing

Submillimetre instruments use heterodyne techniques to observe spectral lines at high resolution. Heterodyne receivers take a weak incoming signal and mix it with a reference frequency known as the local oscillator (LO). This results in a signal of lower (intermediate) frequency (IF) which can be amplified and more easily sampled. The current JCMT receivers have a standard IF of 6 GHz for the Nāmakanui insert ʻŪʻū, 5 GHz (HARP) and 4 GHz (for the retired RxA) - although these can be adjusted as needed by the user.This mixing produces two frequencies for each value of the LO ($\nu_{\rm sky} = \nu_{\rm LO} \pm \nu_{\rm IF}$) which are known as the upper sideband (USB) and lower sideband (LSB).

In a single-sideband receiver (denoted SSB) one of these is suppressed. Double-sideband receivers (denoted DSB) have the two frequencies superimposed. In Sideband Separating mixers (denoted 2SB) both the USB and LSB are generated separately.

Reducing the noise in the system is crucial to the efficiency of these receivers. The required integration time is directly proportional to the total noise in the system given by $T_{\rm sys}$ via the radiometer formula

$$T_{\rm A}^{*}({\rm rms}) = \frac{2\,T_{\rm sys}\,\kappa}{\sqrt{\Delta\nu\,t}},$$

where the terms are as follows:

- $\Delta\nu$ is the frequency resolution of each channel.
- t is the total integration time (time spent integrating on the source and the reference position).
- κ is the telescope specific backend degradation factor.
- The factor of 2 is required for position-switch or beam-switch observations where half the time is spend off source at the reference position. For frequency switched observations it is 1.414 instead.
- $T_{\rm sys}$ is the system temperature in kelvin and describes the total noise of the system. It includes contributions from the background sky temperature, the Earth's atmosphere, ground spillover picked up by the side lobes and the noise from the receiver itself (known as receiver temperature, $T_{\rm rx}$). The receiver temperature can be determined by measuring the effect of looking at hot and cold loads, where the cold load is black-body source.
- $T_{\rm A}^{*}({\rm rms})$ is the rms value of the noise observed in the spectrum in kelvin.

3.2 Heterodyne instruments

The heterodyne instruments described below are those currently operational, soon to be operational or recently retired at the JCMT. For information on previous instruments see the JCMT web pages.

In the following $\eta_{\rm MB}$ is the main-beam efficiency and $\eta_{\rm fss}$ is the forward spillover and scattering efficiency. For the latest and historical calibrations visit ACSIS Beam Efficiencies.

Tip

Both T_{sys} and T_{rx} are stored in the raw data files. This information is stored in the .MORE.ACSIS.TSYS and .MORE.ACSIS.TRX extensions respectively, and can be viewed using HDSTRACE (see Section 2.4).

3.2.1 Nāmakanui frontend

The Nāmakanui instrument is a spare receiver for the Greenland Telescope (GLT) on loan for use on the JCMT courtesy of ASIAA. Nāmakanui contains three inserts: ʻAlaʻihi, ʻŪʻū, and ʻĀweoweo operating around 86, 230, and 345 GHz. Each of the three inserts receive dual linear polarizations which are labelled using a zero-based system: P0, P1. Each insert has a separate optical path and so only one receiver/frequency range can be used at any one time.

3.2.2 ʻŪʻū frontend (in commissioning)

ʻŪʻū is one of three inserts held within the Nāmakanui instrument. ʻŪʻū is a dual linear polarizations Sideband Separating (2SB) instrument. It operates between 215 and 270.6 GHz. It achieved first light on 2019 October 5. The most-common observing mode for ʻŪʻū is stare mode, but ʻŪʻū is also able to take jiggle and raster maps.

Receiver	FWHM	η_{MB}	η_{fss}
ʻŪʻū	$20''$	0.57	*

3.2.3 ʻĀweoweo frontend (in commissioning)

ʻĀweoweo is another of three inserts held within the Nāmakanui instrument. ʻĀweoweo, like ʻŪʻū, is also a dual linear polarizations Sideband Separating (2SB) instrument. It operates across a frequency range that spans between 276 and 371 GHz. ʻĀweoweo saw first light on 2021 January 14. Due to its sensitivity ʻĀweoweo's most-common observing mode is stare. For larger areas, HARP becomes more efficient for jiggle and raster mapping.

3.2.4 ʻAlaʻihi frontend (in commissioning)

ʻAlaʻihi is the other insert held within the Nāmakanui instrument. Unlike ʻŪʻū and ʻĀweoweo, ʻAlaʻihi is a Single SideBand (SSB) instrument. It will operate at frequencies around 77 to 88 GHz with dual linear polarizations. Its primary function at the JCMT will be as part of VLBI observations.

3.2.5 HARP frontend

The Heterodyne Receiver Array Programme (HARP) is single-sideband receiver (SSB) comprised of a 16-receptor array arranged on a 4×4 grid [3]. It operates in the B-band between 325 and 375 GHz. The receptors are adjacent on the focal plane but separated on the sky by 30 arcseconds (or approximately twice the beamwidth). The footprint of the full array is 2 arcminutes. Note that not all receptors are operational; historically between 12 and 15 receptors have been available at any given time. The tracking receptor is chosen to be the best performing of the four internal (2×2) receptors in the grid.

You can find the number of the tracking receptor and the number of working receptors in the FITS header values `REFRECEP` and `N_MIX` respectively.

In addition to stare observations with the tracking receptor, the most common observing modes for HARP are jiggles and raster maps. These observing modes are described in Section 3.3 and an example image from each mode is shown in Figure 3.2.

Receiver	FWHM	$\eta_{\rm MB}$	$\eta_{\rm fss}$
HARP	$14''$	0.63	0.88

3.2.6 RxA3 frontend (retired June 2018)

Receiver A3 (RxA3) was a single-element double-sideband (DSB) receiver. It operates in the A-band between 211 and 276 GHz. In addition to stare observations, the most common observing mode for RxA3 was raster mapping. In 2016 RxA3 was updated to RxA3M. The instrument was retired in 2018.

Receiver	FWHM	$\eta_{\rm MB}$	$\eta_{\rm fss}$
RxA3	$19\rlap{.}''7$	0.57	0.80

RxA3M had an additional calibration step (automatically applied by our pipelines if you use a version after June 2019) to normalise the flux levels of observations between the upper and lower sideband. If you need more control of this in the pipeline, please see the calibration recipe parameters in Appendix G. The older RxA3 also had some variation in flux levels between the sidebands, but we do not have a calibration correction for this.

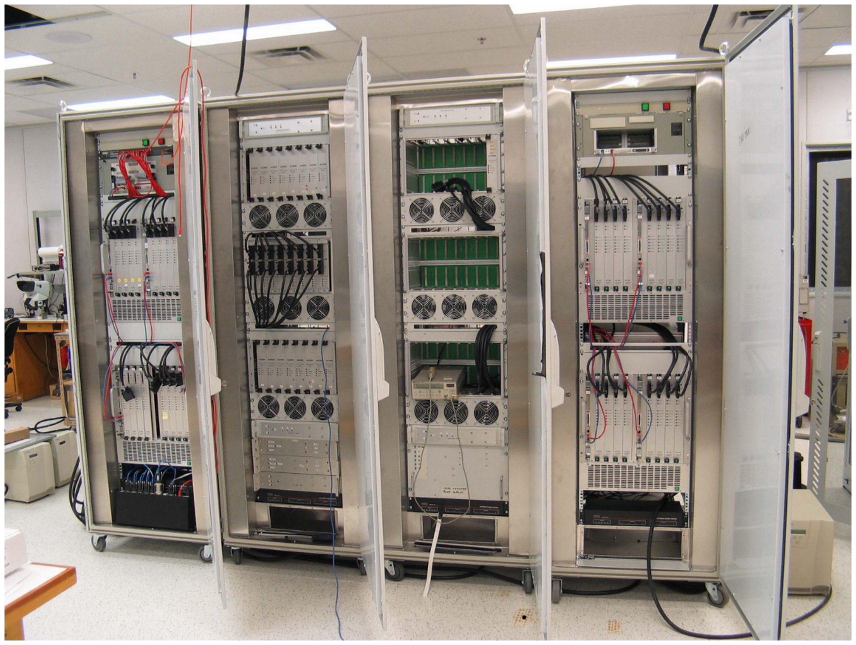


Figure 3.1: Front view of ACSIS at the JCMT. Although designed to work with HARP, ACSIS can be used with any JCMT heterodyne receiver.

3.2.7 ACSIS backend

Since 2006 September, the backend for heterodyne instruments has been the Auto-Correlation Spectrometer and Imaging System known simply as ACSIS (see Figure 3.1). Given the various combinations of ACSIS's 32 down-converter modules and available IF outputs from the different instruments, a number of bandwidth modes and corresponding frequency resolutions are available. Listed below are the most common ones. Visit the JCMT ACSIS webpage for a full list.

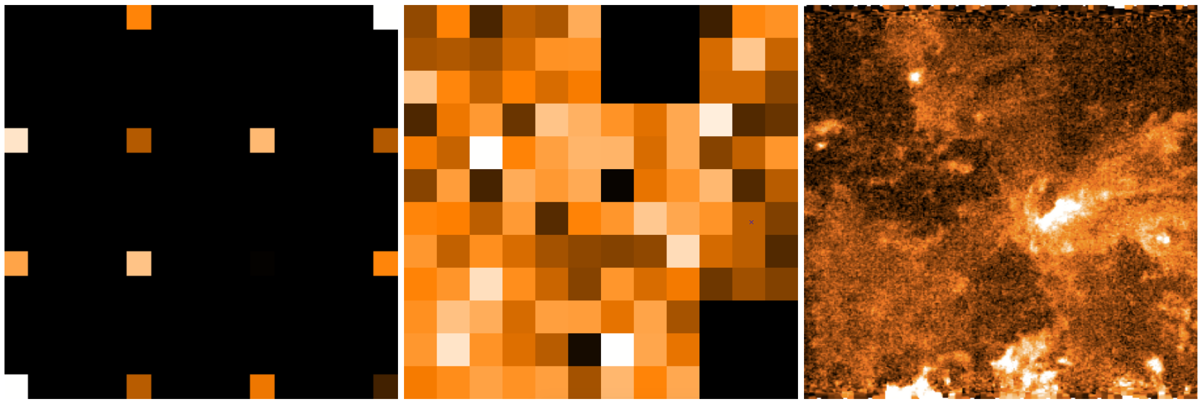


Figure 3.2: (Left) Stare mode with HARP. (Right) Jiggle mapping with HARP. (Bottom) Raster mapping with HARP.

Subbands	**Bandwidth mode**	**Channel Spacing**
1	250 MHz	0.0305 MHz (HARP/RxA/RxW)
1	1000 MHz	0.488 MHz (HARP/RxA/RxW)
1	1860 MHz (1600,1800)	0.977 MHz (HARP)
		0.488 MHz (RxA/RxW)
1	440 MHz (400,420)	0.061 MHz (HARP)
		0.0305 MHz (RxA/RxW)
2	any 250 MHz	0.061 MHz (HARP)
		0.0305 MHz (RxA/RxW)
2	any 1000 MHz	0.977 MHz (HARP)
		0.488 MHz (RxA/RxW)

There are a number of special configurations available with ACSIS. These make use of multiple sub-bands to cover lines at different frequencies. You can find the subsystem number for a given observation by the FITS header `SUBSYSNR`.

You will find each subsystem number corresponds to a different molecule or transition (given by the FITS headers `MOLECULE` and `TRANSITI`). For example, $C^{18}O$ (3-2)/^{13}CO (3-2) and DCN (5-4)/HCN (4-3) for HARP, ^{12}CO/^{13}CO/$C^{18}O$ for ʻŪʻū, and with the now retired RxA the common pairing was SiO (6_0-5_0)/SO (4_3-3_4).

3.3 Observing modes

3.3.1 On-source

Stare A stare, or sample, observation is as simple as it sounds. When you open reduced cubes of stare observations in Gaia you will see a map consisting of one pixel for each receptor; one in the case of ʻŪʻū, ʻĀweoweo and RxA, and up to 16 with HARP. The left-hand panel of Figure 3.2 shows a stare observation (taken by HARP) opened in GAIA. You can also view the central spectrum using linplot (see Appendix F) or SPLAT.

Jiggle A jiggle map is a common HARP observing mode. They are designed to fill in the 30″spacing between the HARP receptors resulting in a $2' \times 2'$ map. This is achieved by 'jiggling' the secondary mirror in a 4×4 or 5×5 pattern to give a HARP4 or HARP5 map respectively. The HARP4 pattern has a 7.″5 pixels while the HARP5 has 6″pixels, slightly under- and over-sampling compared with Nyquist. Other jiggle patterns are also available to be selected with the OT. The central panel of Figure 3.2 shows a HARP 3×3 jiggle pattern.

Each of these patterns can be pixel-centered, where the target co-ordinates fall on one of the central four pixels, or map-centered where the target co-ordinates lie at the centre of the map between the four central pixels. See Figure 3.3 for an illustration of the different jiggle patterns. Although designed for HARP this mode can be used with the single-element receivers.

Raster Raster mapping, or scan mapping, is designed for mapping large areas with maximum efficiency. The telescope continuously takes data while scanning across the source. When HARP is used for raster mapping the array is rotated 14.°04 to the direction of the scan in order to give a fully sampled map with 7.″3 pixels. This will result in jagged edges to your reduced map which you may wish to trim. It is common when using HARP to repeat the map scanning along the other axis to give a 'basket-weave' pair of maps. Each one of the pair will appear as an individual observation. The right-hand panel of Figure 3.2 shows a HARP raster map.

3.3.2 Reference

Position-switch In position-switch (PSSW) mode, the whole telescope moves off source and on to the reference position. This allows a large offset to the reference position, useful in crowded environments such as the Galactic Plane. The main disadvantage is that it takes a longer time resulting in less-accurate sky subtraction or non-flat baselines.

Beam-switch In beam-switch (BSW) mode, the secondary mirror chops onto the reference position. This results in fast and accurate sky subtraction but it is limited to targets which have a blank reference position within 180″(the maximum chop distance of the SMU).

Frequency-switch In frequency-switch (FSW) mode the telescope does not move but the centre of the spectral band is shifted, typically by 8 or 16 MHz, to a line-free region of the spectrum. It is very efficient with no off-source time and can be useful when no emission-free reference position can be found.

Tip

The location of of the reference point can be derived from the FITS headers `SKYREFX` and `SKYREFY` with respect to the base position given by `BASEC1` and `BASEC2`, respectively. Details may be found in SUN/259.

3.3.3 Hybrid

Subsystems are normally arranged to cover different spectral lines and would be processed independently. However, there is also a mode in which the subsystems overlap in frequency, called **hybrid**. Such data are combined to yield a broader spectral coverage, say where there multiple lines in proximity. The number of subsystems is up to a maximum of four.

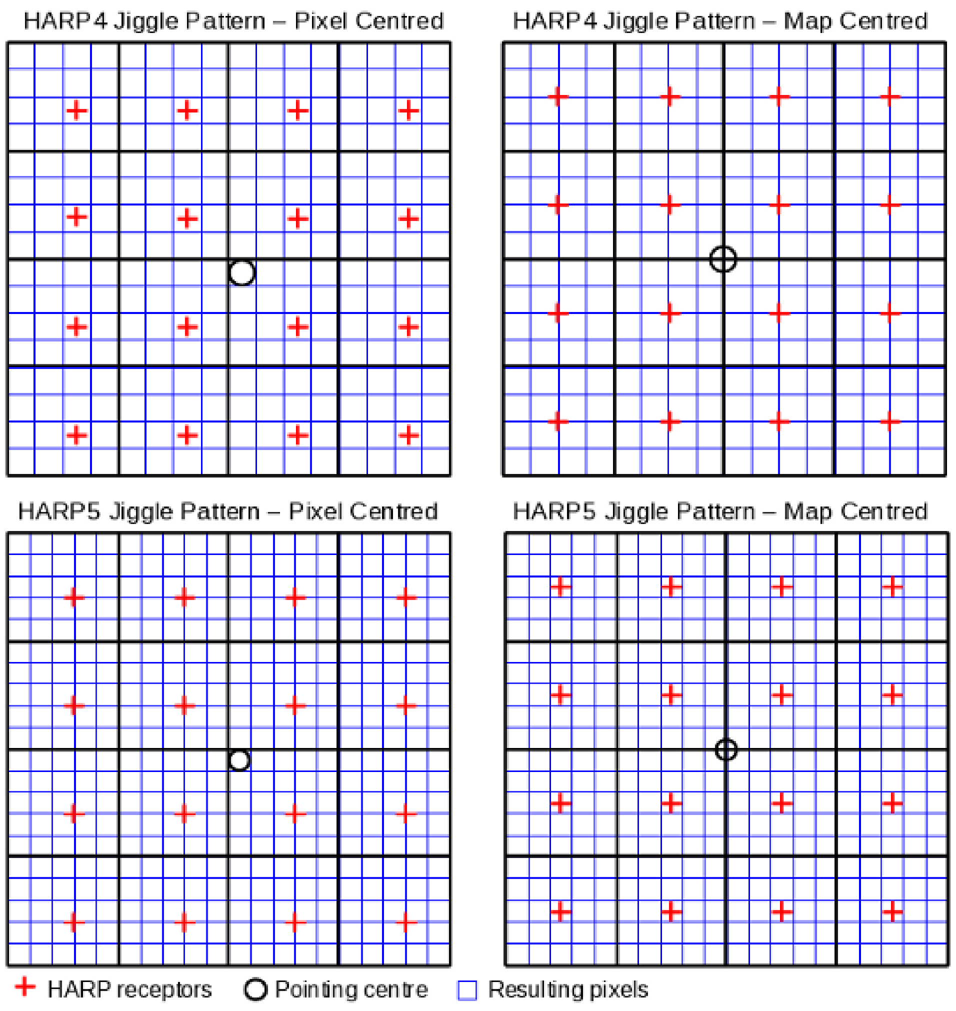

HARP4 Jiggle Pattern – Pixel Centred
HARP4 Jiggle Pattern – Map Centred
HARP5 Jiggle Pattern – Pixel Centred
HARP5 Jiggle Pattern – Map Centred
HARP receptors
Pointing centre
Resulting pixels

Figure 3.3: Top row: HARP4 jiggle patterns produce a map of 16×16 pixels each of 7.″5. Bottom row: HARP5 jiggle patterns produce a map of 20×20 pixels each of 6″.

Chapter 4
Raw ACSIS Data

4.1 Data format

Raw ACSIS data follow this naming convention:

aUTDATE_OBSNUMBER_SUBSYSTEM_SUBSCAN

For example, `a20131115_00055_03_0001.sdf`. The first 'a' represents ACSIS. The UT date and observation number (OBSNUM) serve as an identifier for any given night of observing. ACSIS observations may include up to eight frequency settings given by the different subsystems (01–08, or 00 in very old data), while long observations may require multiple subscans to avoid an excessive file size (currently 384 GB, previously 512 GB).

Within each data file each receptor is named. The naming system is zero-based. For HARP the receptor are labelled: H00, H01, H03... H015. For the Nāmakanui inserts the names follow:

Instrument_Insert_Polarisation_Sideband

For example ʻŪʻū receptors are: NU0L, NU0U, NU1L, NU1U, with U denoting the insert. For ʻĀweoweo, W denotes the insert: e.g. NW0L, NW0U, NW1L, NW1U; and for ʻAlaʻihi, A denotes the insert.

4.2 Visualising raw data

Use the Starlink application GAIA to visualise your raw data. The is initiated by:

```
    % gaia rawfile
```

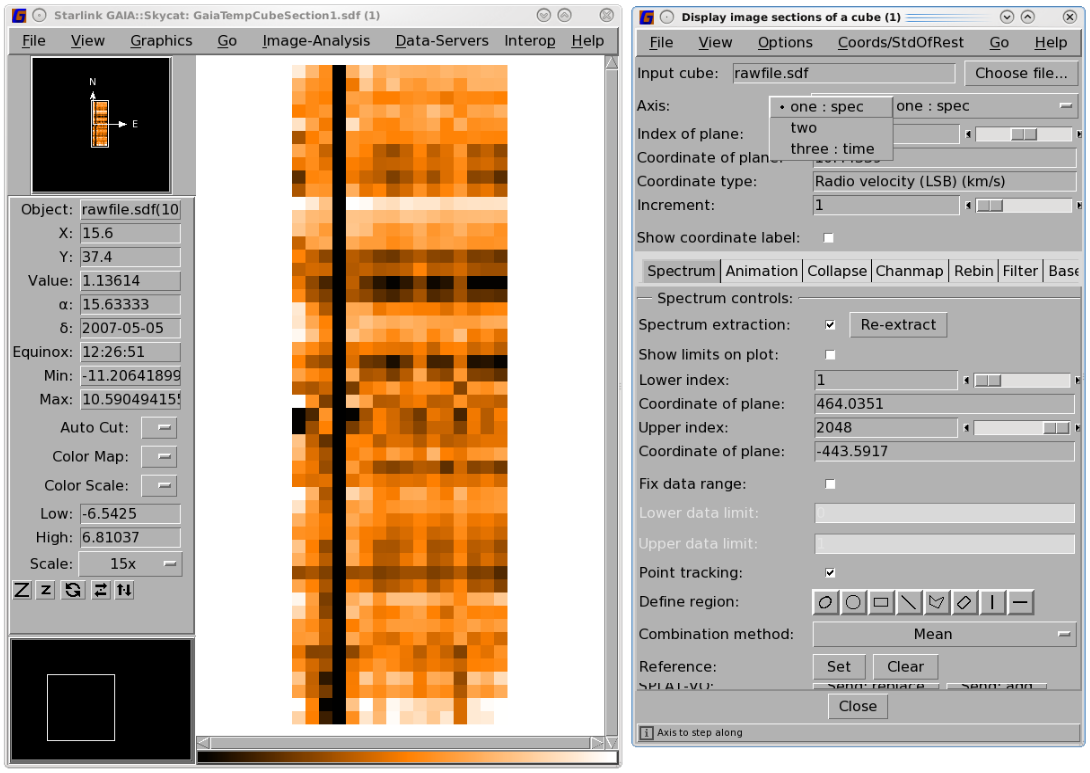
Loading a file in GAIA produces two to three windows (see Figure 4.1 and Figure 4.2). The main window shows a map of the HARP receptors (along the x-axis; note the 16 pixels) for a given sample of the observation (time on the y-axis). You can track the performance of an individual receptor by following a column from bottom to top to check its consistency. In Figure 4.1 H03 is dead for the whole observation. The spectrum for a receptor at one of the time slices can be seen by clicking on the pixel. This will bring up another window—the "`Spectral plot`" (see Figure 4.3). This spectrum will be replaced when you click on a different pixel.

You can change the way the data is displayed in the "`Display image sections of a cube`" window by changing the **Axis**. Selecting Axis number two will display spectra against time while Axis number three gives spectra against receptors. You can scroll through your data by moving the **Index of plane** slider.

A second way to scroll through your spectrum is to click and drag the vertical red bar on the "`Spectral plot`" window. As you do so array shown in the main window will automatically update.

You will likely want to change the auto cut of the colour scale, the colour scheme and the zoom factor—all of which are controlled by buttons on the sidebar in the main window.

See the GAIA manual for full details.

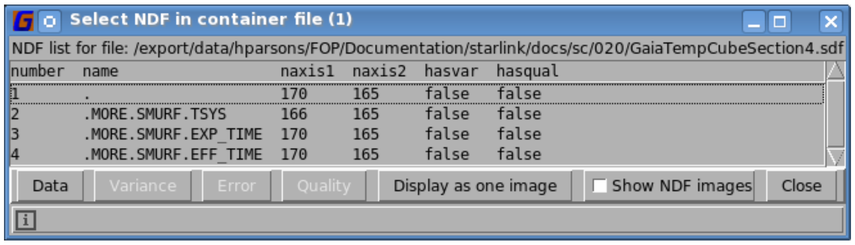

Starlink GAIA::Skycat: GaiaTempCubeSection1.sdf (1)
File View Graphics Go Image-Analysis Data-Servers Interop Help
N
E
Auto Cut:
Color Map:
Color Scale:
Display image sections of a cube (1)
File View Options Coords/StdOfRest Go Help
Choose file...
Spectrum Animation Collapse Chanmap Rebin Filter Base
Spectrum controls:
Re-extract
Show limits on plot:
Fix data range:
Lower data limit:
Upper data limit:
Point tracking:
Define region:
Set Clear
Close

Object: rawfile.sdf(10
X: 15.6
Y: 37.4
Value: 1.13614
α: 15.63333
δ: 2007-05-05
Equinox: 12:26:51
Min: -11.20641899
Max: 10.59049415
Low: -6.5425
High: 6.81037
Scale: 15x

Input cube: rawfile.sdf
Axis: one : spec
- one : spec
- two
- three : time

Index of plane:
Coordinate of plane:
Coordinate type: Radio velocity (LSB) (km/s)
Increment: 1
Show coordinate label:
Spectrum extraction: ✓
Lower index: 1
Coordinate of plane: 464.0351
Upper index: 2048
Coordinate of plane: -443.5917
Point tracking: ✓
Combination method: Mean
Reference:
SPLAT-VO:
Axis to step along

Figure 4.1: Initial GAIA windows displayed upon loading a raw data file. The main window on the left shows a map of receptors at given time-samples during the observation. You may have to zoom in multiple times by clicking the large **Z** button on the side panel. On the right, the "`Display image sections of a cube`" window enables you to navigate the time axis.



NDF list for file: /export/data/hparsons/FOP/Documentation/starlink/docs/sc/020/GaiaTempCubeSection4.sdf

```
number  name                 naxis1  naxis2  hasvar  hasqual
1       .                    170     165     false   false
2       .MORE.SMURF.TSYS     166     165     false   false
3       .MORE.SMURF.EXP_TIME 170     165     false   false
4       .MORE.SMURF.EFF_TIME 170     165     false   false
```

Figure 4.2: Select the NDF from the container file. This window is displayed upon loading a new file in GAIA. In this example, you can choose from NDFs holding the data, the system temperatures, the effective times, and the exposure times.

4.3 Identifying and masking bad data

You can mask bad receptors in the time-series data with the KAPPA command chpix. In the HARP example below all data for Receptors 7 and 8 (on the second axis) are turned off by setting them to BAD. Note the commas in the `SECTION` parameter which specify which axis is being referred to; here the first (spectra) and third (time) axes are unchanged so no range is defined. You will have to repeat the command for non-contiguous receptor numbers.

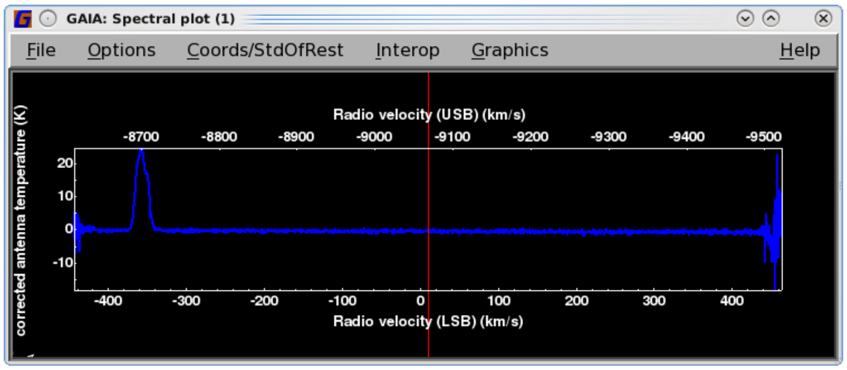


Figure 4.3: The "`Spectral plot`" window displaying its time-varying signal, appears automatically once a pixel is clicked in the main window. The vertical red line indicates the time slice that is currently selected in the "`Display image sections of a cube`" window. This can be dragged across the spectrum to scroll through the spectra. Examine exposure time map etc. in ancillary arrays.

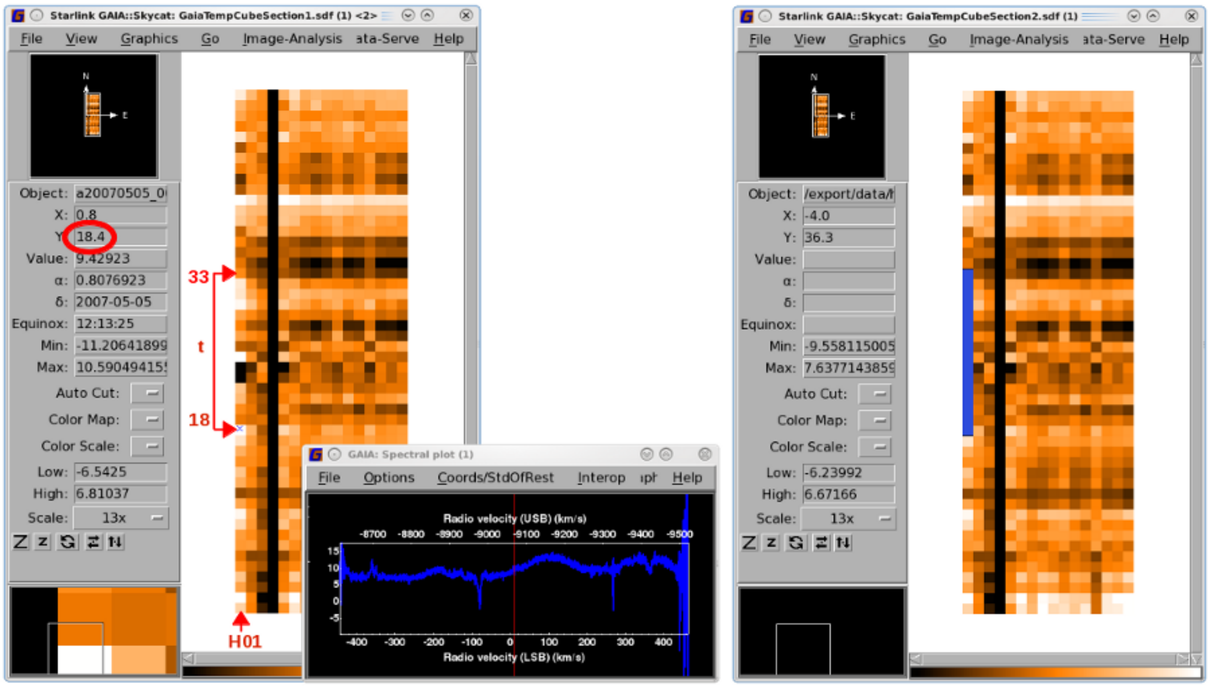
```
% chpix in=raw out=raw_masked section="',7:8,'" newval=bad
```

You can also mask a subset of the time stream for a particular receptor. The example below masks out time Steps 18 to 33 for HARP receptor H01. See Figure 4.4 for instructions on how to identify the affected time range.

```
% chpix in=raw out=raw_masked section="',1:1,18:33'" newval=bad
```

Note that the numbering for the HARP receptors here is from 1 to the number of enabled receptors; this is in contrast to numbering from 0 that you may encounter with the pipeline. (You may examine the .MORE.ACSIS.RECEPTORS component of the file with HDSTRACE to see which receptors were enabled.)

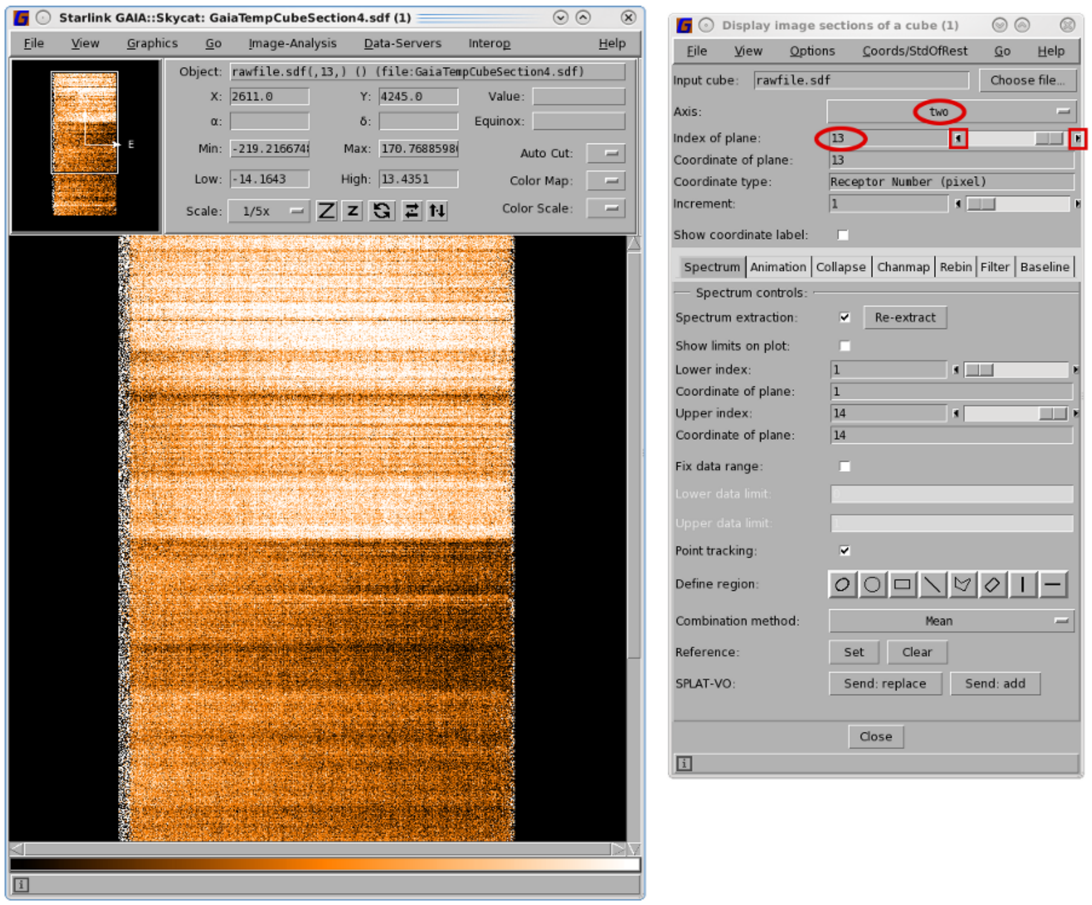
For jiggle maps, where the receptors do not cover multiple sky positions, bad receptors can be identified in the reduced cubes. For raster maps, however, bad receptors are most easily identified in the time-series data. Once you have opened your raw cube in GAIA it is useful to select the second axis (receptor) which gives spectral dispersion along the *x*-axis and time along the *y*-axis. You can use the cube control panel to step through each receptor looking for bad spectra (see Figures 4.4 and 4.5).


Starlink GAIA::Skycat: GaiaTempCubeSection1.sdf (1) <2>
File View Graphics Go Image-Analysis ata-Serve Help
N
E
33
t
18
H01
GAIA: Spectral plot (1)
File Options Coords/StdOfRest Interop pl Help
Radio velocity (USB) (km/s)
-8700 -8800 -8900 -9000 -9100 -9200 -9300 -9400 -9500
15 10 5 0 -5
-400 -300 -200 -100 0 100 200 300 400
Radio velocity (LSB) (km/s)
Starlink GAIA::Skycat: GaiaTempCubeSection2.sdf (1)
File View Graphics Go Image-Analysis ata-Serve Help

Object: a20070505_0
X: 0.8
Y: 18.4
Value: 9.42923
α: 0.8076923
δ: 2007-05-05
Equinox: 12:13:25
Min: -11.20641899
Max: 10.59049415
Auto Cut:
Color Map:
Color Scale:
Low: -6.5425
High: 6.81037
Scale: 13x

Object: /export/data/
X: -4.0
Y: 36.3
Value:
α:
δ:
Equinox:
Min: -9.558115005
Max: 7.6377143859
Auto Cut:
Color Map:
Color Scale:
Low: -6.23992
High: 6.67166
Scale: 13x

Figure 4.4: The left-hand GAIA window shows raw HARP data before masking. Clicking on a pixel will show the spectra. The spectrum shown is from receptor H01 at Step 18 (see circled Y value). To mask this receptor in time from Step 18 to 33—at which point the bad baselines disappear—use chpix. The right-hand GAIA window shows the raw data after masking where the time range in question is now blank.



Object: rawfile.sdf(,13,) () (file:GaiaTempCubeSection4.sdf)
X: 2611.0 Y: 4245.0 Value:
α: δ: Equinox:
Min: -219.216674 Max: 170.7688598 Auto Cut:
Low: -14.1643 High: 13.4351 Color Map:
Scale: 1/5x Color Scale:

Input cube: rawfile.sdf
Axis: two
Index of plane: 13
Coordinate of plane: 13
Coordinate type: Receptor Number (pixel)
Increment: 1
Show coordinate label: ☐

Spectrum controls:
Spectrum extraction: ✓
Show limits on plot: ☐
Lower index: 1
Coordinate of plane: 1
Upper index: 14
Coordinate of plane: 14
Fix data range: ☐
Lower data limit:
Upper data limit:
Point tracking: ✓
Define region:
Combination method: Mean
Reference:
SPLAT-VO:

Figure 4.5: In this example, the second axis (receptor number) has been selected. Use the arrows outlined in red on the **Index of plane** bar to scroll through the receptors. Here you can see Receptor 13 (see circled **Index of plane** value) which goes noisy two-thirds the way through the observation.

Chapter 5
The ACSIS Pipeline

The ORAC-DR pipeline [4] is a generic automated data reduction pipeline that can process your raw JCMT data and return advanced data products: baselined single observation cubes, mosaicked and co-added cubes, moments map and clump catalogues. [15]

It has advanced algorithms for the common routines such as baseline subtraction. The data processing is performed using standard KAPPA, SMURF, and CUPID applications, the main ones of which are described in later chapters.

5.1 Recipes and primitives

Reduction recipes in ORAC-DR consist of a series of stand-alone processes, known as primitives. These primitives are linked together to form data reduction recipes. Each primitive can be fed different input data depending on the nature of the recipe in question and may sometimes be omitted altogether.

There are four main science recipes available, each tailored to different type of observation:

- REDUCE_SCIENCE_NARROWLINE
- REDUCE_SCIENCE_BROADLINE
- REDUCE_SCIENCE_GRADIENT
- REDUCE_SCIENCE_LINEFOREST

A summary of these recipes is given below.

There are also variants of the recipes with the following suffixes.

_POL	polarimetry
_QL	Quick Look which merely runs makecube to turn the raw time-series spectra into a spectral cube for display during data acquisition.
_SUMMIT	A limited reduction to keep pace with data acquisition at the JCMT. It excludes any bad spectra rejection, quality assurance, or iterative baseline determination.

RECIPE	**DESCRIPTION OF EMISSION**	**BASELINE METHOD**

NARROWLINE	One or more narrow lines are expected across the band. Select this recipe if the expected lines are less than about 8 km s^{-1} wide.	Smoothing:
		spatial = 5×5 pixels
		frequency = 10 channels

BROADLINE	This recipe is designed for wide lines that extend over a large fraction of the band. The line is typically too weak to see in a single observation so a pre-determined baseline window and linear baselines are used.	Uses the outer 10% of each end of the spectra to fit a single-order polynomial.

GRADIENT	Typically one moderately wide line is expected, for which the center velocity varies significantly across the field. The baseline window changes across the field. Nearby galaxies often fall in this category. The expected lines should be wider than about 8 km s^{-1} and probably not wider than 20% of the available bandwidth	Smoothing:
		spatial = 3×3 pixels
		frequency = 25 channels

LINEFOREST	A forest of lines is expected across the band. Bright, nearby star-formation sources may fall in this category. This recipe also creates a separate moments map for each line (as defined by the parameter `PER_LINE`).	Smoothing:
		spatial = none
		frequency = 10 channels

5.2 The workflow

You can follow this commentary via the flow chart in Figure 5.1.

Two notations are used in the following list:
• **a_cube** to mean the regridded cube of a single observation.
• **g_cube** to mean the group cube which is a co-add of all the **a_cube** files.

Note that the pipeline will co-add data into a group file whenever it encounters observations with identical LO frequencies, base positions and bandwidths. You can also force a set of observations, such as ones of the same object taken on different nights, to be regarded as a single group to be combined via the `oracdr -onegroup` command-line option.

(**1**) The raw data are copied to the local directory. Typically, subsystems are treated as individual and separate observations by ORAC-DR except for hybrid-mode observations.

(**2**) Because data acquisition is asynchronous, time slices are not necessarily written in sequential order. The next step sort the time-series data into time order. This makes it easier to search for intermittent bad data.

(**3**) The noisy ends of the spectral band dwarf most astronomical signal, and hence are removed as follows. The spectra are collapsed along the receptor using the 'sigma' estimator to form a single spectrum. A constant value background is then fit to the resulting spectrum. The fitting regions are used to determine where the spectrum gets noisier (i.e. higher RMS values in the RMS spectrum). These high-noise regions are then trimmed from the ends in the frequency axis. There is also an alternative to trim specified percentages through the `TRIM_` recipe parameters (see Table G.2).

(**4**) The DC-level offset between corresponding sub-band observations is determined using the median of all the spectra. The DC offset is then subtracted from the sub-band spectra, and the resulting sub-band spectra are mosaicked together to form single time-series cube.

(**5**) Regions or individual spectra containing high- and low-frequency interference from local sources are identified and flagged. Bad detectors are identified by comparing the deviation from linearity of each detector's baseline.

(**6**) Strong spikes ($\pm$150) are replaced by bad values.

(**7**) Quality assurance checks are run on the raw cubes. See Appendix H for a description of the checks performed. Any time-slices failing any of these checks gets flagged as bad and are not included in the group co-add that follows.

(**8**) Once all the raw observations have undergone the initial processing, the individual time-series files are combined to form a **g_cube**.

(**9**) The **g_cube** is smoothed by an amount specified by the particular science recipe being used. See Section 5.1 for details on the different recipes.

(**10**) mfittrend is run to find the baseline regions on the smoothed **g_cube**. The ranges are determined automatically by setting `auto=true`. A baseline mask is written out.

(**11**) The baseline mask is then applied to the unsmoothed **g_cube** and mfittrend is re-run. This time however, the emission regions have been masked out, so all remaining data are included in the fit by setting `auto=false`. The resulting baseline is subtracted.

(**12**) Moments maps and noise maps are made from the baseline-subtracted **g_cube**.

(**13**) If another iteration is required, continue to Step (**14**). If not, the processing is complete.

(**14**) The baseline mask is converted back to the time series with unmakecube. There it is applied to the time-series data for each observation.

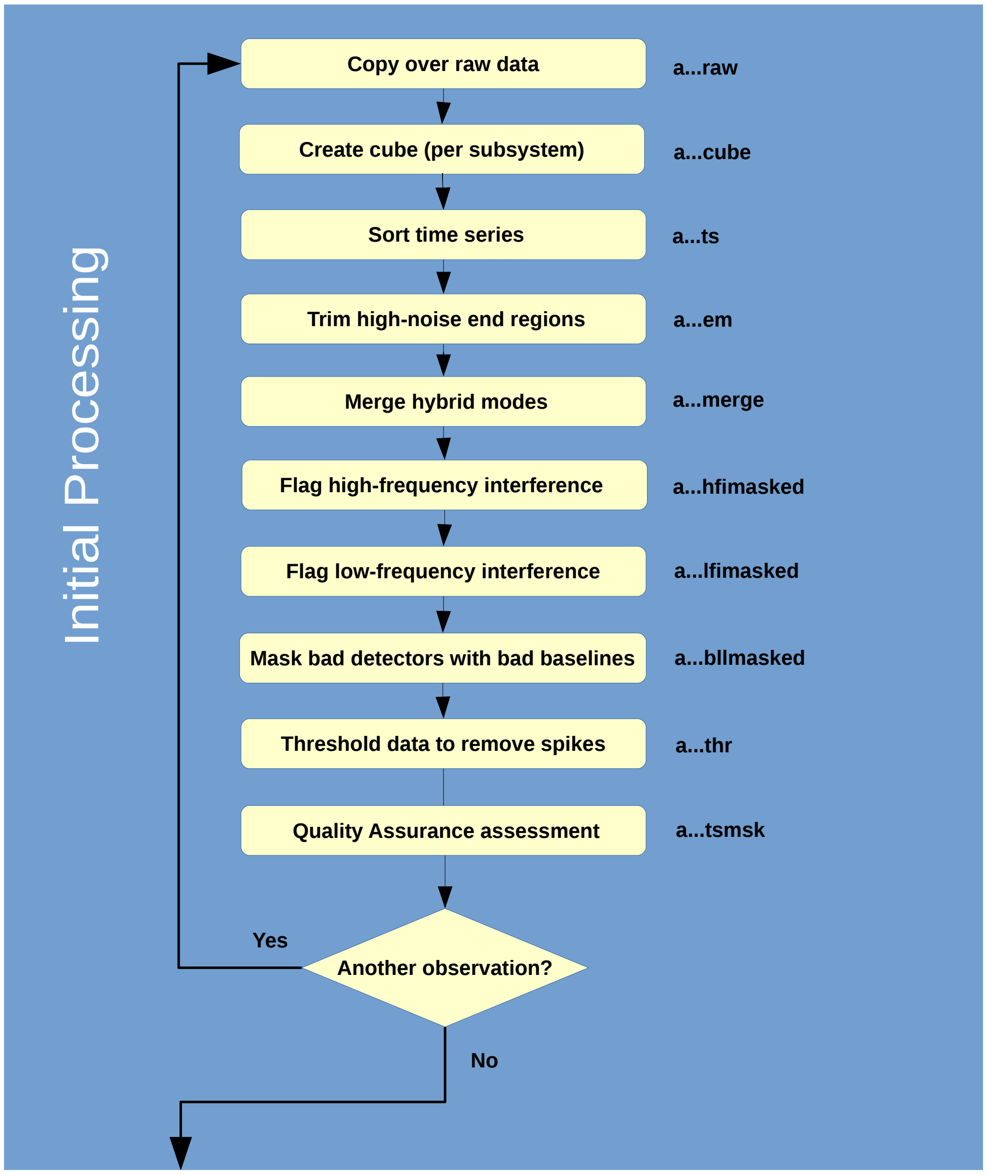
(**15**) A baseline is fit to the masked time-series data using mfittrend and the result is subtracted from the unmasked time-series data. This can be a higher-order polynomial (up to 15th order), although a linear fit is often used.

(**16**) A flat field, i.e. the relative responsivities of the detectors, is optionally created and applied. It involes making cubes for each each receptor combining all observations on a given date to improve the accuracy. There is a choice of analysis methods, but an iterative approach of thresholding—to exclude the noisy baseline that would dilute the comparisons—worked best on most suitable observations. Those with low signal-to-noise or signal only originating in a small percentage of the area mapped are not amenable to determining the receptor-to-receptor responses. (See Table G.5).

(**17**) An **a_cube** is made from the baseline subtracted data for each observation.

(**18**) The individual baseline-subtracted time series are combined to make a new **g_cube**.

(**19**) Return to Step (**9**).


Initial Processing
Copy over raw data
a...raw
Create cube (per subsystem)
a...cube
Sort time series
a...ts
Trim high-noise end regions
a...em
Merge hybrid modes
a...merge
Flag high-frequency interference
a...hfimasked
Flag low-frequency interference
a...lfimasked
Mask bad detectors with bad baselines
a...bllmasked
Threshold data to remove spikes
a...thr
Quality Assurance assessment
a...tsmsk
Yes
Another observation?
No

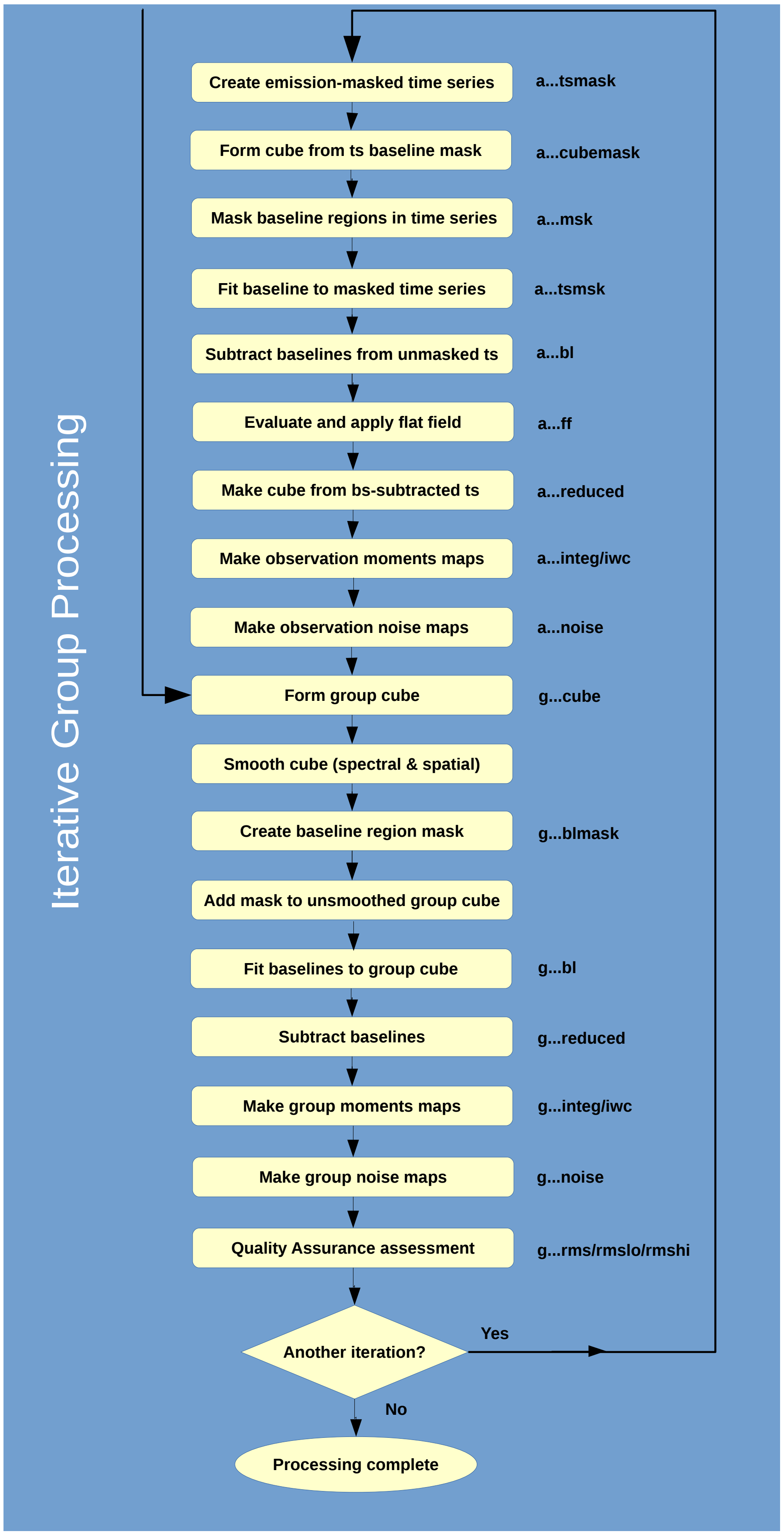

Iterative Group Processing
Create emission-masked time series
a...tsmask
Form cube from ts baseline mask
a...cubemask
Mask baseline regions in time series
a...msk
Fit baseline to masked time series
a...tsmsk
Subtract baselines from unmasked ts
a...bl
Evaluate and apply flat field
a...ff
Make cube from bs-subtracted ts
a...reduced
Make observation moments maps
a...integ/iwc
Make observation noise maps
a...noise
Form group cube
g...cube
Smooth cube (spectral & spatial)
Create baseline region mask
g...blmask
Add mask to unsmoothed group cube
Fit baselines to group cube
g...bl
Subtract baselines
g...reduced
Make group moments maps
g...integ/iwc
Make group noise maps
g...noise
Quality Assurance assessment
g...rms/rmslo/rmshi
Another iteration?
Yes
No
Processing complete

Figure 5.1: Flow chart of the processes employed by the ORAC-DR ACSIS pipeline, specifically the REDUCE_SCIENCE_NARROWLINE and REDUCE_SCIENCE_GRADIENT recipes. A few minor processing steps are omitted to avoid the diagram being too long. Other recipes may have additional steps or omit steps as appropriate. Because of space constraints baseline is sometimes abbreviated to 'bs' and time series to 'ts' in the step descriptions. The prefix and suffix of the file created at each stage is shown to the right of the process box. Note that suffix 'a' refers to an individual observation, while 'g' refers to a group (or co-added) file.

Chapter 6
Running the Pipeline

6.1 Checking and changing the science recipe

You can find out which recipe is set in the data header via the RECIPE keyword in the FITS header of any of your raw files. You can use either of the options below:

```
% fitsval a20130609_00059_01_0001 RECIPE
% fitslist a20130609_00059_01_0001 | grep RECIPE
```

You can override the recipe set in the FITS header by listing any different one on the command line when starting ORAC-DR. For example:

```
% oracdr -files mylist -log x REDUCE_SCIENCE_GRADIENT
```

6.2 Setting recipe parameters (optional)

You can tailor the recipe parameters by supplying a .ini file (called myparams.ini in the following example). This file contains the recipe name (which must match the one assigned to your data, whether from the header or any different one specified on the command line) followed by the options you wish to specify in the following format:

```
[REDUCE_SCIENCE_NARROWLINE]
MOMENTS_LOWER_VELOCITY = -30.0
MOMENTS_UPPER_VELOCITY = 155.0
PIXEL_SCALE = 6.0
SPREAD_METHOD = gauss
SPREAD_WIDTH = 9
SPREAD_FWHM_OR_ZERO = 6
REBIN = 0.635,2.0
```

Notice the two values given for the REBIN option. This means that two maps will be produced, one at each of the specified velocity resolutions. See Appendix G for a full list of recipe parameters.

The .ini file can contain an arbitrary number of [] constructs for different recipes, qualified by object name or ORAC-DR internal headers.

While the use of recipe parameters is optional, once you have some experience with the pipeline you are encouraged to tailor the recipe parameters for your reductions to get the best out of the recipe and your data.

6.2.1 Setting recipe parameters by object name

You can further select the recipe parameters by object name, as given by the OBJECT FITS header converted to uppercase with spaces removed. Here are some examples.

```
[REDUCE_SCIENCE_BROADLINE:NGC253]
REBIN = 10.0,20.0,30.0
PIXEL_SCALE = 15.0
```

This would apply the two recipe parameters to the NGC 253 target when processed by REDUCE_SCIENCE_BROADLINE, while

```
[REDUCE_SCIENCE_BROADLINE:ARP*]
REBIN = 10.0,20.0,30.0
PIXEL_SCALE = 15.0
```

would apply the same parameter values for any beginning with the Arp designation.

6.2.2 Setting recipe parameters by ORAC-DR internal header

The pipeline translates a selection of metadata from the FITS headers into ORAC-DR internal headers. You can choose different sets of recipe parameters from the internal headers. The available headers can be inspected with translatehdr supplied with one of your raw data files.

```
% $STARLINK_DIR/Perl/bin/translatehdr a20151213_00067_01_0001.sdf
```

Here is an example of how to select by internal header.

```
[REDUCE_SCIENCE_NARROWLINE#SAMPLE_MODE=GRID]
LV_IMAGE = 1
LV_AXIS = skylon
```

would apply these parameters only to data whose sample mode was grid.

You can select by an arbitrary number of internal headers to narrow the selection. The internal headers must follow any object name as in the following example.

```
[REDUCE_SCIENCE_NARROWLINE:HH21#SPECIES=C-18-O#BANDWIDTH_MODE=250MHzx4096]
FINAL_LOWER_VELOCITY = -100
FINAL_UPPER_VELOCITY = 100
MOMENTS_LOWER_VELOCITY = 1.2
MOMENTS_UPPER_VELOCITY = 6
FLATFIELD = 0
```

`SPECIES` defines the molecule. The rationale for this might be disable estimation of the flat field because the signal is much weaker for the molecule C^{18}O. Other common values are `CO` for ^{12}CO and `13-CO` for ^{13}CO. There is also a `TRANSITION` header to select by the transition.

The velocity limits for the moments might be set narrower than for other molecules. You might want to set a different velocity range in the final spectral cubes (set by the `FINAL_LOWER_VELOCITY` and `FINAL_UPPER_VELOCITY` parameters) depending on the spectral resolution given by `BANDWIDTH_MODE` (see Figure 3.2.7 for a list).

6.3 Setting quality-assurance parameters (optional)

There is a set of quality-assurance (QA) parameters that are applied by default when you run the pipeline. These are found in the text file `$ORAC_DATA_CAL/qa.ini`.

You can set your own QA parameters by creating a local *.ini* file (called `myqa.ini` in the following examples). You can then call this local QA file via `-calib qaparams=myqa.ini` when starting the pipeline.

The example below illustrates the format of this QA file and highlights some of the main parameters you may consider tweaking.

```
[default]
BADPIX_MAP=0.1
GOODRECEP=8
TSYSBAD=550
FLAGTSYSBAD=0.5
TSYSMAX=550
TSYSVAR=1.0
RMSVAR_RCP=0.5
```

The text in the square brackets describes the data to which the QA parameters should be applied. This is followed by the QA parameters and their values.

The parameters listed under the [default] header will get picked up for *all* observations unless overridden by other header descriptions. Extra header may describe a frequency range (e.g. `[default 200:300]`), a particular molecule and transition (e.g. `[default CO32]`), an instrument (e.g. `[RXA]`), or a legacy survey (e.g. `[GBS]`). You can also include combinations (e.g. `[GBS_13CO32]`). Note that for any data taken by RxA or RxW, `GOODRECEP` must be set to 1.

The example file below sets up a different set of parameters for two transitions of CO. Any QA parameters not explicitly set will revert to those specified in the default `qa.ini` file.

```
[C18O32]
RES_CHAN=10
VELRES=1.0
TSYSBAD=650

[CO21]
RES_CHAN=10
VELRES=0.5
TSYSBAD=550
GOODRECEP=1
```

See Appendix H for a description of all available QA parameters.

6.4 Specifying bad receptors (recommended)

If certain receptors are known to be bad for the full length of an observations (e.g. dead receptors), processing time can be reduced by not having the pipeline attempt to reduce those data. The ability to mask bad receptors also provides an additional tool for those wishing to examine data in more detail (e.g. when looking at data from an instrument that has yet to be fully commissioned).

Information regarding bad receptors is stored in a master text file called index.bad_receptors, located in your ORAC_DATA_CAL directory. This file lists dates and the corresponding receptors which are not operational. By default, when the pipeline runs it searches this file and discards the appropriate receptors. However, this master list is grossly incomplete so you should select another method for specifying bad receptors.

We recommend using the index option, called via -calib bad_receptors=index when starting the pipeline. This creates an index file called index.bad_receptors_qa in your ORAC_DATA_OUT directory, where any bad receptors identified during your reductions are indexed. This file is appended each time you run the pipeline and in this way you build up your own independent list.

If an index file exists, then, by default, the pipeline will look for bad-receptor information in both $ORAC_DATA_CAL/index.bad_receptors and $ORAC_DATA_OUT/index.bad_receptors_qa.

bad_receptors	**Description**
masterorindex	Use both the master index.bad_receptors and pipeline-generated index.bad_receptors_qa files [Default].
master	Use the master index.bad_receptors file in $ORAC_DATA_CAL.
index	Use the index.bad_receptors_qa index file in $ORAC_DATA_OUT as generated by the pipeline.
file	Reads a file called bad_receptors.lis, which should contain a space-separated list of receptor names in the first line. This file must be located in $ORAC_DATA_OUT and invoked by -calib bad_receptors=file.
list	A colon-separated list of receptor names can be supplied, such as -calib bad_receptors=H01:H06, or -calib bad_receptors=NU1L,NU1U. You can append any of the other options to the end of your list, such as -calib bad_receptors=H14:index

Table 6.1: The options available for the -calib bad_receptors flag when running the pipeline.

You can inquire which receptors are present in your data by inspecting a data file's FITS header. This can be done using fitslist or fitsval to look at the RECPTORS keyword.

```
% fitsval a20210520_00060_02_raw0001 RECPTORS
H01 H02 H03 H04 H05 H06 H07 H08 H09 H10 H11 H12 H13 H15
```

These receptors may not necessarily all contain good data. For reduced spectral cubes generated by makecube, the RECPUSED keyword reports only the receptors actually used to form the spectral cube. See Section 7.2.3.

6.5 Starting the pipeline

(**1**) Initialise the pipeline software. This may be as simple as

```
% oracdr_acsis
```

(**2**) Define environment variables to ensure the data are read from, and written to, the right place. Many are set automatically when the pipeline is initialised but others must be set manually. Details of the optional variables are given in **SUN/260** but you should specify where to find the raw data and where to write any files that are created.

```
% setenv ORAC_DATA_IN <path to raw data>
% setenv ORAC_DATA_OUT <where you want reduced data to go>
```

These extra definitions can be avoided if you initialise with the following command.

```
% oracdr_acsis -cwd
```

The `cwd` option requests that the current working directory is used for output, which is most likely what you require.

If you are supplying a text file listing the raw data `$ORAC_DATA_IN` should be the location of the files listed, unless they are given as absolute paths.

If you wish to keep the intermediate files produced by the pipeline you should also set `ORAC_KEEP`.

```
% setenv ORAC_KEEP 1
```

However, this still excludes temporary files made during a recipe step, whose names have the `oractemp` prefix. To retain these set

```
% setenv ORAC_KEEP temp
```

Retention of temporary and intermediate will require plenty of storage, but it can be invaluable during debugging.

(**3**) Now you are ready to run the pipeline. In this example, the recipe name on the end ensures that all data being processed using REDUCE_SCIENCE_NARROWLINE.

```
% oracdr -files list.txt -batch -log xf -calib qaparams=myqa.ini \
  bad_receptors=index -recpars mypar.ini -nodisplay REDUCE_SCIENCE_NARROWLINE
```

Below is a list of commonly used command line options when running the ORAC-DR pipeline from outside the EAO. For a full list of all possible options and their descriptions see **SUN/230**.

6.6 Pipeline output

The pipeline will produce a group file for each object being processed. If the pipeline is given data from multiple nights, all those data will be included in the group co-add using inverse variance weighting.

Files created from individual observations are prefixed with a, while group (co-added) maps are prefixed with ga. Table 6.3 describes the files created. In addition PNG images are made of the reduced files at a variety of resolutions.

Tip

Even if just a single observation is processed, a group file is created so long as the observation passes the QA check.

Flag	Description
Supplying data	
-files <mylist>	Input data provided by a text file. Supply the file name (relative to current directory, or the full path) of an ASCII text file containing a list of all observation files to be reduced, one file per line.
Looping	
-loop file	Loop over all the lines in the file supplied by the `-files` option. (This is the default if a list of files has been given.)
Group processing	
`-batch`	Delays group processing until the individual files have been reduced. (This is the default if not in QL or summit mode, or processing today's data.)
`-onegroup`	Forces all the observations into the same group. Useful if building up a map from rasters with different target co-ordinates or from different dates.
Supplying parameter files	
`-calib`	Under the calibration flag you can specify QA parameters by including `qaparams=<myqa.ini>` and bad receptors via `bad_receptors=index`. If you do not set the `qaparams` option but have a file called `qa.ini` in your output directory, the QA settings will come from this file.
-recpars	Allows you to provide recipe parameters as a `.ini` file.
Logs/Display	
`-nodisplay`	Shorten processing times by turning off all graphical output.
`-log sxfh`	Send the log to a screen (`s`), an xwindow (`x`), file (`f`) or html (`h`). The logfile (`f`) has the name `.oracdr_`*NNNN*`.log` where *NNNN* is the current process ID. Only include the options you want. This defaults to `xf`.

Table 6.2: The options available on the command line when running the ORAC-DR pipeline from outside the EAO.

6.6.1 Why have multiple reduced files been generated?

You will find the pipeline splits large reduced cubes (e.g. from raster maps) into smaller tiles of <512 MB. This is to save on computer memory. It does this from the central position outwards so you may find some of the smaller edge tiles are strange, often narrow, rectangles. These tiles can be pasted together with the KAPPA command paste but beware of the final file size. For example,

```
% paste ga20140601_15_1_reduced\*.sdf tiledmap
```

Note, there is a recipe parameter CHUNKSIZE which adjusts the maximum tile size. If you make it larger than 512 it is possible to generate a single reduced cube.

In addition to these NDFs a number of log files are created.

Finally there are HDS scratch `t*.sdf` files, and temporary files `oractemp*` which can be NDFs or small text files such as file lists. These should be removed automatically at the end of processing, unless something has gone wrong.

Default Files	
`cube001`	Baselined cube
`integ`	Integrated intensity image
`rimg`	Representative image (same as `integ` file), used to form rimg PNG
`sp001`	Spectrum taken from position of peak intensity in the integ file
`rsp`	Representative spectrum (same as `sp001`), used to form rsp PNG
`iwc`	Intensity weighted co-ordinate image
`noise`	Noise map
`reduced00`*n*	Final trimmed, baselined cube of the *n*th tile. There may be a few of these as large maps will be split into tiles of 512 MB.
`rmslo`	Low-frequency noise
`rmshi`	High-frequency noise
Extra files kept with ORAC_KEEP = 1	
`em001`	Ends of the frequency axis have been trimmed off
`thr001`	Time-series data with large spikes removed
`ts001`	Time-sorted time-series data
`tss001`	Time-series data with median DC signal removed
`blmask001`	Baseline mask cube. Regions of emission have been masked out.
`bl001`	Baselined cube
`linteg00`*n*	Integrated intensity image formed around the *n*th line.

Table 6.3: Table of the files written out by the science pipeline. Each of these files is generated for both the individual observations and the group files.

Default Files	
`.oracdr_*.log`	The ORAC-DR log file.
`log.flat`	Flat-field ratios for each receptor and observation.
`log.group`	The files contributing to each group.
`log.noisestats`	Noise statistics for each observation and group.
`log.qa`	Quality assurance reports
`log.removedobs`	Observations removed from each group (e.g. due to failing QA)

Table 6.4: Table of the log files written by the science pipeline.

Chapter 7 Regridding your Data

If you prefer to reduce your data manually, regridding your data is the best place to start. (See Appendix D for an explanation of what regridding is.)

To convert your raw time-series cubes with spectral channel number, receptor index, time-slices axes into a spatial cube with RA, Dec (or Galactic longitude, Galactic latitude) and spectral axes use the SMURF application makecube. In this chapter we will show an example of processing data using makecube, discuss what to look out for and introduce the various parameters with which to tailor your reduction.

7.1 Running Makecube

The SMURF application used for regridding your raw data is makecube.

```
% makecube in=a20070505_00041_01_0001.sdf out=cube autogrid

Processing data from instrument 'HARP' for object 'W49' from the following
observation  :
   20070505 #41 grid/pssw

   Projection parameters used:
      CRPIX1 = 0
      CRPIX2 = 0
      CRVAL1 = 287.555875 ( RA = 19:10:13.410 )
      CRVAL2 = 9.10386111111111 ( Dec = 9:06:13.90 )
      CDELT1 = -0.00166666666666667 ( -6 arcsec )
      CDELT2 = 0.00166666666666667 ( 6 arcsec )
      CROTA2 = 0

   Output cube pixel bounds: ( -8:11, -8:11, -1024:1023 )
   Output cube WCS bounds:
        Right ascension: 19:10:09.156 -> 19:10:17.259
        Declination: 9:05:16.90 -> 9:07:16.90
        Radio velocity (LSB): -443.8134 -> 464.2568 km/s
```

The `autogrid` option allows the software to determine the optimal projection parameters for regridding. This is usually fine for stares and jiggles but is not so good for raster maps. See Section 7.2.5 for more discussion.

The screen report generated by makecube includes the input files, instrument, source, UT date and observation number, and observing mode. The pixel size can be checked in the projection parameters section, while the bounding box of the resulting cube (RA, Dec and velocity) is reported in the output section.

You can visualise your map by opening it with GAIA—see Figure 7.1.

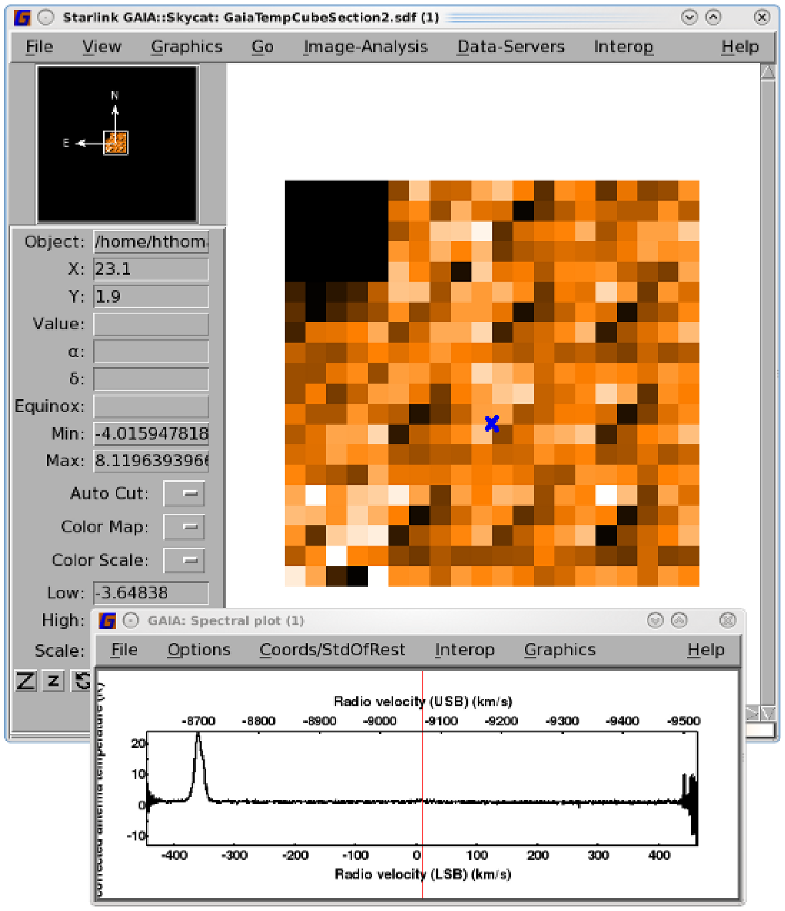


Figure 7.1: The regridded cube displayed with GAIA. The 5×5 jiggle pattern is evident in the map as is the missing receptor in the top-left corner. The smaller window shows the spectrum for the pixel selected by the blue cross on the map.

7.1.1 Reducing multiple observations

You can supply makecube a group in a text file containing a list of all the raw files you want reduced. These filenames should include the full path if they are not in the local directory.

When you pass such group files to makecube (and any Starlink command) you should prefix the filename with an up-caret (ˆ).

```
% makecube in=^filelist.txt
```

Alternatively you can use wild cards. The example below reads in all the raw files for observation 10.

```
% makecube in=rawdata/a20140101_00010\*
```

7.2 Makecube options

7.2.1 Specifying spectral bounds

To save disk space and processing time you can specify the region of the spectra to be processed with SPECBOUNDS. If you know your line lies close to 0 km s^{-1}, you may chose to process only the central 100 km s^{-1} as in the example below.

```
% makecube rawfile specbounds=\"-50.0 50.0\"
```

7.2.2 Masking receptors

When combining several maps with masked bad receptors one can use the BADMASK parameter of makecube. This option determines the way in which bad pixels are propagated from input to output. The one that produces the best map is badmask=and, which uses all the data so an output pixel will be bad only if all the input pixels that contribute to it are bad. This generates an output map with the lowest noise and fewest bad pixels. However, for large datasets the memory requirements may be excessive.

Other options are badmask=or (default) and badmask=first. Or means that an output pixel will be bad if any of the input pixels that contribute to it are bad and is the default. The flag first means that the decisions on which pixels will be flagged as bad will be made with the first input spectrum while the rest are ignored.

7.2.3 Including/excluding receptors

To make a cube with only certain receptors you can use the detectors option on the command line. This allows you to either include or exclude specific receptors without the need for masking. The example below makes a HARP map using all HARP receptors except H04 and H11.

```
% makecube in=rawfile autogrid out=map detectors='"-H04,H11"'
```

This alternative example makes a map using only receptor H10. Maps made with single receptors may be used to evaluate the receptor-to-receptor flat-field.

```
% makecube in=rawfile autogrid out=map detectors='"H10"'
```

When reduced by makecube—either by hand or through the pipeline—it is possible to find out which *good* receptors contributed to the reduced cube by inspecting that file's FITS header. This can be done using fitslist or fitsval to examine the RECPUSED parameter. Here are examples for ʻŪʻū and HARP respectively.

```
% fitsval a20200808_00084_01_reduced001 RECPUSED
NU0L NU1L
% fitsval ga20210520_60_1_reduced001 RECPUSED
H12 H13 H04 H03 H11
```

7.2.4 Defining an output grid

You can use the REF option to specify a reference NDF that defines an output grid. This NDF can have only two spatial dimensions, even though your data may be in three dimensions.

```
% makecube in=rawfile ref=myreffile
```

If you do not supply a reference file makecube will revert to the AUTOGRID option.

If autogrid=true, the output grid is determined automatically so that as many data samples as possible fall close to the centre of pixels in the output cube.

If autogrid=false, REFLON and REFLAT are set to the first pointing BASE position, CROTA2 is set to minus the MAP_PA value in the FITS header (converted to the requested sky co-ordinate system), PIXSIZE is set to 6 arcseconds, and REFPIX1 and REFPIX2 are both set to zero. See **SUN/258** for a full description of each of these parameters.

7.2.5 Data-regridding options

There are several regridding options available to you through the makecube command. These are specified with the SPREAD parameter on the command line. For instance,

```
% makecube in=rawfile spread=sincsinc params="[6,9]"
```

The default method is Nearest, which assigns the input pixel completely to the nearest output pixel. This is the fastest method available but is not optimal. In particular it may result in blank pixels in HARP maps due to missing receptors (see Figure 7.2).

We recommend you select one of the other methods available which spread the emission to neighbouring pixels. These include Gauss and SincSinc which are the most common, although the latter sometimes gives noisy edges with negative values in some pixels. SincCos will give neater edges. For each of these you will need to provide the size of the convolution function through the params option as in the example above. See **SUN/258** for more details.

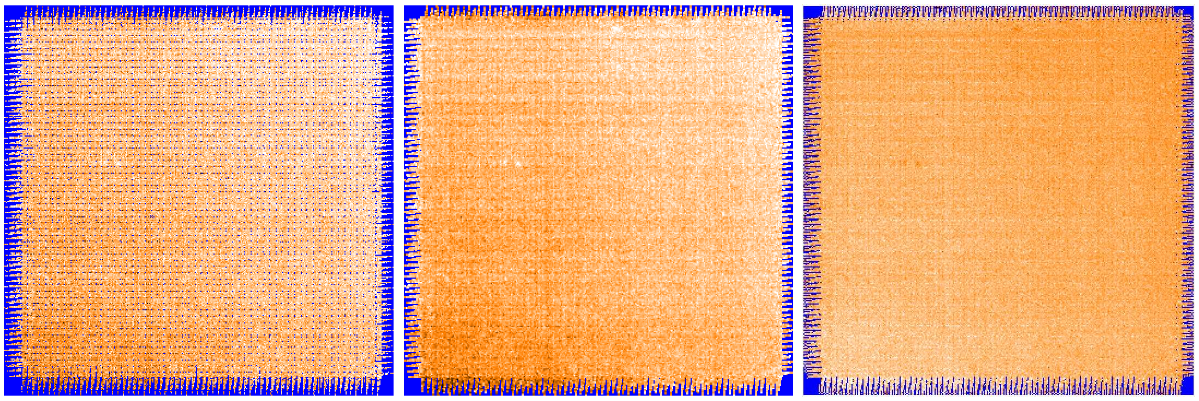


Figure 7.2: Raster map regridded with makecube and spread = nearest (left), Gauss (centre) and SincSinc (right). Blank pixels are coloured blue. You can see the Nearest option has resulted in multiple blank pixels throughout the map.

7.2.6 Generating a variance map

The option GENVAR can be used to generate a variance array for your cube. This is important if you are adding multiple maps with different variances as it allows them to be accurately weighted. There are three options for GENVAR:

Spread The output variance values are based on the spread of input data values contributing to each output pixel. Note that this option is not available if parameter SPARSE is set `true`. If the BADMASK value is `or` or `first`, then a single variance value will be produced for each output spectrum. However, when BADMASK is `and`, an independent variance value is calculated for each channel in each output spectrum.

Tsys The output variance values are based on the system noise temperature values supplied in the input NDFs. Since each input spectrum is characterised by a single Tsys value, each output spectrum will have a constant variance value (i.e. all channels in an output spectrum will have the same variance value). [Default].

None No output variance values are created.

7.2.7 Sparse grid

If one has observed a number of offset positions which are not on a regular grid or are widely spaced, it is of little use to make a normal cube with makecube, which then would have a large size and which would be mostly empty. Then one can use the makecube parameter SPARSE. When `sparse=true` a one-dimensional array is created which stacks spectra in the order in which they were observed (from left to right when viewed with GAIA).

```
% smurfhelp makecube param sparse
```

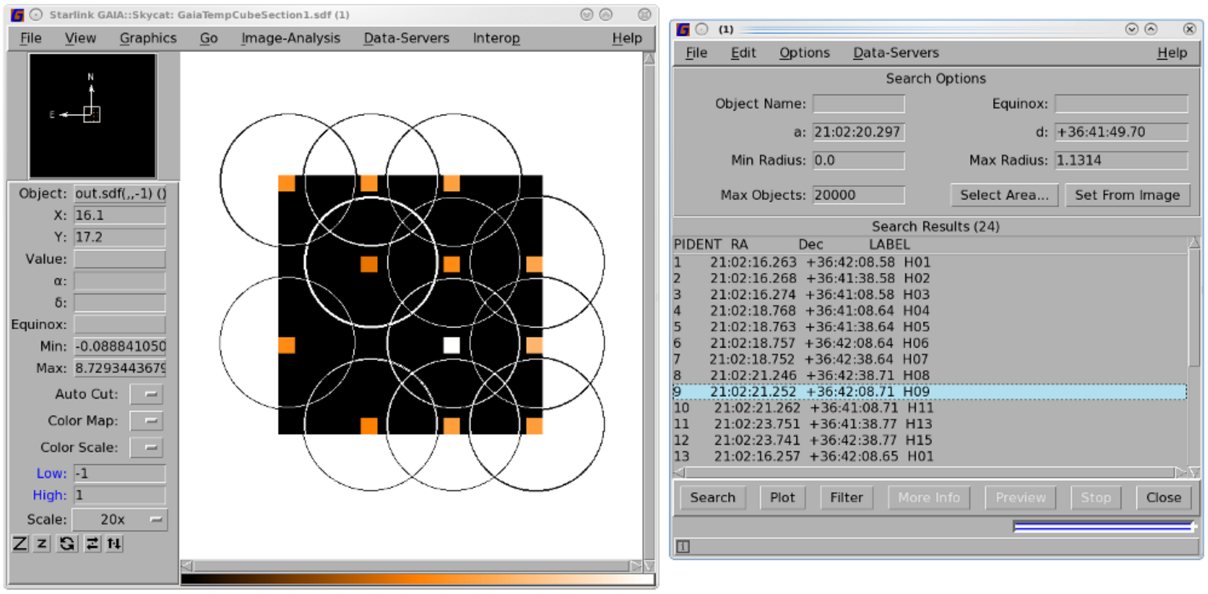
This is generates a one-dimensional array where spectra are displayed by GAIA in the order where they were observed (from left to right).

7.2.8 Creating a catalogue of receptor positions

You can use the option OUTCAT to generate a catalogue of the spatial positions of the receptors used to make your final cube. The label associated with each row in the catalogue is the detector name. The detector positions in the catalogue are ordered as follows: all the positions for the first input NDF come first, followed by those for the second input NDF, etc. Within the group of positions associated with a single input NDF, the positions for the first time slice come first, followed by the positions for the second time slice, and so on.

```
% makecube in=rawfiles outcat=cat
```

The catalogue produced will be in FITS format. In the example above a file called `cat.FIT` is created. When viewing your map with GAIA this catalogue can be overlaid to identify the receptors contributing to each pixel (see Figure 7.3).


Starlink GAIA::Skycat: GaiaTempCubeSection1.sdf (1)
File View Graphics Go Image-Analysis Data-Servers Interop Help
N
E
Auto Cut:
Color Map:
Color Scale:
(1)
File Edit Options Data-Servers Help
Search Options
Select Area...
Set From Image
Search Plot Filter More Info Preview Stop Close

Object: out.sdf(,,-1) ()
X: 16.1
Y: 17.2
Value:
α:
δ:
Equinox:
Min: -0.088841050
Max: 8.7293443679
Low: -1
High: 1
Scale: 20x

Object Name:
Equinox:
a: 21:02:20.297
d: +36:41:49.70
Min Radius: 0.0
Max Radius: 1.1314
Max Objects: 20000

Search Results (24)

PIDENT	RA	Dec	LABEL
1	21:02:16.263	+36:42:08.58	H01
2	21:02:16.268	+36:41:38.58	H02
3	21:02:16.274	+36:41:08.58	H03
4	21:02:18.768	+36:41:08.64	H04
5	21:02:18.763	+36:41:38.64	H05
6	21:02:18.757	+36:42:08.64	H06
7	21:02:18.752	+36:42:38.64	H07
8	21:02:21.246	+36:42:38.71	H08
9	21:02:21.252	+36:42:08.71	H09
10	21:02:21.262	+36:41:08.71	H11
11	21:02:23.751	+36:41:38.77	H13
12	21:02:23.741	+36:42:38.77	H15
13	21:02:16.257	+36:42:08.65	H01

Figure 7.3: The receptor positions overlaid on a regridded cube of a HARP stare observation. This is viewed in GAIA by selecting **Data-Servers** > **Local Catalogs** and selecting your OUTCAT filename (here cat.FITS). Notice that dead receptors are excluded from the list.

7.2.9 Tile dimensions

For large data sets, it may be beneficial to break the output array up into a number of smaller rectangular tiles. This can be controlled by the TILEDIMS parameter. The files can be stitched together later using the KAPPA application paste which may be used any time the tiles to be abutted have the same pixel co-ordinates. Each tile may also have a rim of extra data to allow for future co-adding, the width of which is set by the TILEBORDER parameter.

Chapter 8
Analysing your Cube

8.1 Smoothing your data

Frequency or velocity smoothing is performed using KAPPA:block. It uses a rectangular box filter with each output pixel either the mean or the median of the input pixels within the filter box. The box size can be set to 1 for dimensions which do not require smoothing. In the example below the spatial pixels remain untouched while the velocity axis is smoothed using a box of five channels.

```
% block in=mycube out=rebinnedcube box="[1,1,5]"
```

Two-dimensional NDFs can also be smoothed using gausmooth. This smooths using a symmetrical Gaussian point spread function (PSF) of specified width(s) and orientation. Each output pixel is the PSF-weighted mean of the input pixels within the filter box.

```
% gausmooth in=mymap out=smoothedmap fwhm=3
```

Note that the `fwhm` option is given in the number of pixels. For example, to smooth a map with 4″pixels by a 12″Gaussian you would set `fwhm=3`.

8.2 Removing a baseline

Baselines can be removed using KAPPA:mfittrend. This routine can fit polynomials up to order 15, or cubic splines, to each spectrum in your cube. You can specify the ranges to fit via the `ranges` parameter, or you can allow them to be determined automatically (`auto`), say if the number of observations is large.

```
% mfittrend in=cube ranges=\"-50 -10 80 160\" order=1 axis=3 out=baselinedcube
```

If using automatic range determination you should set the `numbin` option. This is the number of bins in which to compress the trend line. A single line or average may be noisy so binning is used to improve the signal-to-noise in order to enhance features that should be masked.

Ideally the bin size should match your line width to get maximum single to noise between the emission and the background. Be aware that bins that are too big can dilute weaker features. Likewise, bins that are too small will get noisy at the edges of broad lines. We recommend you experiment with the bin size to find the best fit for your data.

To exclude features that are not part of the baseline trend you can use the `clip` option. Use `clip` to provide an array of standard deviations for progressive clipping of outliers within each bin. Once the data have been binned a trend is fit within each bin and the outliers clipped, the trend is then re-fitted and the outliers again clipped etc. By default this will clip at 2, 2, 2.5 and 3 σ successively but you may wish to experiment with your own clip levels. The `clip` option is only used when `auto=true`.

For convenience GAIA has a **Baseline** tab in its cube control panel—see Section 10.1 for an example. This invokes mfittrend.

8.2.1 Advanced method

These steps give a more in depth way of determining the baselines and mimic the pipeline.

(1) Perform a block smooth on your reduced cube. Here the smoothing is quite aggressive with a box filter of five pixels along the spatial axes and fifteen pixels along the frequency axis. This is to really pull out all emission features. You can tweak these to suit your data, e.g a narrower box might be needed if you have very narrow lines.

```
% block in=cube out=temp1 box="[5,5,15]" estimator=mean
```

(2) Perform mfittrend on your smoothed cube with a high-order polynomial and automatic range determination.

```
% mfittrend in=temp1 out=temp2 axis=3 order=3 auto method=single \
  variance subtract=false mask=blmask
```

(3) Add the baseline mask to your original unsmoothed cube.

```
% add in1=cube in2=blmask out=temp3
```

(4) Perform mfittrend on your masked unsmoothed cube with a lower-order polynomial. This time fit the full range by setting `auto=false`.

```
% mfittrend in=temp3 out=temp4 axis=3 order=1 auto=false ranges=\! \
  method=single variance=true subtract=false
```

(5) Subtract the baseline from the original unsmoothed cube.

```
% sub in1=cube in2=temp4 out=baselinedcube
```

8.2.2 What about findback?

The CUPID routine findback offers an alternative for background subtraction however there are more limitations. It applies spatial filtering to remove structure with a size scale less than a specified box size. Within each box the values are replaced by the minimum of the input values. The data are then filtered again and this time replaced with the maximum in each box.

Using findback results in a baseline trend that hugs the lower limit of the noise and can contain sharp edges. These problems can be mitigated but extra steps are required. See **SUN/255** for more details.

8.3 Collapsing your map

An integrated intensity map can be created by collapsing your cube along the frequency/velocity axis using KAPPA:collapse. For each output pixel, all input pixels between the specified bounds are collapsed and combined using one of a selection of estimators. Amongst others, the estimators include the mean, integrated value, maximum value, the mean weighted by the variance, RMS value and total value. See **SUN/95** for a full list.

The example below includes the options `low` and `high` which can be specified to limit the range over which the cube is collapsed on the given axis (in this case from $-5\,\mathrm{km\,s^{-1}}$ to $+10\,\mathrm{km\,s^{-1}}$). The `variance` flag indicates the variance array should be used to define weights and generate an output variance. The option `wlim` sets the fraction of pixels that must be 'good' in the collapse range for a valid output pixel to be generated. You may find you need to reduce the value for `wlim` below its default of 0.3 for some data.

```
% collapse in=cube axis=vrad estimator=integ variance=true low=-5.0 high=10.0 \
  wlim=0.5 out=integmap
```

If you do not know the axis name you can give the number instead (1=RA, 2=Dec, 3=vrad).

For convenience GAIA has a **Collapse** tab in its cube control panel—see Section 10.6 for an example. This invokes collapse.

8.3.1 Advanced method

These steps mimic the pipeline and show the procedure for generating an integrated map collapsed only over regions with emission greater than $3\,\sigma$.

(1) Use findclumps to run a clump-finding algorithm on your cube. Specify the $3\,\sigma$ limit in your configuration file. See Section 9.5 for more details on clump-finding and Section 8.9 for how to determine your RMS.

```
% findclumps in=cube rms=<medianrms> config=^myconfig.dat \
  method=clumpfind out=temp1 outcat=mycat deconv=no
```

(2) Divide the map by itself to create a clump mask that has clump regions set to 1.

```
% div in1=temp1 in2=temp1 out=temp2
```

(3) Set the bad data to zero using KAPPA:nomagic.

```
% nomagic in=temp2 out=temp3 repval=0
```

(4) Multiply your reduced cube by the clump mask.

```
% mult in1=cube in2=temp3 out=maskedcube
```

(5) Collapse your masked reduced cube.

```
% collapse in=maskedcube out=integ estimator=integ wlim=0.1 variance
```

8.3.2 Moments maps

You can use collapse to generate the moments maps by selecting the appropriate `estimator` option—Integrated value (`Integ`) for the integrated map (zeroth moment), Intensity-weighted co-ordinate (`Iwc`) to get the velocity field (first moment), or Intensity-weighted dispersion (`Iwd`) to get the velocity dispersion (second moment).

Tip

You can also use the PICARD recipe CREATE_MOMENTS_MAPS. See Appendix A for details. This recipe follows the advanced method outlined above.

8.4 Temperature scales

Your cube will come out of makecube and the pipeline, in units of T_A^*. To convert this to main-beam brightness temperature, T_{MB} divide your cube by the main-beam efficiency, η_{MB}. Alternatively, to convert to corrected radiation temperature, T_R^*, divide by the forward scattering and spillover efficiency, η_{fss}. See Section 3.2 for efficiency values for RxA and HARP, and the JCMT webpages for further information and historical numbers.

You can use the KAPPA command cdiv to divide any NDF by a constant.

```
% cdiv in=harpcube scalar=0.63 out=harpcube_Tmb
```

8.5 Changing the pixel size

You may want to *regrid* your data onto larger pixels, for instance to allow direct comparison with data from other observatories. In the example below a JCMT map is regridded and aligned to match a Herschel map. See Appendix B for instructions on converting your FITS file to NDF format.

```
% wcsalign in=jcmtmap out=regridmap lbnd=! ubnd=! ref=herschelmap rebin conserve=f
```

Alternatively, to *resample* your data onto smaller pixels you should use the KAPPA command sqorst. In the example below 'map' has a pixel size of 8″, while 'resamplemap' has a pixel size of 4″– the number of pixels has been doubled by applying a factor of 2 to the spatial axes. A factor of 1 was applied to the frequency (third) axis leaving it unchanged.

```
% sqorst in=map out=resamplemap factors="[2,2,1]" conserve
```

`mode=factors` is the default setting so it was not specified in the example above. The example below however, uses `mode=pixelscale` to define a pixel size for the third axis; here the spectra are rebinned into 2 km s^{-1} channels.

```
% sqorst in=map out=resamplemap mode=pixelscale pixscale=2 axis=3
```

Remember you can check your current pixel size and channel spacing with ndftrace (see Section 2.4).

8.6 Mosaicking cubes

If you pass raw files covering different regions of the sky to makecube it will automatically mosaic them together. This can be heavy work for your processor so you may wish to make individual cubes and combine them during post-processing. To co-add multiple reduced maps you should use wcsmosaic.

Note that this is not the same as stitching the tiles of a reduced cube from a single makecube command. Such tiles share common pixel co-ordinates and so can be combined with paste (Section 6.6.1). Separately created cubes will not have the same pixel co-ordinates, and therefore the world co-ordinates are required to align the tesserae in the mosaic of an extended region.

```
% wcsmosaic in=^maplist out=mosaic lbnd=! ubnd=! variance
```

By selecting the lower bound (`lbnd`) and upper bound (`ubnd`) to be default (!) you are including all of the input tiles. You can change these if you want to only mosaic a sub-region of the input maps. As with makecube there are a number of regridding options available to you describing how to divide an input pixel between a group of neighbouring output ones (the default is `SincSinc`).

Note that by default wcsmosaic aligns reduced cubes along the spectral axis using the heliocentric standard of rest (the ORAC-DR pipeline also does this). If you want to align using a non-heliocentric standard of rest, you should set the AlignStdOfRest attribute for your NDFs using wcsattrib before running wcsmosaic. The following example sets the alignment to the kinematic LSR.

```
% wcsattrib <list_of_separate_cubes> set AlignStdOfRest LSRK
```

where <list_of_separate_cubes> is group of cubes specified as a comma-separated list or as one per line in a text file.

Tip

Take care not to combine data separated on the sky as wcsmosaic will attempt to create a vast cube that encompasses all the input files until it runs out of resources.

Tip

You can also use the PICARD recipe MOSAIC_JCMT_IMAGES. This will correctly combine all NDF extensions. See Appendix A for details.

8.7 Cropping your map

Cropping an ACSIS map can be fiddly. The simplest way is to create an ARD mask in GAIA and apply it to your map. The steps below work on both two- and three-dimensional data.

(1) Create your ARD mask. You may find it easiest to do this with GAIA. See Figure 8.2 for an illustrated guide.

(2) Trim off the third axis from your mask using ndfcopy.

```
% ndfcopy mask trim trimwcs
```

(3) Apply the saved ARD mask to your map. Setting `inside=false` will mask the pixels outside rather than inside your mask. The masked pixels will appear blank when you open it with GAIA; see the left-hand panel of Figure 8.2.

```
% ardmask map inside=false ard=ardmask.txt out=maskmap
```

(4) Trim off the blank pixels using ndfcopy with the `trimbad` option. This leaves just your selected region; see the right-hand panel of Figure 8.2.

```
% ndfcopy maskmap trimbad
```

An alternative method is to use ndfcopy while defining the section of the map or cube you wish to extract as an output cube. See Section 2.6.

8.7.1 Supplying a template

If you wish to extract only a region which overlaps with an existing file you have, e.g. from a different observing campaign or telescope, you can also use ndfcopy but with the `like` option.

```
% ndfcopy in_full out_cropped like=map2 likewcs
```

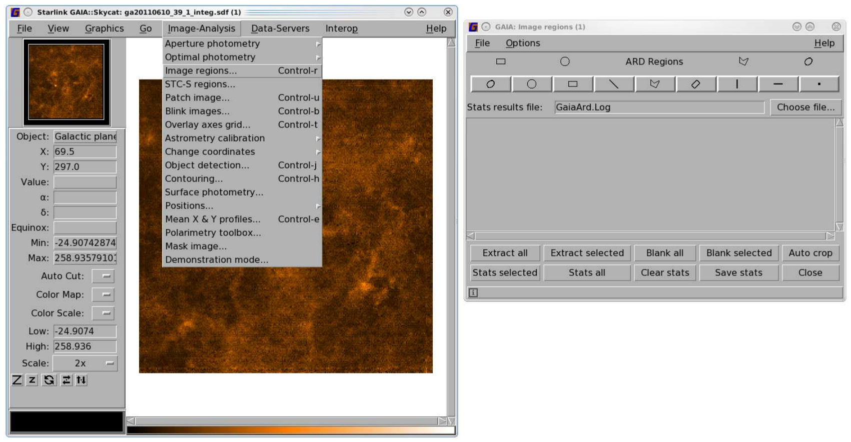
The shape of the file supplied by `like` will determine the shape of the output file. This shape can be in either pixel indices or the current WCS Frame. If the WCS Frame is required, include the parameter `likewcs` to the command line otherwise it can be omitted.

Defining an ARD region in GAIA

Open your map with GAIA.

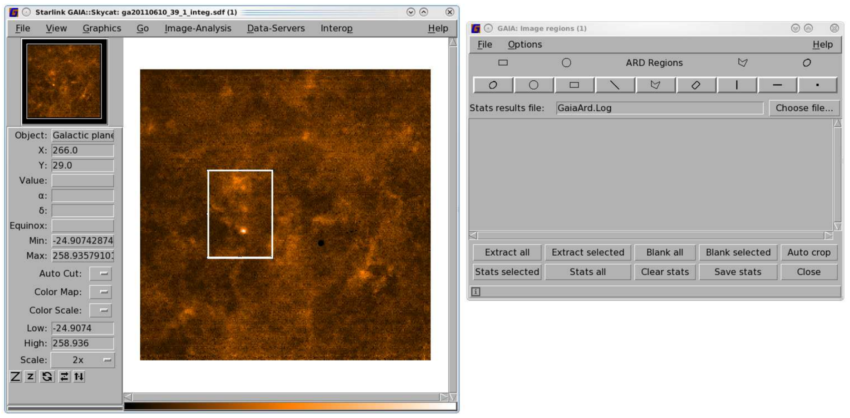
```
% gaia map
```

In GAIA go to **Image-analysis>Image regions**.


Starlink GAIA::Skycat: ga20110610_39_1_integ.sdf (1)
File View Graphics Go Image-Analysis Data-Servers Interop Help
Aperture photometry
Optimal photometry
Image regions... Control-r
STC-S regions...
Patch image... Control-u
Blink images... Control-b
Overlay axes grid... Control-t
Astrometry calibration
Change coordinates
Object detection... Control-j
Contouring... Control-h
Surface photometry...
Positions...
Mean X & Y profiles... Control-e
Polarimetry toolbox...
Mask image...
Demonstration mode...
Auto Cut:
Color Map:
Color Scale:
GAIA: Image regions (1)
File Options Help
ARD Regions
Extract all Extract selected Blank all Blank selected Auto crop
Stats selected Stats all Clear stats Save stats Close
Choose file...

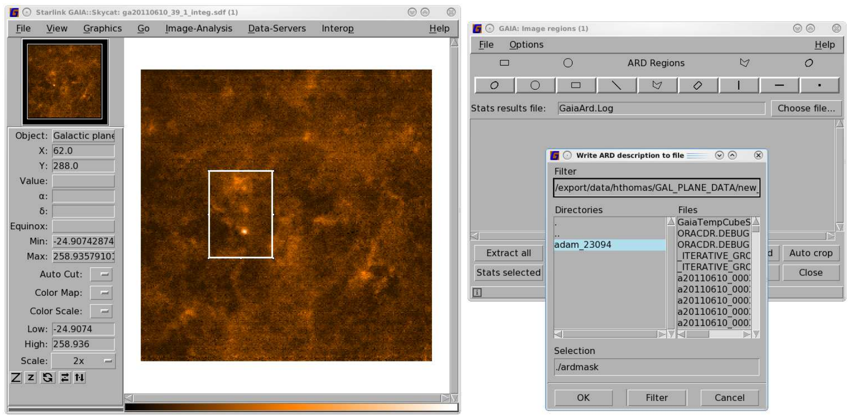
Object: Galactic plane
X: 69.5
Y: 297.0
Min: -24.90742874
Max: 258.9357910
Low: -24.9074
High: 258.936
Scale: 2x
Stats results file: GaiaArd.Log

Select the shape of the ARD region you wish to define and drag it on your map by holding the mouse button down, dragging the shape out, then releasing the button.


Starlink GAIA::Skycat: ga20110610_39_1_integ.sdf (1)
File View Graphics Go Image-Analysis Data-Servers Interop Help
Auto Cut:
Color Map:
Color Scale:
GAIA: Image regions (1)
File Options Help
ARD Regions
Extract all Extract selected Blank all Blank selected Auto crop
Stats selected Stats all Clear stats Save stats Close
Choose file...

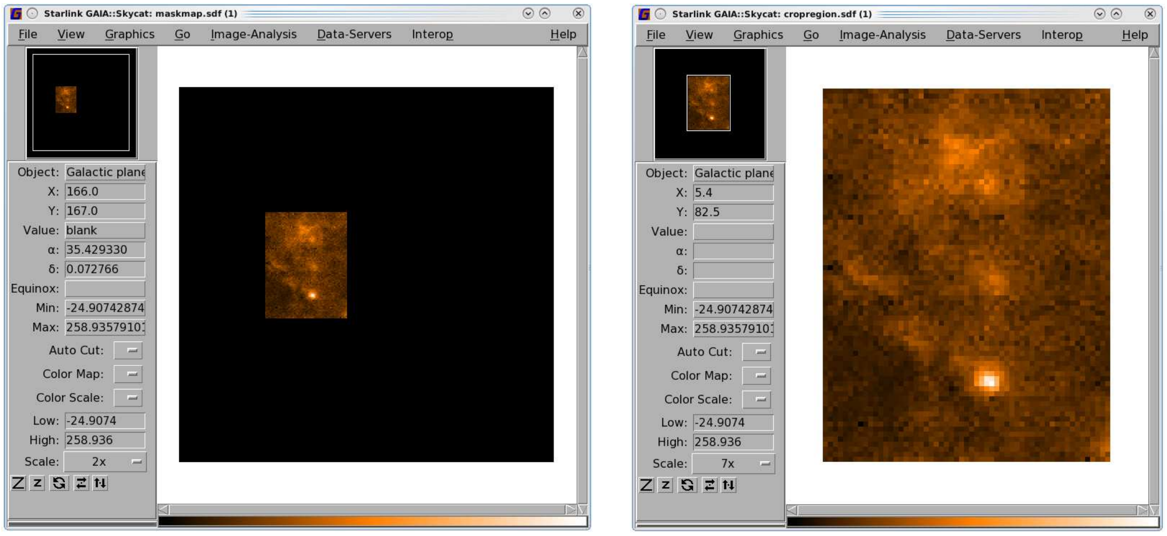
Object: Galactic plane
X: 266.0
Y: 29.0
Min: -24.90742874
Max: 258.9357910
Low: -24.9074
High: 258.936
Scale: 2x
Stats results file: GaiaArd.Log

In the "`Image regions`" window go to **File>Save ARD description** to save your ARD mask for later use.



Object: Galactic plane
X: 62.0
Y: 288.0
Min: -24.90742874
Max: 258.9357910
Low: -24.9074
High: 258.936
Scale: 2x
Stats results file: GaiaArd.Log
Filter: /export/data/hthomas/GAL_PLANE_DATA/new
Selection: ./ardmask

Figure 8.1: Defining an ARD region using GAIA.



Object: Galactic plane
X: 166.0
Y: 167.0
Value: blank
α: 35.429330
δ: 0.072766
Equinox:
Min: -24.90742874
Max: 258.93579101
Low: -24.9074
High: 258.936
Scale: 2x

Object: Galactic plane
X: 5.4
Y: 82.5
Value:
α:
δ:
Equinox:
Min: -24.90742874
Max: 258.93579101
Low: -24.9074
High: 258.936
Scale: 7x

Figure 8.2: (Left) After applying the ARD mask, the cropped region is still surrounded by blank pixels. (Right) The blank pixels have been trimmed using ndfcopy with the `trimbad` option.

8.8 Filling holes in your map

You may come across empty pixels in your map due to dead receptors or receptors which have failed quality assurance in the pipeline. The second half of 2013 in particular had only twelve working receptors for HARP.

You can fill these holes using KAPPA:fillbad. This replaces the bad values with a smooth function derived from neighbouring values, derived by iterating to a solution of Laplace's equation.

The default values for fillbad will fill the holes but smooths by five pixels along each axis. If you prefer not to smooth in the spectral axis, set the parameter `size` to [*s*,*s*,0] where *s* is the scale length in pixels. The zero means do not smooth along the third axis.

The scale length should have a value about half the size of the largest region of bad data to be replaced. Since the largest bad regions apart from the cube peripheries are two pixels across, a size of 1 is appropriate.

```
% fillbad in=holeymap out=filledmap size="[1,1,0]"
```

Figure 8.3 shows an initial map with holes and the final filled map.

8.9 Checking the noise

You can use KAPPA:stats to check the noise in your map. It will report the statistics for all pixels so be aware that noisy edges and strong signal will contribute to the standard deviation. You can mitigate this by trimming any noisy edges and by examining the square root of the VARIANCE component (hereafter referred to as the error component[1]) although bright emission will still increase your noise. You can select the error component with `comp=err` on the command line.

[1]No such component exists in the NDF.

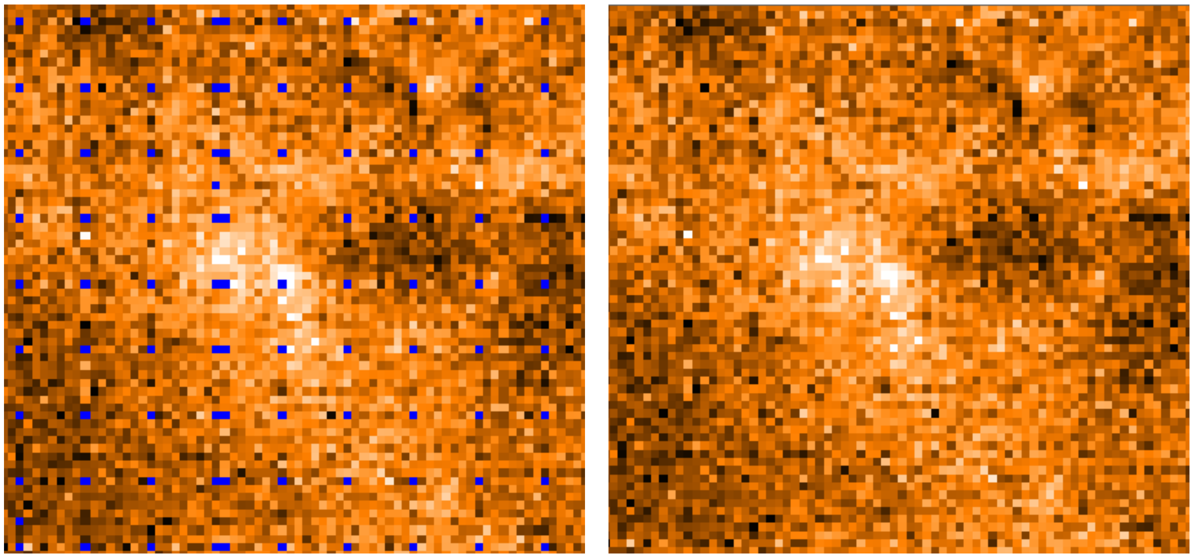


Figure 8.3: Filling holes in your HARP map with fillbad. Compare the original map (left) with the filled map (right). A blue pixel indicates an undefined spectrum.

```
% stats file comp=err order
```

Including the option `order` allows the ordered statistics such as the median and percentiles to be reported. Note that percentile options will need to be specified via the `percentiles` parameter.

Tip

Remember the mean or median values will give you the noise when using the error component.

8.9.1 Visualising the noise

You can plot the noise or error component of your map using the KAPPA command histogram. This allows you to visualise the distribution with more ease. Again the `comp=err` option is used.

```
% histogram mosaic comp=err range=! numbin=200
```

The output is shown in Figure 8.4. An alternative to setting the number of bins via the `numbin` parameter is to set the width of each bin using the `width` parameter.

It is also useful to view the noise map itself. Figure 8.5 shows how to select the error component in GAIA. Other options available from the "`Select NDF in container file`" window include the exposure time, the effective time and, if `spread=nearest` when running makecube, the system temperature. To return to your main image select the top level and click the **Data** button.

8.9.2 HARP flatfielding

The individual detectors in the HARP array do not respond exactly the same to the incoming signal, returning different temperatures for the same flux. If uncorrected, this can lead to artificial net-like

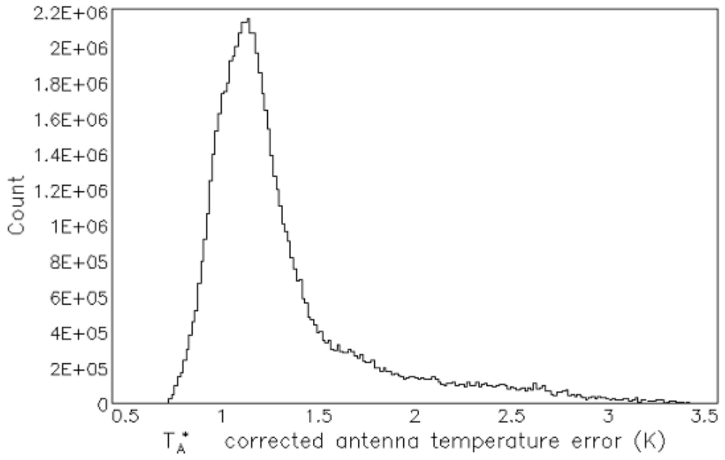
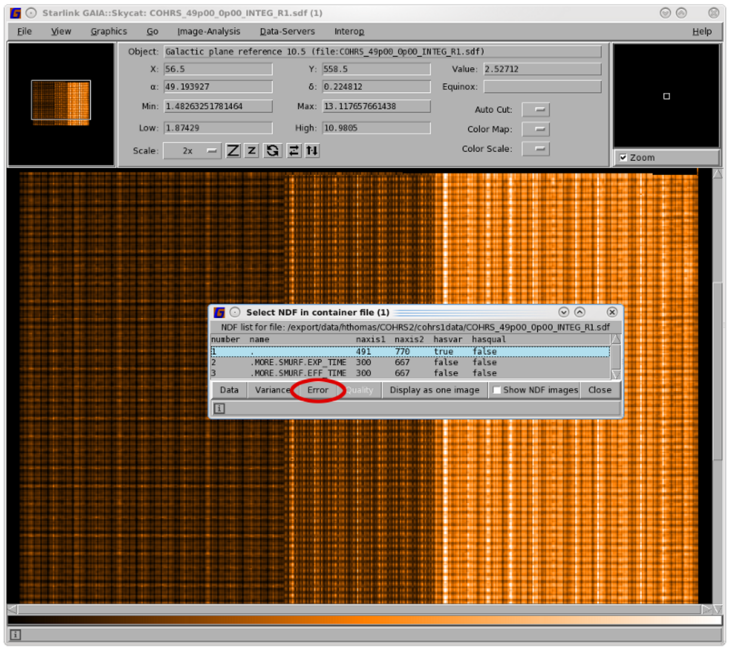


Figure 8.4: The error component plotted with histogram.

Tip

You can write out the error component into a new file using ndfcopy.

```
% ndfcopy map comp=err noisemap
```



```
Object: Galactic plane reference 10.5 (file:COHRS_49p00_0p00_INTEG_R1.sdf)
X: 56.5                 Y: 558.5                Value: 2.52712
α: 49.193927            δ: 0.224812             Equinox:
Min: 1.48263251781464   Max: 13.117657661438
Low: 1.87429            High: 10.9805
Scale: 2x
```

```
NDF list for file: /export/data/hthomas/COHRS2/cohrs1data/COHRS_49p00_0p00_INTEG_R1.sdf
number  name                 naxis1  naxis2  hasvar  hasqual
1       .                    491     770     true    false
2       .MORE.SMURF.EXP_TIME 300     667     false   false
3       .MORE.SMURF.EFF_TIME 300     667     false   false
```

Figure 8.5: The noise map of a mosaic of tiles taken in differing weather conditions.

patterns in the reduced spectral cube. As with any flatfield correction, the goal is to expose to a uniform source to measure the relative responses of the detectors. However, this is not possible with HARP observations, and the approach is assume that during an observation the detectors see approximately the same signal from the various sources. For extended emission at low galactic latitudes this is a reasonable assumption. For isolated compact sources it is likely to be invalid.

It may be possible to use a flatfield determined on the same night in a high-signal region and apply that to a low-signal or compact-source observation. The flatfield can change during the night, but it might better than no flatfield.

The method of Curtis et al. (2010)[9] creates a map for each detector and integrates the measured flux in the main spectral line. Use the DETECTORS parameter to create a spectral cube from just that detector.

```
% makecube in=rawfile out=cube_H01 detectors="'H01'" autogrid
```

The simplest way to sum the flux across a line run stats over a chosen spectral range, such as $-5\,\mathrm{km\,s^{-1}}$ to $7.2\,\mathrm{km\,s^{-1}}$.

```
% stats cube_H01"(,,-5.0:7.2)"
```

If there might be bad pixels present, the mean is more accurate than the sum. This assumes that the line is located within these bounds across the whole spatial region. It is wise not to push too far into the wings of the line and baseline errors and noise have a greater impact on the integrated flux.

The receptors are normalised to the flux of the reference receptor, near the centre, normally H05, but given by the FITS header REFRECEP.

There are other methods in ORAC-DR, one to cope with multiple lines. It sums all the signal above some threshold, such as the median plus four standard deviations, to exclude the baseline noise. Since the flat ratio biases this sum, ORAC-DR iterates to converge to a solution.

Chapter 9
Advanced Analysis

9.1 Changing co-ordinate frames

Co-ordinate systems of regridded Starlink files are described by the world co-ordinate system, (WCS). There are a number of basic co-ordinate systems (or frames) common to all NDFs; these are described by their DOMAIN (essentially their name) and are PIXEL, AXIS, GRID and FRACTION. Additional frames can be stored in the WCS component. Two common additional frames are SKY and SPECTRUM. SKY refers to two-dimensional frames (known as SkyFrames) which describe sky co-ordinates. SPECTRUM refers to one-dimensional frames (known as SpecFrames) that describe a position within a spectrum.

The choice of co-ordinates within the SkyFrame or SpecFrame is called the system. For example, for SkyFrames this may be Galactic or FK5, while for SpecFrames this may be velocity or frequency.

The current frame and the system can both be changed using the KAPPA applications wcsframe and wcsattrib. The current frame for data processed with makecube (either manually or by the pipeline) typically has a DOMAIN of SKY-DSBSPECTRUM. The compound name refers to the first two axes of the frame having sky co-ordinates and the third axis having dual side-band spectral units. You can check this with ndftrace:

```
% ndftrace file | grep Domain
```

You can change the attributes of a cube using wcsattrib. To change the SkyFrame co-ordinate system set the System attribute.

```
% wcsattrib file set system galactic
```

To change between velocity and frequency you also use the System attribute. For SKY-DSBSPECTRUM the software knows which axis is being referred to based on the option you supply (the third in this case).

```
% wcsattrib file set system freq
% wcsattrib file set system vrad
```

Once a cube has been collapsed and is two-dimensional its Frame becomes SKY. If you need to change frames manually use wcsframe.

```
% wcsframe file sky
```

For offset co-ordinates you will need to set the system `skyrefis` to origin. The final two lines in the example below convert the offset units to arcseconds instead of radians.

```
% wcsattrib file set skyrefis origin
% wcsattrib file set 'format(1)' 's.*'
% wcsattrib file set 'format(2)' 's.*'
```

9.2 Hybrid data

To merge two hybrid observations run wcsalign on both files.

```
% wcsalign in="'rawfile_01.sdf,rawfile_02.sdf'" insitu lbnd=! ubnd=! ref=!
```

Next trim each subsystem to remove noisy ends of the spectra in the overlap region. You can get the pixel bounds from ndftrace (here 1024). The example below trims 35 channels from the 'left' of rawfile_01 and 24 channels from the 'right' of rawfile_02.

```
% ndfcopy in='rawfile_01(35:1024,,)' out=trimfile_01
% ndfcopy in='rawfile_02(0:1000,,)'  out=trimfile_02
```

At this point the subsystems are aligned pixel for pixel. So a subtraction with sub will generate the differences in the trimmed overlap region analysed by stats from which the median is found, which is 0.04 in the example below.

```
% sub trimfile_01 trimfile_02 overlap_12
% stats overlap_12 order
% cadd trimfile_02 0.04 offsetfile_02
```

You can then apply the offset to the second subsystem with cadd. Then the first subsystem can then be merged using wcsmosaic.

```
% wcsmosaic in='"rawfile_01,offsetfile_02"' out=rawfile12 lbnd=! method=nearest accept
```

You should check the final spectra to make sure there is no noise bump in the middle where they overlap.

Another technique to locate the noisy edges is apply collapse to all the spectra in an observation using the `estimator=sigma` option then smooth. The composite formed will show a smooth noise profile, increasing dramatically at its end. The trim limits can be estimated manually, or a linear trend fit with mfittrend will return the bounds of the flat low-noise portion of the spectrum in its `aranges` parameter.

```
% collapse in=rawfile_01 out=profile axis=spec wlim=0.0 estimator=sigma trim
% block in=profile out=profile_sm box=25
% mfittrend in=profile_sm out=junk axis=1 method=region auto order=0
% parget aranges mfittrend
```

In the example above, parget should return just one pair of pixel co-ordinates, which can then form part of an NDF section to extract the non-noisy part of rawfile_01 using ndfcopy. If there is more than one range you want to use the lowest and highest values.

9.3 Position-velocity diagram

You can create a position-velocity map by collapsing over the skylat or skylon axis of a reduced cube. Use the KAPPA command collapse with `axis=skylat`:

```
collapse in=reduced.sdf axis=skylat estimator=sum out=pv
```

You can view your pv map with GAIA or display.

To obtain a position-velocity map at an arbitrary angle, there is a PVSLICE in the DATACUBE package. This can be invoked directly, or its recipe extracted from the PVSLICE script, if you know the co-ordinates of the ends of the slice and do not need the graphics.

9.4 Creating channel maps

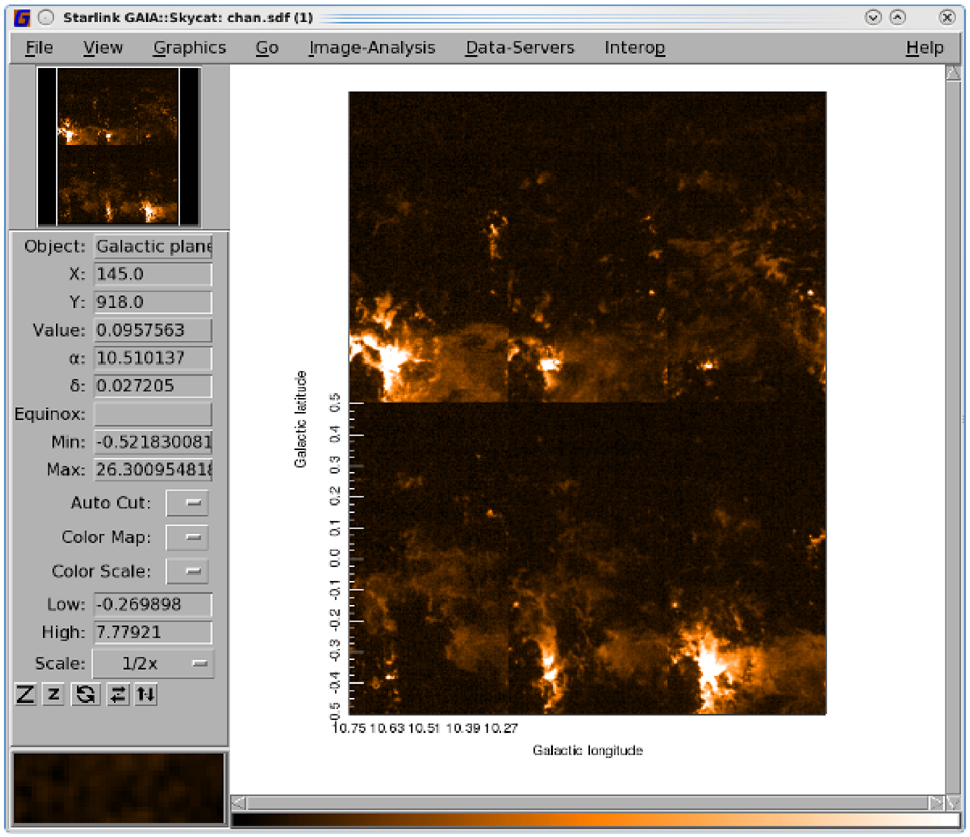
The KAPPA application chanmap creates a two-dimensional channel map from a cube. It collapses along the nominated axis (with many of the same parameters as collapse). The number of channels to create is given by `nchan` and these are divided evenly between your ranges. The arrangement of the resulting panels is determined by the shape parameter. The collapsed slices are tiled with no margins to form the final image.

```
% chanmap in=cube axis=3 low=-10 high=5 nchan=6 shape=3 estimator=mean
```

This grid of channel maps is filled from left to right, and bottom to top. It can be viewed with GAIA (see Figure 9.1) or with display.

```
% display chanmap mode=faint noaxes lut=$STARLINK_DIR/bin/kappa/smooth3_lut
```

Channel maps can also be created using GAIA. See Section 10.2 for instructions.



Object: Galactic plane
X: 145.0
Y: 918.0
Value: 0.0957563
α: 10.510137
δ: 0.027205
Equinox:
Min: -0.521830081
Max: 26.30095481
Auto Cut:
Color Map:
Color Scale:
Low: -0.269898
High: 7.77921
Scale: 1/2x

Figure 9.1: Displaying a channel map with GAIA.

9.5 Clump finding

The CUPID application findclumps can be used to generate a clump catalogue. It identifies clumps of emission in one-, two- or three-dimensional NDFs. You can select from the clump-finding algorithms

FellWalker, GaussClumps, ClumpFind or Reinhold. You must supply a configuration file which lists the options for whichever algorithm you chose (see the CUPID manual for a full list). The result is returned as a catalogue in a text file and as a NDF pixel mask showing the clump boundaries (see Figure 9.2). See Appendix E for descriptions of the various algorithms.

```
% findclumps in=map out=clumpmap outcat=clumps.FIT logfile=clumps.log \
  config=^myconfig.dat method=clumpfind rms=0.2 shape=polygon
```

The shape option allows findclumps to create a shape that should be used to describe the spatial coverage of each clump in the output catalogue. It can be set to " None", " Polygon" or " Ellipse". The shape - as described in the output catalogue—is in "STC-S" format. STC-S is a textual format developed by the IVOA for describing regions within a WCS—see here for details.. The STC-S descriptions are added as extra columns to the output catalogue.

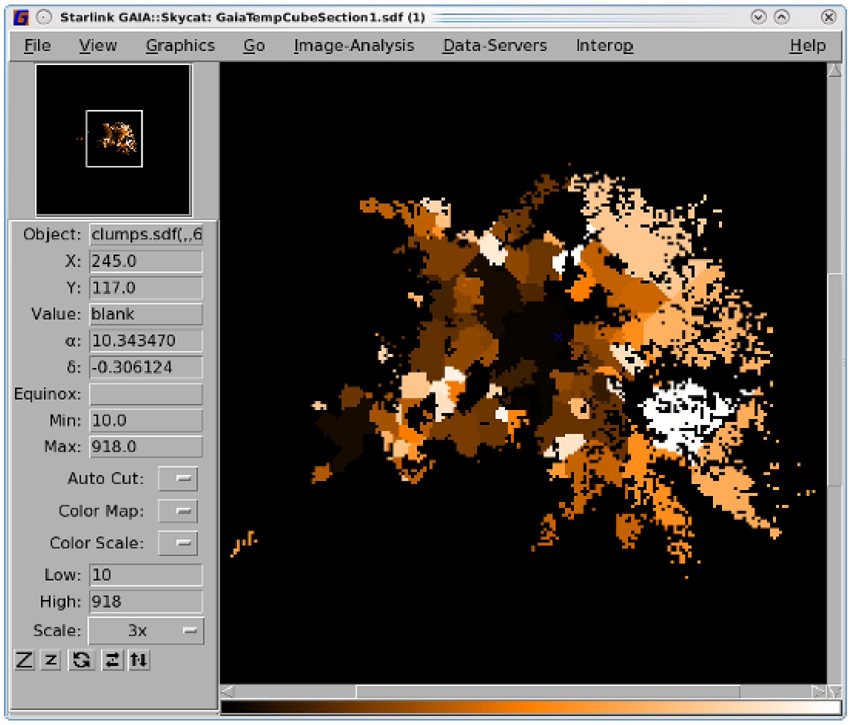


Figure 9.2: A velocity slice of the result of three-dimensional clump finding viewed with GAIA. This output NDF contains boundaries of the clumps, with the value reflecting the number of pixels contained within that clump.

Polygon Each polygon will have, at most, 15 vertices. For two-dimensional data the polygon is a fit to the clump's outer boundary (the region containing all good data values). For three-dimensional data the spatial footprint of each clump is determined by rejecting the least significant 10% of spatial pixels, where "significance" is measured by the number of spectral channels that contribute to the spatial pixel. The polygon is then a fit to the outer boundary of the remaining spatial pixels.

Ellipse All data values in the clump are projected onto the spatial plane and "size" of the collapsed clump at four different position angles—all separated by 45°—is found. The ellipse that generates the same sizes at the four position angles is then found and used as the clump shape.

You can plot the clump outlines over an image using GAIA. See Section 10.4 for instructions.

9.6 Using SPLAT to identify telluric emission

Occasionally you may see unexpected emission within your data. An example is shown in Figure 9.3. In this figure, we use the average spectrum feature (see Section 10.5)—and in this example—we see expected bright emission around 7 km s^{-1}. In addition to the expected bright emission, we see some faint emission around -30 km s^{-1}.

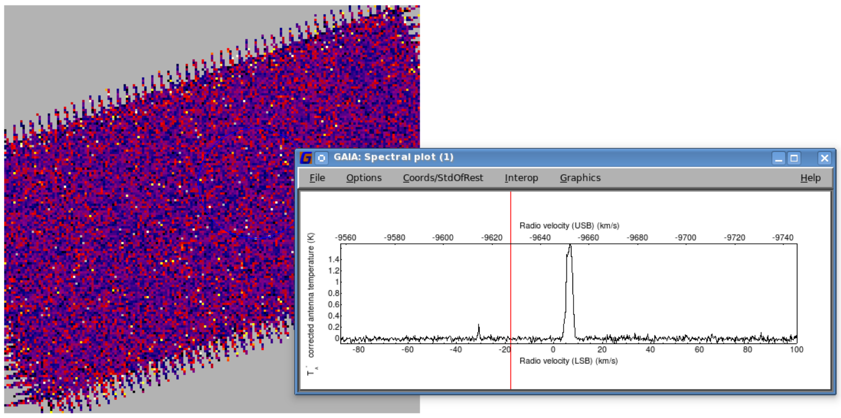

GAIA: Spectral plot (1)
File Options Coords/StdOfRest Interop Graphics Help
Radio velocity (USB) (km/s)
-9560 -9580 -9600 -9620 -9640 -9660 -9680 -9700 -9720 -9740
1.4 1.2 1 0.8 0.6 0.4 0.2 0
T*A corrected antenna temperature (K)
-80 -60 -40 -20 0 20 40 60 80 100
Radio velocity (LSB) (km/s)

Figure 9.3: An example of a spectrum with bright expected emission around 7 km s^{-1} and unexpected emission around -30 km s^{-1}. The data are displayed using GAIA.

We can use SPLAT to investigate this emission further (see Section 10.8 on how to open up a spectrum in SPLAT).

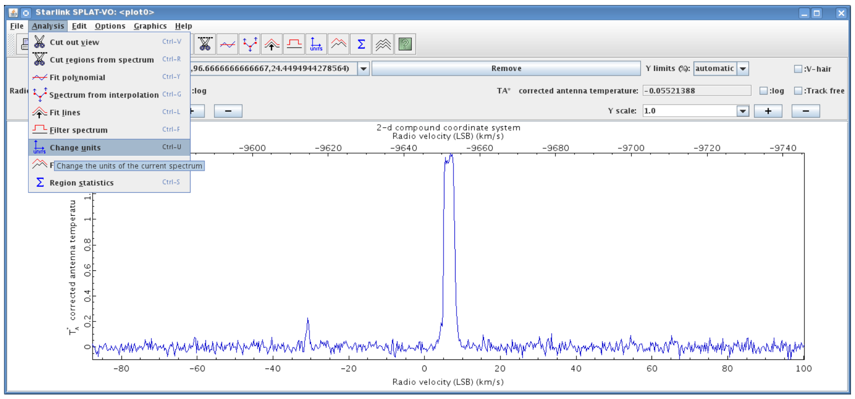
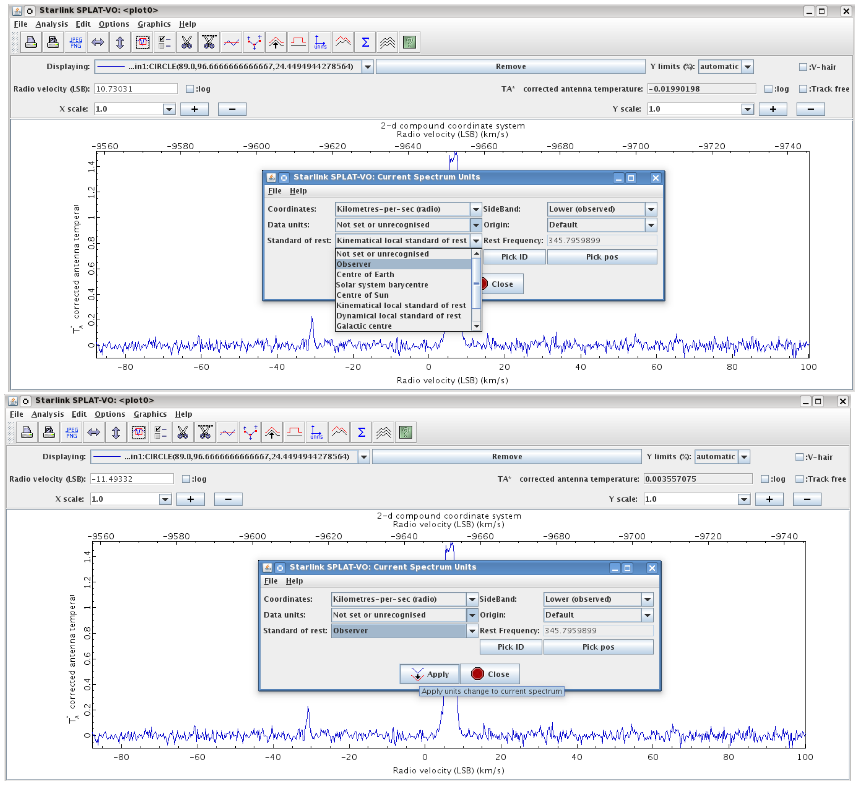

Starlink SPLAT-VO: <plot0>
File Analysis Edit Options Graphics Help
Cut out view Ctrl-V
Cut regions from spectrum Ctrl-R
Fit polynomial Ctrl-Y
Spectrum from interpolation Ctrl-G
Fit lines Ctrl-L
Filter spectrum Ctrl-F
Change units Ctrl-U
Change the units of the current spectrum
Region statistics Ctrl-S
96.6666666666667,24.4494944278564)
Remove
Y limits (%): automatic
:V-hair
:log
TA* corrected antenna temperature: -0.05521388
:log :Track free
Y scale: 1.0
2-d compound coordinate system
Radio velocity (LSB) (km/s)
-9600 -9620 -9640 -9660 -9680 -9700 -9720 -9740
T*A corrected antenna temperatu
0 0.2 0.4 0.6 0.8 1
-80 -60 -40 -20 0 20 40 60 80 100
Radio velocity (LSB) (km/s)

Figure 9.4: Displaying a channel map with SPLAT.


Starlink SPLAT-VO: <plot0>
File Analysis Edit Options Graphics Help
Remove
V-hair
Track free
log
2-d compound coordinate system
Radio velocity (LSB) (km/s)
T_A^* corrected antenna temperat
Starlink SPLAT-VO: Current Spectrum Units
File Help
Pick ID
Pick pos
Apply
Close
Apply units change to current spectrum
Not set or unrecognised
Observer
Centre of Earth
Solar system barycentre
Centre of Sun
Kinematical local standard of rest
Dynamical local standard of rest
Galactic centre

Displaying: ...in1:CIRCLE(89.0,96.6666666666667,24.4494944278564)
Y limits (%): automatic
Radio velocity (LSB): 10.73031
TA* corrected antenna temperature: -0.01990198
X scale: 1.0
Y scale: 1.0

Coordinates: Kilometres-per-sec (radio)
Data units: Not set or unrecognised
Standard of rest: Kinematical local standard of rest
SideBand: Lower (observed)
Origin: Default
Rest Frequency: 345.7959899

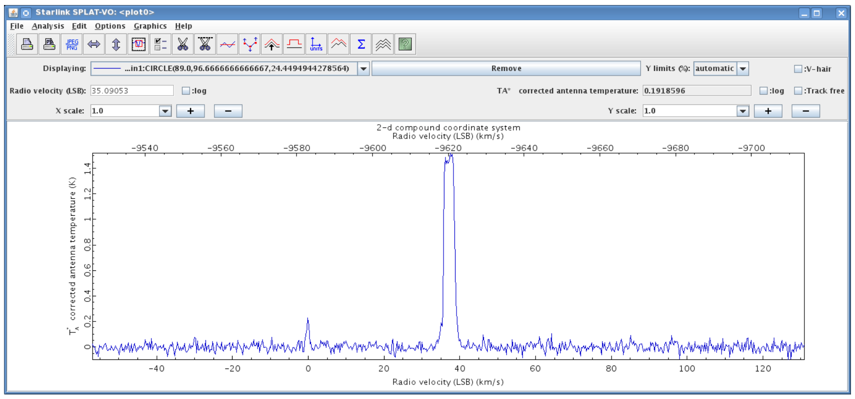
Displaying: ...in1:CIRCLE(89.0,96.6666666666667,24.4494944278564)
Y limits (%): automatic
Radio velocity (LSB): -11.49332
TA* corrected antenna temperature: 0.003557075
X scale: 1.0
Y scale: 1.0

Coordinates: Kilometres-per-sec (radio)
Data units: Not set or unrecognised
Standard of rest: Observer
SideBand: Lower (observed)
Origin: Default
Rest Frequency: 345.7959899

Figure 9.5: Select **Observer** from the **Standard of rest** Menu and select **Apply**.

It is possible to use SPLAT to change the units to **Standard of rest: Observer** by first going to **Analysis** and clicking on **Change units**.

We then use the drop-down menu next to **Standard of rest** to select **Observer** from the list, and select **Apply**.


Starlink SPLAT-VO: <plot0>
File Analysis Edit Options Graphics Help
Remove
V-hair
Track free
log
2-d compound coordinate system
Radio velocity (LSB) (km/s)
T_A^* corrected antenna temperature (K)

Displaying: ...in1:CIRCLE(89.0,96.6666666666667,24.4494944278564)
Y limits (%): automatic
Radio velocity (LSB): 35.09053
TA* corrected antenna temperature: 0.1918596
X scale: 1.0
Y scale: 1.0

Figure 9.6: Spectrum from the observer rest frame. We see the unexpected line is found at 0 km s^{-1}, and is likely telluric contamination.

Now we see the unexpected line is at 0 km s^{-1}, and is therefore likely to be contamination from telluric lines. To confirm a telluric line we can compare the frequency of the unidentified line to the frequency of known telluric lines. To get the frequency of the line again go to **Analysis** and click on **Change units** and then change the units to **Coordinates: Gigahertz**. Finally read off the position of the line.

Chapter 10
Using GAIA

10.1 Removing a baseline with GAIA

(1) Select the spectrum you want to use as your template for the entire cube.

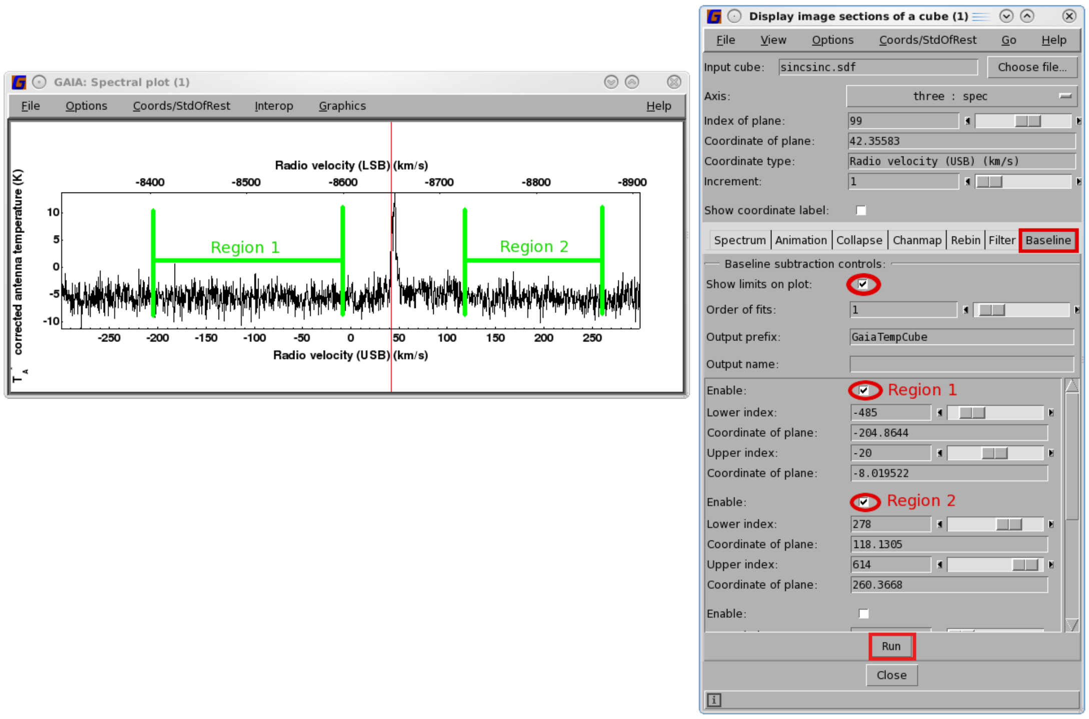
(2) Select the Baseline tab in the "`Display image sections of a cube`" window.

(3) Select the order of the baseline you want to fit and remove.

(4) Check the **Show limits on plot** button to interactively draw your baseline windows. You can click and drag the edges of these limit lines in the "`Spectral plot`" window.

(5) Check **Enable** for each new baseline window you want to define.

(6) Click **Run**.



Input cube: sincsinc.sdf
Axis: three : spec
Index of plane: 99
Coordinate of plane: 42.35583
Coordinate type: Radio velocity (USB) (km/s)
Increment: 1
Show coordinate label: ☐

Show limits on plot: ✓
Order of fits: 1
Output prefix: GaiaTempCube
Output name:

Enable: ✓ Region 1
Lower index: -485
Coordinate of plane: -204.8644
Upper index: -20
Coordinate of plane: -8.019522

Enable: ✓ Region 2
Lower index: 278
Coordinate of plane: 118.1305
Upper index: 614
Coordinate of plane: 260.3668

Enable: ☐

Figure 10.1: Removing a baseline with GAIA.

10.2 Creating channel maps with GAIA

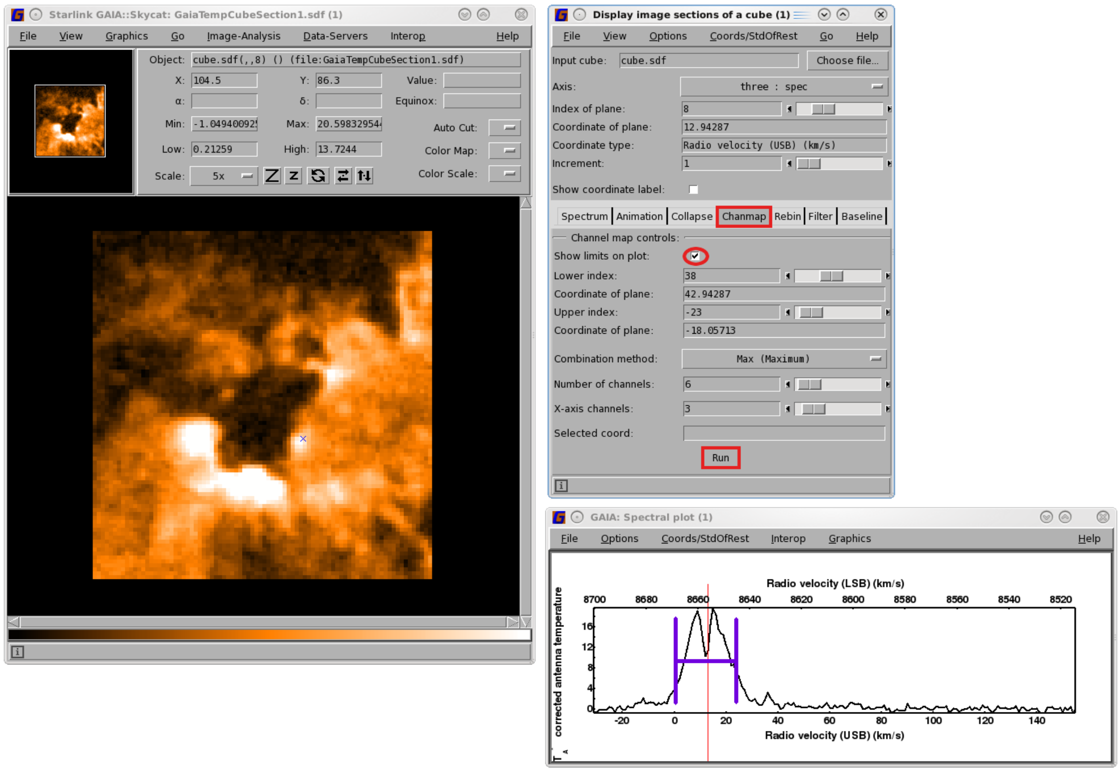
You can display channel maps in GAIA by selecting the region of a spectrum you wish to collapse over. You can use a spectrum from any of the pixels in the cube and the region selected will be applied to the whole map.

(**1**) Select the spectrum you want to use as your template for the entire cube.

(**2**) Select the **Chanmap** tab in the "`Display image sections of a cube`" window.

(**3**) Check the **Show limits on plot** box to interactively draw the range over which to collapse your cube. You can click and drag the end-bars of the limit lines in the "`Spectral plot`" window.

(**4**) Select the collapse method (Max is chosen in the example).


Starlink GAIA::Skycat: GaiaTempCubeSection1.sdf (1)
File View Graphics Go Image-Analysis Data-Servers Interop Help
Display image sections of a cube (1)
File View Options Coords/StdOfRest Go Help
Choose file...
Spectrum Animation Collapse Chanmap Rebin Filter Baseline
Channel map controls:
Run
GAIA: Spectral plot (1)
File Options Coords/StdOfRest Interop Graphics Help
Radio velocity (LSB) (km/s)
8700 8680 8660 8640 8620 8600 8580 8560 8540 8520
16 12 8 4 0
-20 0 20 40 60 80 100 120 140
Radio velocity (USB) (km/s)
corrected antenna temperature

Object: cube.sdf(,,8) () (file:GaiaTempCubeSection1.sdf)
X: 104.5 Y: 86.3 Value:
α: δ: Equinox:
Min: -1.04940092 Max: 20.59832954
Low: 0.21259 High: 13.7244
Scale: 5x
Auto Cut: Color Map: Color Scale:

Input cube: cube.sdf
Axis: three : spec
Index of plane: 8
Coordinate of plane: 12.94287
Coordinate type: Radio velocity (USB) (km/s)
Increment: 1
Show coordinate label: ☐
Show limits on plot: ✓
Lower index: 38
Coordinate of plane: 42.94287
Upper index: -23
Coordinate of plane: -18.05713
Combination method: Max (Maximum)
Number of channels: 6
X-axis channels: 3
Selected coord:

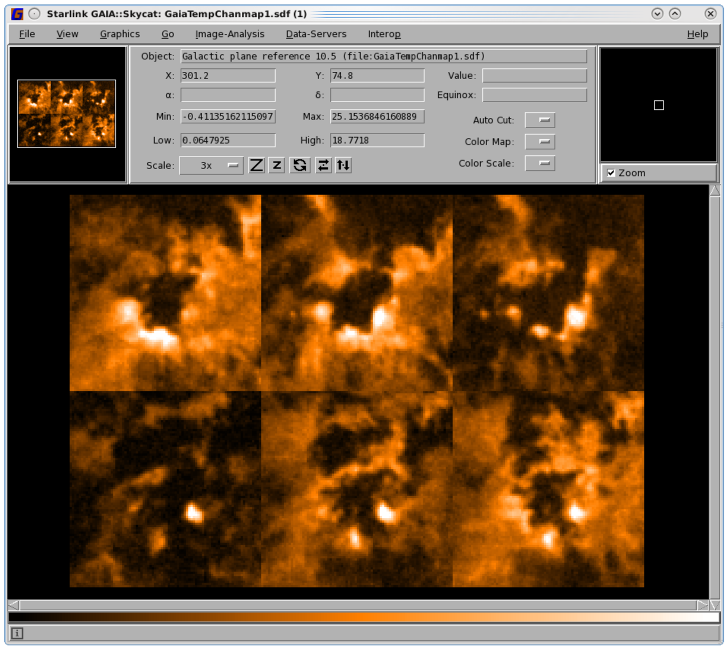
Figure 10.2: Creating a channel map with GAIA.

(**5**) Use the slider bars to select the total number of channels you want to generate and the number of x-axis channels. The x-axis channel number sets the aspect ratio for the resulting display grid.

(**6**) Click **Run**. The result is shown in the figure below.

10.3 Contouring with GAIA

(**1**) Open the map you wish to contour over.


Starlink GAIA::Skycat: GaiaTempChanmap1.sdf (1)
File View Graphics Go Image-Analysis Data-Servers Interop Help
Zoom

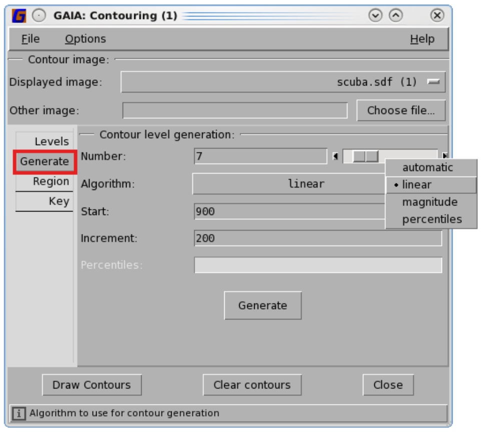
Object: Galactic plane reference 10.5 (file:GaiaTempChanmap1.sdf)
X: 301.2
Y: 74.8
Value:
α:
δ:
Equinox:
Min: -0.41135162115097
Max: 25.1536846160889
Auto Cut:
Low: 0.0647925
High: 18.7718
Color Map:
Scale: 3x
Color Scale:

Figure 10.3: Channel map created with GAIA.

(2) Select **Image-Analysis**>**Contouring** from the menu bar across the top of the main window.

(3) Select the file you wish to contour in the "`Contouring`" window.

(4) Generate your contours in the **Generate** tab found on the left-hand side of the "`Contouring`" window. The example below defines linearly spaced contours starting at 900K. Clicking the **Generate** button will return you to the Levels tab.

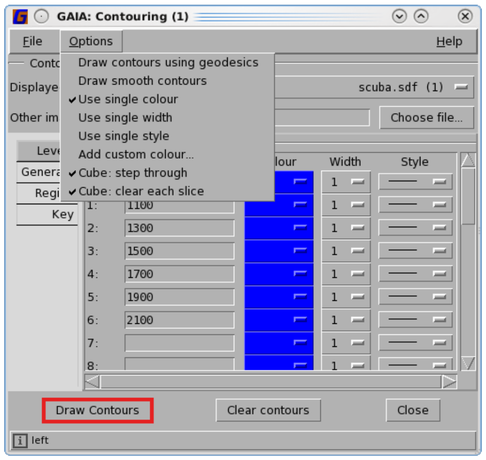


Contour image:
Displayed image: scuba.sdf (1)
Other image:
Choose file...
Contour level generation:
Number: 7
Algorithm: linear
Start: 900
Increment: 200
Percentiles:
Algorithm to use for contour generation

Figure 10.4: Setting contours with GAIA.

You can also input or edit the contour levels manually in the **Levels** tab.

(5) Customise the look of your contours under the **Options** menu. You can experiment with the other tabs (Region and Key) for options concerning contour area and legend.



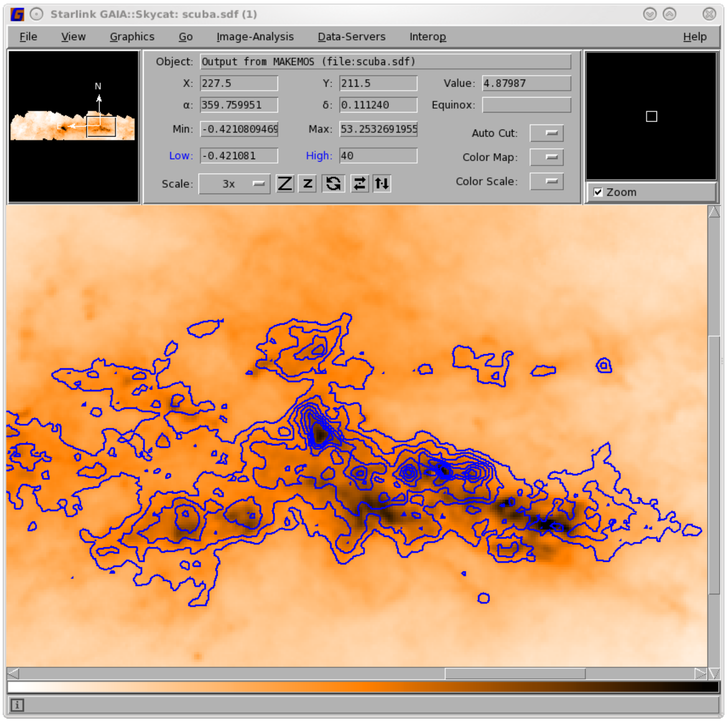
Displayed: scuba.sdf (1)

	Level	Width
1:	1100	1
2:	1300	1
3:	1500	1
4:	1700	1
5:	1900	1
6:	2100	1
7:		1
8:		1

left

Figure 10.5: Formatting contours with GAIA.

(6) Click the **Draw Contours** button to make the contours appear over your map in the main window. If you are contouring over a cube you can scroll through the velocity axis whilst the contours remain fixed on top.



Object: Output from MAKEMOS (file:scuba.sdf)
X: 227.5 Y: 211.5 Value: 4.87987
α: 359.759951 δ: 0.111240 Equinox:
Min: -0.4210809469 Max: 53.2532691955
Low: -0.421081 High: 40
Scale: 3x

Figure 10.6: Contoured map in GAIA.

(7) To add a second set of contours select **File**>**New window** in the top menu of the "`Contouring`" window. Here you can define a second image to be contoured and specify new levels and appearance. Open as many new contouring windows as necessary.

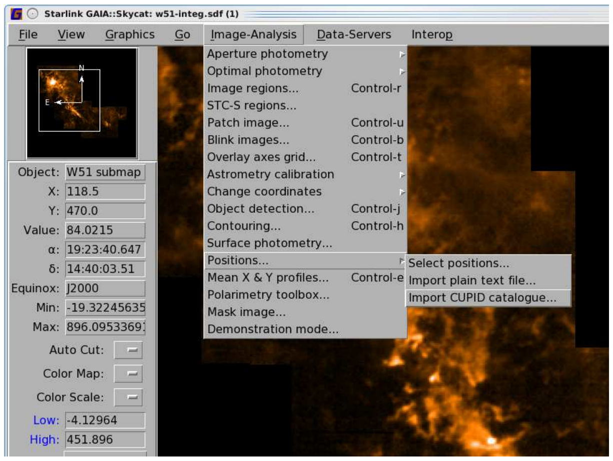
To save the graphic, there is a **File** > **Print**, but some people prefer a tool with a capture facility such as XV.

10.4 Overlaying clumps and catalogues with GAIA

GAIA can display two- or three-dimensional clump catalogues that have been generated by the CUPID routine findclumps (see Section 9.5). Clump catalogues in this format are also available for download from the JCMT Science Archive.

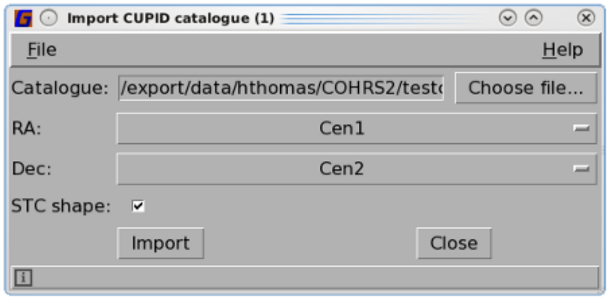
(**1**) Open your cube for three-dimensional clump finding or your integrated map for two-dimensional clump finding.

(**2**) Select **Image-Analysis**>**Positions**>**Import CUPID catalogue** from the menu bar across the top of the main window. Note that for two-dimensional catalogues an alternative route is to select **Data-Servers**>**Local catalogs**. In this case you can skip Step 3.


Starlink GAIA::Skycat: w51-integ.sdf (1)
File View Graphics Go Image-Analysis Data-Servers Interop
Aperture photometry
Optimal photometry
Image regions... Control-r
STC-S regions...
Patch image... Control-u
Blink images... Control-b
Overlay axes grid... Control-t
Astrometry calibration
Change coordinates
Object detection... Control-j
Contouring... Control-h
Surface photometry...
Positions...
Mean X & Y profiles... Control-e
Polarimetry toolbox...
Mask image...
Demonstration mode...
Select positions...
Import plain text file...
Import CUPID catalogue...
Auto Cut:
Color Map:
Color Scale:

Object: W51 submap
X: 118.5
Y: 470.0
Value: 84.0215
α: 19:23:40.647
δ: 14:40:03.51
Equinox: J2000
Min: -19.32245635
Max: 896.0953369
Low: -4.12964
High: 451.896

Figure 10.7: Importing a CUPID catalogue with GAIA.

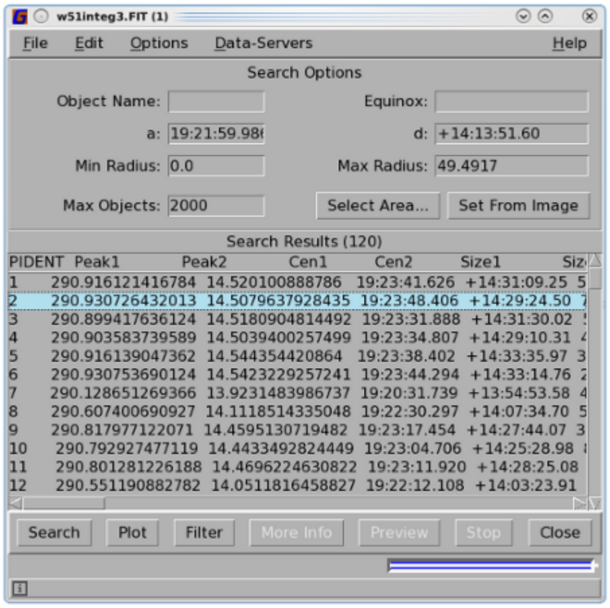


Catalogue: /export/data/hthomas/COHRS2/test
RA: Cen1
Dec: Cen2
STC shape: ✓

Figure 10.8: Catalogue window in GAIA.

(**3**) In the "`Import CUPID catalogue`" window, select a file with the **Choose file**... button. For polygon shapes tick the STC shape box. You can change the RA/Dec co-ordinates from Cen1/Cen2, which give the central position of the clumps, to Peak1/Peak2 which give the position of the peak within them.

(**4**) A catalogue window for your FITS file will appear listing all the sources and their positions and extents.

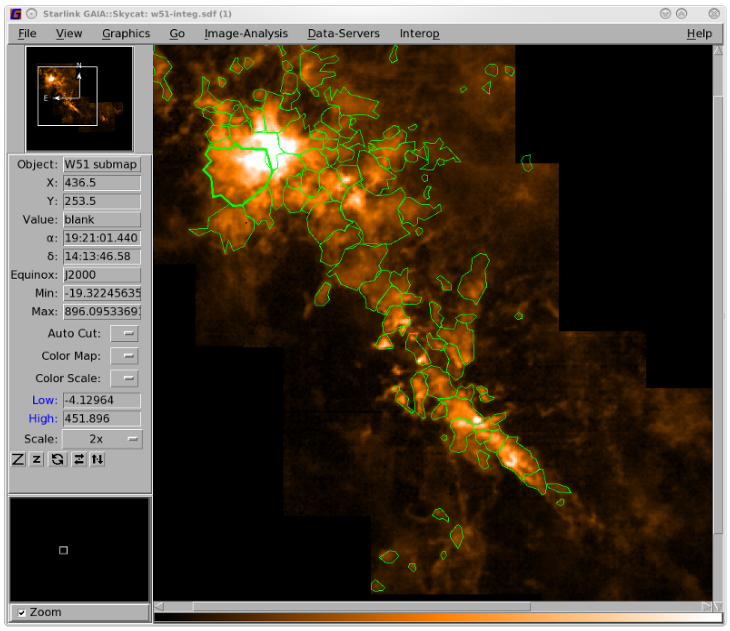


```
Search Options
Object Name:                 Equinox:
a: 19:21:59.98               d: +14:13:51.60
Min Radius: 0.0              Max Radius: 49.4917
Max Objects: 2000

Search Results (120)
PIDENT  Peak1           Peak2            Cen1          Cen2          Size1        Siz
1       290.916121416784  14.520100888786   19:23:41.626  +14:31:09.25  5
2       290.930726432013  14.5079637928435  19:23:48.406  +14:29:24.50  7
3       290.899417636124  14.5180904814492  19:23:31.888  +14:31:30.02
4       290.903583739589  14.5039400257499  19:23:34.807  +14:29:10.31  4
5       290.916139047362  14.544354420864   19:23:38.402  +14:33:35.97  3
6       290.930753690124  14.5423229257241  19:23:44.294  +14:33:14.76  2
7       290.128651269366  13.9231483986737  19:20:31.739  +13:54:53.58  4
8       290.607400690927  14.1118514335048  19:22:30.297  +14:07:34.70  5
9       290.817977122071  14.4595130719482  19:23:17.454  +14:27:44.07  3
10      290.792927477119  14.4433492824449  19:23:04.706  +14:25:28.98
11      290.801281226188  14.4696224630822  19:23:11.920  +14:28:25.08
12      290.551190882782  14.0511816458827  19:22:12.108  +14:03:23.91
```

Figure 10.9: Catalogue window in GAIA.

(**5**) Outlines of your clumps, or symbols at the peak positions, will be automatically overlaid on your map. If this does not happen, click the **Plot** button on the catalogue window. When you click on a clump from the catalogue list the outline of that clump will appear in bold on your map.



```
Object:  W51 submap
X:       436.5
Y:       253.5
Value:   blank
α:       19:21:01.440
δ:       14:13:46.58
Equinox: J2000
Min:     -19.32245635
Max:     896.0953369
Low:     -4.12964
High:    451.896
Scale:   2x
```

Figure 10.10: Clumps outlined overlaid on integrated map in GAIA.

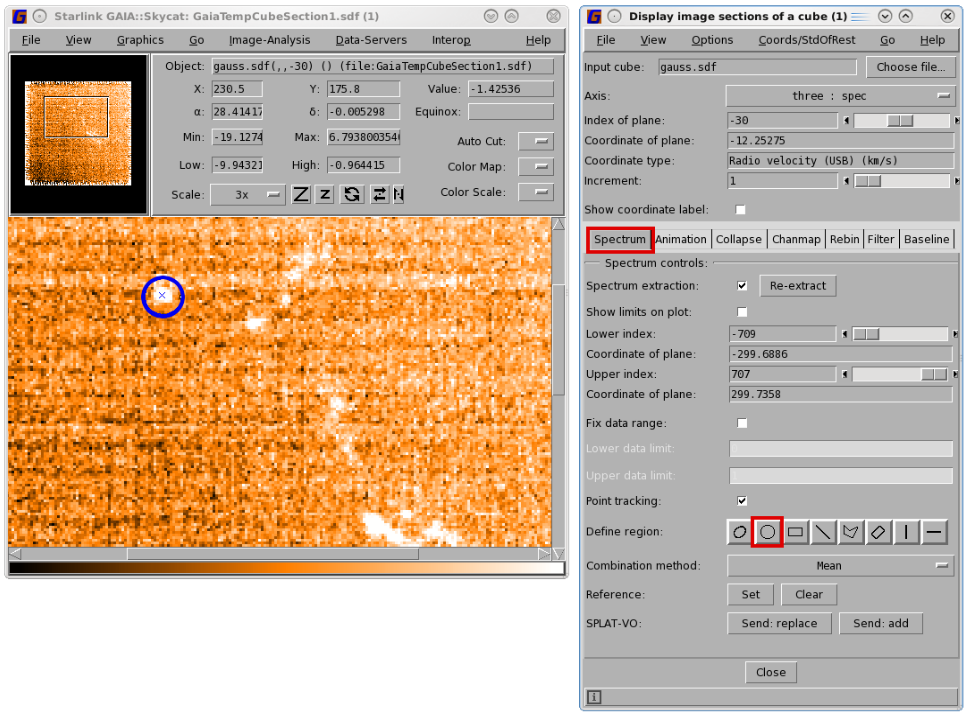
(**6**) If you are displaying a three-dimensional catalogue over a cube, it will only display clumps which include data from the current slice. The clumps shown will update as you move through the cube.

10.5 Displaying average spectrum with GAIA

(1) Select the **Spectrum** tab in the "`Display image sections of a cube`" window.

(2) Define the shape of your region by selecting one of the **Define region** buttons (a circle is chosen in the example below).

(3) Select the combination method (Mean is chosen in the example below).


Starlink GAIA::Skycat: GaiaTempCubeSection1.sdf (1)
File View Graphics Go Image-Analysis Data-Servers Interop Help
Auto Cut:
Color Map:
Color Scale:
Display image sections of a cube (1)
File View Options Coords/StdOfRest Go Help
Choose file...
Spectrum Animation Collapse Chanmap Rebin Filter Baseline
Re-extract
Set Clear
Send: replace Send: add
Close

Object: gauss.sdf(,,-30) () (file:GaiaTempCubeSection1.sdf)
X: 230.5 Y: 175.8 Value: -1.42536
α: 28.41417 δ: -0.005298 Equinox:
Min: -19.1274 Max: 6.793800354
Low: -9.94321 High: -0.964415
Scale: 3x

Input cube: gauss.sdf
Axis: three : spec
Index of plane: -30
Coordinate of plane: -12.25275
Coordinate type: Radio velocity (USB) (km/s)
Increment: 1
Show coordinate label: ☐
Spectrum controls:
Spectrum extraction: ☑
Show limits on plot: ☐
Lower index: -709
Coordinate of plane: -299.6886
Upper index: 707
Coordinate of plane: 299.7358
Fix data range: ☐
Lower data limit: 0
Upper data limit: 1
Point tracking: ☑
Define region: ○
Combination method: Mean
Reference:
SPLAT-VO:

Figure 10.11: Displaying an average spectrum with GAIA.

(4) Draw the shape on your map by clicking and dragging the mouse. The "`Spectral plot`" window will automatically update to show your combined spectrum. You can re-position and resize your shape at any time. You can see from Figure 10.12 that the averaged spectrum gives a much clearer profile of the source.

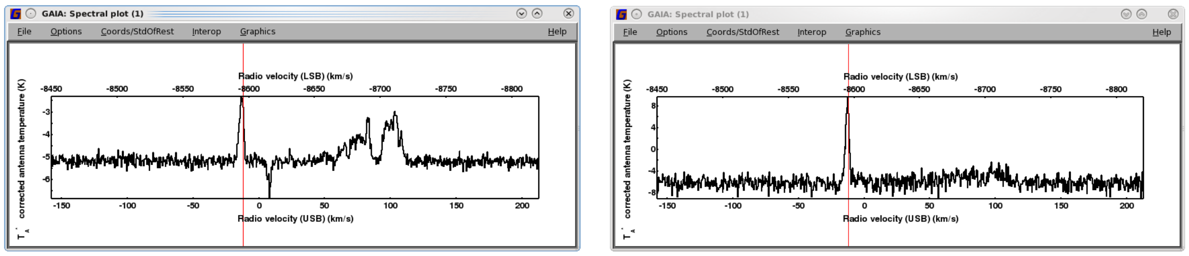


Figure 10.12: (Left) Average of the spectra within the blue circle in the figure above. (Right) Single spectrum from the centre of the blue circle above.

10.6 Collapsing your cube with GAIA

(1) Select the axis you want to collapse you to collapse over by selecting from the **Axis** drop-down list in the "`Display image sections of a cube`" window.

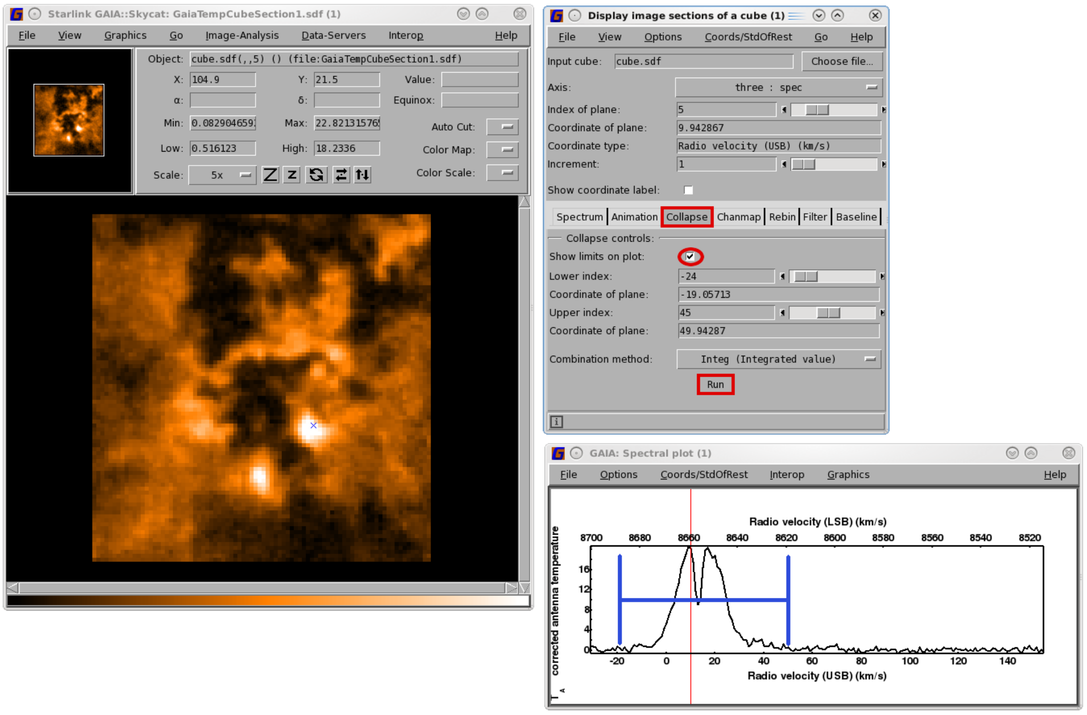
(2) Select the spectrum you want to use as a template for your cube.

(3) Select the Collapse tab in the "`Display image sections of a cube`" window.

(4) Check the **Show limits on plot** button to interactively select your collapse region. You can click and drag the edges of these limit lines in the "`Spectral plot`" window. Position these around the region you wish to collapse over.

(5) Select the collapse method via the **Combination method** drop-down list (**Integ** is selected in the example below).

(6) Click **Run**. The main window will automatically update to show your collapse image.


Starlink GAIA::Skycat: GaiaTempCubeSection1.sdf (1)
File View Graphics Go Image-Analysis Data-Servers Interop Help
Auto Cut:
Color Map:
Color Scale:
Display image sections of a cube (1)
File View Options Coords/StdOfRest Go Help
Choose file...
Spectrum Animation Collapse Chanmap Rebin Filter Baseline
Run
GAIA: Spectral plot (1)
File Options Coords/StdOfRest Interop Graphics Help
Radio velocity (LSB) (km/s)
8700 8680 8660 8640 8620 8600 8580 8560 8540 8520
corrected antenna temperature
16 12 8 4 0
-20 0 20 40 60 80 100 120 140
Radio velocity (USB) (km/s)

Object: cube.sdf(,,5) () (file:GaiaTempCubeSection1.sdf)
X: 104.9 Y: 21.5 Value:
α: δ: Equinox:
Min: 0.082904659 Max: 22.82131576
Low: 0.516123 High: 18.2336
Scale: 5x

Input cube: cube.sdf
Axis: three : spec
Index of plane: 5
Coordinate of plane: 9.942867
Coordinate type: Radio velocity (USB) (km/s)
Increment: 1
Show coordinate label: ☐

Collapse controls:
Show limits on plot: ✓
Lower index: -24
Coordinate of plane: -19.05713
Upper index: 45
Coordinate of plane: 49.94287
Combination method: Integ (Integrated value)

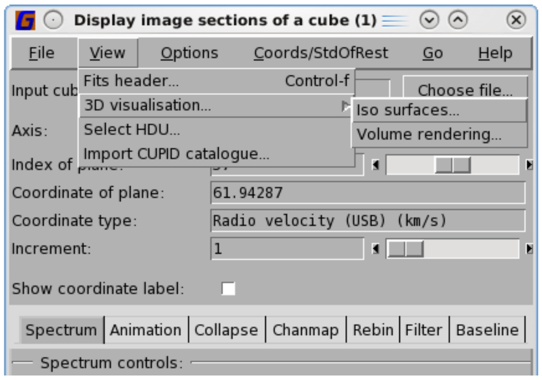
Figure 10.13: Collapsing your cube using GAIA.

10.7 Three-dimensional visualisation with GAIA

(1) Select **View**>**3D Visualisation**>**Iso surfaces...**/**Volume rendering** in the "`Display image sections of a cube`" window.

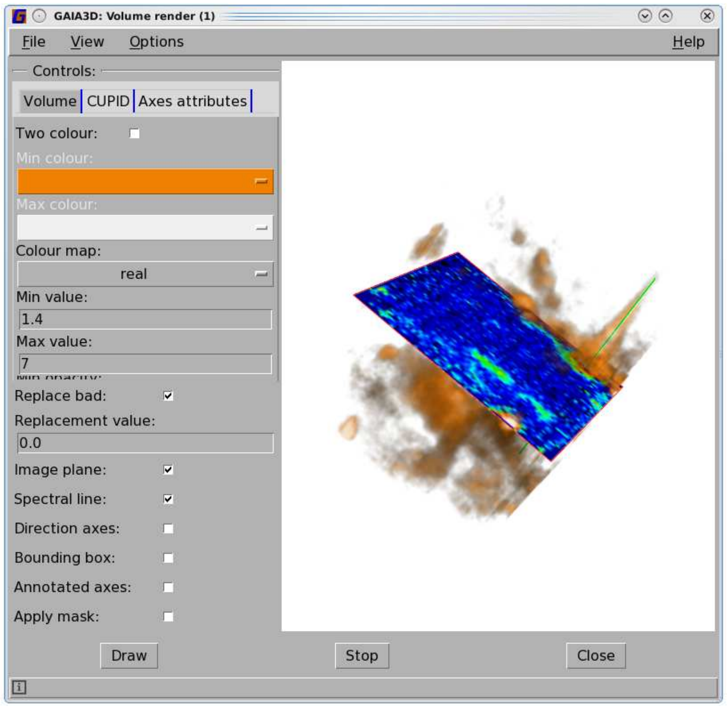
(2) Click and drag the image display to change the orientation in the "`Volume render`" window.

(3) Include axes labelling, the image plane and other features using the check boxes on the side bar.



Input cub
Axis:
Index of plane:
Coordinate of plane: 61.94287
Coordinate type: Radio velocity (USB) (km/s)
Increment: 1
Show coordinate label: ☐

Figure 10.14: Selecting the **Volume rendering** menu in GAIA.



Two colour: ☐
Min colour:
Max colour:
Colour map: real
Min value: 1.4
Max value: 7
Replace bad: ✓
Replacement value: 0.0
Image plane: ✓
Spectral line: ✓
Direction axes: ☐
Bounding box: ☐
Annotated axes: ☐
Apply mask: ☐

Figure 10.15: Three-dimensional visualisation in GAIA.

10.8 Sending spectra to SPLAT

GAIA has very limited analysis functionality for spectra. Whereas SPLAT is a sophisticated graphical spectral-analysis tool. Any spectrum displayed in GAIA (a single-position spectrum or a spectrum averaged over some region 10.5) can be sent (via a protocol called SAMP) to SPLAT for more-detailed spectral analysis.

SPLAT offers the ability to further process, fit, or identify spectral lines. You can also plot different spectra in the same window and make publishable files. The full SPLAT documentation can be found here in **SUN/243**.

Here are some instructions on how to do this.

(1) Start SPLAT, if it is not already running.

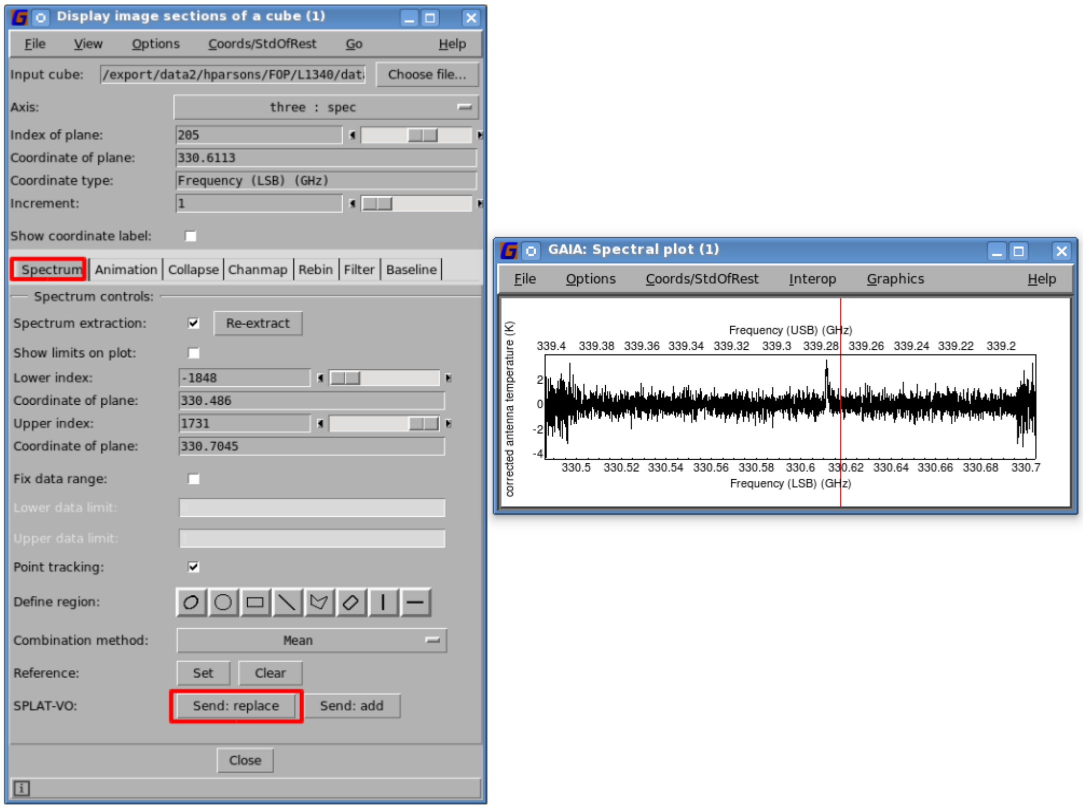
```
% splat &
```

(2) Click on the **Interop** menu item in SPLAT. If the **SAMP control** icon has a red circle, it means that the SAMP communication hub is running. If, however, the icon has a white circle, you need to start a hub. To do this press the **Register with HUB** button at the bottom of the window, and then press the **Start internal hub** button in the window that pops up.

(3) If you had GAIA already running before the hub was active, then in GAIA select the **Interop**>**Register** to register your GAIA with the hub.

(4) Select the spectrum you wish to send from GAIA.

(5) Click on SPLAT-VO **Send: replace** or **Send: add** button near the bottom of the "`Display image sections of a cube`" window. The spectrum should appear in a SPLAT window.



Input cube: /export/data2/hparsons/FOP/L1340/dat
Axis: three : spec
Index of plane: 205
Coordinate of plane: 330.6113
Coordinate type: Frequency (LSB) (GHz)
Increment: 1
Show coordinate label: ☐
Spectrum extraction: ✓
Show limits on plot: ☐
Lower index: -1848
Coordinate of plane: 330.486
Upper index: 1731
Coordinate of plane: 330.7045
Fix data range: ☐
Lower data limit:
Upper data limit:
Point tracking: ✓
Define region:
Combination method: Mean
Reference:
SPLAT-VO:

Figure 10.16: Sending spectrum in GAIA to SPLAT.

SAMP can also transmit images and catalogues between compliant applications. For example, you might have a selected catalogue of sources observed elsewhere in TOPCAT and want to plot the locations over your HARP maps in GAIA.

Chapter 11
Using SPLAT

SPLAT is a graphical spectral-analysis tool. SPLAT offers tools for further processing, fitting, or identification of spectral lines. One can also plot different spectra in the same window and make publishable files. The full SPLAT documentation can be found in **SUN/243**.

11.1 Opening a spectrum in SPLAT

There are two options for opening up a spectrum in SPLAT. First it is possible to send a spectrum to SPLAT using GAIA as described in Section 10.8.

Alternatively spectra can be displayed in directly in SPLAT using:

```
% splat filename &
```

for a single position cube or

```
% splat 'filename(xpos,ypos,)' &
```

to display the spectrum at the pixel position (xpos,ypos). Or simply initialise SPLAT and open the file directly from the gui.

```
% splat  &
```

When SPLAT is initially started you will get a main window appear and also a spectral window.

11.2 Display synopsis of spectrum

When you initially open a specrum in SPLAT the synopsis of that spectrum will automatically be overlaid on the spectrum displayed, if the metadata are known to SPLAT. The following are a list of possible properties than can be displayed, with a brief description. Names in parenthesis are either the FITS keywords or AST attributes used to obtain the values.

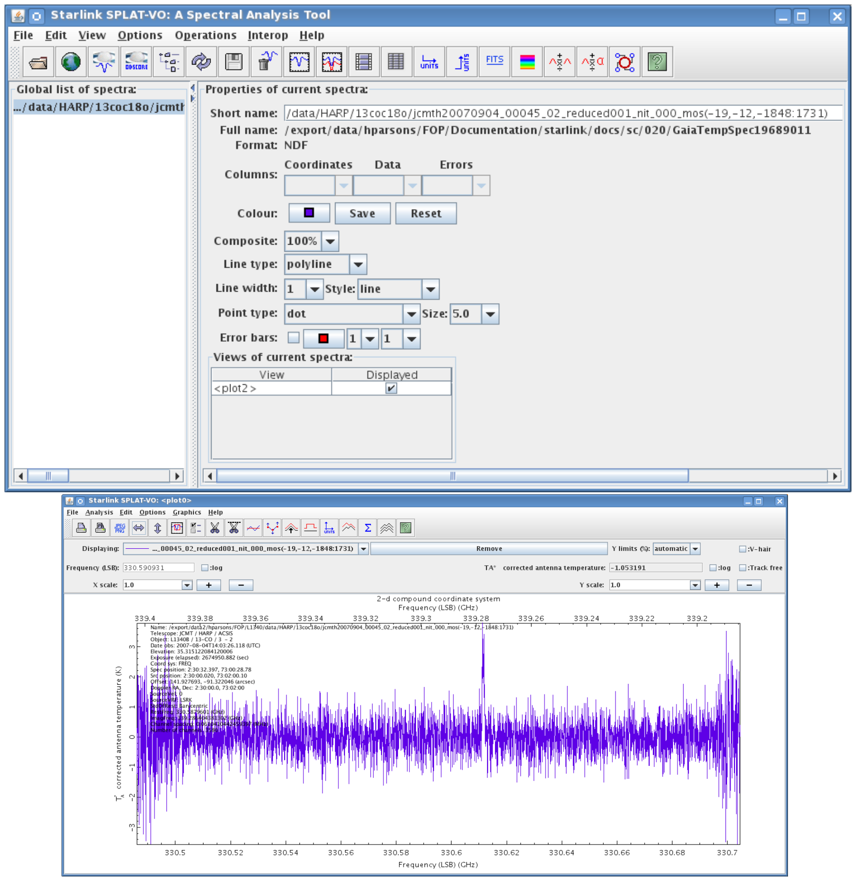
(1) Name: the short name

(2) Telescope: telescope, instrument and its backend (TELESCOP/INSTRUME/BACKEND)

(3) Object: target, molecule and molecular transition being observed (OBJECT/MOLECULE/TRANSITI)

(4) Date obs: date of the observation (DATE-OBS or Epoch in UTC)

(5) Elevation: observatory elevation (ELSTART or 0.5∗(ELSTART+ELEND))


Starlink SPLAT-VO: A Spectral Analysis Tool
File Edit View Options Operations Interop Help
Global list of spectra:
Properties of current spectra:
Save
Reset
Starlink SPLAT-VO: <plot0>
File Analysis Edit Options Graphics Help
Remove
2-d compound coordinate system
Frequency (LSB) (GHz)
T_A^* corrected antenna temperature (K)

Global list of spectra: .../data/HARP/13coc18o/jcmth

Short name: /data/HARP/13coc18o/jcmth20070904_00045_02_reduced001_nit_000_mos(-19,-12,-1848:1731)
Full name: /export/data/hparsons/FOP/Documentation/starlink/docs/sc/020/GaiaTempSpec19689011
Format: NDF
Columns: Coordinates Data Errors
Composite: 100%
Line type: polyline
Line width: 1 Style: line
Point type: dot Size: 5.0
Error bars: 1 1

Views of current spectra:

View	Displayed
<plot2>	✓

Displaying: ..._00045_02_reduced001_nit_000_mos(-19,-12,-1848:1731)
Y limits (%): automatic
Frequency (LSB): 330.590931
TA* corrected antenna temperature: -1.053191
X scale: 1.0
Y scale: 1.0

Name: /export/data2/hparsons/FOP/L1340/data/HARP/13coc18o/jcmth20070904_00045_02_reduced001_nit_000_mos(-19,-12,-1848:1731)
Telescope: JCMT / HARP / ACSIS
Object: L1340B / 13-CO / 3 - 2
Date obs: 2007-08-04T14:03:26.118 (UTC)
Elevation: 35.315122084120006
Exposure (elapsed): 2674950.882 (sec)
Coord sys: FREQ
Spec position: 2:30:32.397, 73:00:28.78
Src position: 2:30:00.020, 73:02:00.10
Offset: 141.927693, -91.322046 (arcsec)
Doppler RA, Dec: 2:30:00.0, 73:02:00
SourceVel: 0
SourceVRF: LSRK
StdOfRest: Barycentric
RestFreq: 330.5879601 (GHz)

Figure 11.1: The main SPLAT window and spectral window.

(6) Exposure: exposure time for the spectrum (EXTIME)

(7) Exposure (median): exposure time for the spectrum (EXP_TIME)

(8) Exposure (elapsed): the time elapsed during exposure (INT_TIME otherwise DATE-END − DATE-OBS)

(9) Exposure (effective): effective exposure time (EXEFFT)

(10) Coords sys: the co-ordinate system code (System)

(11) Spec position: position of spectrum on the sky (EXRAX, EXDECX)

(12) Src position: position of the observation centre (EXRRA, EXRDEC)

(13) Img centre: centre of the originating image (EXRA, EXDEC)

(14) Offset: offset of spectrum from the observation centre (EXRRAOF, EXRDECOF or EXRAOF, EXDECOF)

(15) Doppler RA Dec: reference position (RefRA,RefDec)

(16) SourceVel: source velocity (SourceVel)

(17) SourceVRF: source velocity reference frame (SourceVRF)

(18) SourceSys: system of the source velocity (SourceSys)

(19) StdOfRest: standard of rest (StdOfRest)

(20) RestFreq: rest frequency (RestFreq

(21) ImageFreq: image sideband equivalent of the rest frequency (ImagFreq)

(22) Channel spacing: channel spacing (derived value)

(23) Number of channels: Number of co-ordinate positions

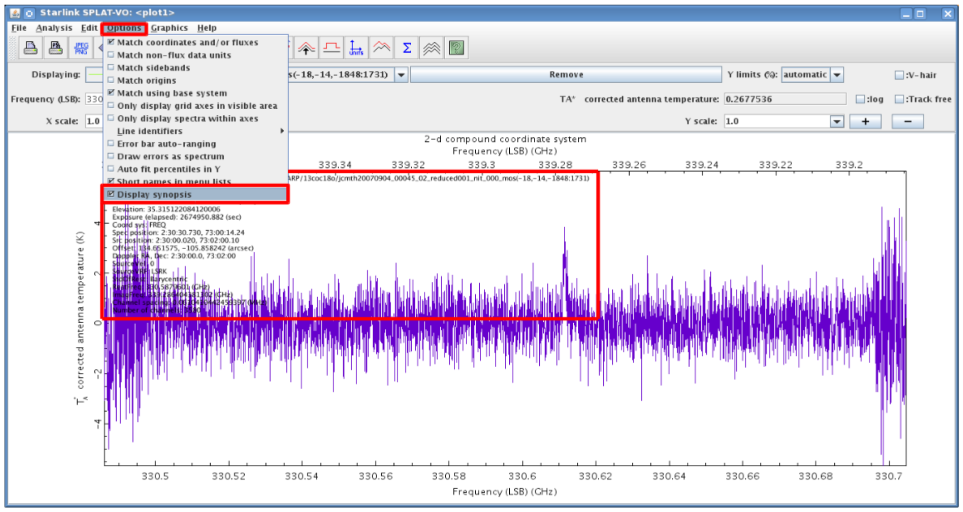
(24) TSYS: system temperature (`TSYS`)

(25) TSYS (median): median system temperature (`MEDTSYS`)

(26) TSYS (est): system temperature (derived value)

(27) TRX: receiver temperature (`TRX`)

To show or remove the synopsis of a spectrum simply click on the **Options** button in the spectral view window and select or deselect **Display Synopsis**. To move the position of the synopsis simply click and drag the text box to the desired location.


Starlink SPLAT-VO: <plot1>
File
Analysis
Edit
Options
Graphics
Help
☑ Match coordinates and/or fluxes
☐ Match non-flux data units
☐ Match sidebands
☐ Match origins
☑ Match using base system
☐ Only display grid axes in visible area
☐ Only display spectra within axes
Line identifiers
☐ Error bar auto-ranging
☐ Draw errors as spectrum
☐ Auto fit percentiles in Y
☑ Short names in menu lists
☑ Display synopsis
Displaying:
Remove
Frequency (LSB):
X scale:
Y limits:
automatic
Y scale:
:log
:V-hair
:Track free
2-d compound coordinate system
Frequency (LSB) (GHz)
T*A corrected antenna temperature (K)
Frequency (LSB) (GHz)

```
Elevation: 35.315122084120006
Exposure (elapsed): 2674950.882 (sec)
Coord sys: FREQ
Spec position: 2:30:30.730, 73:00:14.24
Src position: 2:30:00.020, 73:02:00.10
Offset: 134.651575, -105.858242 (arcsec)
Doppler RA, Dec: 2:30:00.0, 73:02:00
SourceVel: 0
SourceVRF: LSRK
StdOfRest: Barycentric
RestFreq: 330.5879601 (GHz)
ImagFreq: 339.28 [illegible] (GHz)
Channel spacing: [illegible]
Number of channels: [illegible]
```

Figure 11.2: How to display or remove the synopsis of a spectrum in the spectral window.

11.3 Changing units of a spectrum in SPLAT

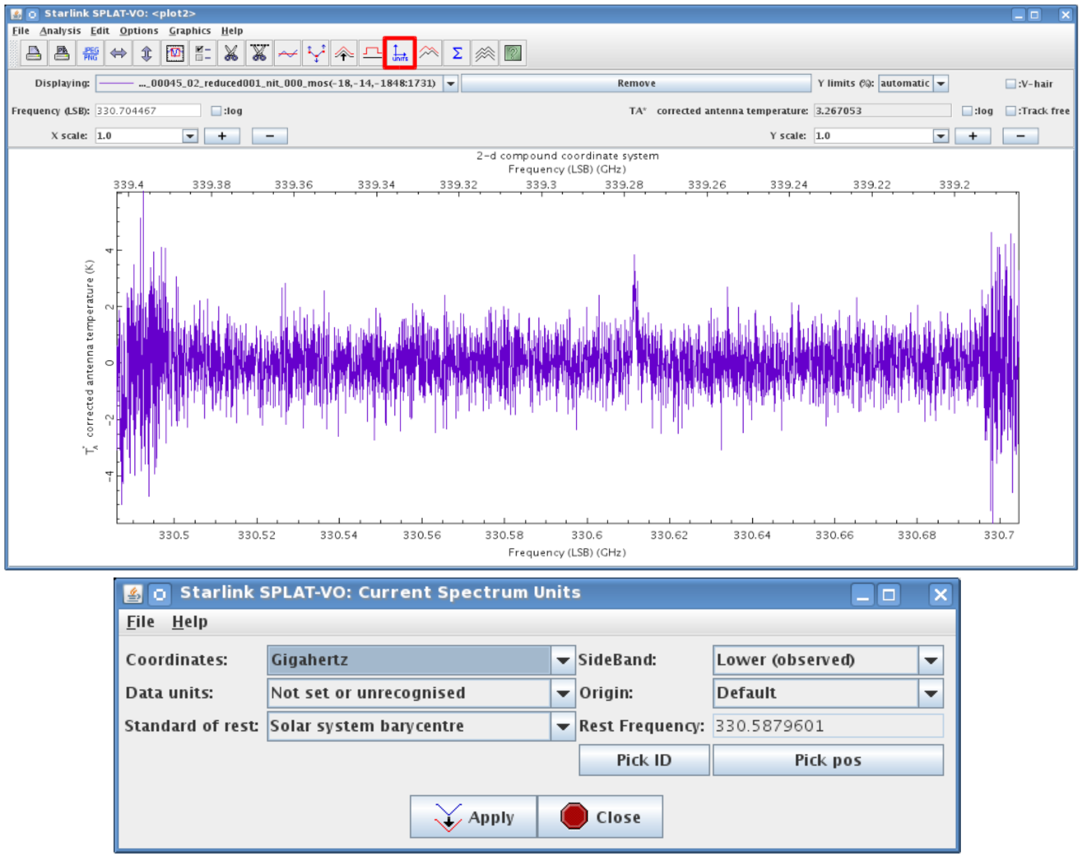
Sometime you may wish to change the units of a spectrum in SPLAT.

(1) Click on **Change the units of the current spectrum** button in your current spectral window.

(2) A new window will appear called "`Current Spectrum Units`".

(3) From the Coordinates drop-down menu select the desired unit (in this case we select **Kilometers-per-sec (radio)**). Then click **Apply**.

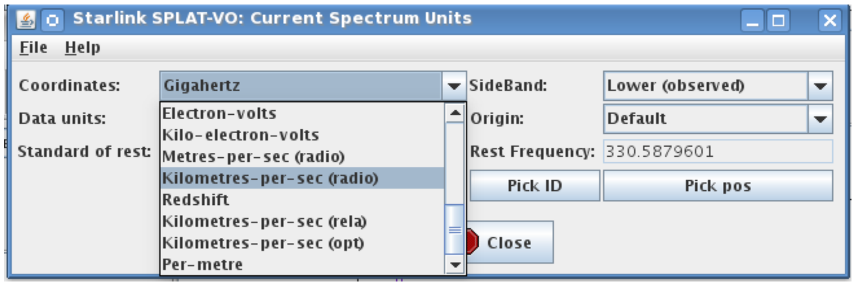
(4) You will now see that the spectrum is displayed in **Radio Velocity (km/s)**.



Displaying: ..._00045_02_reduced001_nit_000_mos(-18,-14,-1848:1731)
Y limits (%): automatic
:V-hair
Frequency (LSB): 330.704467
:log
TA* corrected antenna temperature: 3.267053
:log
:Track free
X scale: 1.0
Y scale: 1.0

Coordinates: Gigahertz
SideBand: Lower (observed)
Data units: Not set or unrecognised
Origin: Default
Standard of rest: Solar system barycentre
Rest Frequency: 330.5879601

Figure 11.3: Top: Select the **Change the units of the current spectrum** button. Note that in this example the current units are Frequency. Bottom: the "`Current Spectrum Units`" window.



Coordinates: Gigahertz
SideBand: Lower (observed)
Data units:
Origin: Default
Standard of rest:
Rest Frequency: 330.5879601

Electron-volts
Kilo-electron-volts
Metres-per-sec (radio)
Kilometres-per-sec (radio)
Redshift
Kilometres-per-sec (rela)
Kilometres-per-sec (opt)
Per-metre

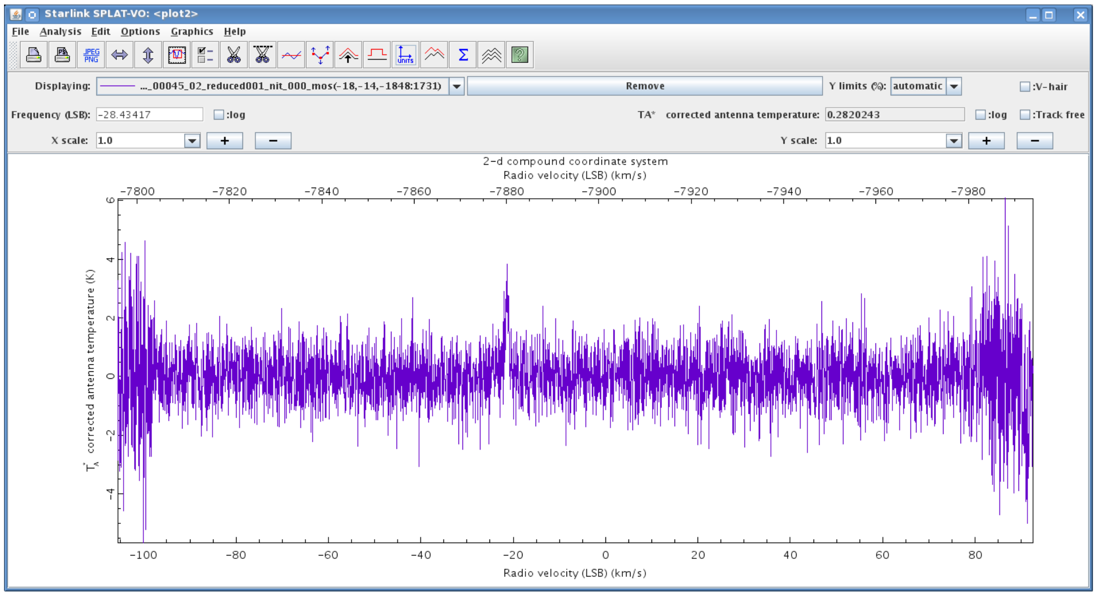
Figure 11.4: Changing the units of the spectrum via the drop-down menu to $\mathrm{km\,s^{-1}}$.

11.4 Cropping a spectrum in SPLAT

(1) Click on **Cut out the regions of the current spectrum** button above the spectrum you wish to work on. This will open a new window.

(2) Select **Add** from the cut region window.

(3) Click and drag the regions you wish to remove from the spectrum in the spectrum window. As we had a second region to remove we click on the **Add** button a second time. In this example we select


Starlink SPLAT-VO: <plot2>
File Analysis Edit Options Graphics Help
Remove
2-d compound coordinate system
Radio velocity (LSB) (km/s)
-7800 -7820 -7840 -7860 -7880 -7900 -7920 -7940 -7960 -7980
6 4 2 0 -2 -4
T_A^* corrected antenna temperature (K)
-100 -80 -60 -40 -20 0 20 40 60 80
Radio velocity (LSB) (km/s)

Displaying: ..._00045_02_reduced001_nit_000_mos(-18,-14,-1848:1731)
Y limits (%): automatic
:V-hair
Frequency (LSB): -28.43417 :log
TA* corrected antenna temperature: 0.2820243 :log :Track free
X scale: 1.0
Y scale: 1.0

Figure 11.5: Spectrum with units of km s^{-1}.

the two noisy edge regions from the spectrum to remove.

(4) Select the **Remove Selected** button.

(5) A new spectrum with the noisy edges removed will now appear in the main SPLAT window. Click on your newly created spectrum to view.

11.5 Rebinning a spectrum in SPLAT

(1) Click on **Apply a filter to the current spectrum** button above the spectrum you wish to work on. This will open a new window.

(2) Select the number of channels over which you want to bin—this is the **Width**. In this example the spectrum has a resolution of 0.055 km s^{-1}. We bin by 18 channels to rebin the spectrum to 1 km s^{-1} channels.

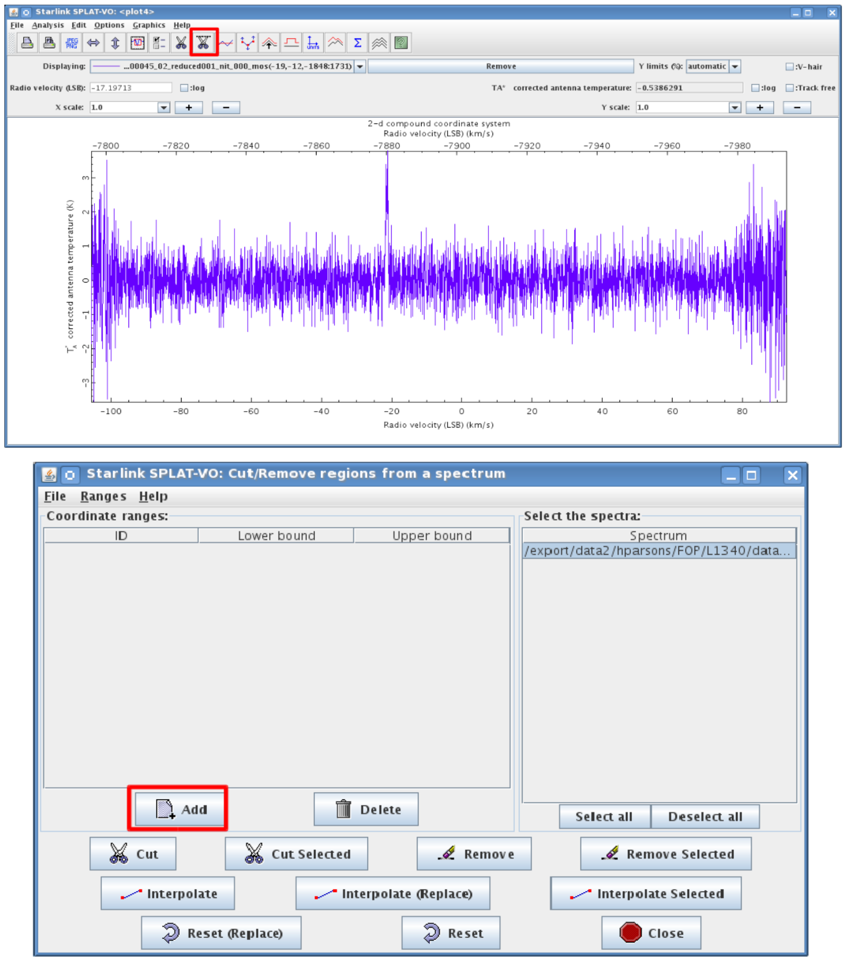
(3) Select **Filter (Replace)**. This will create a new spectrum in the main SPLAT window and will also replace the existing spectrum displayed in the spectral window with the new binned spectrum.

11.6 Estimating the noise in a spectrum using SPLAT

(1) Click on **Get statistics on region of spectrum** button above the spectrum you wish to work on. This will open a new window.

(2) Select **Add** from the "`Region statistics`" window.

(3) Click and drag the regions you calculate the statistics over. As we wish to include baselines eithersie of the spectral line seen in the data we click on the **Add** button a second time.


Starlink SPLAT-VO: <plot4>
File Analysis Edit Options Graphics Help
2-d compound coordinate system
Radio velocity (LSB) (km/s)
T_A^* corrected antenna temperature (K)
Starlink SPLAT-VO: Cut/Remove regions from a spectrum
File Ranges Help
Add
Delete
Select all
Deselect all
Cut
Cut Selected
Remove
Remove Selected
Interpolate
Interpolate (Replace)
Interpolate Selected
Reset (Replace)
Reset
Close

Displaying: ...00045_02_reduced001_nit_000_mos(-19,-12,-1848:1731)
Y limits (%): automatic
Radio velocity (LSB): -17.19713
TA* corrected antenna temperature: -0.5386291
X scale: 1.0
Y scale: 1.0

Coordinate ranges:

ID	Lower bound	Upper bound

Select the spectra:

Spectrum
/export/data2/hparsons/FOP/L1340/data...

Figure 11.6: Top: Select the cut button. Bottom: Select the **Add** button.

(4) You will see the two selected spectral regions appear in a table in the "`Region statistics`" window with some basic statistic already computed for each region created.

(5) Click on **Selected stats** or **All stats** to get statistics of the regions combined. In this example we see the standard deviation in the regions selected is 0.19K.

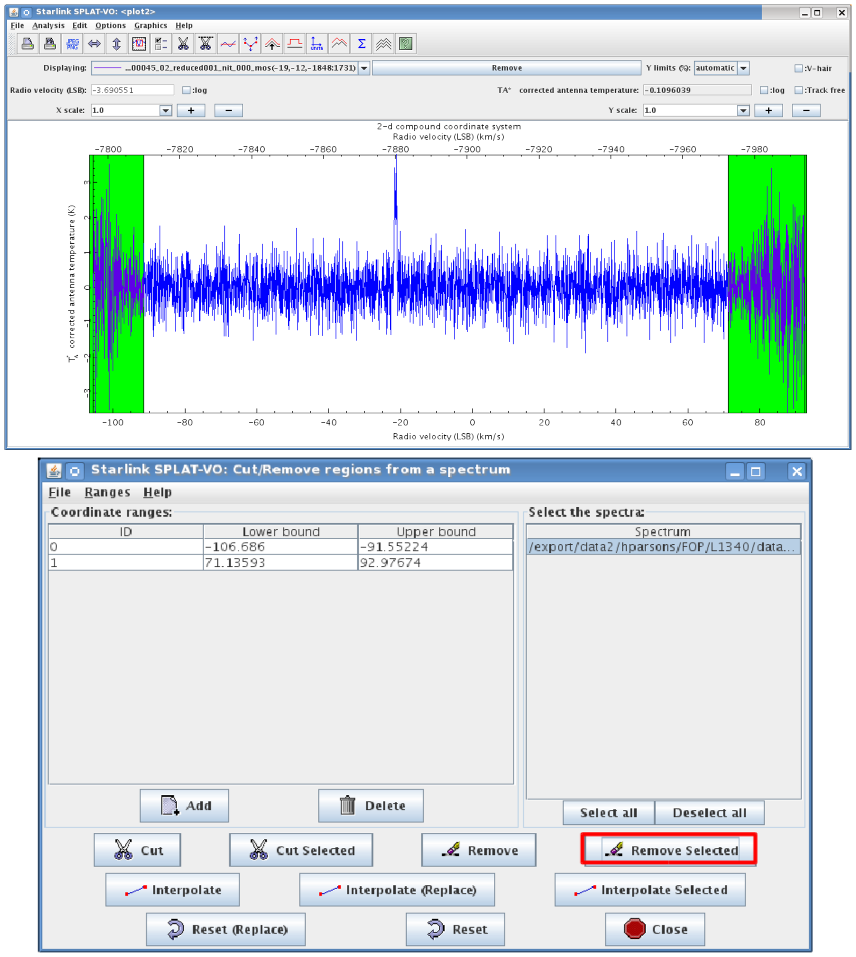
11.7 Fitting a line in a spectrum using SPLAT

Your data might have lines for which you want a quick estimate of its line strength and width.

(1) Click on **Fit spectral lines using a variety of functions** button above the spectrum you wish to work on. This will open a new window.

(2) Click on the **Add** button in the "`Measure spectral lines`" window.

(3) Click and drag the box that appears in the spectral window to encompass the line for which you wish to calculate a fit for.


Starlink SPLAT-VO: <plot2>
File Analysis Edit Options Graphics Help
2-d compound coordinate system
Radio velocity (LSB) (km/s)
corrected antenna temperature (K)
Radio velocity (LSB) (km/s)
Starlink SPLAT-VO: Cut/Remove regions from a spectrum
File Ranges Help
Add
Delete
Select all
Deselect all
Cut
Cut Selected
Remove
Remove Selected
Interpolate
Interpolate (Replace)
Interpolate Selected
Reset (Replace)
Reset
Close

Displaying: ...00045_02_reduced001_nit_000_mos(-19,-12,-1848:1731) Remove Y limits (%): automatic :V-hair

Radio velocity (LSB): -3.690551 :log TA* corrected antenna temperature: -0.1096039 :log :Track free

X scale: 1.0 Y scale: 1.0

Coordinate ranges:

ID	Lower bound	Upper bound
0	-106.686	-91.55224
1	71.13593	92.97674

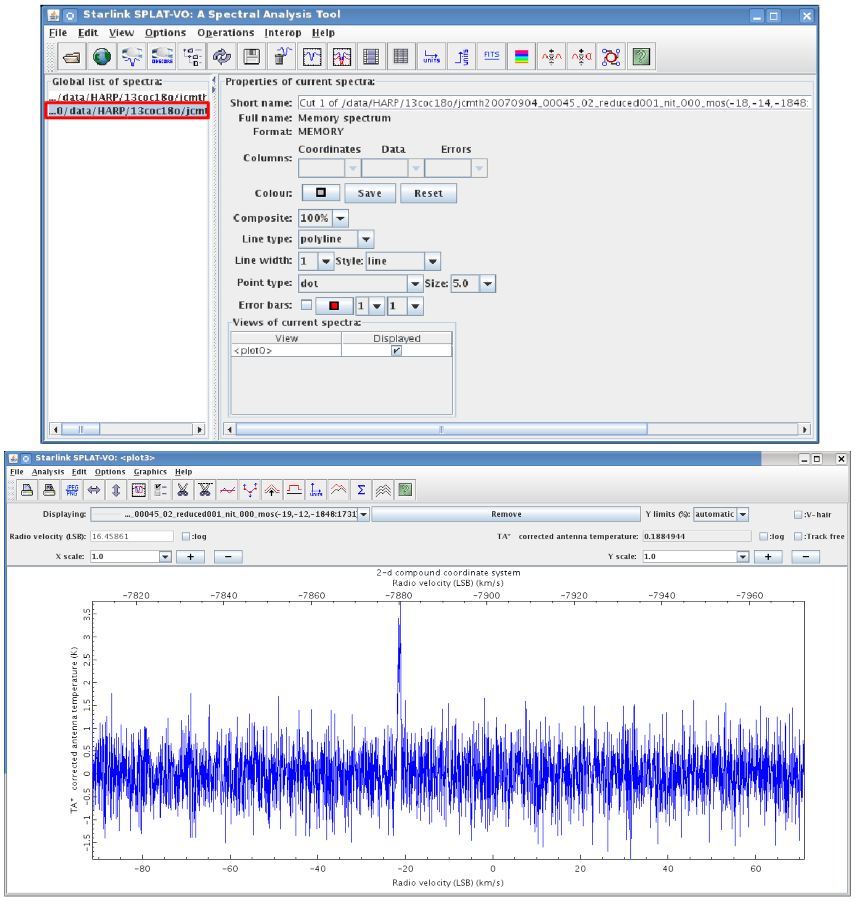
Select the spectra:

Spectrum
/export/data2/hparsons/FOP/L1340/data...

Figure 11.7: Top: The two regions selected. Bottom: remove the selected region by clicking the **Remove Selected** button.

(**4**) Click on the **Fit** button to produce both a **Quick** and **Gaussian** fit to the spectral line.

(**5**) We see the results of the fit are provided in the "`Measure spectral lines`" window, and the two fits can be seen in the spectral line window.

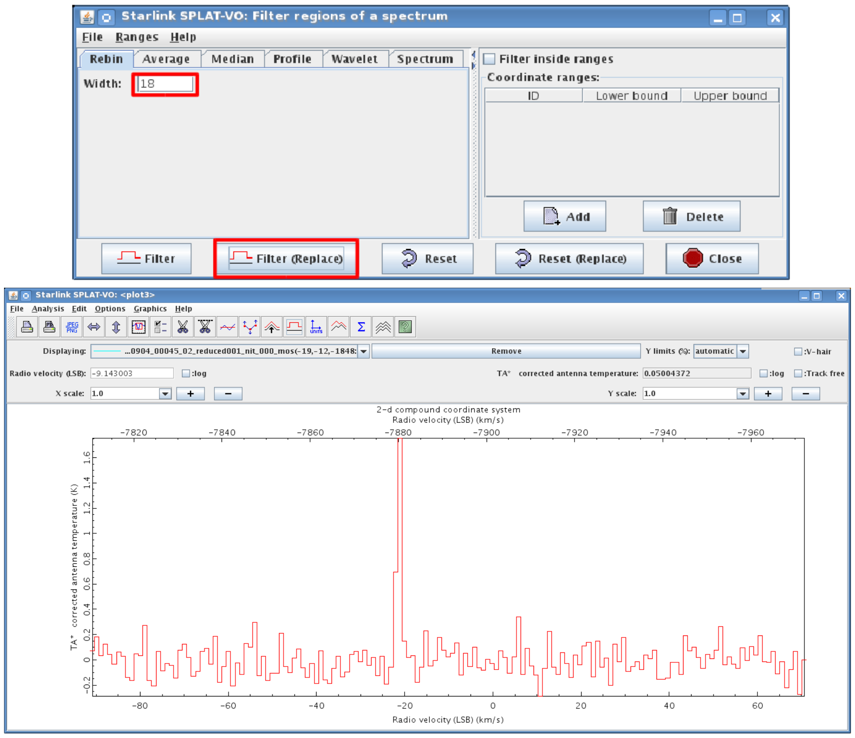

Starlink SPLAT-VO: A Spectral Analysis Tool
File Edit View Options Operations Interop Help
Global list of spectra:
.../data/HARP/13coc18o/jcmth
...0/data/HARP/13coc18o/jcmt
Properties of current spectra:
Colour: Save Reset
Views of current spectra:
Starlink SPLAT-VO: <plot3>
File Analysis Edit Options Graphics Help
Remove
:V-hair
:Track free
2-d compound coordinate system
Radio velocity (LSB) (km/s)
TA* corrected antenna temperature (K)

Short name: Cut 1 of /data/HARP/13coc18o/jcmth20070904_00045_02_reduced001_nit_000_mos(-18,-14,-1848:
Full name: Memory spectrum
Format: MEMORY
Columns: Coordinates Data Errors
Composite: 100%
Line type: polyline
Line width: 1 Style: line
Point type: dot Size: 5.0
Error bars: ☐ 🟥 1 1

View	Displayed
<plot0>	✓

Displaying: ..._00045_02_reduced001_nit_000_mos(-19,-12,-1848:1731
Y limits (%): automatic
Radio velocity (LSB): 16.45861 ☐ :log
TA* corrected antenna temperature: 0.1884944 ☐ :log
X scale: 1.0
Y scale: 1.0

Figure 11.8: Top: Select the newly created spectrum in the main SPLAT window. The spectrum with the selected noisy-edge-data removed will then appear.


Starlink SPLAT-VO: Filter regions of a spectrum
File Ranges Help
Rebin Average Median Profile Wavelet Spectrum
Filter inside ranges
Coordinate ranges:
ID Lower bound Upper bound
Add Delete
Filter Filter (Replace) Reset Reset (Replace) Close
Starlink SPLAT-VO: <plot3>
File Analysis Edit Options Graphics Help
Remove
:V-hair :log :Track free
2-d compound coordinate system
Radio velocity (LSB) (km/s)
TA* corrected antenna temperature (K)

Width: 18

Displaying: ...0904_00045_02_reduced001_nit_000_mos(-19,-12,-1848:

Y limits (%): automatic

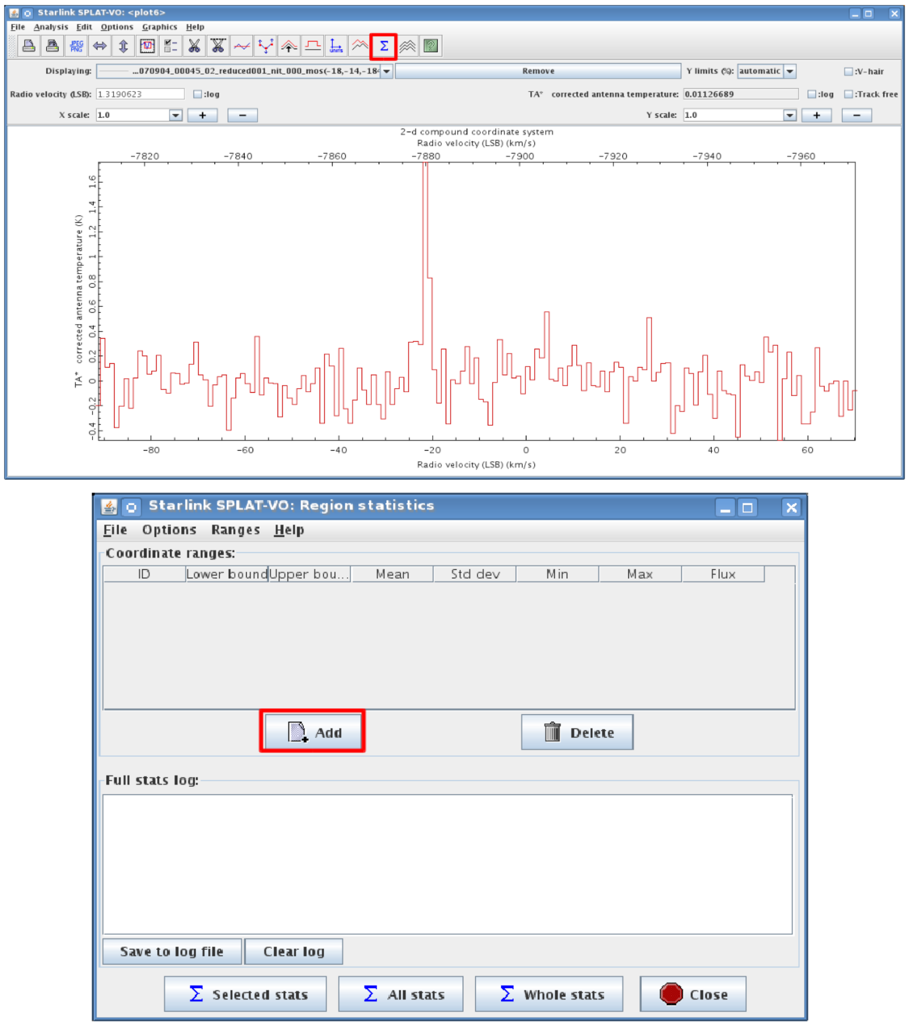
Radio velocity (LSB): -9.143003

TA* corrected antenna temperature: 0.05004372

X scale: 1.0

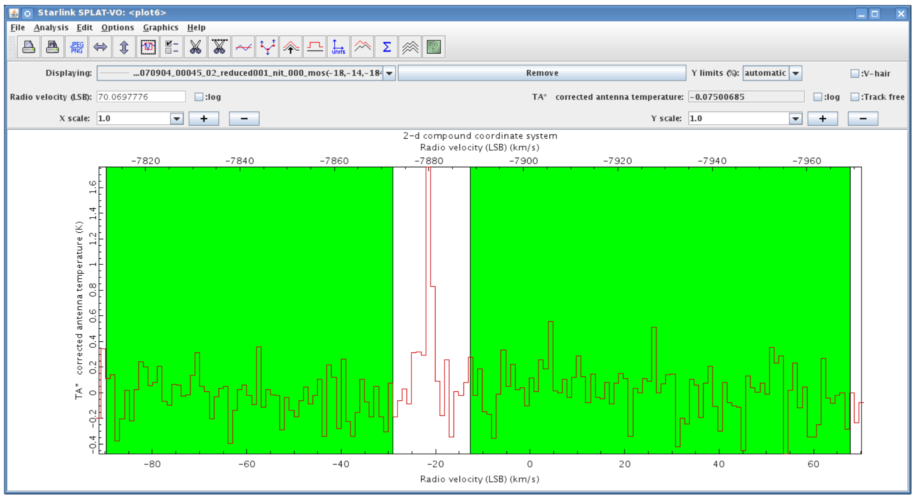
Y scale: 1.0

Figure 11.9: Top: The SPLAT "`Filter regions of a spectrum`" window. Bottom: The final rebinned spectrum.


Starlink SPLAT-VO: <plot6>
File Analysis Edit Options Graphics Help
Remove
2-d compound coordinate system
Radio velocity (LSB) (km/s)
TA* corrected antenna temperature (K)
Starlink SPLAT-VO: Region statistics
File Options Ranges Help
Coordinate ranges:
ID Lower bound Upper bou... Mean Std dev Min Max Flux
Add
Delete
Full stats log:
Save to log file
Clear log
Selected stats
All stats
Whole stats
Close

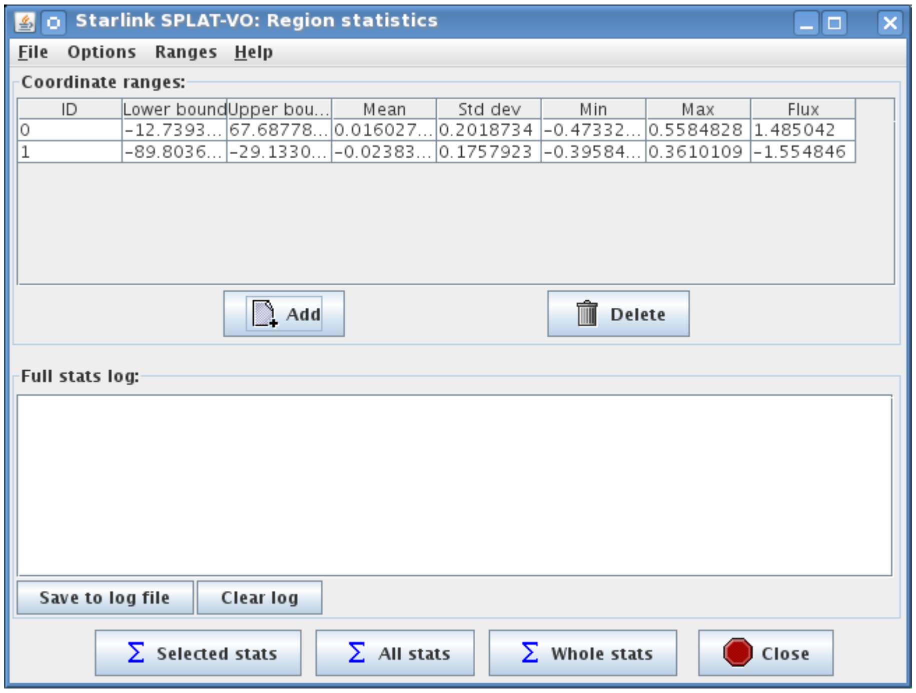
Displaying: ...070904_00045_02_reduced001_nit_000_mos(-18,-14,-18
Y limits (%): automatic
Radio velocity (LSB): 1.3190623
TA* corrected antenna temperature: 0.01126689
X scale: 1.0
Y scale: 1.0

Figure 11.10: Top: Select the statistics button. Bottom: Select the **Add** button.


Starlink SPLAT-VO: <plot6>
File Analysis Edit Options Graphics Help
Remove
2-d compound coordinate system
Radio velocity (LSB) (km/s)
TA* corrected antenna temperature (K)

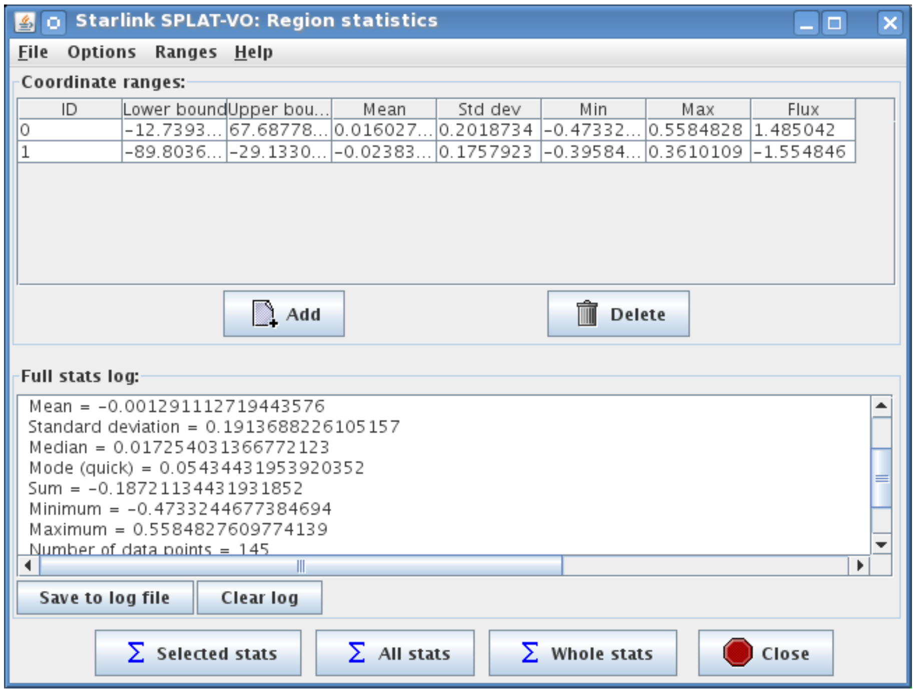
Displaying: ...070904_00045_02_reduced001_nit_000_mos(-18,-14,-18
Y limits (%): automatic
Radio velocity (LSB): 70.0697776
TA* corrected antenna temperature: -0.07500685
X scale: 1.0
Y scale: 1.0

Figure 11.11: Click and drag the regions over which you wish to estimate the noise.



ID	Lower bound	Upper bou...	Mean	Std dev	Min	Max	Flux
0	-12.7393...	67.68778...	0.016027...	0.2018734	-0.47332...	0.5584828	1.485042
1	-89.8036...	-29.1330...	-0.02383...	0.1757923	-0.39584...	0.3610109	-1.554846

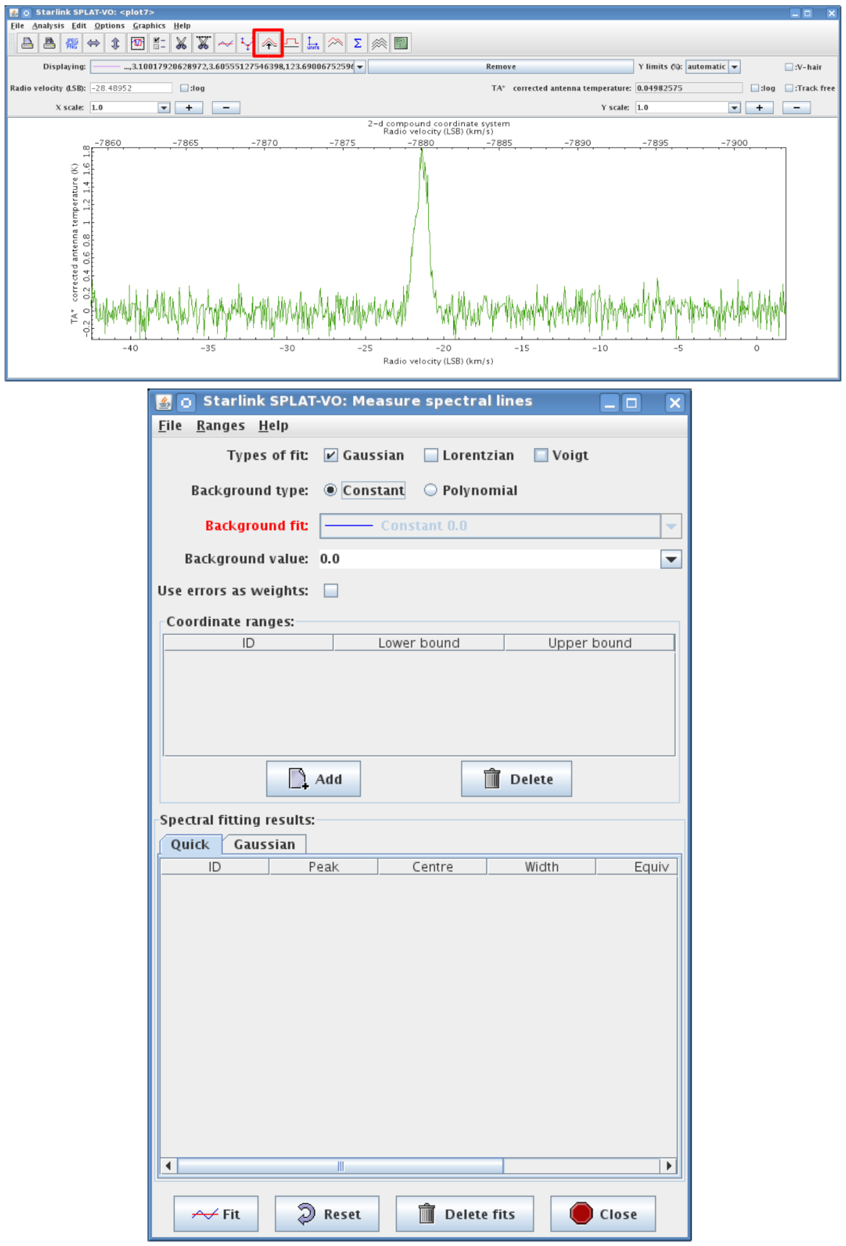
Figure 11.12: Two spectral regions have been selected and are displayed in the "`Region statistics`" window.



ID	Lower bound	Upper bou...	Mean	Std dev	Min	Max	Flux
0	-12.7393...	67.68778...	0.016027...	0.2018734	-0.47332...	0.5584828	1.485042
1	-89.8036...	-29.1330...	-0.02383...	0.1757923	-0.39584...	0.3610109	-1.554846

```
Mean = -0.001291112719443576
Standard deviation = 0.1913688226105157
Median = 0.017254031366772123
Mode (quick) = 0.054344319539203552
Sum = -0.18721134431931852
Minimum = -0.4733244677384694
Maximum = 0.558482760977413 9
Number of data points = 145
```

Figure 11.13: Statistics for both regions are displayed in the "`Region statistics`" window.


Starlink SPLAT-VO: <plot7>
File Analysis Edit Options Graphics Help
Remove
X scale: + −
Y scale: + −
2-d compound coordinate system
Radio velocity (LSB) (km/s)
TA* corrected antenna temperature (K)
Starlink SPLAT-VO: Measure spectral lines
File Ranges Help
Coordinate ranges:
Add
Delete
Spectral fitting results:
Quick
Gaussian
Fit
Reset
Delete fits
Close

Displaying: ...,3.10017920628972,3.60555127546398,123.6900675259(

Y limits (%): automatic

☐ :V-hair

Radio velocity (LSB): -28.48952 ☐ :log

TA* corrected antenna temperature: 0.04982575 ☐ :log ☐ :Track free

X scale: 1.0

Y scale: 1.0

Types of fit: ☑ Gaussian ☐ Lorentzian ☐ Voigt

Background type: ● Constant ○ Polynomial

Background fit: Constant 0.0

Background value: 0.0

Use errors as weights: ☐

ID	Lower bound	Upper bound

ID	Peak	Centre	Width	Equiv

Figure 11.14: Top: Select the spectral-fitting button. Bottom: Click on the **Add** button.

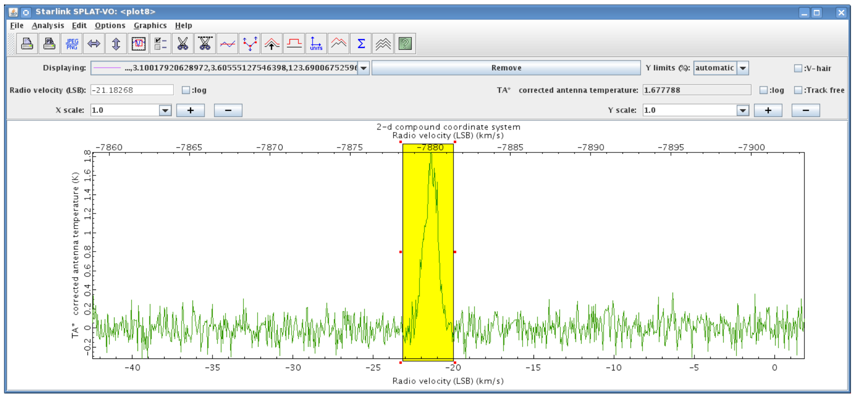
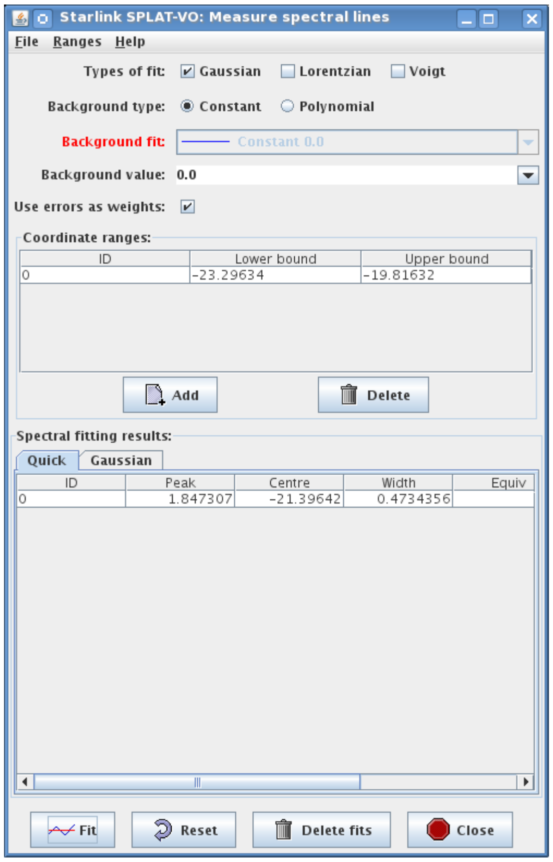

Starlink SPLAT-VO: <plot8>
File Analysis Edit Options Graphics Help
Displaying: ...,3.10017920628972,3.60555127546398,123.69006752596
Remove
Y limits (%): automatic
:V-hair
Radio velocity (LSB): -21.18268
:log
TA* corrected antenna temperature: 1.677788
:log
:Track free
X scale: 1.0
Y scale: 1.0
2-d compound coordinate system
Radio velocity (LSB) (km/s)
-7860 -7865 -7870 -7875 -7880 -7885 -7890 -7895 -7900
TA* corrected antenna temperature (K)
-0.2 0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8
-40 -35 -30 -25 -20 -15 -10 -5 0
Radio velocity (LSB) (km/s)

Figure 11.15: Click and drag the region containing the line which you wish to be fitted.


Starlink SPLAT-VO: Measure spectral lines

File Ranges Help

Types of fit: [✓ Gaussian] Lorentzian Voigt

Background type: [✓ Constant] Polynomial

Background fit: Constant 0.0

Background value: 0.0

Use errors as weights: ✓

Coordinate ranges:

ID	Lower bound	Upper bound
0	-23.29634	-19.81632

Add Delete

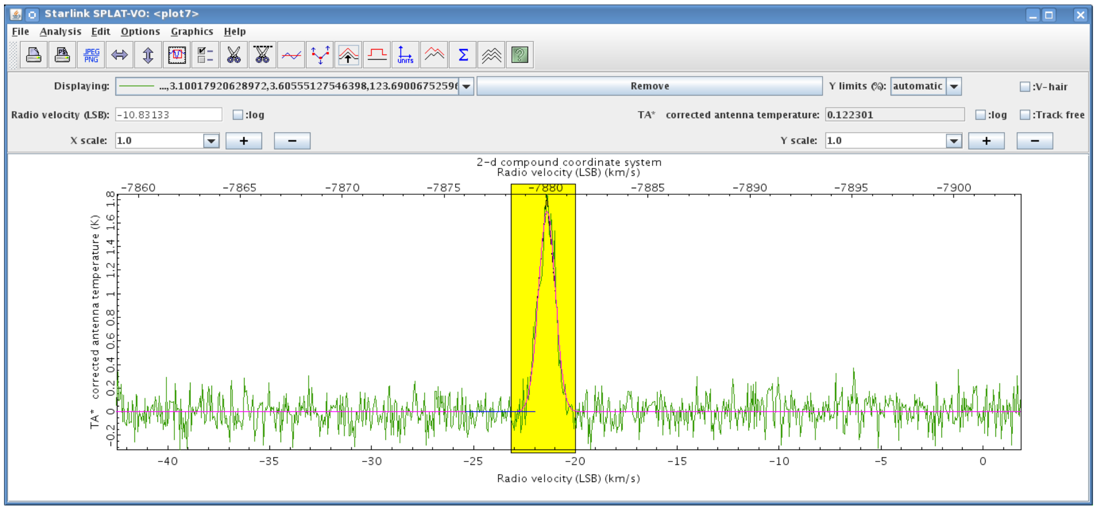
Spectral fitting results:

Quick Gaussian

ID	Peak	Centre	Width	Equiv
0	1.847307	-21.39642	0.4734356	

Fit Reset Delete fits Close

Figure 11.16: In the "`Measure spectral lines`" window, we see the both the **Quick** fit (black dashed line) and the **Gaussian** fit (pink line) of the spectral line.


Starlink SPLAT-VO: <plot7>
File Analysis Edit Options Graphics Help
Remove
+
−
2-d compound coordinate system
Radio velocity (LSB) (km/s)
−7860 −7865 −7870 −7875 −7880 −7885 −7890 −7895 −7900
TA* corrected antenna temperature (K)
−0.2 0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8
−40 −35 −30 −25 −20 −15 −10 −5 0
Radio velocity (LSB) (km/s)

Displaying: ...,3.10017920628972,3.60555127546398,123.69006752596
Y limits (%): automatic
:V-hair
Radio velocity (LSB): -10.83133
:log
TA* corrected antenna temperature: 0.122301
:log
:Track free
X scale: 1.0
Y scale: 1.0

Figure 11.17: In the spectral window we see the both the **Quick** fit (black dashed line) and the **Gaussian** fit (pink line) of the spectral line.

Chapter 12
Getting your Data from CADC

The JCMT Science Archive is hosted by The Canadian Astronomy Data Centre (CADC). Both raw data and data processed by the science pipeline are made available to PIs and co-Is through the CADC Advanced Search interface linked from the JCMT landing page (`https://www.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/en/jcmt/`).

To access proprietary data you will need to have your CADC username registered by the EAO and thereby associated with the project code.

When searching the JCMT Science Archive, be sure to select the correct search option from the 'JSA Queries' tab. Here you can select public versus proprietary, raw versus reduced, and SCUBA-2 versus ACSIS data.

The format and file naming conventions are different in the JSA for *reduced* data compared with merely running ORAC-DR. Reduced observations will be in FITS format with the `.fits` file extension. File names begin with the `jcmth` prefix followed by the UT date, observation number, subsystem number, the product name and subscan number, the grouping algorithm, and finally a version number (in practice zero), with adjacent elements separated by an underscore. The product name of most interest will be `reduced`, but others include `rimg` and `rsp` for the representative image and spectrum. The grouping algorithm is `obs` for a single observation, and `nit` where all the good data for an object observed with

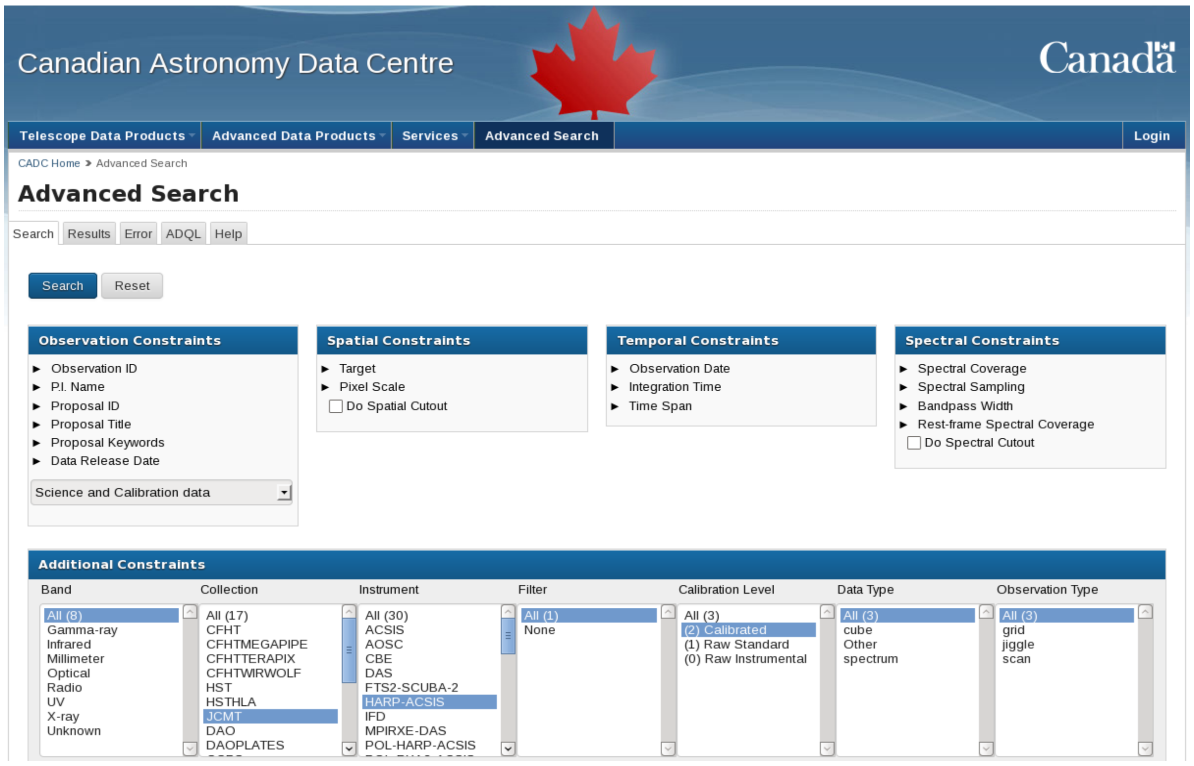


Figure 12.1: Screenshot of the CADC search window for reduced HARP data.

the same instrumental setup during a night are combined to create products of better quality. A further grouping is `pro` where matching observations within a project are combined, which may extend over many nights spanning years.

Here is are some examples: `jcmth20070728_00066_01_reduced002_obs_000.fits`, `jcmth20070728_00066_01_rimg001_nit_000.fits`, and `jcmth20070728_00066_02_rsp001_pro_000.fits`.

For a detailed guide to using the Advanced Search see JSA User's Guide.

Bibliography

[1] Berry D. S., 2013, *CUPID. Version 2.0*, Starlink User Note 255 1.3

[2] Berry D. S., 2015, *FellWalker—a clump-identification algorithm*, Astronomy and Computing, 10, 22 (DOI:10.1016/j.ascom.2014.11.004) E

[3] Buckle, J. V., Hills, R. E., Smith, H., et al., 2009, *HARP/ACSIS: a submillimetre spectral imaging system on the James Clerk Maxwell Telescope*, MNRAS, 399, 1026 (DOI:10.1111/j.1365-2966.2009.15347.x) 3.2.5

[4] Cavanagh B., Jenness T., Economou F., Currie M. J., 2008, *The ORAC-DR data reduction pipeline*, Astron. Nactr., 329, 295 (DOI:10.1002/asna.200710944) 1.2, 1.3, 5

[5] Chapin E. L., et al., 2013, *SMURF – Sub-Millimetre User Reduction Facility*, Starlink User Note 258 1.3

[6] Currie M. J., Privett G. J., Chipperfield A. J., Berry D. S. and Davenhall A.C., 2013, *CONVERT – A Format-conversion Package*, Starlink User Note 55 1.3

[7] Currie M. J., Wallace P. T., Warren-Smith R. F., 1989, *Starlink Standard Data Structures*, Starlink General Paper 38.2 2.1

[8] Currie M. J., Berry D. S, 2013, *KAPPA – Kernel Application Package*, Starlink User Note 95 1.3

[9] Curtis E. I., Richer J. S. and Buckle J. V. 2010, *A submillimetre survey of the kinematics of the Perseus molecular cloud – I. Data*, MNRAS, 401, 455 (DOI:10.1111/j.1365-2966.2009.15658.x) 8.9.2

[10] Draper P. W., Gray N., Berry D. S., Taylor M., 2012, *GAIA – Graphical Astronomy and Image Analysis Tool*, Starlink User Note 214 1.3

[11] Draper P. W., Taylor M. and Allan A., 2011, *CCDPACK – CCD data reduction package*, Starlink User Note 139 1.3

[12] Gibb A. G., Jenness T., Economou F., 2012, *PICARD — a PIpeline for Combining and Analyzing Reduced Data*, Starlink User Note 265 1.3

[13] Dempsey, J. T.,Thomas, H. S. and Currie, M. J., 2013, *CO (3 - 2) High-resolution Survey of the Galactic Plane: R1*, Astrophys.J. Supp., 209, 8 (DOI: 10.1088/0067-0049/209/1/8) I.4, I.1

[14] Jenness T. & Economou F., 2015, *ORAC-DR: A generic data reduction pipeline infrastructure*, Astronomy and Computing, 9, 40 (DOI:10.1016/j.ascom.2014.10.005) 1.2

[15] Jenness T., Currie M. J., Tilanus R., Cavanagh B., Berry D. S., Leech J., Rizzi L., 2015, *Automated reduction of sub-millimetre single-dish heterodyne data from the James Clerk Maxwell Telescope using ORAC-DR*, MNRAS, 453, 73 (DOI:10.1093/mnras/stv1545) 5

Appendix A
PICARD

There are two PICARD recipes which can be used with ACSIS data and are of general interest:

A.1 CREATE_MOMENTS_MAP

This recipe is used to create a moments maps from a cube. It smooths the cube in frequency and spatial extents, then uses the ClumpFind algorithm to find clumps of emission. Everything in the cube not found in a clump is masked out, then the masked cube is collapsed to form the moments map.

Recipe options:

- BASELINE_ORDER: The polynomial order for baseline fitting `[1]`
- CLUMP_METHOD: Method to find emission in the data (`clumpfind/fellwalker/thresh`) `[clumpfind]`
- FREQUENCY_SMOOTH: Smoothing kernel size in channels along the frequency axis for baseline determination `[25]`
- MOMENTS: Which moments maps to create (`integ/iwc/itd`) `[integ]`
- MOMENTS_LOWER_VELOCITY: An optional lower velocity in km s^{-1}, below which no data will be used when creating the moments map `[undef]`
- MOMENTS_UPPER_VELOCITY: An optional upper velocity in km s^{-1}, above which no data will be used when creating the moments map `[undef]`
- MOMENTS_SNR: Selects whether or not to do clump detection on an signal-to-noise cube instead of the signal cube. Enabling this is useful for data taken in varying conditions. `[0]`
- SPATIAL_SMOOTH: Smoothing kernel size in pixels along both spatial axes for baseline determination `[3]`

A.2 MOSAIC_JCMT_IMAGES

This recipe co-add the given files into a single map while correctly handling the exposure time FITS header and other NDF components. The maps are combined using inverse-variance weighting and the output variance is derived from the input variances. You can choose either wcsmosaic or a combination of wcsalign and the CCDPACK application makemos to mosaic the images.

Recipe options:

- MOSAIC_EACH: Flag to indicate whether the data should be mosaicked by individual object or combined into a single output file.

- MOSAIC_TASK: Name of mosaicking task to use, wcsmosaic or makemos. [wcsmosaic]
- MAKEMOS_METHOD: Image combination method for makemos, may be median (default), mean or sigmas. See CCDPACK manual for advice on choosing a method.
- MAKEMOS_SIGMAS: Sigma-clipping threshold if MAKEMOS_METHOD = sigmas. [4]
- MASK_LOWVAR: Flag to indicate that pixels with anomalously-low variances should be removed before mosaicking. [0]
- WCSMOSAIC_METHOD: wcsmosaic and wcsalign pixel-spreading method. [nearest]
- WCSMOSAIC_PARAMS: additional parameters that may be specified with wcsmosaic.

A.3 Running PICARD recipes

There are a number of options available when running PICARD recipes. The example below illustrates most of them.

```
% picard -recpars params.ini MOSAIC_JCMT_IMAGES -files myfilelist.lis -log sfx
```

- The recipe parameters are listed in a file called params.ini. It is called by the recpars option. This file has the same .ini format used by the ORAC-DR pipeline (see Section 6.2).
- The recipe name need to be in uppercase.
- If you need to supply multiple files you can do so by listing them in a file (myfilelist.lis in this example).
- Like the ORAC-DR pipeline, the -log option specifies whether the log file should be written to the screen [s], a file [f] or an xwindow [x].

Tip

The .sdf extension on filenames is required by PICARD.

Tip

If the environment variable ORAC_DATA_OUT is defined, any files created by PICARD will be written in that location. Check there if new files are expected but do not appear in your working directory.

Appendix B
Converting File Formats

B.1 Converting a FITS file to NDF

A FITS file can be converted to NDF using the Starlink package CONVERT. The `convert` instruction sets up the definitions for this package and only needs to be entered once,

```
% convert
```

the relevant one here being `fits2ndf`.

```
% fits2ndf file.fits file.sdf
```

Note that the `.sdf` file extension NDF may be omitted to save typing.

Unless your FITS file is from a recognised source, fits2ndf puts the various FITS extensions into NDF extensions called FITS_EXT_$< n >$, where n counts from one for the first FITS extension. These are located in the top-level MORE component of the NDF. If one of these is the array you want—it should be an NDF too—it will be easier to handle if you copy out of the NDF extension to its own NDF.

```
% ndfcopy in=file.more.fits_ext_1 out=file_data
```

If there is a variance array stored in another extension that you want to attach to your new NDF, a command like the following will do that.

```
% setvar ndf=file_data from=file.more.fits_ext_2
```

fits2ndf does offer a way of mapping FITS extensions to familiar NDF array components DATA, VARIANCE, and QUALITY through the `EXTABLE` file, avoiding the ndfcopy and possible setvar steps.

B.2 Converting an NDF file to FITS

The CONVERT package can also be used to convert an NDF file to FITS format.

```
% convert
% ndf2fits file file.fits
```

There are options to control whether or not to export all of the NDF extensions (`PROEXTS`) and history records (`PROHIS`), and whether or not to merge the contents of the FITS airlock (`PROFITS`).

For historical reasons the specific set of headers encoding the world co-ordinate system varies from package to package. There is a parameter to select the appropriate encoding for the destination of the FITS file. A common one in the submillimetre world is the CLASS package used for spectroscopic reductions.

```
% ndf2fits ga20091011_37_1_reduced \* encoding=fits-class
```

This will create a `ga20091011_37_1_reduced.fit` file with the CLASS-style of WCS headers.

Appendix C Scripting your Reduction

You may wish to write a script to process your data rather than running each task independently. In this case your script should start with:

```
#!/bin/csh -f
source $STARLINK_DIR/etc/login >& /dev/null
source $STARLINK_DIR/etc/cshrc >& /dev/null
kappa >& /dev/null
```

Appendix D
Regridding versus Resampling

Sometimes you may wish to put your data onto a new pixel-size scheme. This may either involve regridding (often called rebinning) the data onto larger pixel sizes, or resampling the data onto smaller pixels, both using KAPPA.

The key difference between rebinning and resampling is whether you iterate over the input or output pixels. Rebinning divides each input pixel value between a group of neighbouring output pixels, whereas resampling allocates each output pixel a value sampled from the input array.

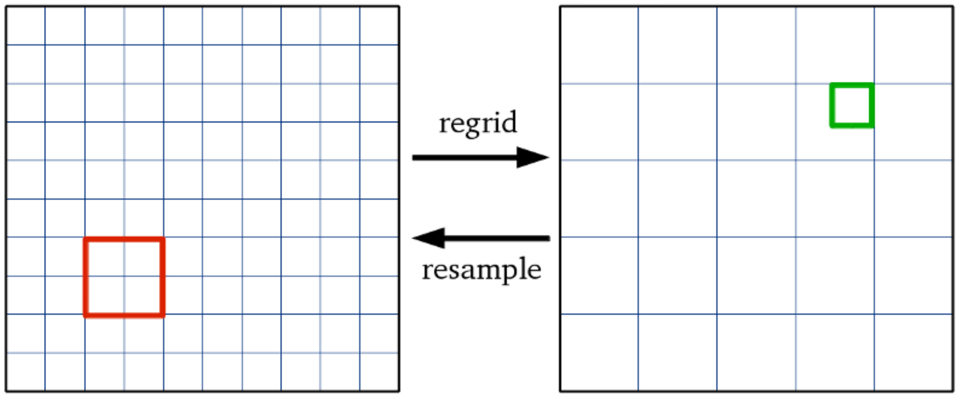


Figure D.1: Regridding versus resampling.

resampling—Resampling of the grid of input data is performed by transforming the co-ordinates of the centre of each output pixel into the co-ordinate system of the input grid. Since the resulting co-ordinates will not, in general, coincide with the centre of an input pixel, sub-pixel interpolation is performed between the neighbouring input pixels. This produces a resampled value, which is then assigned to the output pixel. Since resampling does not integrate, the total data value in the input image will not, in general, be conserved. Resampling has the advantage of ensuring that the output image is filled (provided the input array contains good data).

rebinning—Rebinning of the grid of input data is performed by transforming the co-ordinates of the centre of each input pixel into the co-ordinate system of the output grid. The input pixel value is then divided up and assigned to the output pixels in the neighbourhood of the central output co-ordinates. A choice of schemes are provided for determining how each input pixel value is divided up between the output pixels. In general, each output pixel may be assigned values from more than one input pixel. All contributions to a given output pixel are summed to produce the final output pixel value. Rebinning can leave gaps in the output array if the output array's pixels are smaller than those of the input array.

Appendix E
Clump-finding Algorithms

GaussClumps
Based on the algorithm described by Stutzki & Güsten (1990, ApJ 356, 513). This algorithm proceeds by fitting a Gaussian profile to the brightest peak in the data. It then subtracts the fit from the data and iterates, fitting a new ellipse to the brightest peak in the residuals. This continues until the integrated data sum in the fitted Gaussians reaches the integrated data sum in the input array, or a series of consecutive fits are made which have peak values below a given multiple of the noise level. Each fitted ellipse is taken to be a single clump and is added to the output catalogue. In this algorithm, clumps may overlap. Any input variance component is used to scale the weight associated with each pixel value when performing the Gaussian fit. The most significant configuration parameters for this algorithm are: `GaussClumps.FwhmBeam` and `GaussClumps.VeloRes` which determine the minimum clump size.

ClumpFind
Described by Williams et al (1994, ApJ 428, 693). This algorithm works by first contouring the data at a multiple of the noise, then searches for peaks of emission which locate the clumps, and then follows them down to lower intensities. No a priori clump profile is assumed. In this algorithm, clumps never overlap. Clumps which touch an edge of the data array are not included in the final list of clumps.

Reinhold
Based on an algorithm developed by Kim Reinhold at JAC. See **SUN/255** for more information on this algorithm. The edges of the clumps are first found by searching for peaks within a set of one-dimensional profiles running through the data, and then following the wings of each peak down to the noise level or to a local minimum. A mask is thus produced in which the edges of the clumps are marked. These edges however tend to be quite noisy, and so need to be cleaned up before further use. This is done using a pair of cellular automata which first dilate the edge regions and then erode them. The volume between the edges are then filled with an index value associated with the peak position. Another cellular automata is used to removed noise from the filled clumps.

FellWalker
David Berry devised an algorithm[2] that walks up hills along the line of greatest gradient until a significant peak is reached. It then assigns all pixels visited along the route to the clump associated with the peak. Such a walk is performed for every pixel in the data array which is above a specified background level.

See **SUN/255** for more information on findclumps and a full list of the configuration options for each algorithm.

Appendix F
Viewing your Data with KAPPA

This appendix offers a few examples of creating different types of plots using KAPPA. The applications highlighted here are display and contour for maps, and linplot and clinplot for spectra (See the KAPPA manual for more details).

The plotting attributes for each of these applications is controlled in the same way. For full details of all the formatting options as well as setting up your graphics device, see the 'Plotting Styles and Attributes' section in the KAPPA manual.

F.1 Setting up xwindows

- Clear my graphics window and reset default values.

```
% gdclear
```

- Set my display device to xwindows.

```
% gdset xw
```

- Create an xwindow of the desired dimensions.

```
% xmake -width 1500 -height 1000 xwindows
```

- To create a display window which plots black on a white background, change the order of black and white in the palette list for your graphics device. Check this using gdstate.

```
% gdstate
```

  Ensure white is entry 0 and black in entry 1.

```
% palentry 0 White
% palentry 1 Black
```

F.2 Format axes

- Use wcsattrib to set how your axes should be formatted. Here the format of the first and second axes is set to two decimal places for Galactic co-ordinates. Find out more about the Format attribute under wcsattrib in the KAPPA manual.

```
% wcsattrib map set format"(1)" .2
% wcsattrib map set format"(2)" .2
```

- For FK5 co-ordinates you can use the same method to set exactly how the co-ordinates should be displayed. The example below sets R.A. units to $0^h00^m00^s$ and Dec. units to $00'00''$.

```
% wcsattrib fk5map set format"(1)" ghms
% wcsattrib fk5map set format"(2)" gdms
```

F.3 Plotting a two-dimensional image

- Display `map.sdf`. Do draw axes and a border, but do not draw a title or a grid. The default cut parameters for `mode=scale` include all the data.

```
% display map border axes style='"drawtitle=0,grid=0"' mode=scale
```

  Depending on your default settings you may get a map like the one below.

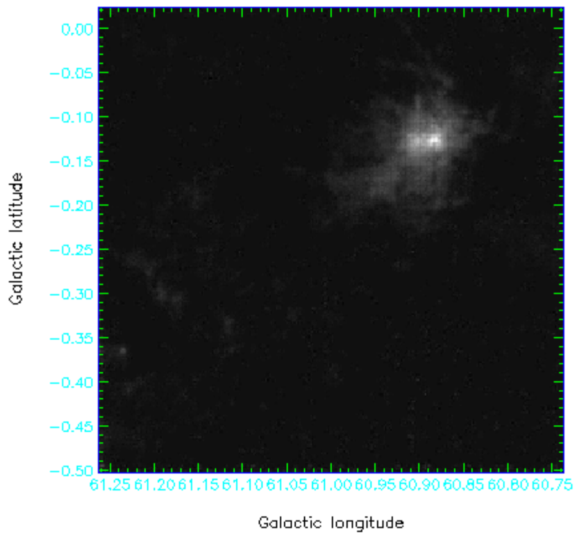


Figure F.1: First attempt at displaying an integrated map with KAPPA:display.

- Re-draw calling a style file and changing the scaling.

```
% display integ.sdf axes style=^style.dat mode=scale low=0 high=100
```

  Below is a copy of the style file (`style.dat`) used to make Figure F.2.

```
border=1
tickall=1
majticklen=0.01
drawtitle=0
color(axis)=white
color(numlab)=black
color(border)=black
color(ticks)=white
size(numlab)=1
size(textlab)=1.5
nogrid=1
textlabgap(1)=0.01
textlabgap(2)=0.01
```

- Make the colour scale negative and then replot.

```
% lutneg
% display integ.sdf axes style=^style.dat mode=scale low=0 high=100
```

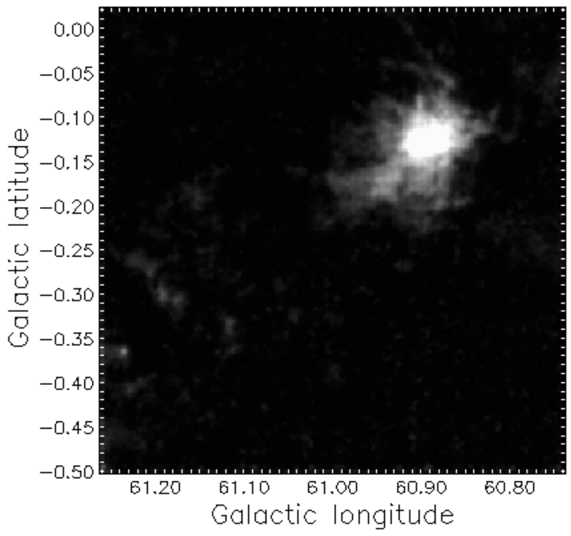


Figure F.2: Displaying an integrated map with KAPPA:display using a style file to format the plotting attributes.

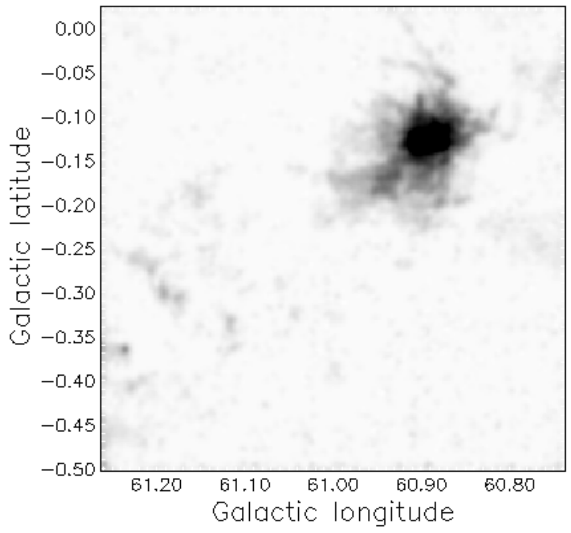


Figure F.3: Reversing the colour scale in KAPPA:display.

- There are a number of colour palettes available in Starlink. You can find them in the `$STARLINK_DIR//bin/kappa/` directory.

  ```
  % ls $STARLINK_DIR/bin/kappa/*lut.sdf
  ```

  You can create your own colour scheme using the KAPPA routine lutedit. See the KAPPA manual for more details.

- Replot in colour by selecting one of these palettes via the `lut` option.

  ```
  % display integ.sdf axes style=^style.dat mode=scale low=0 high=100 \
    lut=$STARLINK_DIR/bin/kappa/random3_lut
  ```

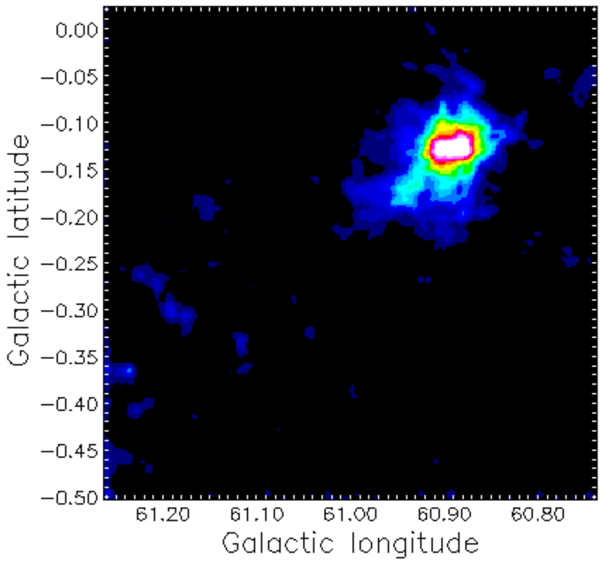


Figure F.4: Displaying an integrated map in colour with KAPPA:display.

F.4 Plotting spectra

- Plot the spectrum from `cube.sdf` at the position l=60.°87, b=0.°18. Use existing `style.dat` except override the text label size which is too big for these axes labels.

```
% linplot "cube(60.87,0.18,)" style="'^style.dat,size(textlab)=1'"
```

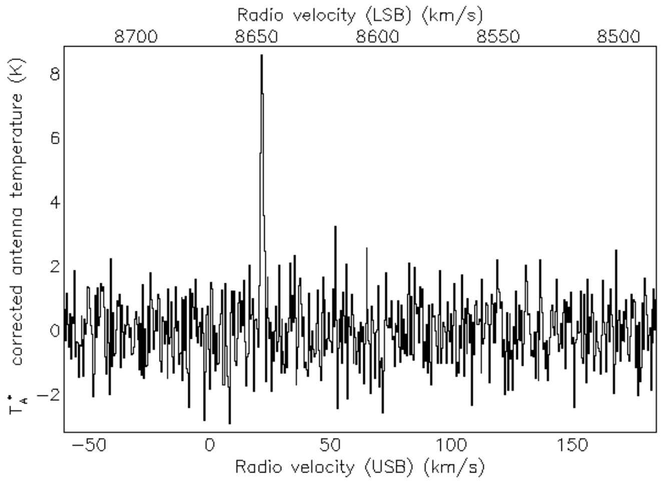


Figure F.5: Displaying a spectrum with KAPPA:linplot.

- Re-plot, but this time only show the velocity range 10 to 35 km s^{-1}. You can do it this way:

```
% linplot "cube(60.87,0.18,-10:50)"  style="'^style.dat,size(textlab)=1'" \
  mode=histogram
```

or using `xleft` and `xright`.

```
% linplot "cube(60.87,0.18,10:35)"  style="'^style.dat,size(textlab)=1'" \
  mode=histogram xleft=10 xright=35 lmode='"Extend,15,15"'
```

In the example above, the additional option `lmode` is included. This sets how the upper and lower limits for the y-axis are defined. This example expands the minimum and maximum values by 15% of the data range. In additional `extended`, the available options are `range`, `percentile` and `sigma`.

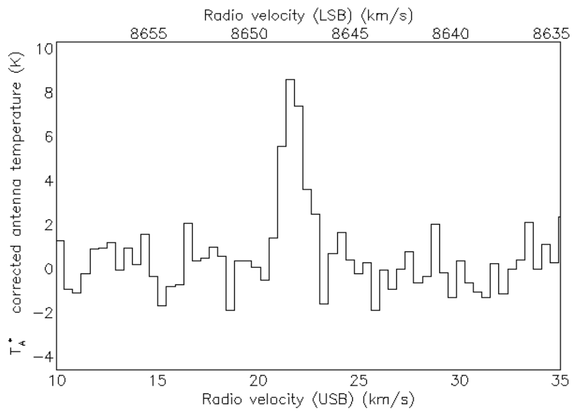


Figure F.6: Displaying a zoomed-in spectrum with KAPPA:linplot.

- To plot a second spectrum over this one, re-call linplot but with the `noclear` option to keep the original plot underneath. The formatting defaults to the last time linplot was run, but with the one-off temporary change of setting the colour of the line to red. The plus sign means the change will not be remembered next time linplot is run.

```
% linplot "cube2(60.87,0.18,-10:50)" mode=hist style="+colour(line)=red" \
  noclear
```

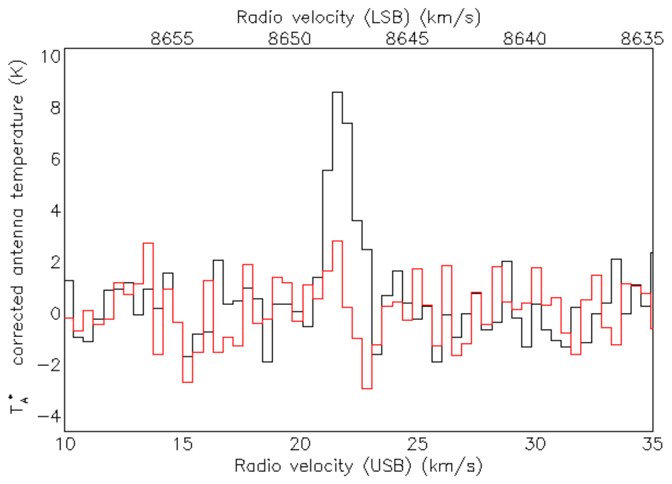


Figure F.7: Displaying two super-imposed spectra with KAPPA:linplot.

F.5 Plotting a grid of spectra

- To plot multiple spectra from a cube you can use clinplot.

```
% clinplot "cube(61.06~5,0.15~4,10.:30.)" nokey style=^style.dat reflabel \
  specstyle="'colour(textlab)=red,colour(numlab)=red'" mode=hist
```

This example plots a grid of 5×4 pixels around l=61°.06 and b=0°.15. Each spectrum has a range of 10 to 30 km s^{-1}. Note the decimal points are necessary for the spectral limits, otherwise the spectrum would extend from pixel co-ordinates 10 to 30. The `specstyle` option allows you to format the spectral axes inside the main plot. The formatting options are the usual plotting attributes. The flag `reflabel` annotates the interior spectral axes.

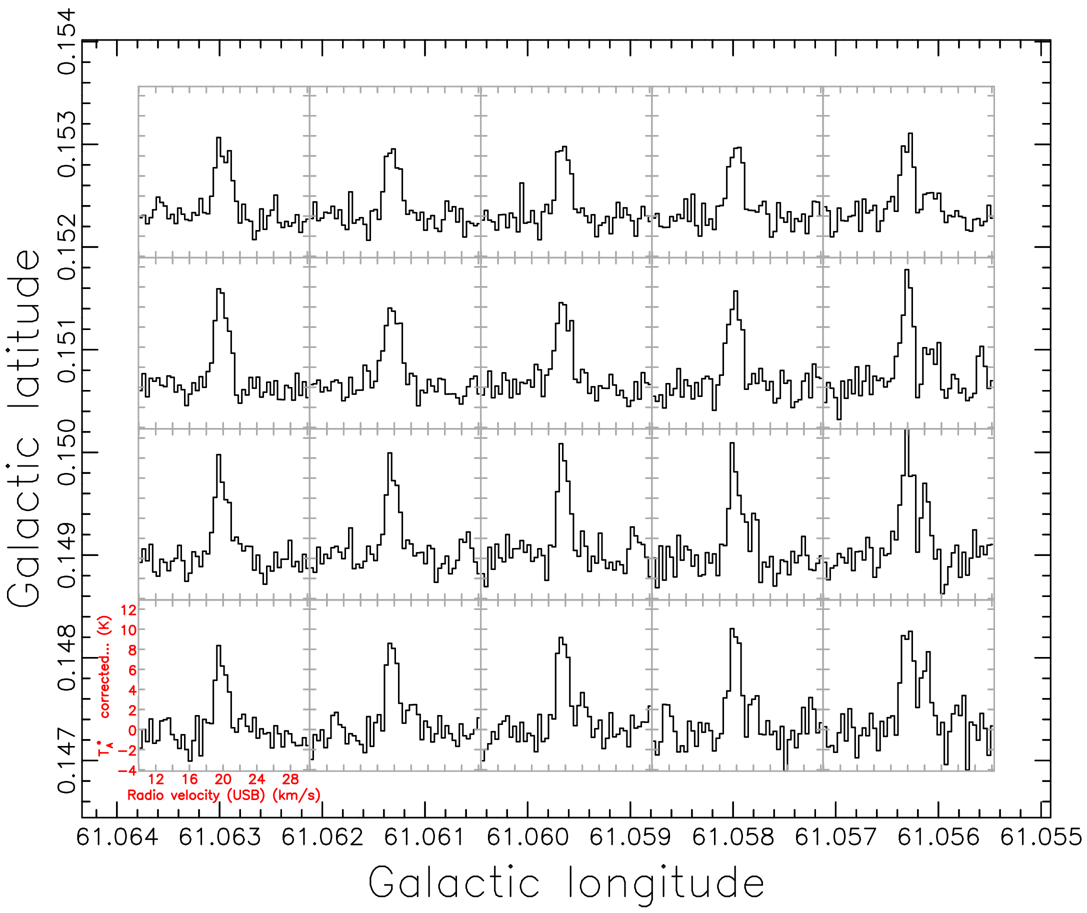


Figure F.8: Displaying a grid of spectra with clinplot.

F.6 Plotting two images side by side

- Create a 2×1 grid of frames.

```
% picgrid 2 1
```

- Select the first frame.

```
% picsel 1
```

- Display `integ.sdf` in the first frame.

```
% display integ axes style=^style.dat mode=scale low=0 high=100
```

- Select the second frame.

```
% picsel 2
```

- Display `subinteg.sdf`, a cut-out of integ.sdf, in the second frame.

```
% display subinteg axes style=^style.dat mode=scale low=0 high=100
```

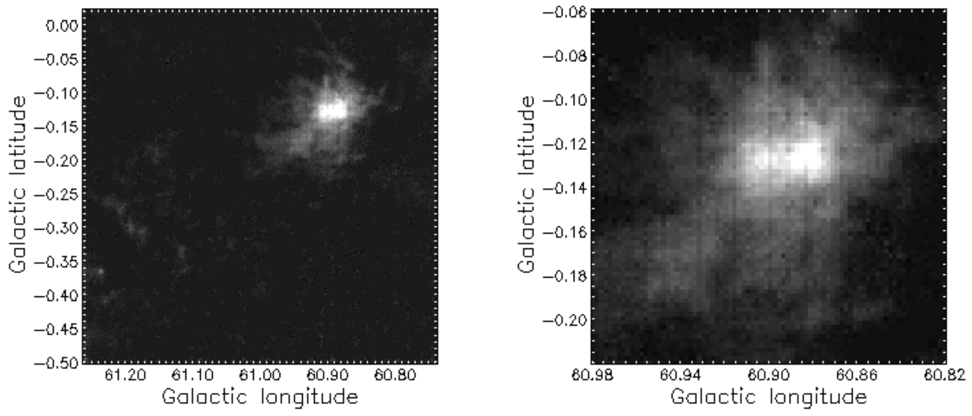


Figure F.9: Displaying an image and a magnified portion of that image with KAPPA:display.

- Still in the second frame, plot contours over the map from a smoothed version of subinteg.sdf.

```
% contour smoothsub.sdf noaxes noclear nokey ncont=5 mode=free \
  heights="[40,80,120,160,200]" pens='"width=3.0,colour=red"'
```

F.7 Selecting a different graphics device

- You can view all available graphics devices with `gdnames`.

```
% gdnames
```

- Instead of using `gdset` to define a new graphics device for all your applications, you can add the `device` option to select a different one for just this command. In the example below the output goes to a PostScript file called `myplot.ps`. The option `pscol_l` specifies landscape mode with colour. Setting a name for your output PostScript file is optional; if left unset it will default to `pgplot.ps`.

```
% linplot "cube(60.85,0.18,-10:50)" mode=hist device="pscol_l;myplot.ps"
```

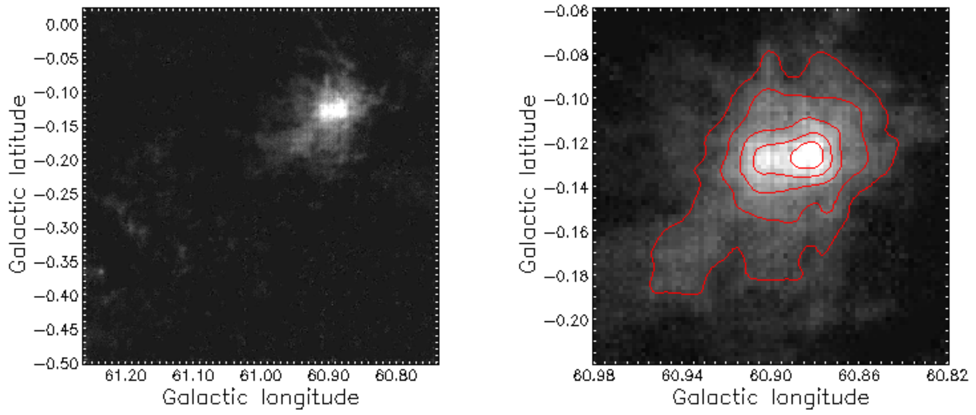


Figure F.10: Displaying an image and a magnified portion of that image overlaid with smooth contours created with KAPPA:display and contour.

- If you want to combine the results of several applications into a single plot, you need to specify an encapsulated PostScript device, e.g. epsf_l. Once you have run all the application you can stack your list of PostScript files to produce a single file.

```
% psmerge plot*.ps > final.ps
```

Tip

You can abbreviate any of the command-line options so long as the application can distinguish it from the other options. For example, `mode=histogram` can be abbreviated all the way to `mode=h`.

Appendix G
Classified Recipe Parameters

The recipes REDUCE_SCIENCE_NARROWLINE, REDUCE_SCIENCE_GRADIENT, REDUCE_SCIENCE_LINEFOREST, and REDUCE_SCIENCE_BROADLINE support the following parameters, except where noted. Recipe parameters should be supplied in an external file and called by the command line option `-recpars <params.ini>`. For an example recipe-parameter file see Section 6.2.

The parameters are classified for easier identification.

Parameter	**Description**
ALIGN_SIDE_BAND	Whether or not to combine sidebands in makecube.
CLUMP_METHOD	Method for identifying emission clumps: `Clumpfind`, `Fellwalker`, or `Thresh`.
CUBE_WCS	The co-ordinate system in which to regrid cubes.
FINAL_LOWER_VELOCITY	Lower velocity limit for all the cube products.
FINAL_UPPER_VELOCITY	Upper velocity limit for all the cube products.
ITERATIONS	Number of iterations. Further iterations refine the identification of emission to exclude from baseline subtraction. One iteration is usually sufficient.
PIXEL_SCALE	The pixel scale, in arcseconds, of cubes.
REBIN	Comma-separated list of velocity resolutions to rebin the final spectral cube. The rebinned cubes are in addition to the full-resolution cube. The last can be compressed too (cf. VELOCITY_BIN_FACTOR).
SPREAD_FWHM_OR_ZERO	Depending on the spreading method, this parameter controls the number of arcseconds at which the envelope of the spreading function goes to zero, or the full-width at half-maximum for the Gaussian envelope.
SPREAD_METHOD	The method to use when spreading each input pixel between a group of neighbouring output pixels when regridding cubes.
SPREAD_WIDTH	The number of arcseconds on either side of the output position which receive contributions from the input pixel.
VELOCITY_BIN_FACTOR	Average contiguous sets of velocity channels by this integer factor, each forming one channel in the reduced output. This compression is intended for the high-resolution ACSIS modes, to save significant storage and processing time, but it also yields better baseline subtraction.

Table G.1: The recipe parameters used to define the properties of the recipe products.

Parameter	Description
FRACTION_BAD	The maximum fraction of bad values permitted in a receptor (or receptor's subband for a hybrid observation) permitted before the a receptor is deemed to be bad.
RESTRICT_LOWER_VELOCITY	Trims all data to this lower velocity, not just at the end for the products.
RESTRICT_UPPER_VELOCITY	Trims all data to this upper velocity, not just at the end for the products.
TRIM_MINIMUM_OVERLAP	The minimum number of desired channels that should overlap after trimming hybrid-mode observations.
TRIM_PERCENTAGE	The percentage of the total frequency range to trim from either end. This parameter only takes effect if both `TRIM_PERCENTAGE_LOWER` and `TRIM_PERCENTAGE_UPPER` are undefined.
TRIM_PERCENTAGE_LOWER	The percentage of the total frequency range to trim from the lower end of the frequency range.
TRIM_PERCENTAGE_UPPER	The percentage of the total frequency range to trim from the higher end of the frequency range.

Table G.2: The recipe parameters used to exclude areas of noisy spectrum or bad receptors.

There are a number of ways to define the baseline regions:

- as a percentage of the spectrum width at either end of the spectrum (see BASELINE_EDGES);
- as a set of velocity ranges expected or known to be free of emission lines (see BASELINE_REGIONS); or if both of these arguments of corresponding recipe parameters is undefined,
- use the whole spectrum smoothing spectrally and spatially (see FREQUENCY_SMOOTH and SPATIAL_SMOOTH) with feature detection to mask lines (see BASELINE_METHOD).

The first two are suitable for broadline emission. The second is desirable in the presence of many lines. The third is the default and most appropriate for a single narrow line.

Parameter	**Description**
BASELINE_EDGES	Percentage of the full range to fit on either edge of the spectra for baselining purposes. If set to a non-positive value and BASELINE_REGIONS is undefined, then the baseline is derived after smoothing and automatic emission detection. If assigned a negative value, BASELINE_REGIONS, if it is defined, will be used instead to specify where to determine the baseline.
BASELINE_METHOD	Source of the baseline region. Currently only `auto` is recognised. This requests the automated mode where the emission is detected and masked before baseline fitting. Otherwise BASELINE_EDGES or BASELINE_REGIONS (q.v.) will be used.
BASELINE_NUMBIN	The number of channels to which the spectral axis is compressed for automated masking of emission when BASELINE_METHOD=`auto`.
BASELINE_ORDER	The polynomial order to use when baselining cubes.
BASELINE_REGIONS	A comma-separated list of velocity ranges each in the format $v1 : v2$, from where the baseline should be estimated. It is countermanded should BASELINE_EDGES be positive. These can also be used to define where to test baseline linearity if BASELINE_LINEARITY_LINEWIDTH is set to `base`.
FREQUENCY_SMOOTH	The number of channels to smooth in the frequency axis when smoothing to determine baselines. This number should be small ($\sim$10) for narrow-line observations and large ($\sim$25) for broad-line observations.
SPATIAL_SMOOTH	The number of pixels to smooth in both spatial axes when smoothing to determine baselines.

Table G.3: The recipe parameters to control how baselines are determined and fit.

Parameter	**Description**
LV_IMAGE	Permits creation of a longitude-velocity map via primitive _CREATE_LV_IMAGE_
LV_AXIS	Specify the axis to collapse in the creation of the longitude-velocity map.
LV_ESTIMATOR	Specify the collapse statistic in the creation of the longitude-velocity map.
CREATE_MOMENTS_USING_SNR	If set to true (`1`), moments maps will be created using a signal-to-noise map to find emission regions. This is useful when observations were taken under differing sky conditions and have different noise levels.
MOMENTS	Comma separated list of the moments maps to create (integ,iwc).
MOMENTS_LOWER_VELOCITY	The lower velocity range from which the moments maps will be created.
MOMENTS_UPPER_VELOCITY	The upper velocity range from which the moments maps will be created.

Table G.4: The recipe parameters used to specify moments and longitude-velocity products.

Parameter	Description
FLATFIELD	Whether or not to perform flat-fielding.
FLAT_APPLY	Whether or not to apply the calculated flatfield. If set false the ratios are still calculated and logged.
FLAT_METHOD	Flatfield option to ratio voxel by voxel.
FLAT_LOWER	Sets the lower limit by which to restrict the velocity range where there is astronomical signal for FLAT_METHOD.
FLAT_UPPER	Sets the upper limit by which to restrict the velocity range where there is astronomical signal for FLAT_METHOD.
MINSNR	Allow selection of higher signal-to-noise voxels for FLAT_METHOD.

Table G.5: The recipe parameters associated flat fielding.

Parameter	Description
DESPIKE	Whether or not to perform despiking.
DESPIKE_BOX	The size, in pixels, of the box used to both find the"background" and for cleaning spikes.
DESPIKE_CLIP	The clip standard deviations to use when finding spikes in the background-subtracted RMS spectrum.
DESPIKE_PER_DETECTOR	If a spike is not seen in all detectors, consider setting this value to 1.

Table G.6: The recipe parameters associated removal of noise spikes.

Parameter	Description
CHUNKSIZE	Maximum sized chunk used for the group cube.
CUBE_MAXSIZE	Controls the maximum size of the reduced cube.
TILE	A true value (1) performs tiling of the cube to restrict the memory requirements. Such tiled cubes abut each other in pixel co-ordinates and may be pasted together to form the complete spectral cube.

Table G.7: The recipe parameters used to limit computer memory requirements for large datasets or observation fields of view.

Parameter	**Description**
HIGHFREQ_INTERFERENCE	If set to true (1) the spectra for each receptor are analysed to detect high-frequency interference noise, and those spectra deemed too noisy are excluded from the reduced products.
HIGHFREQ_INTERFERENCE_EDGE_CLIP	Used to reject spectra with high-frequency noise. It is the standard deviation to clip the summed-edginess profile iteratively in order to measure the mean and standard deviation of the profile unaffected by bad spectra.
HIGHFREQ_INTERFERENCE_THRESH_CLIP	Used to reject spectra with high-frequency noise. This is the number of standard deviations at which to threshold the noise profile above its median level.
HIGHFREQ_RINGING	Whether or not to test for high-frequency ringing in the spectra. This is where a band of spectra in the time series have the same oscillation frequency and origin with smoothly varying amplitude over time.
HIGHFREQ_RINGING_MIN_SPECTRA	Minimum number of good spectra for ringing filtering to be attempted (see HIGHFREQ_RINGING). The filter needs to be able to discriminate between the normal unaffected spectra from those with ringing. The value should be at least a few times larger than the number of affected spectra.

Table G.8: The recipe parameters associated exclusion of spectra affected by high-frequency noise.

Parameter	**Description**
LOWFREQ_INTERFERENCE	If set to true (1), the spectra for each receptor are analysed to detect low-frequency local interference ripples or bad baselines, and those spectra deemed too deviant from linearity are excluded from the reduced products.
LOWFREQ_INTERFERENCE_EDGE_CLIP	Used to reject spectra with low-frequency interference. It is the standard deviation to clip the profile of summed-deviations from linearity iteratively in order to measure the mean and standard deviation of the profile unaffected by bad spectra. A comma-separated list will perform iterative sigma clipping of outliers, but standard deviations in the list should not decrease.
LOWFREQ_INTERFERENCE_MAX_THRESHOLD	Spectra are deemed to be non-linear if their non-linearity exceeds this threshold.
LOWFREQ_INTERFERENCE_MIN_THRESHOLD	No spectra with non-linearity below this threshold will be rejected.
LOWFREQ_INTERFERENCE_THRESH_CLIP	Used to reject spectra with low-frequency interference. This is the number of standard deviations at which to threshold the non-linearity profile above its median level.

Table G.9: The recipe parameters associated exclusion of spectra affected by non-linear baselines. These are for short periods during an observation where some external signal has affected the baselines. If most or whole of an observtion might be affected see `BASELINE_LINEARITY` and related parameters below. These parameters are not available in the REDUCE_SCIENCE_BROADLINE recipe.

Parameter	Description
BASELINE_LINEARITY	If set to true, receptors with mostly or all non-linear baselines are excluded from the reduced products.
BASELINE_LINEARITY_CLIP	This is used to reject receptors that have non-linear baselines. It is the maximum number of standard deviations above the median rms deviations for which a detector's non-linearity is regarded as acceptable.
BASELINE_LINEARITY_LINEWIDTH	This is used to reject receptors that have non-linear baselines. It is the extent of the source spectral line measured in km s^{-1}, which is excluded from the non-linearity tests.
BASELINE_LINEARITY_MINRMS	This is used to retain receptors that have noisy or slightly non-linear baselines, or transient bad baselines (cf. LOWFREQ_INTERFERENCE). The parameter is the minimum rms deviation from linearity, measured in antenna temperature, for a receptor to be flagged as bad.
BASELINE_LINEARITY_SCALELENGTH	This is used to reject receptors that have non-linear baselines. It is the smoothing scale length in whole pixels. Features narrower than this are filtered out during the background-level determination. It should be should be odd and sufficiently large to remove the noise while not removing the low-frequency patterns in the spectra.

Table G.10: The recipe parameters associated exclusion of receptors affected by non-linear baselines throughout or most of an observation. The parameters are not available in the REDUCE_SCIENCE_BROADLINE recipe.

Parameter	Description
CALCULATE_STANDARD_ALWAYS	If set true (`1`), this will ensure that the `log.standard` file will be created even if the source, molecule, and transition combination is not present in the list of known standard sources. This will affect most non-continuum heterodyne recipes.
NOSIDEBANDCORR	If set to `1`, this will prevent any sideband-calibration correction factor from being applied to the data.
SIDEBAND	Set the sideband (e.g. `USB` or `LSB`) to use when applying the sideband correction. By default the system will use the sideband of the observation, but this recipe parameter can override that default if you are interested in a line in the image sideband.
SIDEBAND_CORR_FACTOR	This allows you to override the sideband-correction factor chosen from the calibration system and provide your own. The data will be multiplied by the value you provide, so you must ensure it is appropriate for the given sideband and LO frequency of your data. The factor should be supplied in floating point.

Table G.11: The recipe parameters associated with calibration of the data. The sideband-ratio-correction recipe parameters are used for controlling the sideband correction, and are currently only available for instrument RxA3M. These corrections do not affect reductions done with REDUCE_SCIENCE_CONTINUUM, but will be used in all other RxA3M science reductions, and in the PICARD recipe CALIBRATE_SIDEBAND_RATIO. The CALCULATE_STANDARD_ALWAYS flag will also affect the non-continuum recipes.

Parameter	**Description**
SUBTRACT_REF_EMISSION	If true, the recipe will attempt to locate and remove reference (off-position) signal that appears as absorption lines.
REF_EMISSION_BOXSIZE	The width (in channels) of the largest reference-spectrum line expected, used to find the background before finding spectral lines. If no value is supplied, an iterative approach determines the largest line width.
REF_EMISSION_COMBINE_DETECTORS	If true, combine all detectors to form the reference spectrum. While it can improve the signal to noise, receptors are strictly viewing slightly different locations and hence each receptor's reference spectrum will be different.
REF_EMISSION_COMBINE_REFPOS	Whether to combine observations by their reference position (`1`), or by observation date (`0`).
REF_EMISSION_MASK_SOURCE	This controls the use of the mask of source emission already detected for baseline estimation. There are pros and cons both using the mask (`1`) or not using it (`0`). The final option, `Both`, attempts to combine the benefits of both methods: the masked spectrum locates the lines, but the unmasked modal spectrum determine the line strengths.
REF_EMISSION_REGIONS	Instead of automated identification of absorption lines, supply a comma-separated list of line extents ($v1$:$v2$). These regions are masked and interpolated, the background determined and subtracted from the unmasked representative spectrum to form an approximation to the reference spectrum.
SUBTRACT_REF_SPECTRUM	If true, the recipe will interpolate across the extents of reference (off-position) lines that you specify, applied to each (reference position or date and/or detector) median spectrum to estimate the respective reference spectrum.
REF_SPECTRUM_COMBINE_DETECTORS	If true, combine all detectors to form the reference spectrum. While it can improve the signal to noise, receptors are strictly viewing slightly different locations and hence each receptor's reference spectrum will be different.
REF_SPECTRUM_COMBINE_REFPOS	Whether to combine observations by their reference position (`1`), or by observation date (`0`).
REF_SPECTRUM_FILE	An estimated reference spectrum in absorption, with other values set to zero, to remove reference lines that other methods fail to excise.
REF_SPECTRUM_REGIONS	A comma-separated list of line extents ($v2$:$v2$) of the locations of absorptions lines to excise.

Table G.12: The recipe parameters associated with the filtering of reference-spectrum absorption-line artefacts, either automatically or specified manually. These methods are still experimental, although the manual identication is more reliable for known lines. Therefore these parameters are currently available only in the REDUCE_SCIENCE_NARROWLINE recipe.

Appendix H Quality Assurance Parameters

Quality assurance parameters are supplied via the command line option `-cal qaparams=<qa.ini>`. For an example QA parameters file see Section 6.3.

Parameter	**Description**
BADPIX_MAP	Maximum fraction of pixels allowed to be bad in the final map.
GOODRECEP	Minimum number of working receptors
TSYSBAD	Upper limit of T_{sys} for raw time-series data.
FLAGTSYSBAD	Flag any receptor with >FLAGTSYSBAD of data points with T_{sys} > TSYSBAD
TSYSMAX	Median T_{sys} > per receptor after flagging by FLAGTSYSBAD must be <TSYSMAX
TSYSVAR	Maximum allowable receptor-to-receptor T_{sys} variation
RMSVAR_RCP	Maximum allowable receptor-to-receptor RMS variation
RMSVAR_SPEC	The average rms noise shall not vary by more than RMSVAR_SPEC from one end of a spectrum to the other.
RMSVAR_MAP	The rms noise in the map over the region with uniform sampling shall not vary by more than RMSVAR_MAP from one pixel to the next.
RMSTSYSTOL	The rms that is measured for each receptor should agree with the value expected from the system temperature, spectral resolution, and integration time to within RMSTOL.
RMSMEANTSYSTOL	Data with an RMS > RMSMEANTSYSTOL% of the mean RMS will get masked out.
CALPEAKTOL	Observations of the spectral-line calibrators shall agree with their expected values for line peak intensity to within ±CALPEAKTOL.
CALINTTOL	Observations of the spectral-line calibrators shall agree with their expected values for integrated intensities to within ±CALINTTOL.
RESTOL	Remaining residuals after subtraction of the baseline shall be no larger than $\sigma \times$RESTOL on average.
RESCHAN12CO	Channels over which to average for residuals for a first line (^{12}CO here).
RESCHAN13CO	Channels over which to average for residuals for a second line (^{13}CO here).
RESTOL_SM	Remaining residuals should be no larger than $\sigma \times$RESTOL_SM on average over any four adjacent channels.
VELRES_12CO	Velocity resolution for the first line (^{12}CO here).
VELRES_13CO	Velocity resolution for the second line (^{13}CO here).

Appendix I
Removal of reference signal

During heterodyne observations, the telescope switches (see Section 3.3.2) between observing your target and a nearby reference position. The latter's spectrum is subtracted from the target data in order to subtract the sky-background signal.

In most cases, this reference location will be devoid of source emission. However, around the Galactic Plane and especially the Galactic Centre, it can be problematic to find a reference position that's pointing at sky, even with a position switch. Should the reference measurement include a source, its signal will be subtracted from all the spectra in your data, appearing one or more absorption lines. If the absorption occurs where your target has strong emission across a wide range of frequencies, the absorption feature can easily be camouflaged. Even it is located where there is weak or no target signal, a strong absorption feature will slightly bias the baseline fitting. So if your data has such artefacts, it is worth trying to correct them.

A simple approach for removal is to interpolate across the absorption feature, but this does assume that the target emission is not varying across the width of the absorption line or lines. If the absorption is located clear of your source's emission, it is possible to merely mask the feature with chpix. The following example masks all spectra witihn the time-series cube ts_cube between 3.5 and 7.8 km s^{-1}.

```
% chpix ts_cube ts_cube_refmasked section="'3.5:7.8,,'" bad
```

The basis of a better method is to attempt to determine the reference spectrum. One such approach is to identify regions with none or minimal target emission in your reduced spectral cube, mask these, then form the median spectrum of the remainder. Regions to include or exclude may be defined with ARD regions in a text file, possibly created with GAIA(see Section 8.7); or using the shape selection in the GAIA cube-manipulation controls, especially the polygon (see Figure I.1). The resultant spectrum may be saved (see Figure I.2). In practice this median spectrum will likely still have some target emission and possibly a non-flat baseline, which need to be removed. One approach is to smooth the spectrum with a wide kernel—essentially a low-pass filter—and then subtract the smoothed spectrum from the original. Finally, since we are only interested in the aborption features, set all the non-line elements to 0. In the Figure I.2 example that is between 4.7 and 6.3 km s^{-1}.

```
% block median_sky_spectrum smoothed_spectrum 21
% sub median_sky_spectrum smoothed_spectrum flat_spectrum
% chpix flat_spectrum ref_spectrum section="':4.7,6.3:'" 0
```

To apply this to your cube, enlarge it to the dimensions of your cube. The value of the AXES parameter will depend on whether you intend to subtract the reference spectrum from the raw time series, or from the position-position-velocity (PPV) reduced cube. For the time series

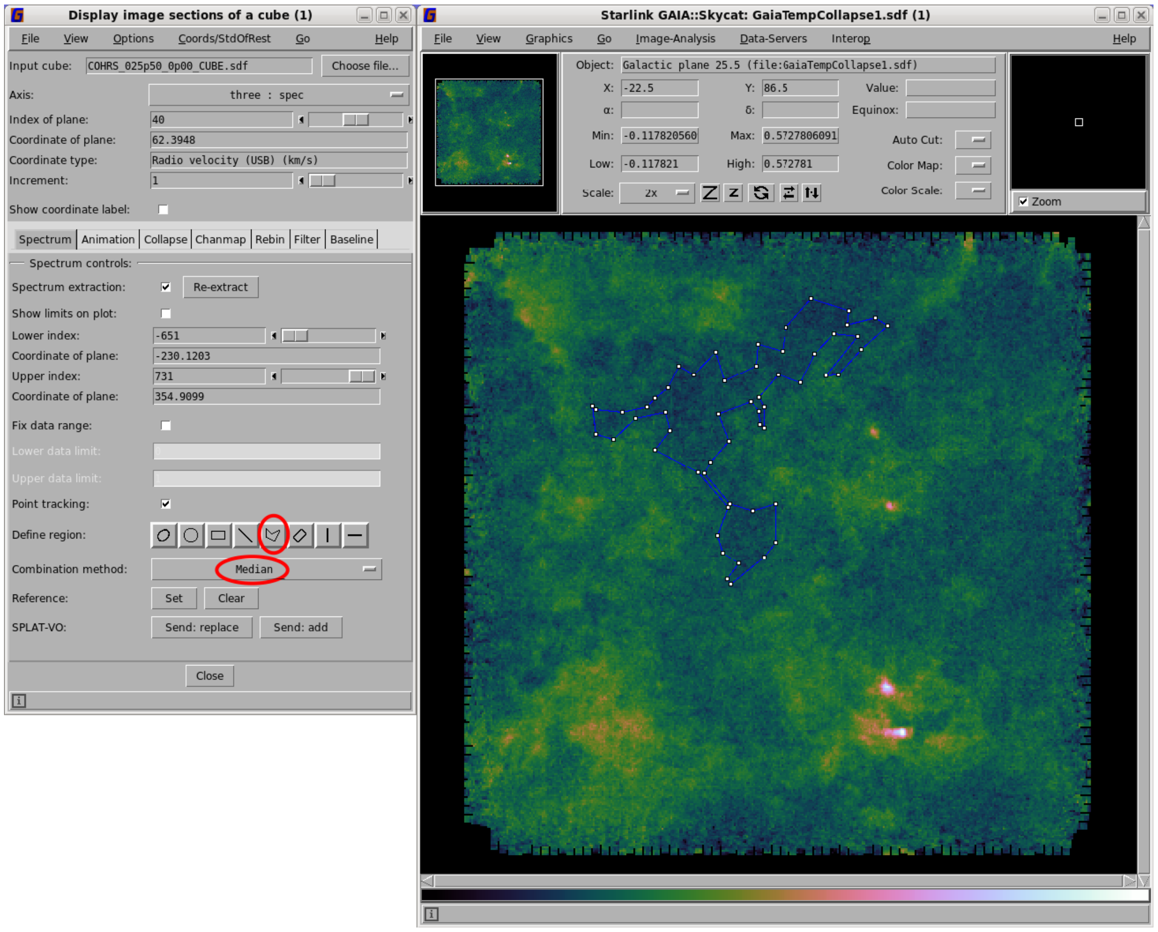
```
% manic ref_spectrum ref_ts_cube axes="[1,0,0]"  lbound="[1,1]" ubound="[14,4202]"
```

and for the PPV

```
% manic ref_spectrum ref_ppv_cube axes="[0,0,1]"  lbound="[-122,-124]" ubound="[126,125]"
```

AXES specifies the axis to retain (1) or new dimensions to grow (0). The pixel bounds of the new axes are given repectively by LBOUND and UBOUND arrays. In the time-series example, there are 14 receptors and 4202 spectra. You can find the bounds of your cube with ndftrace.

```
% ndftrace cube | grep "Pixel bounds"
```


Display image sections of a cube (1)
File View Options Coords/StdOfRest Go Help
Choose file...
Spectrum Animation Collapse Chanmap Rebin Filter Baseline
Spectrum controls:
Re-extract
Set Clear
Send: replace Send: add
Close
Starlink GAIA::Skycat: GaiaTempCollapse1.sdf (1)
File View Graphics Go Image-Analysis Data-Servers Interop Help
Zoom

Input cube: COHRS_025p50_0p00_CUBE.sdf
Axis: three : spec
Index of plane: 40
Coordinate of plane: 62.3948
Coordinate type: Radio velocity (USB) (km/s)
Increment: 1
Show coordinate label:
Spectrum extraction: ✓
Show limits on plot:
Lower index: -651
Coordinate of plane: -230.1203
Upper index: 731
Coordinate of plane: 354.9099
Fix data range:
Lower data limit:
Upper data limit:
Point tracking: ✓
Define region:
Combination method: Median
Reference:
SPLAT-VO:

Object: Galactic plane 25.5 (file:GaiaTempCollapse1.sdf)
X: -22.5 Y: 86.5 Value:
α: δ: Equinox:
Min: -0.117820560 Max: 0.5727806091 Auto Cut:
Low: -0.117821 High: 0.572781 Color Map:
Scale: 2x Color Scale:

Figure I.1: This shows a spectral cube collapsed to two dimensions with the "`Collapse`" tag, on which a polygonal region of background is outlined. First select the median "`Combination method`" and then the polygon "`Define region`" button (both ringed). Click the left mouse button to add points, and double-click to complete the polygon. Vertices may be dragged or new vertices added. See the GAIA help for details.

I.1 Remove the reference signal using ORAC-DR

A number of recipe parameters are provided for the attempted removal of reference signal, and are listed in Table G.12. For Galactic Plann surveys the following combination have worked moderately well.

```
CLUMP_METHOD = clumpfind
SUBTRACT_REF_EMISSION = 1
REF_EMISSION_MASK_SOURCE = both
REF_EMISSION_COMBINE_REFPOS = 1
REF_EMISSION_BOXSIZE = 19
```

You may need to enlarge `REF_EMISSION_BOXSIZ` should your data have narrow velocity channels. An example of using these settings that successfully removed the reference signal from some Galactic Plane data is shown in Figure I.4.

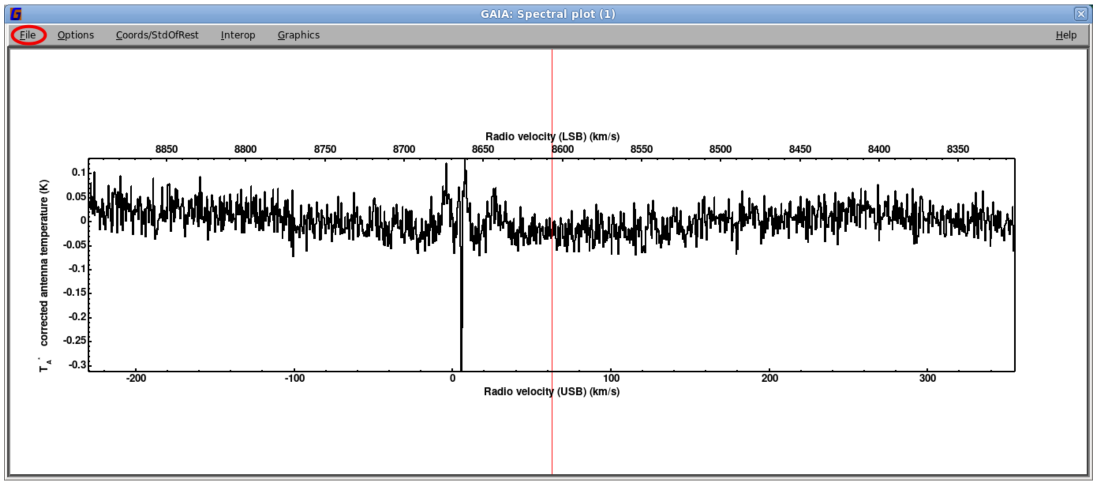

GAIA: Spectral plot (1)
File
Options
Coords/StdOfRest
Interop
Graphics
Help
Radio velocity (LSB) (km/s)
Radio velocity (USB) (km/s)
T_A* corrected antenna temperature (K)

Figure I.2: This shows a median spectrum from a GAIA region defined as in Figure I.1. Use the marked "`File`" menu and choose "`Save as NDF`" to save the created spectrum.

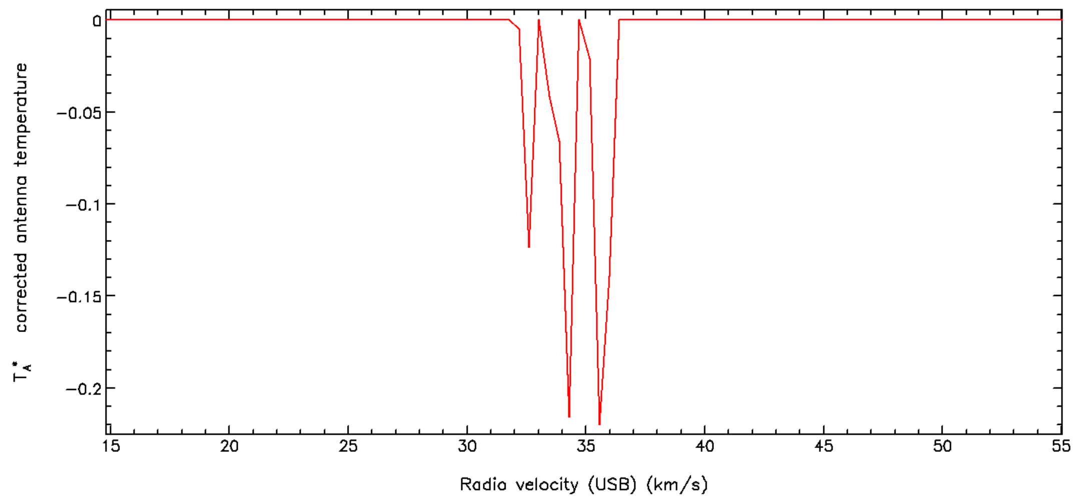


Figure I.3: An extract from an estimated spectrum derived from the methods outlined in the test. Note that beyond the lines, pixel values are zero so as not to add noise to the cube once the reference spectrum is subtracted.

The algorithm used by ORAC-DR does not always remove all of the lines, and in rare cases leaves lines unchanged. For these recalcitrant lines some additional facilities are provided, where you set the velocity limits of the reference absorption line.

```
SUBTRACT_REF_SPECTRUM = 1
REF_SPECTRUM_COMBINE_REFPOS = 1
REF_SPECTRUM_REGIONS = 7.6:13.8
```

`REF_SPECTRUM_REGIONS` sets a comma-separated list of the velocity ranges of the absorption lines.

If that does not work, you can provide your own estimated reference spectrum, containing zeroed data values outside of the line extent, as described earlier and shown in Figure I.3. You then supply its file name using the recipe parameter shown below.

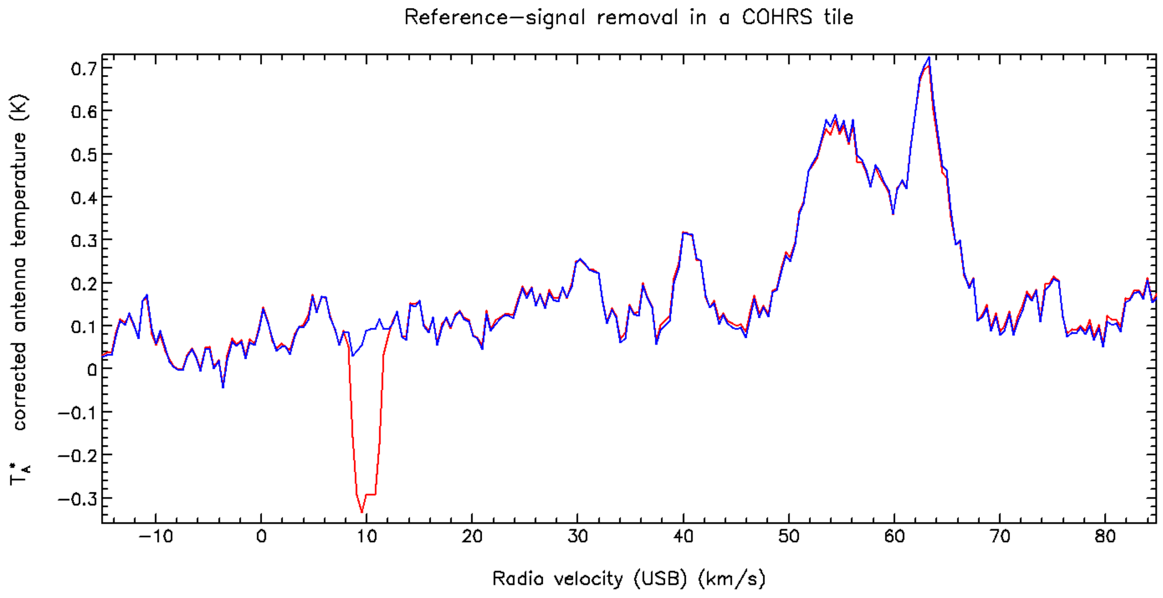


Figure I.4: Part of a mean spectrum collapsed over a COHRS[13] tile in the Galactic Plane. The red curve shows the original uncorrected spectrum and the blue curve is the re-processed spectrum, where the absorption feature is removed within ORAC-DR using the primary set of recipe parameters listed in the text above.

```
REF_SPECTRUM_FILE = ref_spectrum
```

The algorithms used are described in a forthcoming COHRS [13] Second Release paper.