

Figure 2: (*left*) The SMA’s quarter-wave plate (QWP) mounted on a rotation stage. Optical brakes (top and bottom left) determine the position of the QWP. Adjustable flags (small screws on perimeter of the rotation stage) at  $\pm 45^\circ$  from the top optical brake set the positions of the waveplate (from Fig. 1 of Marrone et al., 2008). (*right*) The QWP mounted inside the cabin of an SMA antenna in the path of the beam (personal photo).

## 2.2 Data Reduction with *Miriad*

The data reduction package *Miriad* (Sault et al., 1995) implements and performs most of the required computational tasks for calibrating the data and performing an inverse Fourier transform to obtain maps. When observations are taken the intensity of the visibilities is usually calibrated against measurements of quasars that are taken during the same night, since certain quasars can be assumed to be steady, bright, unchanging sources. The calibration involves calculating gain coefficients to correctly scale the visibilities to their true intensities. The data presented in this paper were already calibrated by Dr. Rao and Dr. Girart, and so the calibration process will not be explained in detail here. In *Miriad*, programs that perform useful operations on the data are broken up into commands called ‘tasks’ that can be scripted using bash scripts or Python. This section will summarize the most important tasks for creating images from the visibilities.

The `invert` task performs a standard two-dimensional fast Fourier transform

(FFT) on a grid whose size can be specified by the user (Sault et al., 2008). It also generates a beam map that gives the response of the interferometer to a point source based on the location of the antennae in the array and the frequency being observed. The beam map is necessary because `invert` alone tends to produce maps of objects that are filled with sidelobes: bumps and peaks and stripes that arise in the map when the *uv*-coverage of the array is irregular (Högbom, 1974; Thompson et al., 2001; Sault et al., 2008). In interferometry the *uv*-coverage is often irregular due to the fact that the placement of dishes is limited to the geography of the region (like on Mauna Kea for the SMA) and specific dishes often need to be excluded from the data due to malfunction (Högbom, 1974). The beam map, because it is made with observations of a point source, can tell us what exactly the pattern of sidelobes is and makes it possible to deconvolve the sidelobe pattern from the dirty map, leaving behind the *approximately* real structure. The process for doing this is called Högbom iteration (Högbom, 1974) or ‘CLEAN’ and is implemented by the `clean` task in *Miriad* (Sault et al., 2008).

The `clean`ing of the dirty map is performed by subtracting the entire dirty beam from the position where the dirty map is maximum. The amplitude of the dirty beam is usually normalized to be a fraction of the maximum of the dirty map, called the loop gain (Högbom, 1974). This subtraction is repeated on what remains of the map until nothing is left. A model is then built by placing point sources at all the positions where a peak was removed at the corresponding amplitude that it was removed at. By then analytically calculating a “clean” beam, which is the response to a point source given the diameter of a single dish, the model can be convolved with the clean beam using the `restor` task to obtain a clean map (Högbom, 1974; Thompson et al., 2001; Sault et al., 2008). Figure 3 demonstrates the `clean`-ing process on the IRC+10216 data presented in the attached paper. Note that the CLEAN algorithm works on extended sources like nebulae or molecular clouds essentially by modelling them as a collection of point sources (Högbom, 1974).

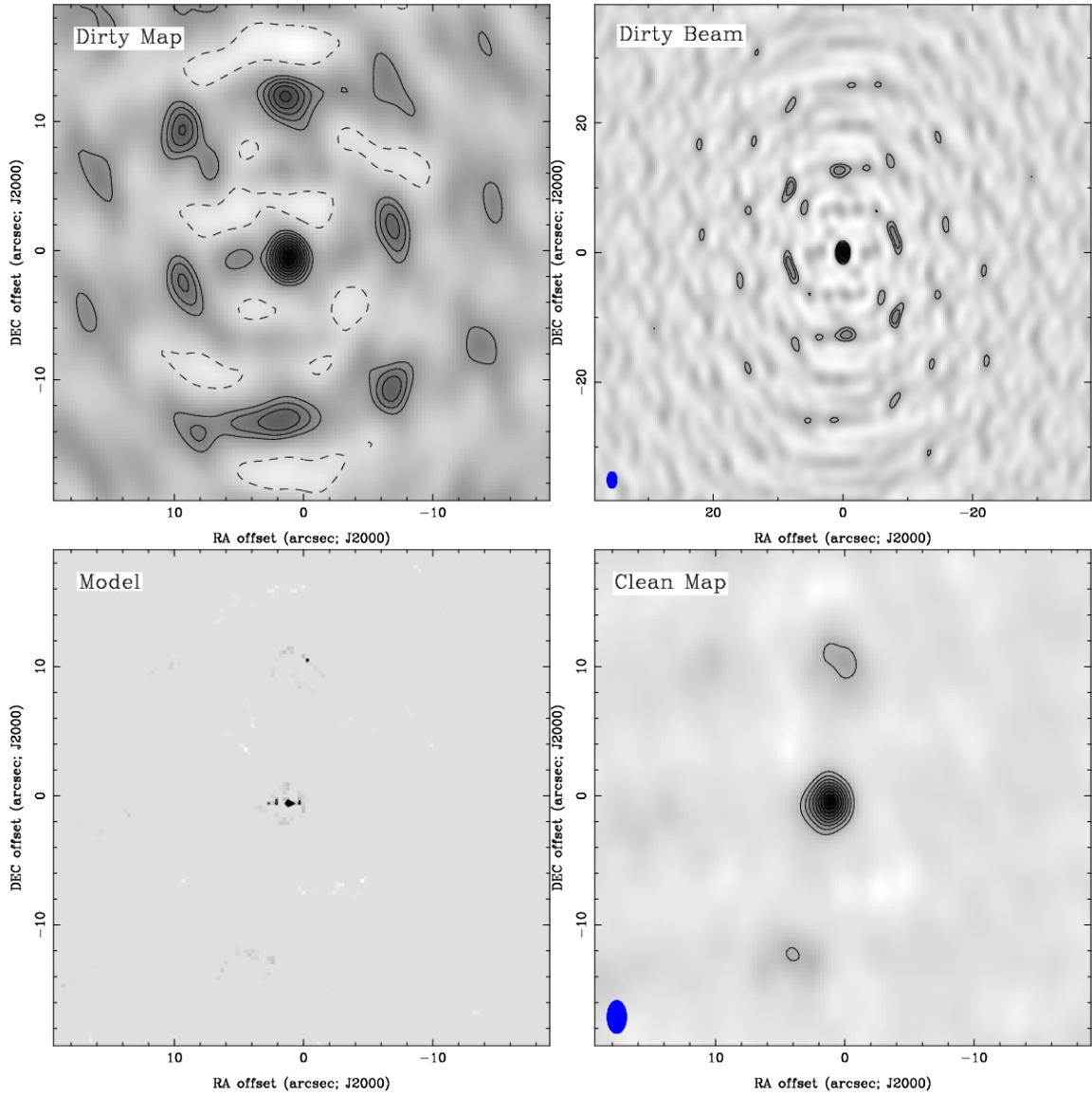


Figure 3: The dirty map (*top left*), dirty beam (*top right*), model obtained from the clean task (*bottom left*), and deconvolved CLEAN map of IRC+10216 (*bottom right*). Contour levels are at -20%, 20%, 30%, 40%, etc. of the maximum. Negative contours are dashed. Pixels show the same data, and darker pixels represent higher intensity. Notice the six peaks around the larger central peak in the dirty map that are also present in the dirty beam but are largely absent in the clean map.

Another task critical to the results presented in the paper is `selfcal` (self-calibration), which makes additional corrections to the gain calibration of the data (Sault et al., 2008). As explained in the paper this is necessary to correct the very small offset between beams of different polarizations that leads to false Stokes  $V$  signals. The `selfcal` task takes the clean model found from `invert-ing` and `clean-ing` the initial visibilities and uses it to find new gain coefficients that better fit the model. The gain coefficients can be further improved by repeating the self-calibration process on a new model generated from the last gain coefficients to be found (Sault et al., 2008). The gain coefficients (which are complex-valued) are searched for and found iteratively by minimizing the following error heuristic I

$$\epsilon^2 = \sum |\mathcal{V}_{ij} - g_i g_j^* \hat{\mathcal{V}}_{ij}|^2, \quad (11)$$

where  $\mathcal{V}_{ij}$  are the measured visibilities found by correlating signals from antenna  $i$  and  $j$ ,  $\hat{\mathcal{V}}_{ij}$  are the model visibilities for the same baseline, and  $g_x$  are the complex gain coefficients (Schwab, 1980; Sault et al., 2008). By finding the gain coefficients that minimize eq. (11), `selfcal` adjusts the measured visibilities until they match the model, and, when models are regenerated after each iteration of `selfcal`, the gain coefficients eventually converge and further iterations do not improve the image (Schwab, 1980; Sault et al., 2008). The self-calibration process can also be applied on visibilities with different polarizations, which is what allows us to correct the offset in each polarized beam.

While *Miriad*'s tasks make it easy to create scripts to automate data reduction and correct the data, the tasks themselves take parameters that must be manually found based on the data set being used, and are usually different across different objects. Using *Miriad* to search the data for Stokes  $V$  signals was initially a monumental challenge since our archive search started with over 25 different sets of data of

different celestial objects taken with different spectral configurations. For example, an early step in the `invert`-ing process is to average together the spectral channels that make up a spectral line (e.g. CO ( $J = 3 \rightarrow 2$ ) at 345.8GHz) so that the inversion process completes successfully. This is done by specifying the number of channels to end with after averaging, the velocity (in km/s) of the channel to start averaging at, and the number of channels to merge together. This involved manually opening the spectra of interest, counting the correct number of channels to cover the entire line, and then switching the x-axis of the displayed spectra to find the corresponding velocity of the starting channel. These values are then passed to the `invert` task like so:

```
1 invert line=vel,18,-43,2,2 vis=[...]
```

where the format is `line=vel,{# of channels},{starting channel},{step},{# to merge}`. The `invert` task has a bug where if the step size does not agree with a hidden rule I was unable to ascertain, it would fail, requiring you to increase the step size and manually recount the number of channels to input as the first number. Since each dataset required a correction script to be written for it, and we wanted to search as much data as possible, inefficiencies like the example above quickly made progress difficult and cumbersome.

To improve the workflow and automate little menial details like counting it was necessary to automate parameter generation for *Miriad*'s tasks. In a language like bash, which *Miriad* supported scripting for natively, such automation would bring its own cumbersome details, and it became clear a language like Python would be better suited for the task. No Python wrapper existed for *Miriad* so I created one and used it to generate any tedious parameters needed for any of the *Miriad* tasks, including detecting when `invert` fails due to the mysterious step bug and automatically changing the step size until the task completes. Other notable features include dumping spectral information to the terminal and parsing it in Python for information needed

to generate other parameters, counting channels automatically, displaying a window with a spectrum that outputs channel information on mouse click, and, in general, allowing one to take advantage of the language features of Python such as loop structures, function definitions, and string substitution. For example, the Python method used to automate the line selection parameter for `invert` is the following:

```
1 def averageVelocityLine(vis, factor):
2     velrange = getVelocityRange(vis)
3     nvels = round(abs(velrange[0] - velrange[1])/factor)
4     startvel = round(velrange[1])
5     return 'vel,{0},{1},{2},{2}'.format(nvels, startvel, factor)
```

*no indent* → which, under the hood, gets the velocity range of the line by using the output of the `uvlist` task in *Miriad* and parsing it for the velocity range before using it to compute the necessary number of channels and building the necessary string in the format that *Miriad* expects.

The Python *Miriad* wrapper works by generating the necessary bash command to run and then passing the command to the terminal. Depending on the task, the output can be displayed in the terminal as usual or can be intercepted by Python and used for other tasks.

Having a Python interface to *Miriad* also made it possible to connect the data to the plotting utilities available in Python, such as `matplotlib` (Hunter, 2007), and also allowed easier access to *Miriad*'s own plotting utilities like `cgdisp` which is used to display contour plots of maps.

Creating an automation suite for *Miriad* in Python and connecting it directly to several different custom plotting methods made it possible to correct more data, find potential Stokes *V* signals across different spectral lines more efficiently, and create the many figures and maps required to convince ourselves of whether the signals were real or not.

## 2.3 Code Repositories

All the code that performs the data reduction, squint correction, and plotting is available online.

The data analysis and correction scripts can be found at [github.com/mef51/SMAData](https://github.com/mef51/SMAData). This repository contains all the archival polarimetric data from the SMA (minus the actual data, scripts only). The folders relevant to the attached paper are `IRC+10216/`, `080106_Ram_Spectra/` (for Orion KL), `141028_ngc7538/` (for NGC7538), and `101014_NGC1333_Ram/` (for NGC1333).

Within that repository is the `squintscripts/` folder which contains all the scripts for generating the figures included in the paper, as well as some extra figures. The `paperplots.py` file contains some utilities for styling the figures. The `GetCOMaps.py` file generates all the maps used in the paper. `CompareImageSpectra.py` generates the before-and-after correction comparison of the spectra (Figure 4 of the attached paper).

The Python wrapper for *Miriad* described in the previous section lives in its own repository due to its potential use for others and can be found at [github.com/mef51/smautils](https://github.com/mef51/smautils). It contains the wrapper as well as a library (`squint.py`) for performing the squint correction detailed in the paper. It is published on the Python Package Index (PyPi) and can be installed by anyone by running the command ‘`pip install smautils`’ on the terminal.

## 3 Acknowledgements

I'd like to thank my supervisor Martin Houde for his advice and guidance during

the last year and for sending me to Hilo, Hawaii, for two weeks as soon as I finished my classes despite my having only a vague idea of what my project would entail and exactly zero research experience in Astronomy. Baptism by fire works.

*I am*

*Dr*

*Prof.*

I'm also grateful to Ramprasad Rao and Josep Miquel Girart who not only made my visit to Hilo productive but also warm and fun. Thank you Josep Miquel and Miquel for accompanying me on my drive up Mauna Kea the night before our respective flights.

*I am*

I'm grateful to my friends, roommates, and the rock climbing crew for giving me things to do that are happily unrelated to work.

## 4 References

- Chandrasekhar, S., & Fermi, E. 1953 ApJ, 118, 113
- Cortes, P. C., Crutcher, R. M. & Watson, W. D. 2005, ApJ, 628, 780.
- Cotton, W. D., Ragland, S., & Danchi, W. C. 2011, ApJ, 736, 96
- Crutcher, R. M. 2012 Annu. Rev. Astron. Astrophys, 50, 29-63
- Davis, L., Jr. 1951, PhRv, 81, 890
- Girart, J. M., Crutcher, R. M. & Rao, R. 1999, ApJ, 525, L109.
- Girart, J. M., Rao, R., Marrone, D. P. 2006 Science, 313, 5788, 812-814
- Girart, J. M., Patel, N., Vlemmings, W. H. T., & Rao, R. 2012, ApJ, 751, L20.
- Glenn, J., Walker, C. K., Bieging, J. H., et al. 1997, ApJ, 487, L89.
- Goldreich, P., Kylafis, N. D. 1981 ApJ, 243, L75-L78
- Greaves, J. S., Holland, W. S., Friberg, P., et al. 1999, ApJ, 512, L139.
- Hamaker, J. P., Bregman, J. D., & Sault, R. J. 1996 A&A Suppl. Ser., 117, 137-147
- Hezareh, T., & Houde, M. 2010, PASP, 122, 786

- Hezareh, T. J., Wiesemeyer, H., Houde, M., Gusdorf, A., Siringo, G. 2013 *A&A*, 558, A45
- Hildebrand, R. G., Kirby, L., Dotson, J. L., Houde, M., Vaillancourt, J. E. 2009 *ApJ*, 696, 567-573
- Högbom, J. A. 1974 *Astron. Astrophys. Suppl.*, 15, 417-426
- Houde, M., Vaillancourt, J. E., Hildebrand, R. H., Chitsazzadeh, S., & Kirby, L., 2009 *ApJ*, 706, 1504
- Houde, M., Hezareh, T., Jones, S., & Rajabi, F. 2013 *ApJ*, 764, 24
- Houde, M. 2014 *ApJ*, 795, 27
- Hull, C. L. H., Plambeck, R. L., Kwon, W., Bower, G. C., Carpenter, J. M., Crutcher, R. M., et al. 2014, *ApJ Suppl.*, 213, 13
- Hunter, J. D. 2007, *Computing in Science & Engineering*, 9, 3, 90-95
- Lai, S.-P., Girart, J. M. & Crutcher, R. M. 2003, *ApJ*, 598, 392.
- Muñoz, D. J., Marrone, D. P., Moran, J. M., & Rao, R. 2012 *ApJ*, 745, 115
- Marrone, D. P. & Rao, R. 2008, *Proc. SPIE* 7020, 70202B
- Nakano, T., Nakamura, T. 1978 *Publ. Astron. Soc. Japan* 30, 671-679
- Sault, R. J., Teuben, P. J. & Wright M. C. H. 1995 *ADASS IV*, ASP Conference Series, 77, 433-436
- Sault, R. J., Hamaker, J. P., & Bregman, J., D. 1996 *A&A Suppl. Ser.*, 117, 149-159
- Sault, R. J., Killeen, N. 2008, *Miriad Users Guide*, Australia Telescope National Facility

Schwab, F. R. 1980, Proc. SPIE 0231, Intl Optical Computing Conf I

Tang, Y.-W., Ho, P. T. P., Koch, P. M., & Rao, R. 2010, ApJ, 717, 1262.

Thompson, A. R., Moran, J. M., & Swenson Jr, G. W. 2001, Interferometry and synthesis in radio astronomy (John Wiley & Sons)

Vanderlinde, K. “Radio Telescope Fundamentals”, slide show presentation, Dunlap Summer School, July 2017