WingFind: A New Algorithm for Analyzing the Velocity Structure of Molecular Clouds

David Chilton

February 11, 2006

Abstract

WingFind is a new algorithm for finding velocity turbulence ("wings") in molecular clouds. It complements *ClumpFind* well and is used to connect turbulence with dense regions of the cloud ("clumps"). It has been tested with data generated by a spline function and applied to data taken of the CO(J=1-0) emission line of MBM16. In comparison with the *ClumpFind* analysis of the molecular cloud MBM16, the gravitational nature of the dense regions of MBM16 is still uncertain.

Contents

Figures

1 Introduction

1.1 The Interstellar Medium and Molecular Clouds

The interstellar medium (ISM hereafter) is all the space that does not consist of stars and objects gravitationally bound to those stars. The ISM consists largely of dust and gas [Ferrière, 2001], but it also contains various non-stellar structures, such as nebulae and molecular clouds. Molecular clouds, loosely bound collections of molecules floating in space, account for half of the mass in the ISM in the solar circle (the volume of space contained within the Sun's orbit). These clouds are important because star formation activities have been observed in all types of molecular clouds. They have been observed to be highly fragmented, showing several discrete "clumps" [Williams et al., 1994]. In fact, Williams states that all clouds observed at high resolution exhibit a clumpy internal structure. Additionally, one study has noted a velocity turbulence in the cloud MBM16, which could be significant in understanding the formation of the clumpy structure of the cloud [Larosa et al., 1999].

Radio astronomers use doppler shifts to determine the velocity that an object is moving relative to the Earth. After tuning the telescope dish to a certain frequency that will get the doppler shift of radiation emitted by a specific element of their choice at a specific velocity, they record the change in the antennae temperature as an indication of the amount of matter in the the target location moving with the chosen velocity. Then, the antennae are set to a new frequency and the measurement is repeated. Each frequency is referred to as a *channel.* MBM16 (a molecular cloud labeled in a paper by Magnani et al. [1985]) was discovered to have velocity turbulence by noting that for several different interstellar coordinates scanned, each had a wide peak spanning several channels.

MBM16 is considered a translucent molecular cloud. The critical meaning of this is that translucent clouds are not gravitationally bound. This is specifically true of MBM16, which was determined to not be gravitationally bound by Magnani et al. [1985]. Since it has a clumpy structure, but is not self gravitating, it is important to understand the structure of the velocity turbulence with respect to the clumps in order to understand how these clumps may have been formed and how the denser, star forming clouds may have formed. Figure 1 shows a wing in the data collected by Magnani et al. [1999]. The data is calibrated using an automatic baselining algorithm to center the 0 in the middle of noise. The broad region around 5 km/s characterizes a wing.

1.2 Methods of Analysis

There are two widely-used algorithms for analyzing the clumpy structure of molecular clouds, *ClumpFind* [Williams et al., 1994] and *GaussClumps* [Stutzki and Güsten, 1990]. A recent paper by Schneider and Brooks [2004] discusses the methods in depth and compares their results. They state that while *GaussClumps*

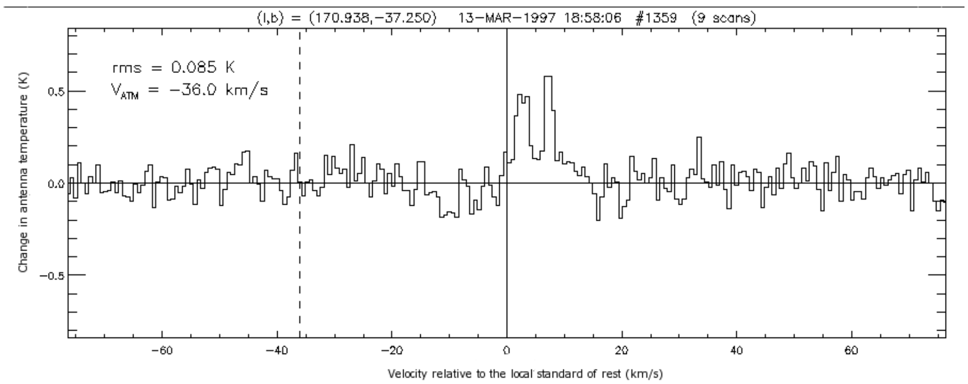

Figure 1: **A wing from the data gathered by Magnani et al. [1999]. It is a sum of the nine scans that touch 170.938 Galactic Latitude and -37.350 Galactic Longitude.**

found more clumps of smaller size than *ClumpFind*, the sum of the masses of the clumps found by each were the same. This implies that although neither method failed to categorize any point as being in a clump, they did differ on assigning the points to a specific clump. Additionally, the *ClumpFind* method searches in a manner more consistent with a manual optical scan of the data. It starts at the highest point of the data and groups clumps together as it moves down from a peak. *ClumpFind* allows the user to specify the lowest value to include and the increment between contour levels to use to search. It is implemented in IDL (Interactive Data Language), a very powerful (and expensive) data analysis suite from Research Systems Inc., and in Fortran. Since *ClumpFind* results were available for MBM16, it was chosen as the clump search method for this analysis.

There is not an existing analytical tool to locate the areas of turbulence, "wings", within the cloud or to connect these wings to the clumps. *WingFind* is an algorithm that attempts to fill this gap. *WingFind* walks along the the data and looks for wings. *WingFind* has been tested on data generated using a spline curve and found to be effective in searching an ideal curve until the signal to noise ratio is approximately 6.25. Below that, the effectiveness decreases linearly. It has been used to analyze the velocity substructure of MBM16 and finds wings in several parts of the clouds, including several neighboring wings that intercept one of the denser clumps of the cloud.

2 *WingFind*

2.1 Design

WingFind was devised to analytically locate wings in the cloud. As described by Listing 1, it walks down the velocity vs antennae temperature spectrum for each galactic coordinate scanned and looks for points above a given multiple (m) of the root-mean-square (rms) value of the spectrum. When it finds a channel at or above $m * rms$, it counts how many channels are at or above $m * rms$ until the spectrum is below $m * rms$ for more than a given number of channels, called the *noise allowance*. If the number of channels found above $m * rms$ was at least a specified number of channels, called the *velocity interval*, the channels are marked as wings. This listing ignores the extra boundary checking code for the sake of simplicity. *WingFind* is a fairly simple algorithm, operating in linear time ($O(n)$), where n is the total number of elements in the cube comprised of latitude vs. longitude vs. velocity.

Listing 1 Search For Wings

```
for all $lat$ in $CloudLatitude$ do
  for all $long$ in $CloudLongitude$ do
    $rms \leftarrow RMS(lat, long)$ {$RMS(lat, long)$ gets the root-mean-squared value of the antennae temperature along the spectrum for a given latitude and longitude}
    for all $channel$ in $CloudChannels(lat, long)$ do
      if $channel \geq multiplier * rms$ then
        $noise\_count \leftarrow 0$
        $band\_count++$
      else
        $noise\_count++$
        if $noise\_count \geq noise\_allowance$ then
          if $band\_count \geq band\_width$ then
            mark as wing
          else
            mark as not wing
          end if
          $band\_count \leftarrow 0$
          $noise\_count \leftarrow 0$
        end if
      end if
    end for
  end for
end for
```

2.2 Implementation Details

2.2.1 Overview

WingFind is implemented in C++, a programming language that is a set of extensions to C to support classes (see below) and other advanced features, as part of a package called *WingClumps*. *WingClumps* supports loading FITS (*Flexible Image Transport System*, the standard data format in astronomy, also used in some medical applications) files and running *WingFind* on the data in the FITS file. Although *WingClumps* is primarily a collection of class templates, it also has portions which must be compiled into a library and linked against to execute. Class templates allow the programmer to design standard operations without knowing ahead of time exactly what type of data, like integers or floating-point numbers, will need to be used. They also allow the developer to support multiple types while only having to write the real code once. This is important when dealing with FITS files, which can have several different data types.

One of the features of C++ is the concept of classes. A class contains data and operations on that data. For example, there could be a class `Car` with data `tires` and the operation `drive`. If `johnsCar` is a `Car` it is called an *instance* of class `Car`. Telling `johnsCar` to `drive` would in turn tell the `tires` to drive. A *static* function or variable is something that is the same for any instance of a class and calling it or changing it effects all instances of that class. If class `Car` had a static variable `numTires`, then changing `numTires` would change it for `johnsCar` and `suziesCar`. If `Car` had a static function `allStop`, then calling `allStop` on `Car` would cause both cars to stop moving.

Data in *WingClumps* is stored in a `valarray`. A `valarray` is a variable-length array which is optimized for numerical operations, and can perform operations easily on all of its data. It is one-dimensional, so *WingClumps* includes a class called `WCMatrix` which provides a three-dimensional index into the one-dimensional data by knowing, through the use of static variables, how long each axis of the three-dimensional matrix is supposed to be. *WingClumps* also provides `WCMatrixCoord`, which provides easy conversion between the one-dimensional index and the three-dimensional spacial coordinates. It also allows for easy distance calculations to find the nearest neighbor. Figure 2 shows a simplified diagram of the classes. There are several more helper classes in the library. The classes shown in the figure are the important ones that matter for implementing a program with the library.

WingClumps uses the CCfits library to access FITS files from C++ (NASA [2004]). It also provides an abstract class, an empty shell defining required functions but not implementing them, `WCFile` which could easily be extended to implement access to other file formats.

2.3 Testing

2.3.1 Ideal Wing Generation

Test data was created with a cubic-spline function. The cubic-spline function uses a few known points to extrapolate the entire curve along a 3^{rd} degree polynomial. The generator randomly chooses if the coordinate is to contain a wing and then randomly selects a starting channel for the wing. It then gives each

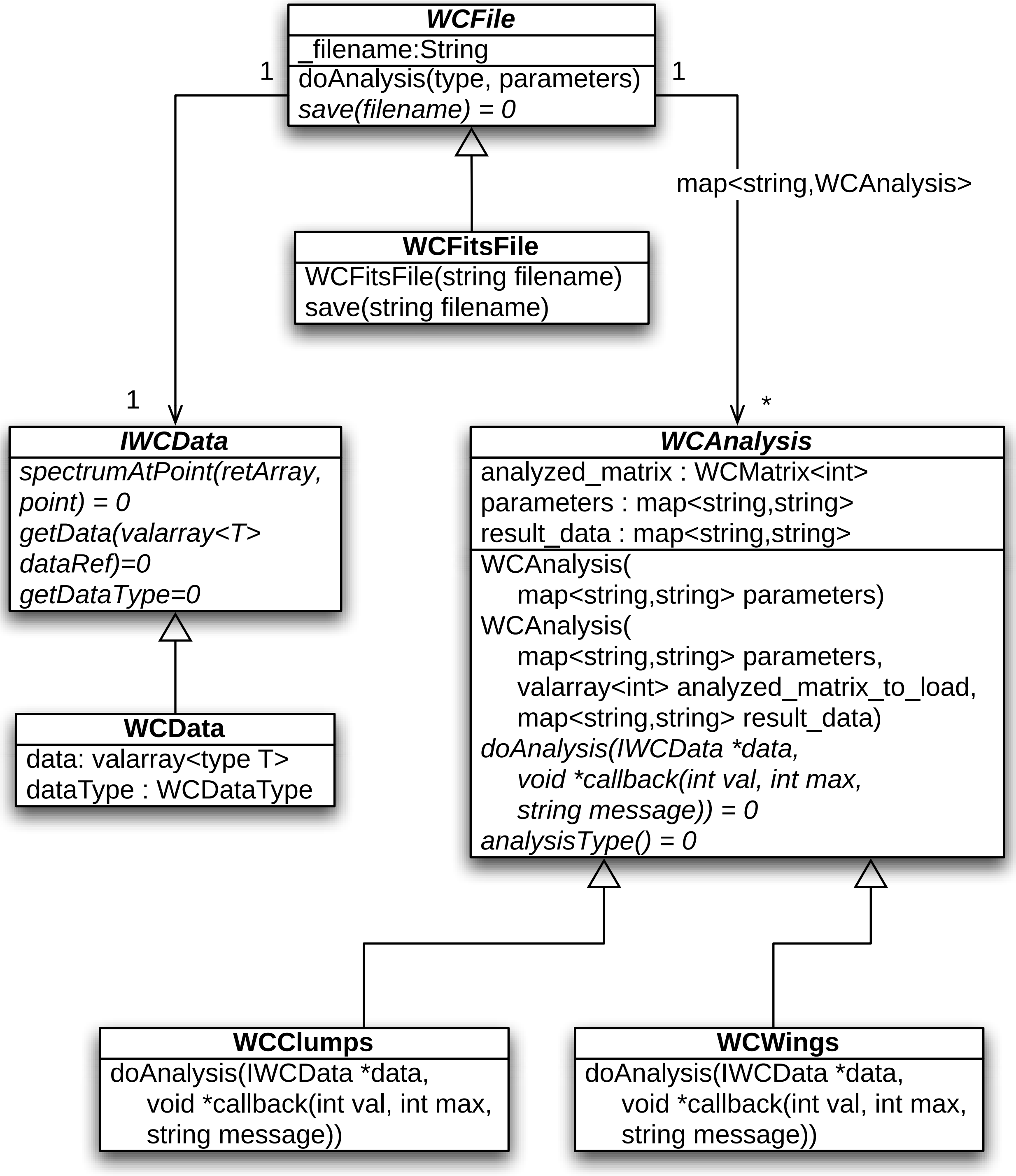

Figure 2: **A basic diagram showing the class relationships.**

point an intensity generated by Equation 1, which defines $q_i(x)$, a function to find a y-value for $x \in (x_{i-1}, x_i]$ (Dahlquist and Björck [1974]). A seed array, $\{0,0,4,4.25,5,4.25,4,0,0\}$, was used for the y_i and there was an interval of 2 between each x_i. The solution to the system 2 gives the values for k_i.

$$\begin{aligned} q_i(x) &= ty_i + (1-t)y_{i-1} + h_i t(1-t)[(k_{i-1} - d_i)(1-t) - (k_i - d_i)t] \qquad (1) \\ &\text{for} \quad i = 1, 2, 3, \dots, m \end{aligned}$$

where

$$\begin{aligned} h_i &= x_i - x_{i-1} \\ d_i &= \frac{y_i - y_{i-1}}{h_i} \\ t &= \frac{x - x_{i-1}}{h_i} \quad \text{for} \quad x \in [x_{i-1}, x_i] \\ \text{and} \quad & k_0, k_1, ..., k_m \quad \text{satisfy the system of equations:} \\ & h_{i+1}k_{i-1} + 2(h_i + h_{i+1})k_i + h_i k_{i+1} = 3(h_i d_{i+1} + h_{i+1} d_i) \qquad (2) \\ & \text{for} \quad i = 1, 2, 3, \dots, m-1 \end{aligned}$$

Since a step of 2 was used to generate the test data, it only has to generate one unknown point in between each pair of known points, and y_i and x_i are given by the seed values. Noise was added to the array by multiplying a random decimal between -1 and 1 by a scaling factor between 0 and 1 and then adding that value to the previous value at a coordinate. The scaling factor is used to select the maximum amount of noise for each test. This gives a good amount of noise for testing the algorithm. Each point in the data set has an amount of noise. Figure

3 shows the ideal wing generated by the cubic-spline without any added noise. The circles mark the input points to define the curve. The axes on the graph have generic units, they can represent any units of temperature and velocity. They represent matrix units, not real world units.

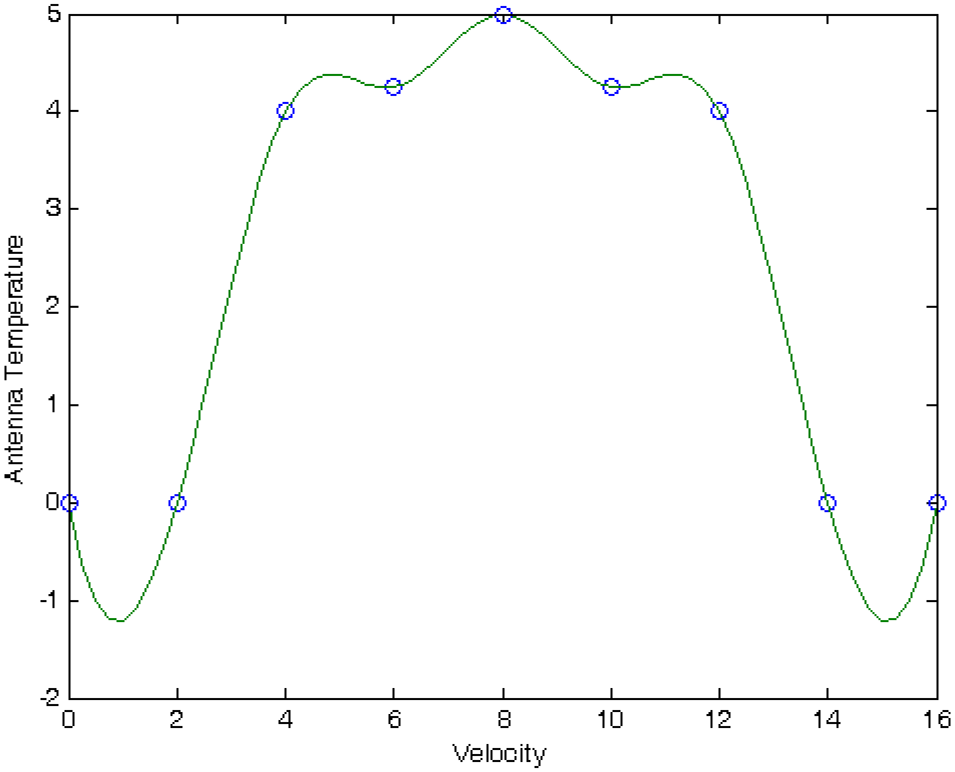

Figure 3: **The Ideal Wing Generated by the Spline Curve**

2.3.2 Results

Each test set generated 242 wings, at random positions in the data set. The generator added varying levels of noise to the data sets. They were tested with multipliers between 1.5 and 2.5, inclusive, at 0.25 increments. The 1.5 multiplier consistently across all noise found approximately the same number of wings as were generated, but at low noise levels it found more than were generated, leading to some doubt as to the results for that multiplier. The multiplier value of 1.75 was effective at finding wings until the noise reached ±0.8 (which is at Noise Level 80 in Figure 4). Figure 4 shows how changing the multiplier affected the effectiveness of the algorithm.

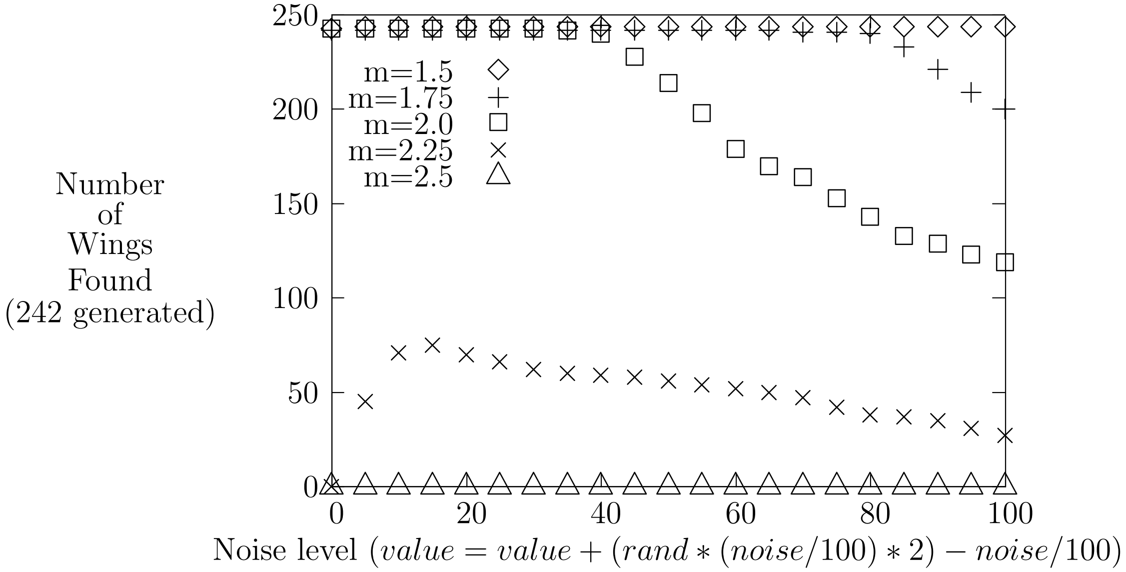

Figure 4: **Effectiveness of *WingFind* as Multiplier Increases**

3 MBM16

3.1 The Cloud

MBM16 is located in the Southern Hemisphere of The Milky Way, near the Taurus Molecular Clouds. It was first recorded by Magnani, Blitz, and Mundy in 1985 (hence the name MBM16, as it was the 16th cloud described in that paper.) Recently more work has been done to characterize the environment of this cloud, including an observation of the velocity turbulence (Larosa et al. [1999]). Spectral data of MBM16 were gathered by Magnani et al. [1996] using the Harvard-Smithsonian 1.2m radio telescope, which picks up signals with wavelengths on the order of a few millimeters. It is tuned to the wavelength of a photon emitted by the energy change between the first and base states of carbon monoxide, called the CO(J=1-0) transition. The original data extended out much further in the velocity dimension, from approximately -60 km/s to 60 km/s, but the cube was clipped to only the interesting channels, from approximately -20 km/s to 20 km/s. Since the data were truncated, the original data were lost in a hard drive crash, so only the truncated set of data was available for analysis, but it was still sufficient.

There are three types of molecular clouds: diffuse, translucent, and dark. They are formally defined by the density, but they are also distinguished by the form of astrochemical interaction which dominates in the cloud. In diffuse clouds, whose CO(J=1-0) emission line is too faint to pick up on a millimeter wavelength telescope, the dominant process is a photoprocess, a chemical process triggered by light and not molecular interactions. Dark clouds, which are star-forming, have a collisional process which dominates. Translucent clouds lie in the middle, with some photo and some collisional interactions. They occupy the transitional space in between [Magnani et al., 1996].

MBM16 is a translucent molecular cloud. It was chosen for study because it was already known to have pronounced clumps and a distinct velocity substructure [Larosa et al., 1999, for example]. The *ClumpFind* algorithm was applied to the raw data with a contour step of 0.50 and starting at 0.0. Shore et al. [2003] postulate that the internal velocity substructure may be caused by a shear flow external to the cloud, and not caused by gravity of the dense regions.

3.2 Clump and Wing Analysis of MBM16

Figure 5 shows how the number of points found to be in a wing versus the number of points found to be in a wing and a clump changed as the cutoff multiplier increased. At the 2.5 multiplier, only 16 points were found to be in a wing, but 12 of them overlapped a clump, giving the 0.75 fraction shown in the figure. This indicates that there may be a connection between the wings and the clumps. However, since the lower multiplier levels find many more points in a wing than in a wing and a clump, there is reason to doubt this conclusion. This conclusion is further complicated by the location of the wings in the cloud as shown in Figures

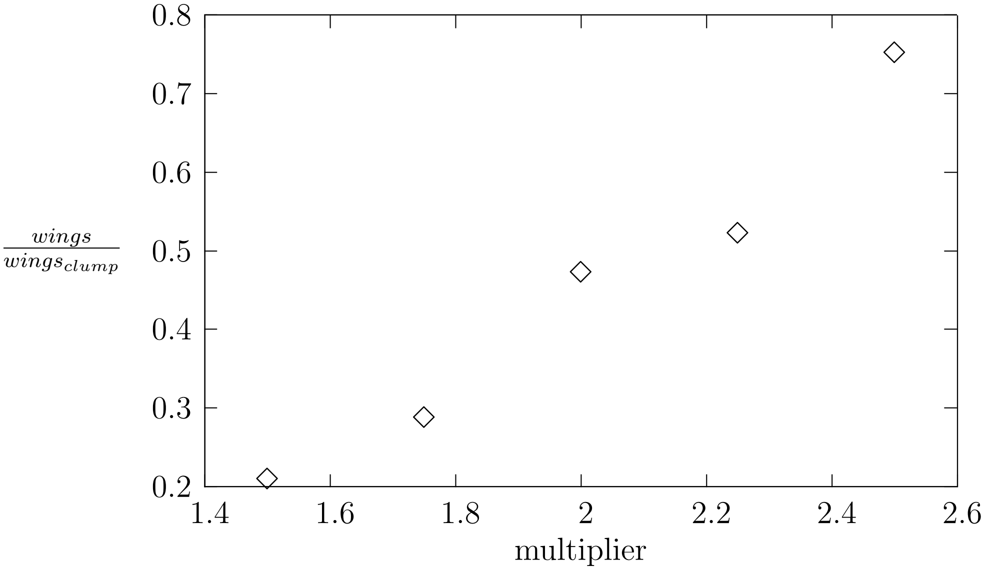

Figure 5: **Fraction of Wings in a Clump as Multiplier Increases**

7 through 11. The lines represent contour levels and the purple stars mark the points that are wings and clumps. The red level is the lowest contour level, green is next, and blue is the highest. Each level represents increasing levels of cloud density. At the 2.5 multiplier level, most of these points are next to the highest part of the contour, not on top of it as would be expected if there were a direct correlation between the clumps and the wings. However, the wings are still close enough to the clumps, and are on top of a definite clump, to make the situation inconclusive either way. The lack of velocity turbulence, especially in the dense northern region, would indicate a gravitational formation of dense regions. However, the presence of velocity turbulence in close proximity to the clumps in the southern region of the cloud is indicative of an external flux causing the dense

regions to form. The contradictory results of this analysis is confounding and inconclusive as to the nature of MBM16.

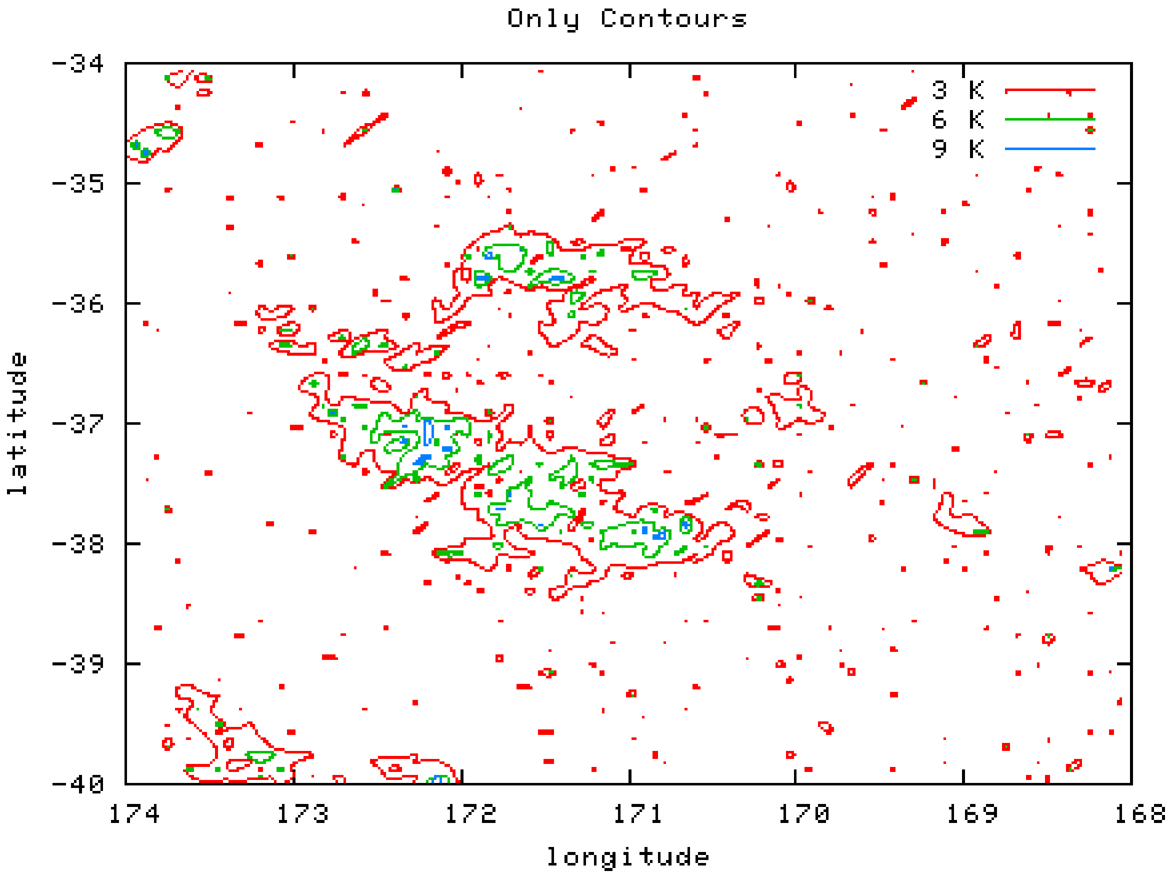

Figure 6: **Just the contour map of MBM16**

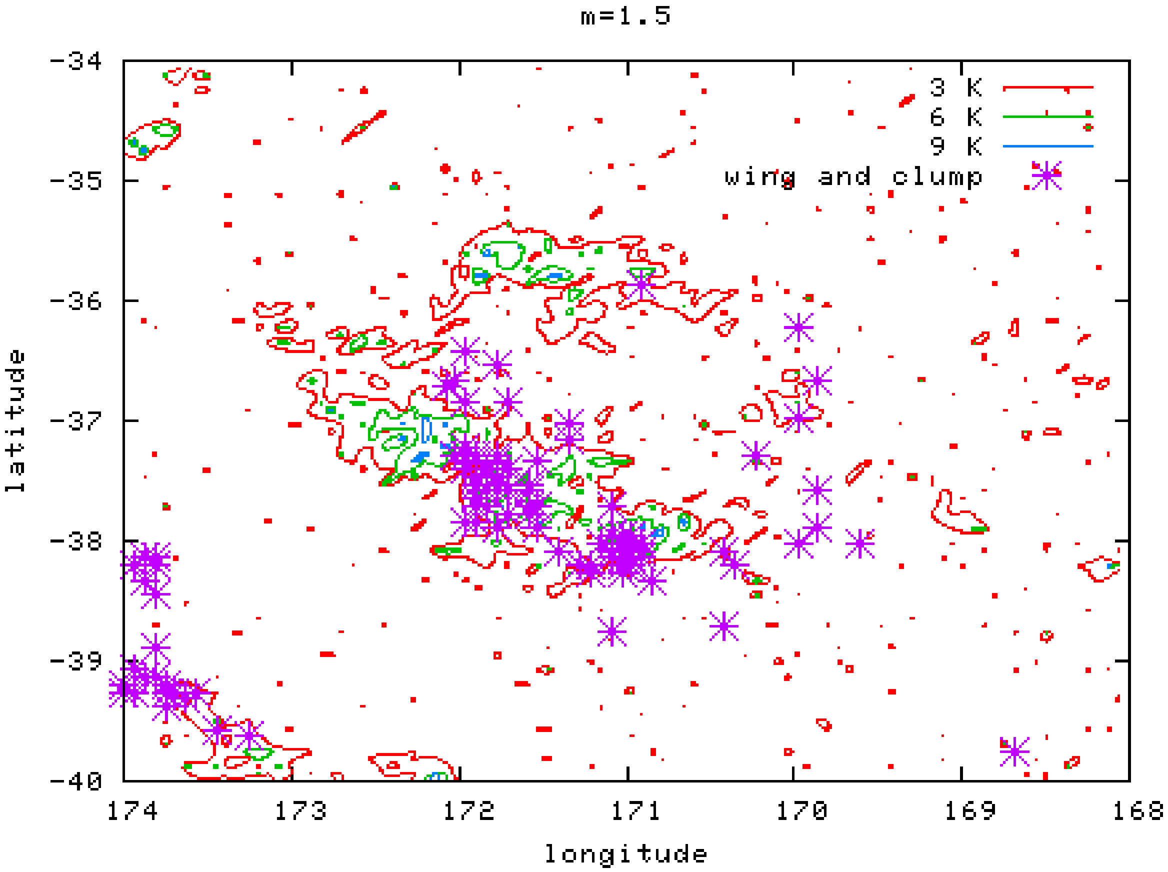

Figure 7: **Wings and Clumps on top of MBM16,** $m = 1.5$

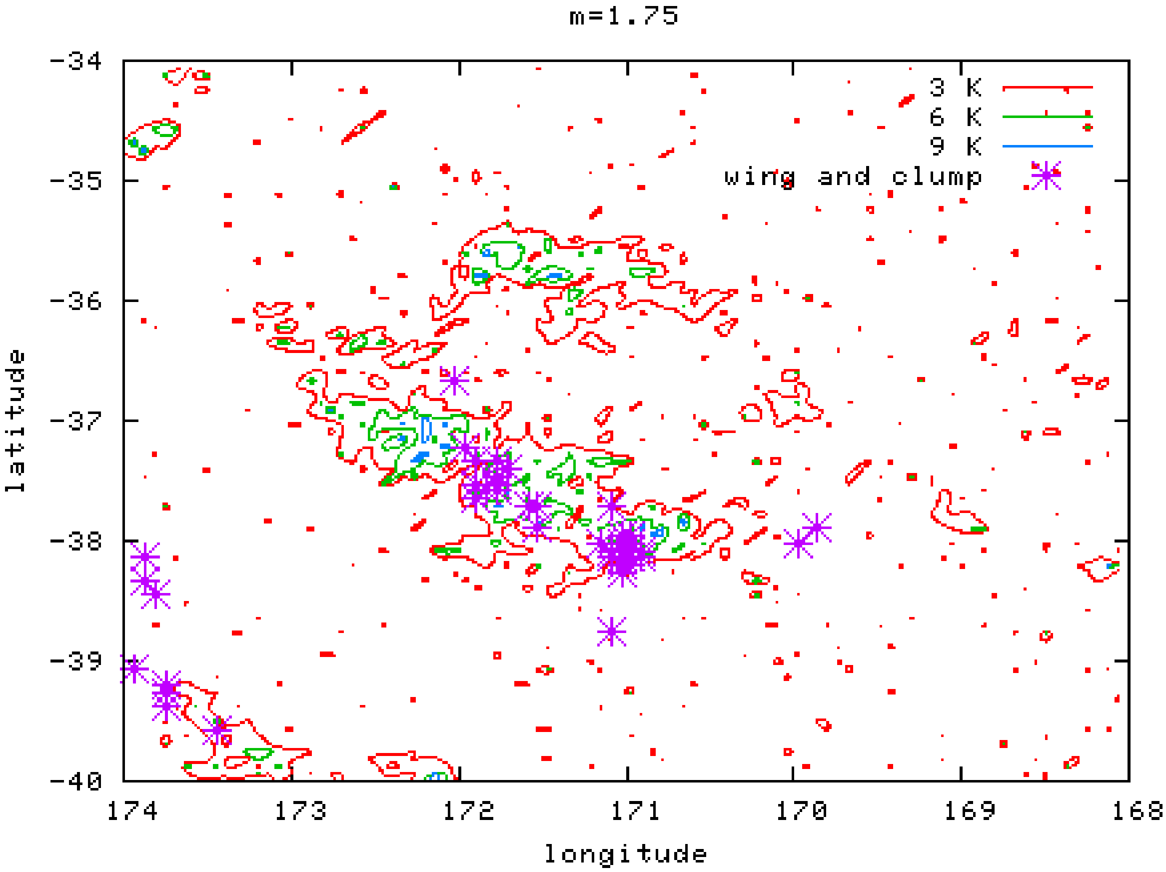

Figure 8: **Wings and Clumps on top of MBM16,** $m = 1.75$

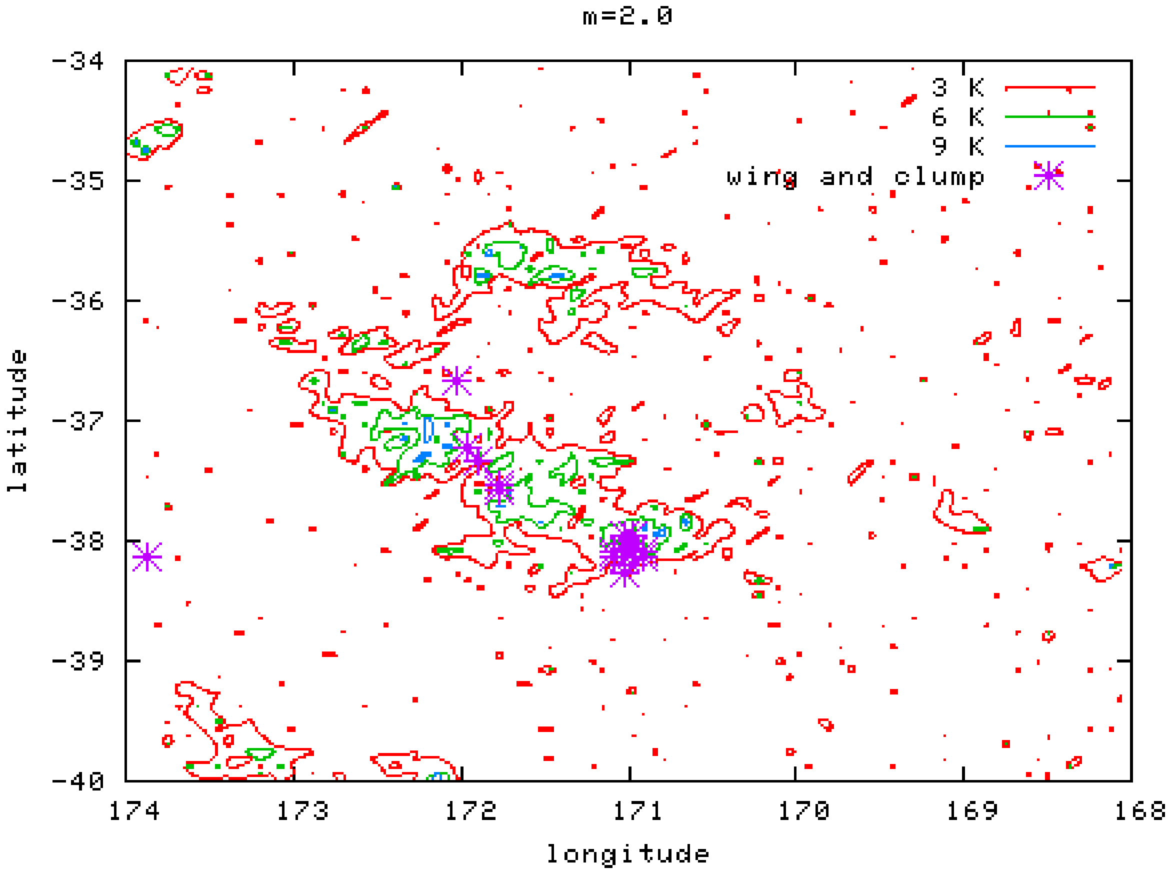

Figure 9: **Wings and Clumps on top of MBM16,** $m = 2.0$

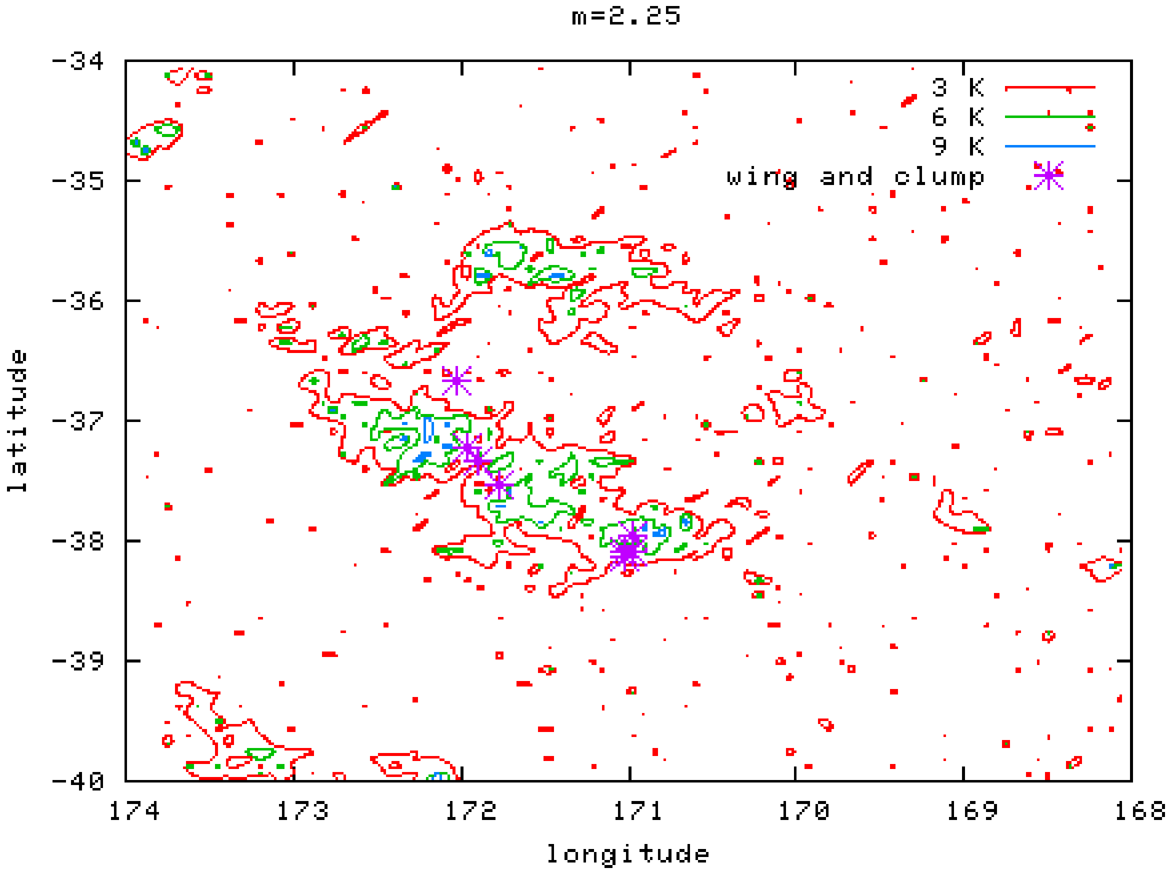

Figure 10: **Wings and Clumps on top of MBM16,** $m = 2.25$

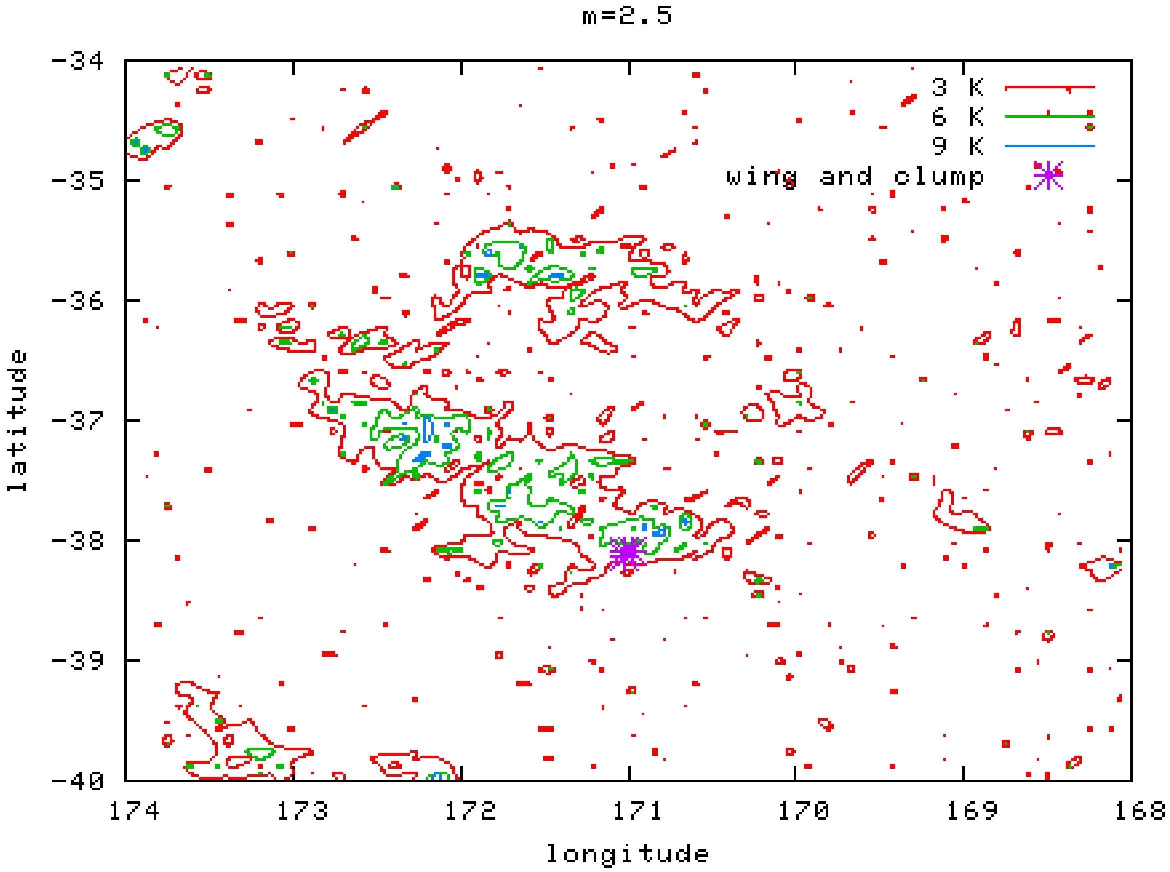

Figure 11: **Wings and Clumps on top of MBM16,** $m = 2.5$

4 Conclusion

WingFind is a new algorithm for locating velocity turbulence ("wings") in molecular clouds. It has been tested and found to be effective on test data generated with a spline function. Using a spline function, effectiveness falls off at a signal to noise ratio of 6.25. The algorithm was then applied to real data from the translucent cloud MBM16. It found many points where wings overlapped clumps, 80% of the points in wings were also in a clump when the cutoff multiplier was 2.5. Since some of the clumps that would be expected to have turbulence, especially in the northern part of the cloud around (171.5, -35.7), did not show wings, this is rather inconclusive as to the nature of the velocity turbulence and its relationship to the gravitational nature of MBM16.

5 Future Work

First and foremost, the *ClumpFind* implementation in *WingClumps* needs to be fully tested and corrected. As it is currently implemented, it is very slow and does not work correctly. To that end the C++ implementation of `search3d` needs to be extensively rethought and streamlined. The current design tends to cause a memory conflict because it is deeply recursive, which causes the program to crash around fifth hour of analysis. Another expansion of *WingClumps* is the development of `WCMatrix` into a more robust class and do away with `WCMatrixCoord`. A program to create a 3D visualization of the clumps and wings, displayed together, is planned, to make it easier to see where they relate. Ideally this would be able to be viewed on a Perspectra$^{\text{TM}}$Viewer (a 3D display that lets the user see the displayed object in true 3D, through the use of a rotating screen). Finally, much work needs to be done to make *WingClumps* a more cohesive suite of tools; it is currently held together by ad hoc shell and Ruby scripts.

6 Acknowledgments

I would like to thank my committee at the University of Tennessee at Chattanooga for their support

- Tatiana Allen, Ph. D. Department of Physics, Geology, and Astronomy
- Stephen Kuhn, Ph. D. Department of Mathematics
- Robert Marlowe, Ph. D. Department of Physics, Geology, and Astronomy
- Andy Novobilski, Ph. D. Department of Computer Science
- Jack Thompson, Ph. D. Department of Computer Science

I would also like to thank Loris Magnani at the University of Georgia for giving a freshman a shot at a project way over his head in the summer of 2002, and for his guidance through the years. Lastly, a great big thanks to the NSF for funding my summer REU project, which led me to do this project.

References

P. Burte. A visualization-based tool in support of high-throughput nuclear magnetic resonance studies. Master's thesis, University of Georgia, 2001.

G. Dahlquist and Å. Björck. *Numerical Methods*. Prentice Hall Inc., 1974.

T. M. Dame, D. Hartmann, and P. Thaddeus. The Milky Way in Molecular Clouds: A New Complete CO Survey. *Astrophysical Journal*, 547:792–813, 2001. doi: 10.1086/318388.

K. M. Ferrière. The interstellar environment of our galaxy. *Reviews of Modern Physics*, 73:1031–1066, 2001.

T. N. Larosa, S. N. Shore, and L. Magnani. A Dynamical Study of the Non-Star-forming Translucent Molecular Cloud MBM 16: Evidence for Shear-driven Turbulence in the Interstellar Medium. *Astrophysical Journal*, 512:761–767, 1999.

L. Magnani, L. Blitz, and L. Mundy. Molecular gas at high Galactic latitudes. *Astrophysical Journal*, 295:402–421, 1985. doi: 10.1086/163385.

L. Magnani, D. Hartmann, and B. G. Speck. A Catalog of Molecular Gas at High Galactic Latitudes. *Astrophysical Journal, Supplement*, 106:447–+, 1996.

L. Magnani, D. Hartmann, and P. Thaddeus. A Survey of High-Latitude Molecular Gas. *Bulletin of the American Astronomical Society*, 31:1477–+, 1999.

NASA. *CCFits Library and Documentation*, 2004. URL `http://heasarc.gsfc.nasa.gov/docs/software/fitsio/ccfits/`.

N. Schneider and K. Brooks. The Bias of Molecular Clump Identification Programs: the Example of the Carina Molecular Clouds. *Publications of the Astronomical Society of Australia*, 21:290–301, 2004. doi: 10.1071/AS04012.

S. N. Shore, L. Magnani, T. N. LaRosa, and M. N. McCarthy. Mechanisms for the Origin of Turbulence in Non-Star-forming Clouds: The Translucent Cloud MBM 40. *Astrophysical Journal*, 593:413–425, 2003. doi: 10.1086/376355.

J. Stutzki and R. Güsten. High spatial resolution isotopic CO and CS observations of M17 SW - The clumpy structure of the molecular cloud core. *Astrophysical Journal*, 356:513–533, 1990. doi: 10.1086/168859.

A. T. Truong, L. Magnani, and D. Hartmann. The Gravitational Stability of MBM16. *Bulletin of the American Astronomical Society*, 29:1220–+, 1997.

J. P. Williams, E. J. de Geus, and L. Blitz. Determining structure in molecular clouds. *Astrophysical Journal*, 428:693–712, 1994.