

An Introduction to

ALMA Spectral Cube Imaging and Analysis

Travis J. Thieme

Postdoctoral Fellow @ ASIAA / EA ARC Taiwan Member

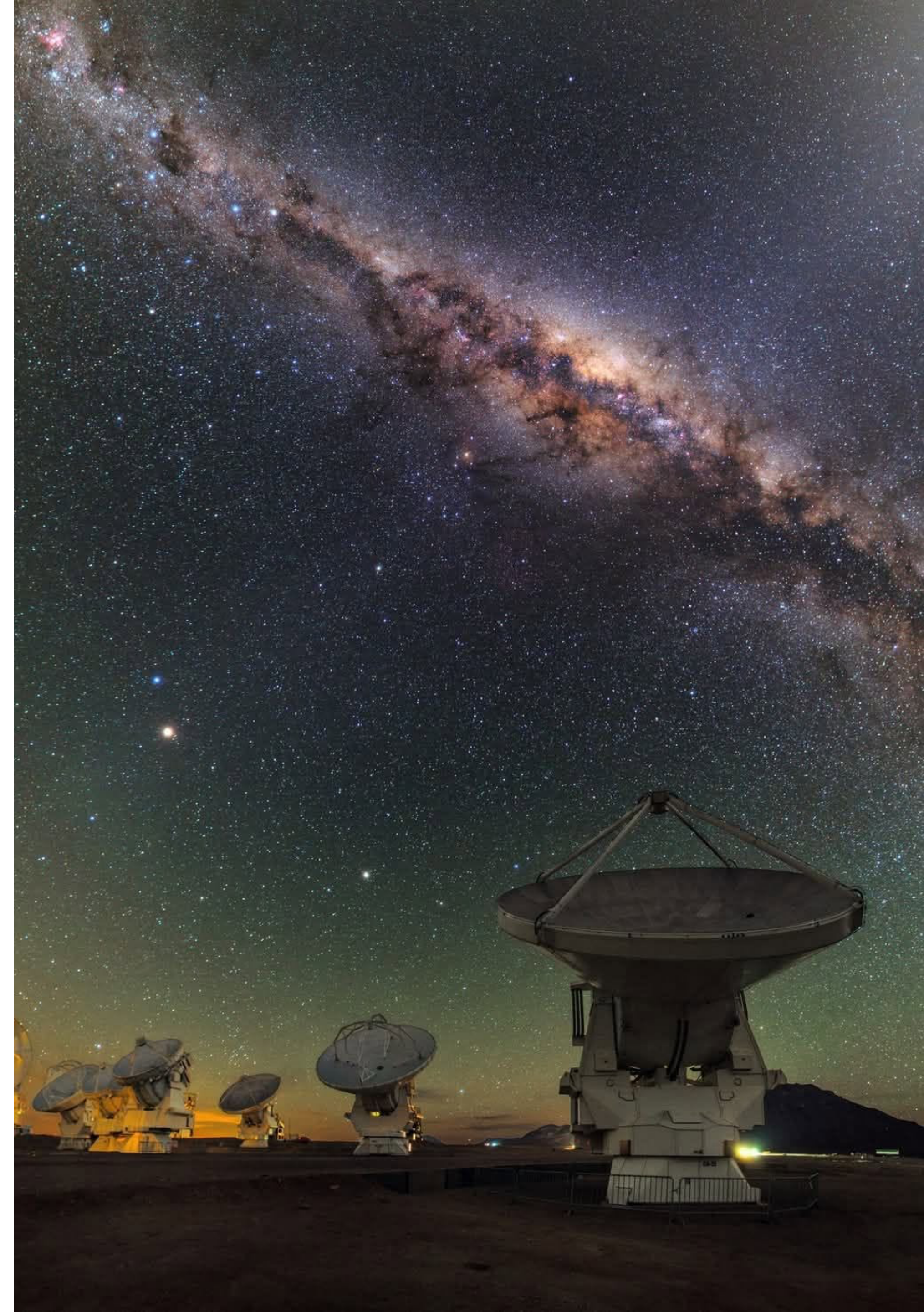


Table of Contents

(1) Introduction to Spectral Line Cubes and Imaging

- Spectral Line Science with ALMA
- What is a Spectral Cube? (*listobs*, *plotms*)
- Continuum Subtraction (*uvcontsub*)
- Cleaning Parameters and Concepts for Spectral Cubes (*tclean*)

(2) Spectral Cube Manipulation

- Headers (*imhead*)
- Regridding (*imregrid*)
- Rebinning (*imrebin*)
- Transposing (*imtrans*)
- Reframing (*imreframe*)

(3) Spectral Cube Analysis

- Moment Maps (*imcollapse*, *immoment*)
- Position-Velocity Diagrams (*impv*)

(4) Spectral Cube Analysis in CARTA

Common
Astronomy
Software
Applications
Documentation:

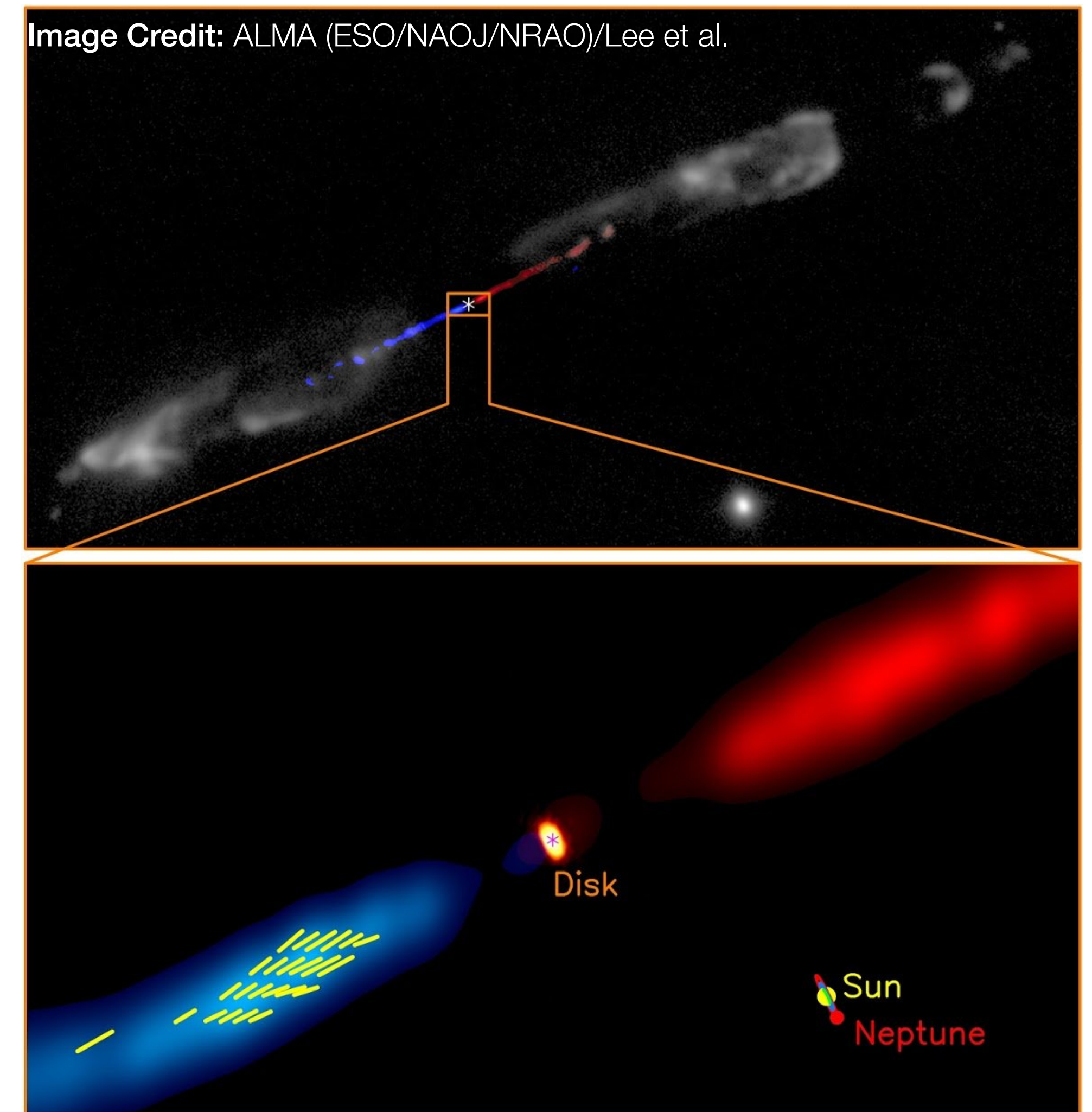
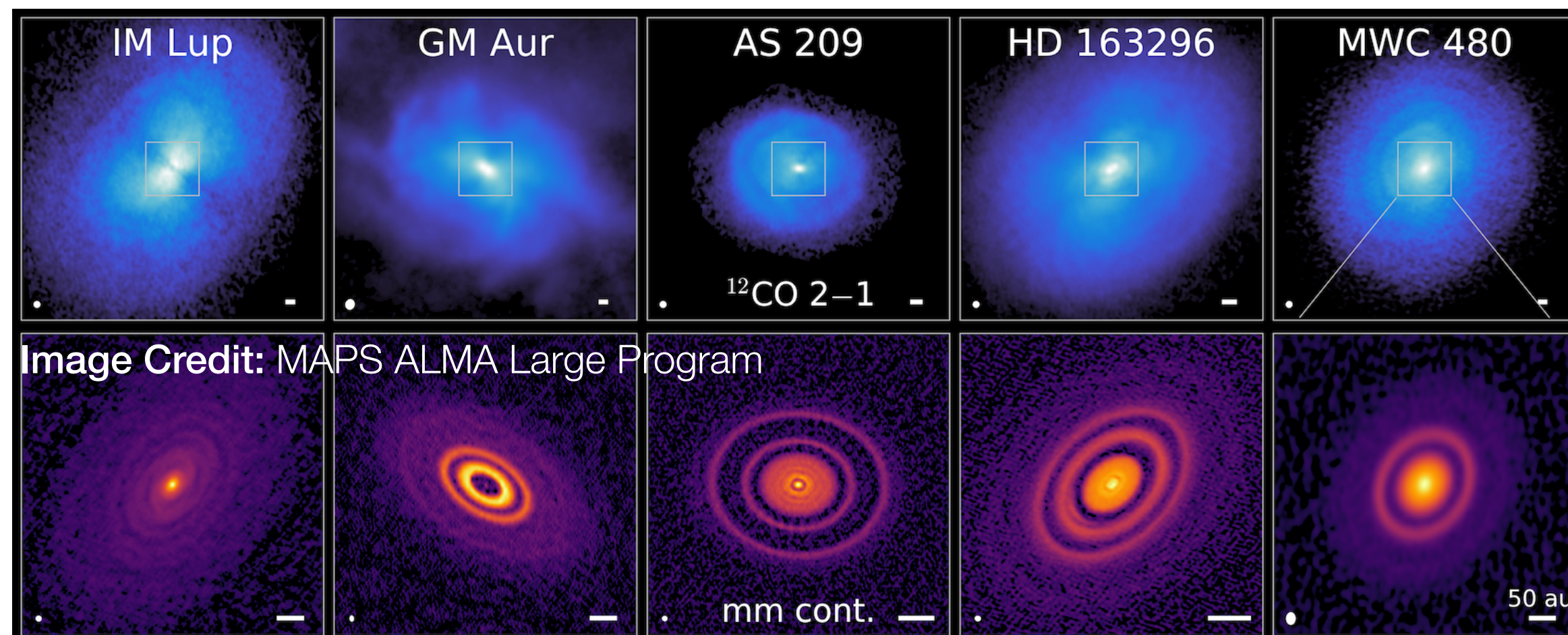
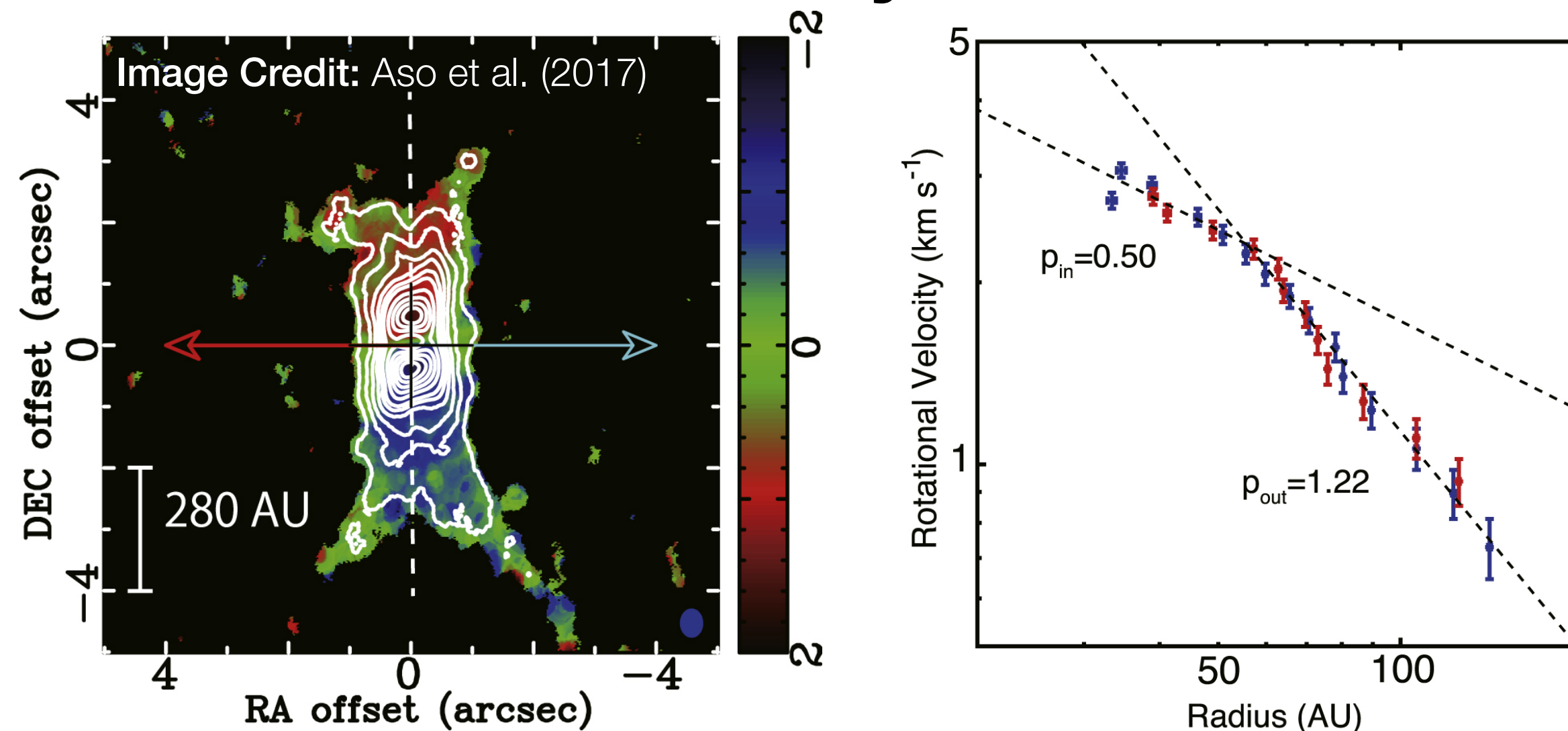


Part 1: Introduction to Spectral Line Cubes and Imaging



Spectral Line Science with ALMA

Galactic Astronomy



Spectral Line Science with ALMA

Galactic Astronomy

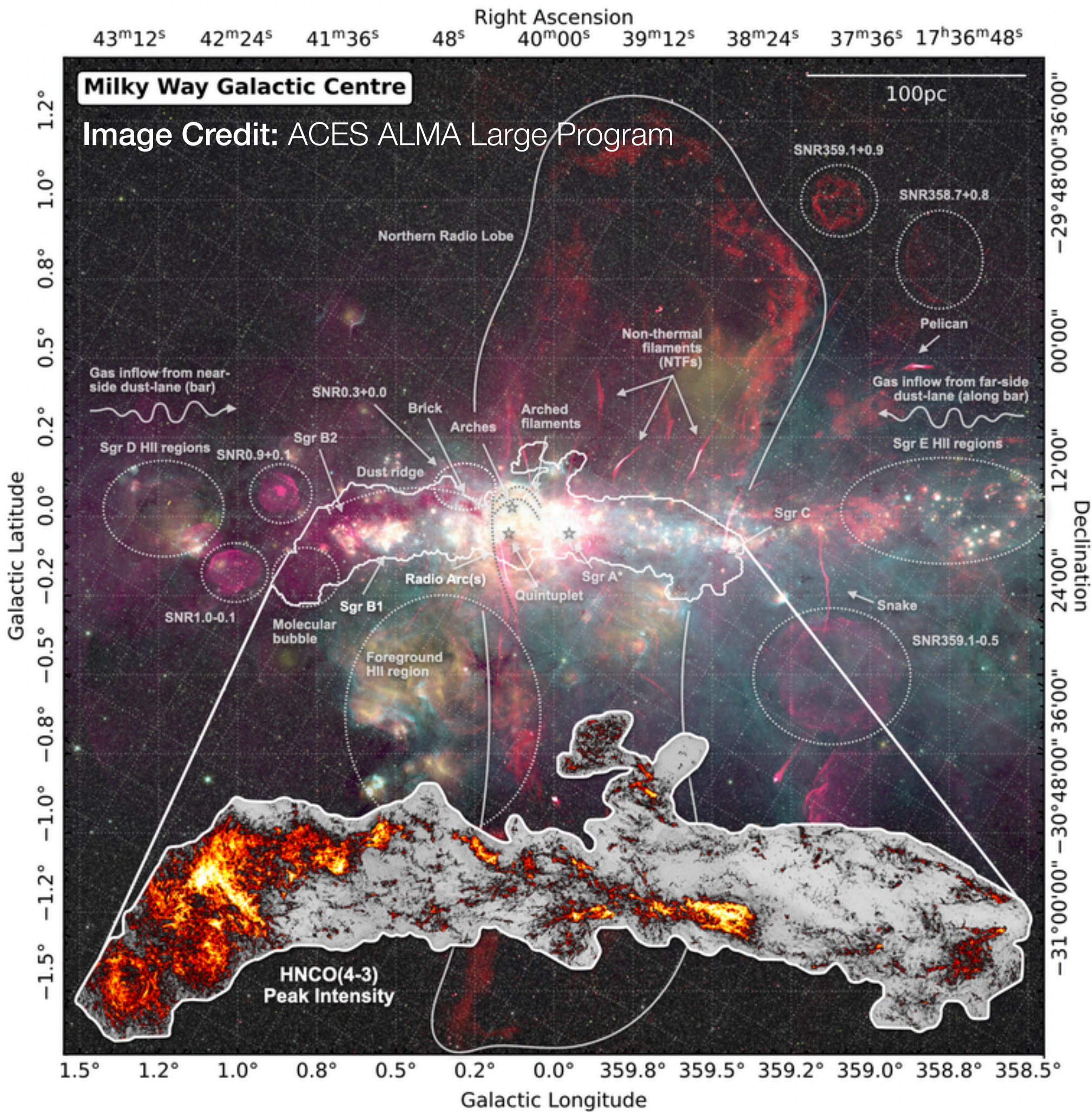
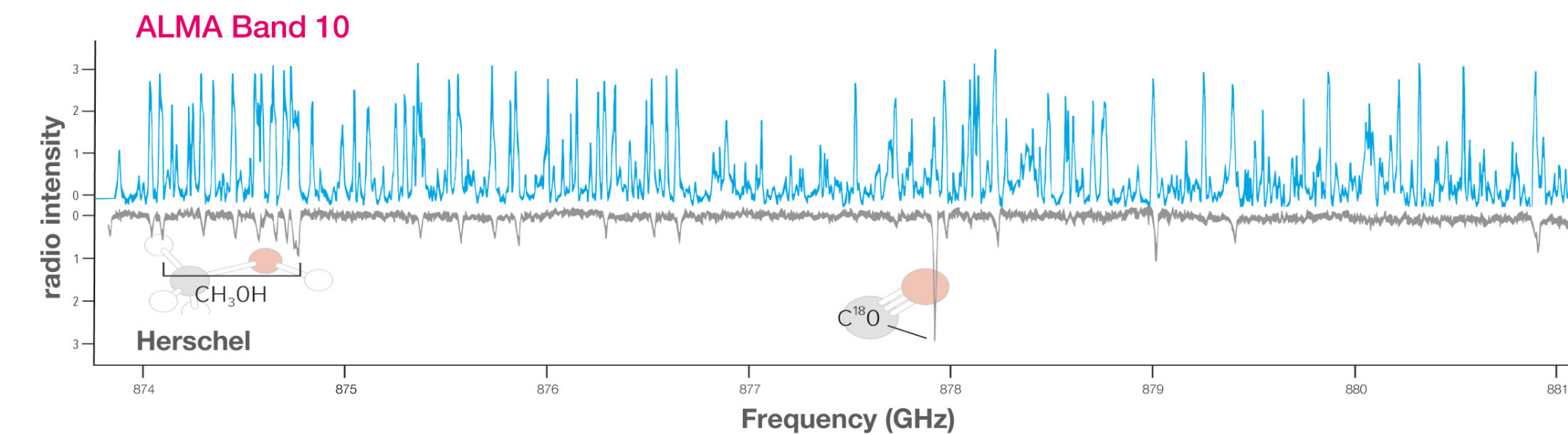


Image Credit: S. Lipinski/NASA & ESA; NAOJ, NRAO/AUI/NSF, B. McGuire et al.



Spectral Line Science with ALMA

Extragalactic Astronomy

Image Credit: NRAO / AUI / NSF / ALMA / ESO / NAOJ /
Herschel / ESA / NASA / JPL-Caltech / NOAO



Image Credit: ESO / ALMA / NAOJ / NRAO / McLeod et al.

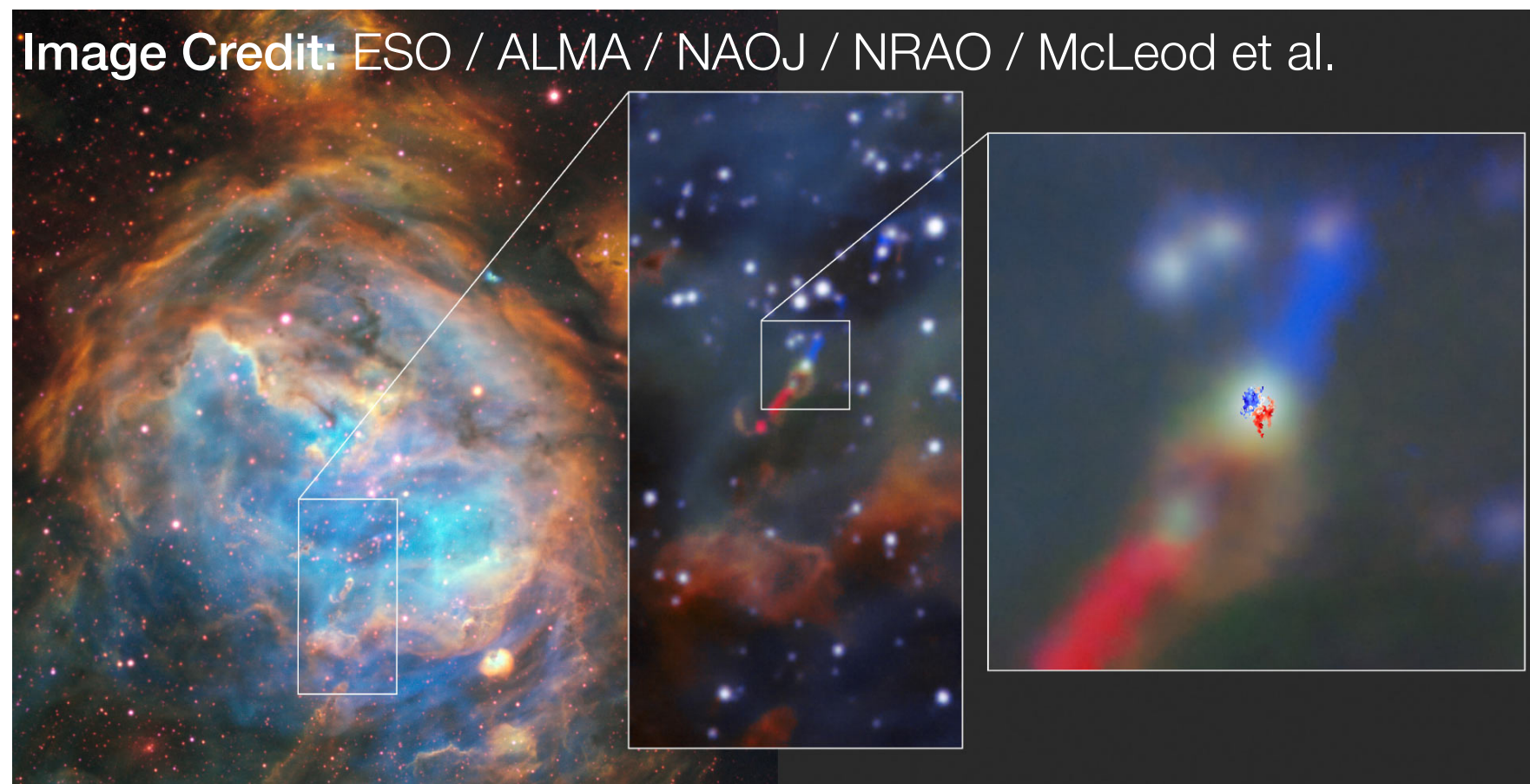
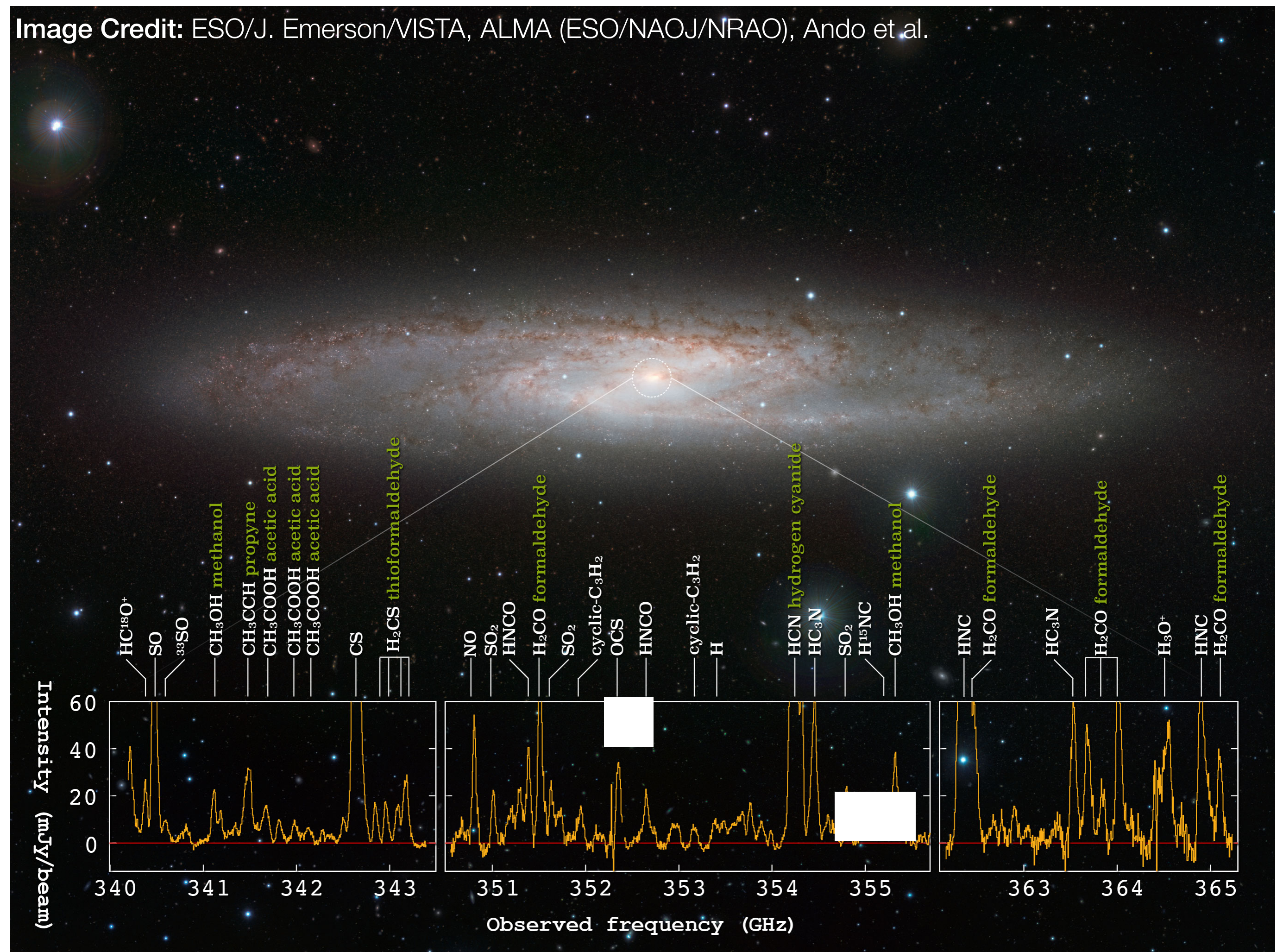
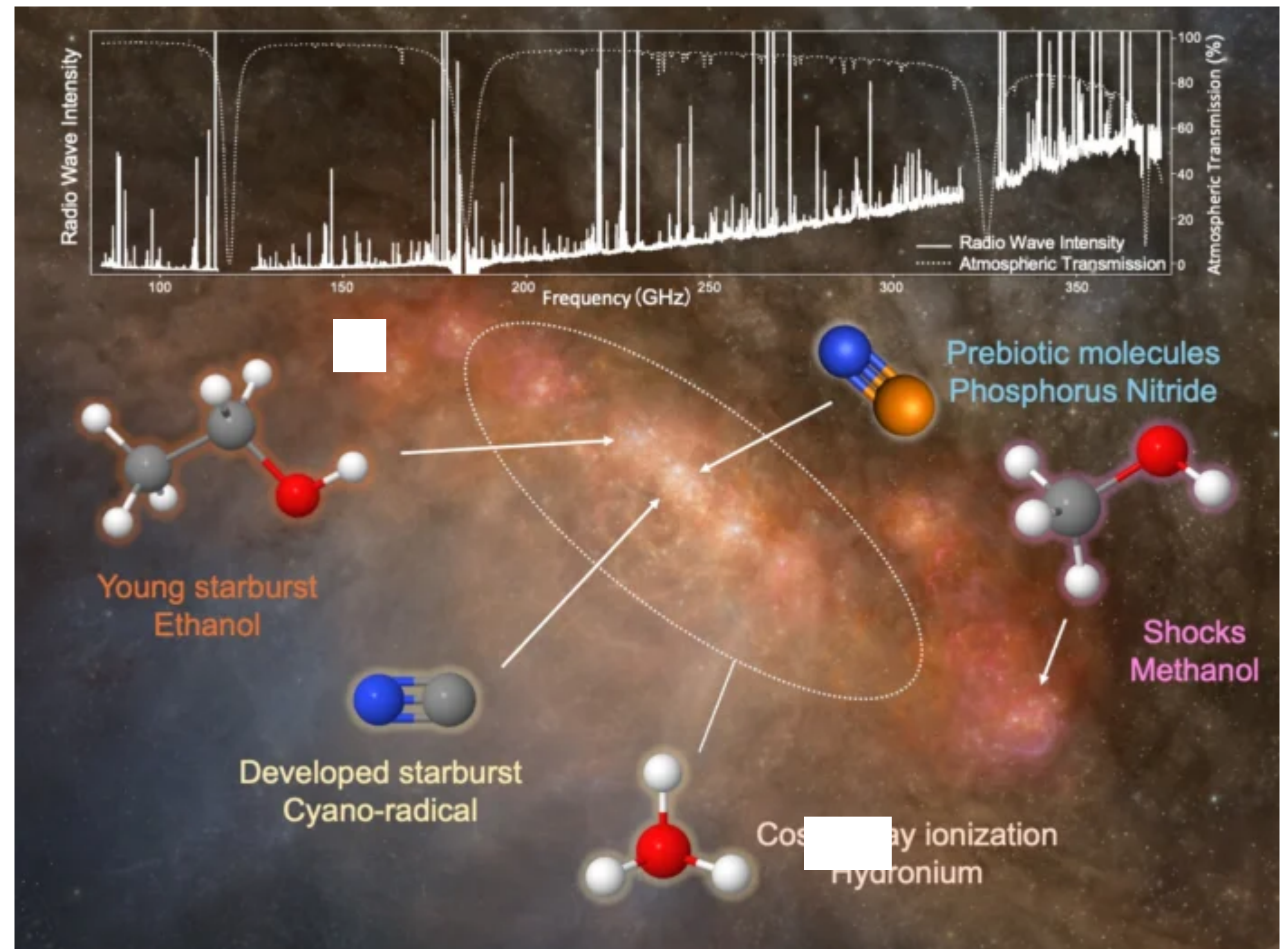
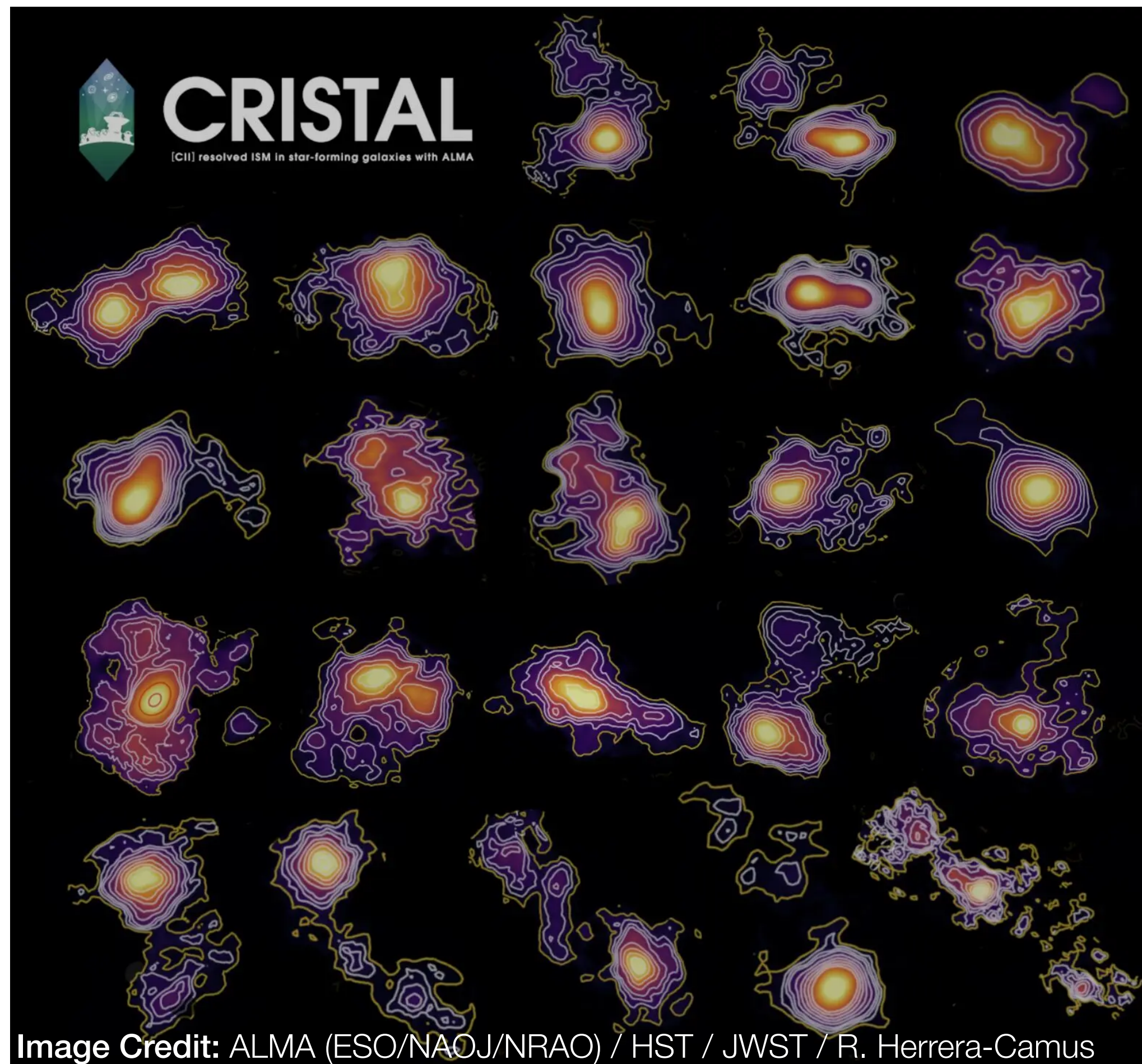


Image Credit: ESO/J. Emerson/VISTA, ALMA (ESO/NAOJ/NRAO), Ando et al.



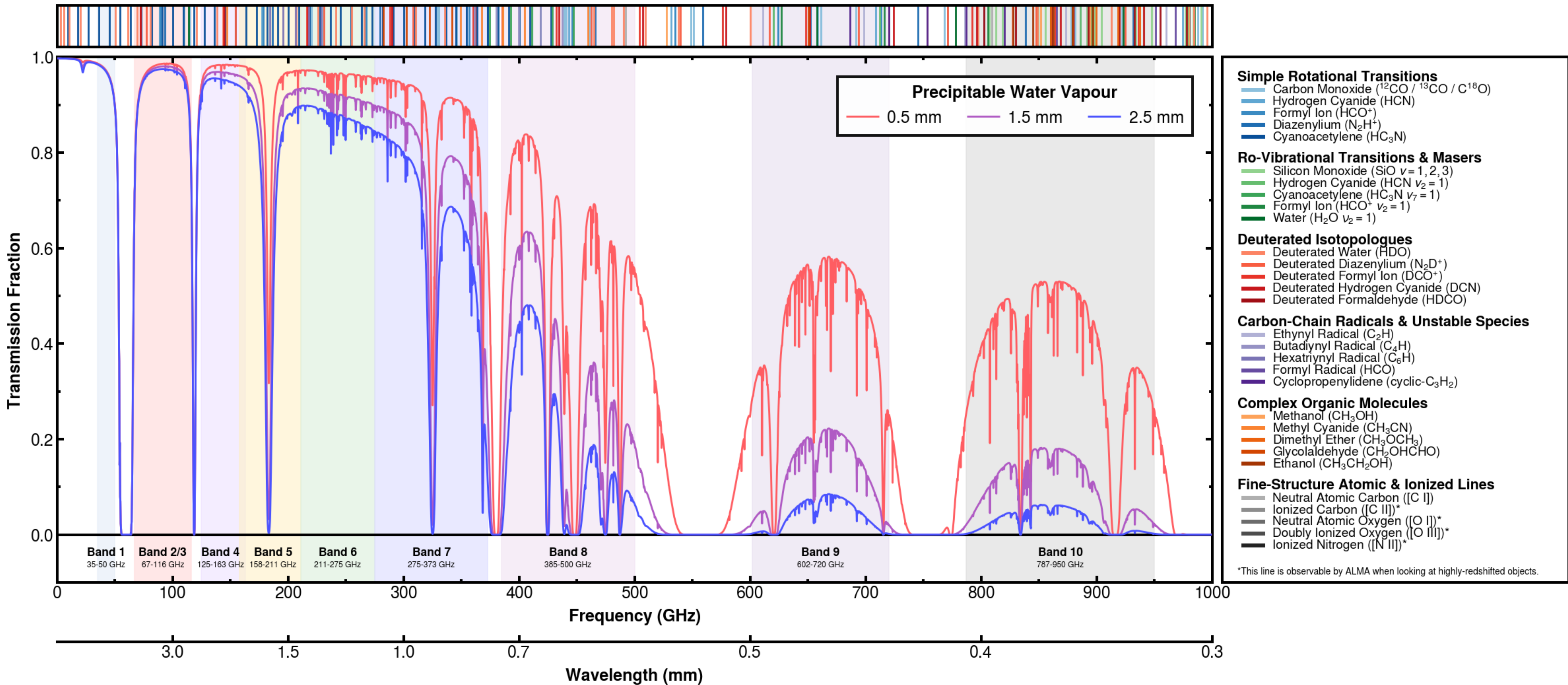
Spectral Line Science with ALMA

Extragalactic Astronomy



Spectral Line Observations with ALMA

Overview of Current Capabilities



Spectral Line Observations with ALMA

Overview of Current Capabilities

Spectral Line

Baseband-1

+ Add spectral window centred on a spectral line

+ Add spectral window manually

Delete all

Fraction	Centre Freq (rest, lsrk)	Centre Freq (sky, bar)	Transition name	Bandwidth, Resolution (smoothed)	Spec. Avg.	Representative Window
1/4	218.222192	GHz218.22240	GHzH2CO 3(0,3)-2(0,2)	58.594 MHz(80 km/s), 122.070 kHz(0.168 km/s) (2-bit)	1	
1/4	218.475632	GHz218.47584	GHzH2CO 3(2,2)-2(2,1)	58.594 MHz(80 km/s), 122.070 kHz(0.168 km/s) (2-bit)	1	
1/4	218.760066	GHz218.76028	GHzH2CO 3(2,1)-2(2,0)	58.594 MHz(80 km/s), 122.070 kHz(0.167 km/s) (2-bit)	1	
1/4	218.44005	GHz218.44026	GHzCH3OH vt=0 4(2,2)-3(1,2)	58.594 MHz(80 km/s), 122.070 kHz(0.168 km/s) (2-bit)	1	

Show image spectral windows

Baseband-2

+ Add spectral window centred on a spectral line

+ Add spectral window manually

Delete all

Fraction	Centre Freq (rest, lsrk)	Centre Freq (sky, bar)	Transition name	Bandwidth, Resolution (smoothed)	Spec. Avg.	Representative Window
1/2	219.560358	GHz219.56057	GHzC18O 2-1	58.594 MHz(80 km/s), 61.035 kHz(0.083 km/s) (2-bit)	1	
1/2	220.398684	GHz220.39890	GHz13CO v=0 2-1	58.594 MHz(80 km/s), 61.035 kHz(0.083 km/s) (2-bit)	1	

Show image spectral windows

Baseband-3

+ Add spectral window centred on a spectral line

+ Add spectral window manually

Delete all

Fraction	Centre Freq (rest, lsrk)	Centre Freq (sky, bar)	Transition name	Bandwidth, Resolution (smoothed)	Spec. Avg.	Representative Window
1 (Full)	230.53800	GHz230.53822	GHzCO v=0 2-1	58.594 MHz(76 km/s), 30.518 kHz(0.040 km/s) (2-bit)	1	

Show image spectral windows

Baseband-4

+ Add spectral window centred on a spectral line

+ Add spectral window manually

Delete all

Fraction	Centre Freq (rest, lsrk)	Centre Freq (sky, bar)	Transition name	Bandwidth, Resolution (smoothed)	Spec. Avg.	Representative Window
1 (Full)	232.00000	GHz232.00022	GHzContinuum	1875.000 MHz(2423 km/s), 31.250 MHz(40.381 km/s) (2-bit)	1	

Show image spectral windows

Frequency Division Mode

FDM is ALMA’s high-spectral-resolution observing mode. It divides each spectral window into as many as **3840 channels**, allowing line profiles and their velocity structure to be resolved.

Flexible Spectral Windows

Each of ALMA’s four basebands can contain independently positioned FDM spectral windows. This allows multiple transitions and line-free continuum channels to be observed simultaneously.

Bandwidth vs. Resolution

FDM spectral-window bandwidths range from **58.6 to 1875 MHz**. Narrower windows provide finer spectral resolution, while wider windows cover broader velocity ranges and may include additional transitions.

Spectral Line Observations with ALMA

Overview of Current Capabilities

Spectral Line

Baseband-1

+ Add spectral window centred on a spectral line

+ Add spectral window manually

Delete all

Fraction	Centre Freq (rest, lsrk)	Centre Freq (sky, bar)	Transition name	Bandwidth, Resolution (smoothed)	Spec. Avg.	Representative Window
1/4	218.222192	GHz 218.22240	H2CO 3(0,3)-2(0,2)	58.594 MHz(80 km/s), 122.070 kHz(0.168 km/s) (2-bit)	1	
1/4	218.475632	GHz 218.47584	H2CO 3(2,2)-2(2,1)	58.594 MHz(80 km/s), 122.070 kHz(0.168 km/s) (2-bit)	1	
1/4	218.760066	GHz 218.76028	H2CO 3(2,1)-2(2,0)	58.594 MHz(80 km/s), 122.070 kHz(0.167 km/s) (2-bit)	1	
1/4	218.44005	GHz 218.44026	CH3OH vt=0 4(2,2)-3(1,2)	58.594 MHz(80 km/s), 122.070 kHz(0.168 km/s) (2-bit)	1	

Show image spectral windows

Baseband-2

+ Add spectral window centred on a spectral line

+ Add spectral window manually

Delete all

Fraction	Centre Freq (rest, lsrk)	Centre Freq (sky, bar)	Transition name	Bandwidth, Resolution (smoothed)	Spec. Avg.	Representative Window
1/2	219.560358	GHz 219.56057	C18O 2-1	58.594 MHz(80 km/s), 61.035 kHz(0.083 km/s) (2-bit)	1	
1/2	220.398684	GHz 220.39890	13CO v=0 2-1	58.594 MHz(80 km/s), 61.035 kHz(0.083 km/s) (2-bit)	1	

Show image spectral windows

Baseband-3

+ Add spectral window centred on a spectral line

+ Add spectral window manually

Delete all

Fraction	Centre Freq (rest, lsrk)	Centre Freq (sky, bar)	Transition name	Bandwidth, Resolution (smoothed)	Spec. Avg.	Representative Window
1 (Full)	230.53800	GHz 230.53822	CO v=0 2-1	58.594 MHz(76 km/s), 30.518 kHz(0.040 km/s) (2-bit)	1	

Show image spectral windows

Baseband-4

+ Add spectral window centred on a spectral line

+ Add spectral window manually

Delete all

Fraction	Centre Freq (rest, lsrk)	Centre Freq (sky, bar)	Transition name	Bandwidth, Resolution (smoothed)	Spec. Avg.	Representative Window
1 (Full)	232.00000	GHz 232.00022	Continuum	1875.000 MHz(2423 km/s), 31.250 MHz(40.381 km/s) (2-bit)	1	

Show image spectral windows

Sampling the Line Profile

Sample the narrowest expected line with at least two independent resolution elements for Nyquist sampling:

$$\Delta v_{\text{res}} \lesssim \frac{\text{FWHM}_{\text{line}}}{2}$$

Use the spectral resolution when checking this requirement.

Velocity Coverage

The spectral window should contain the complete expected velocity extent, with line-free channels on both sides for continuum subtraction and baseline fitting.

Assumptions: Dual-polarization, 2-bit FDM with a 58.594 MHz spectral window per line. Resolutions are Hanning-smoothed. Multiple line windows share one baseband and receive equal correlator resources; lines placed in separate basebands can each retain 30.518 kHz resolution.

Spectral Line Observations with ALMA

Overview of Current Capabilities

ALMA Band	Velocity Coverage (58.594 MHz per line)	Finest Velocity Resolution (1 SPW; 30.518 kHz spectral resolution)	Finest Velocity Resolution (2 SPWs; 61.035 kHz spectral resolution)	Finest Velocity Resolution (3-4 SPWs; 122.070 kHz spectral resolution)
Band 1	351–502 km/s	0.183–0.261 km/s	0.366–0.523 km/s	0.732–1.046 km/s
Band 2	151–262 km/s	0.079–0.137 km/s	0.158–0.273 km/s	0.315–0.546 km/s
Band 3	151–209 km/s	0.079–0.109 km/s	0.158–0.218 km/s	0.315–0.436 km/s
Band 4	108–141 km/s	0.056–0.073 km/s	0.112–0.146 km/s	0.225–0.293 km/s
Band 5	83–111 km/s	0.043–0.058 km/s	0.087–0.116 km/s	0.173–0.232 km/s
Band 6	64–83 km/s	0.033–0.043 km/s	0.067–0.087 km/s	0.133–0.173 km/s
Band 7	47–64 km/s	0.025–0.033 km/s	0.049–0.067 km/s	0.098–0.133 km/s
Band 8	35–46 km/s	0.018–0.024 km/s	0.037–0.048 km/s	0.073–0.095 km/s
Band 9	24–29 km/s	0.013–0.015 km/s	0.025–0.030 km/s	0.051–0.061 km/s
Band 10	18–22 km/s	0.010–0.012 km/s	0.019–0.023 km/s	0.039–0.047 km/s

Sampling the Line Profile
 Sample the narrowest expected line with at least two independent resolution elements for Nyquist sampling:

$$\Delta v_{\text{res}} \lesssim \frac{\text{FWHM}_{\text{line}}}{2}$$

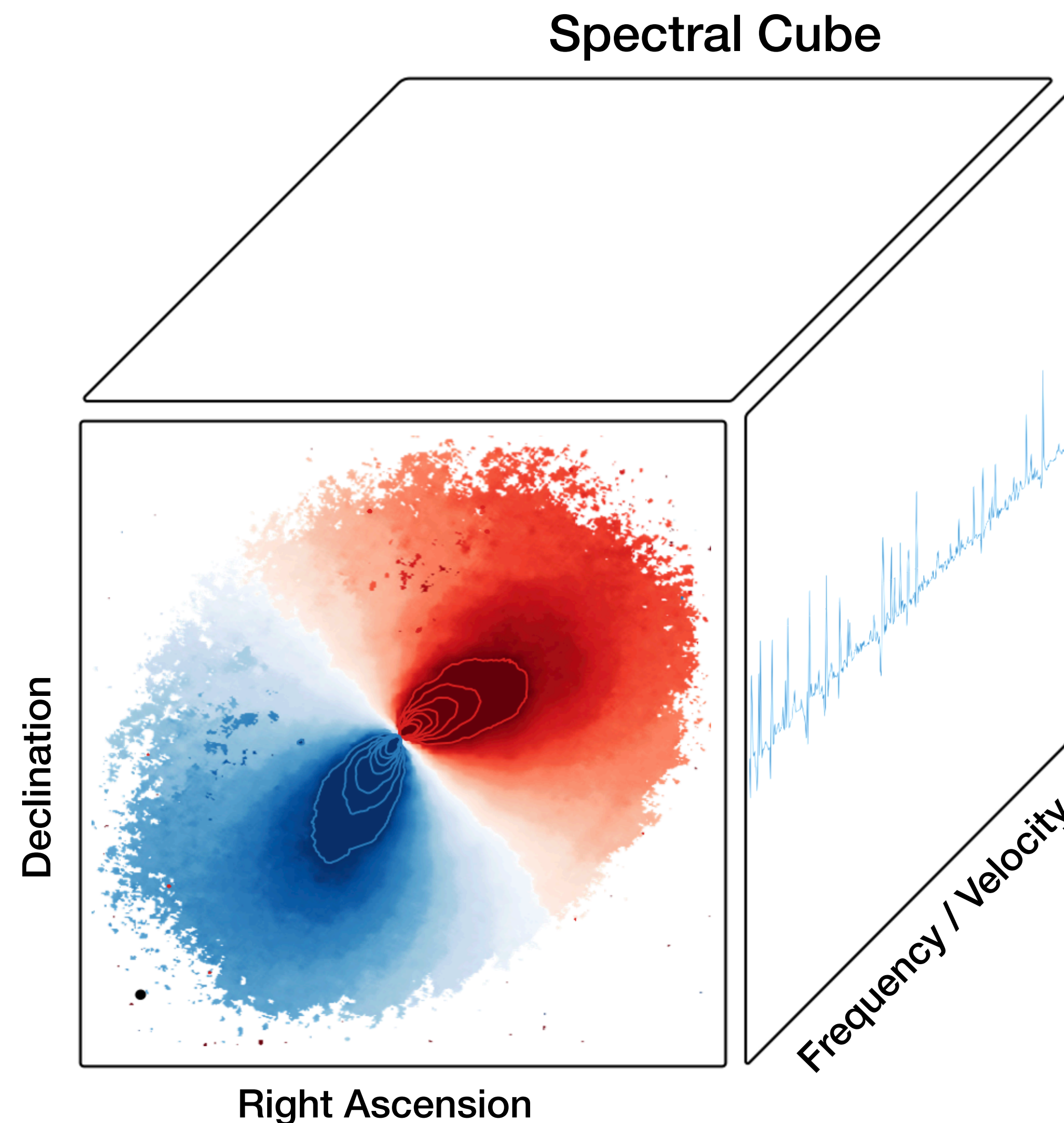
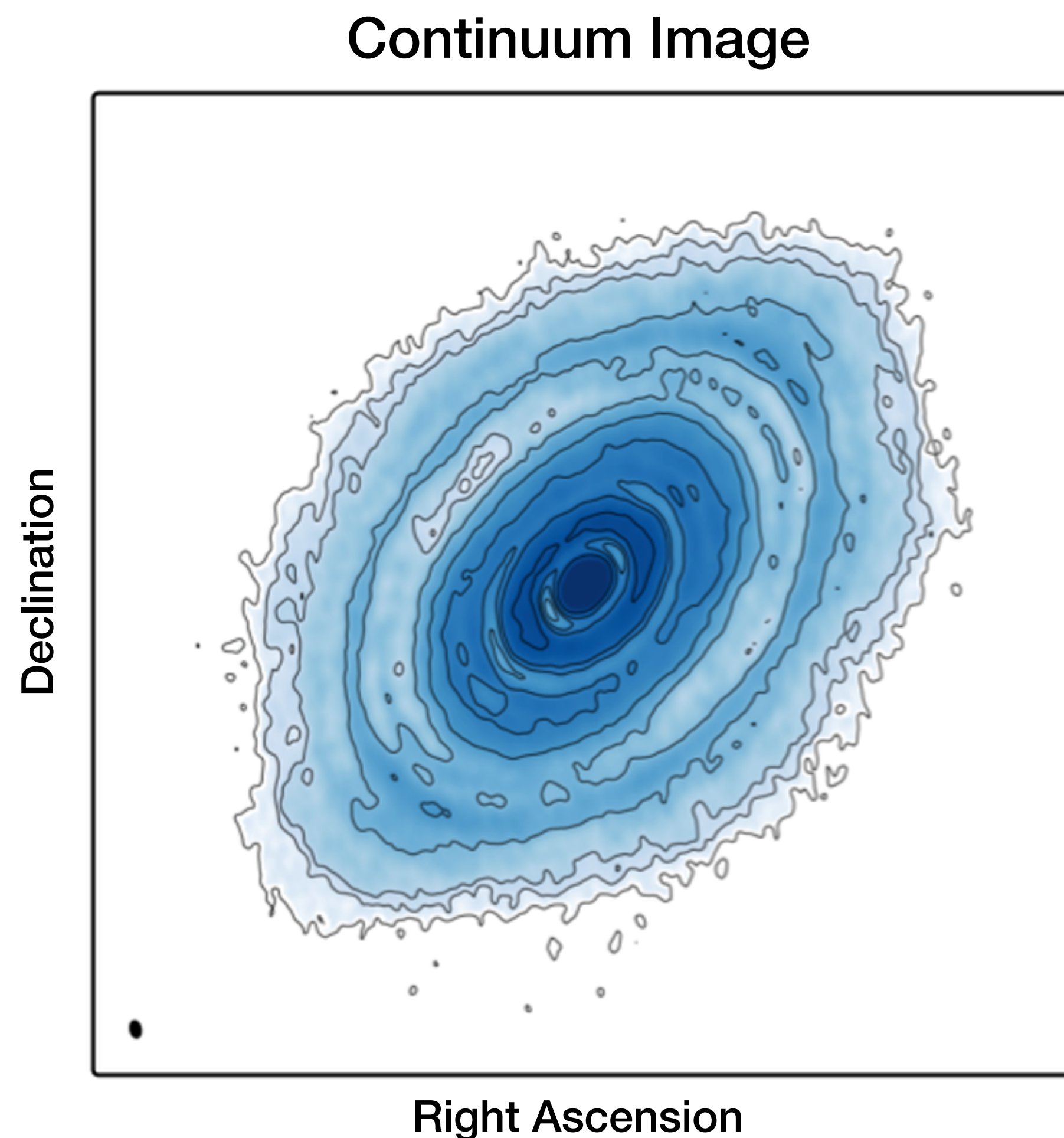
Use the spectral resolution when checking this requirement.

Velocity Coverage
 The spectral window should contain the complete expected velocity extent, with line-free channels on both sides for continuum subtraction and baseline fitting.

Assumptions: Dual-polarization, 2-bit FDM with a 58.594 MHz spectral window per line. Resolutions are Hanning-smoothed. Multiple line windows share one baseband and receive equal correlator resources; lines placed in separate basebands can each retain 30.518 kHz resolution.

What is a Spectral Cube?

Tasks: listobs, plotms



What is a Spectral Cube?

Tasks: listobs, plotms

Example: TWHydra_corrected.ms

```
CASA <1>: inp listobs
# listobs -- Get the summary of a MeasurementSet and list it in the logger or in a file
vis          = ''          # Name of input visibility file (MS)
selectdata   = True        # Data selection parameters
  spw         = ''          # Selection based on spectral-window/frequency/channel.
  field       = ''          # Selection based on field names or field index numbers. Default is all.
  antenna     = ''          # Selection based on antenna/baselines. Default is all.
  uvrange     = ''          # Selection based on uv range. Default: entire range. Default units: meters.
  timerange   = ''          # Selection based on time range. Default is entire range.
  correlation  = ''          # Selection based on correlation. Default is all.
  scan        = ''          # Selection based on scan numbers. Default is all.
  intent      = ''          # Selection based on observation intent. Default is all.
  feed        = ''          # Selection based on multi-feed numbers: Not yet implemented
  array       = ''          # Selection based on (sub)array numbers. Default is all.
  observation  = ''          # Selection based on observation ID. Default is all.
verbose      = True        # Controls level of information detail reported. True reports more than False.
listfile     = ''          # Name of disk file to write output. Default is none (output is written to logger only).
listunfl     = False       # List unflagged row counts? If true, it can have significant negative performance impact.
cachesize    = 50.0        # EXPERIMENTAL. Maximum size in megabytes of cache in which data structures can be held.
```

```
CASA <2>: listobs(vis="data/TWHydra_corrected.ms", overbose=True, listfile="data/TWHydra.listobs.txt", overwrite=True)
```


What is a Spectral Cube?

Tasks: listobs, plotms

Example: TWHydra_corrected.ms

```
MeasurementSet Name: /Users/travisthieme/Research/Science/tutorials/ALMA_Imaging_Workshop_2026/data/TWHYA_BAND7_CalibratedData.ms/TWHydra_corrected.ms MS Version 2
Observer: Unknown Project: T.B.D.
Observation: ALMA
Telescope Observation Date Observer Project
ALMA [ 4.81015e+09, 4.81015e+09]Unknown T.B.D.
ALMA [ 4.81015e+09, 4.81016e+09]Unknown T.B.D.
ALMA [ 4.81016e+09, 4.81017e+09]Unknown T.B.D.
Data records: 126900 Total elapsed time = 16902.1 seconds
Observed from 22-Apr-2011/00:15:36.7 to 22-Apr-2011/04:57:18.8 (UTC)

ObservationID = 0 ArrayID = 0
Date Timerange (UTC) Scan FldId FieldName nRows SpwIds Average Interval(s) ScanIntent
22-Apr-2011/00:15:36.7 - 00:16:07.0 9 0 TW Hya 540 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
00:20:57.8 - 00:30:43.2 13 0 TW Hya 8640 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
00:36:49.2 - 00:44:43.5 18 0 TW Hya 7020 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
00:49:33.9 - 00:59:19.3 22 0 TW Hya 8640 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
01:05:46.8 - 01:13:41.2 27 0 TW Hya 7020 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
01:20:07.2 - 01:29:52.6 31 0 TW Hya 8640 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
01:37:15.6 - 01:40:13.9 36 0 TW Hya 2700 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]

ObservationID = 1 ArrayID = 0
Date Timerange (UTC) Scan FldId FieldName nRows SpwIds Average Interval(s) ScanIntent
22-Apr-2011/02:02:10.9 - 02:02:41.1 44 0 TW Hya 540 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
02:07:36.7 - 02:17:22.0 48 0 TW Hya 8640 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
02:23:35.7 - 02:31:30.0 53 0 TW Hya 7020 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
02:36:23.6 - 02:46:09.0 57 0 TW Hya 8640 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
02:52:40.2 - 03:00:34.6 62 0 TW Hya 7020 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
03:05:14.0 - 03:14:59.4 66 0 TW Hya 8640 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
03:21:21.4 - 03:24:19.6 71 0 TW Hya 2700 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]

ObservationID = 2 ArrayID = 0
Date Timerange (UTC) Scan FldId FieldName nRows SpwIds Average Interval(s) ScanIntent
22-Apr-2011/03:44:46.9 - 03:45:17.1 79 0 TW Hya 540 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
03:50:08.4 - 03:59:53.7 83 0 TW Hya 8640 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
04:06:03.7 - 04:13:58.1 88 0 TW Hya 7020 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
04:18:49.1 - 04:28:34.4 92 0 TW Hya 8640 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
04:34:59.5 - 04:42:53.9 97 0 TW Hya 7020 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
04:47:33.5 - 04:57:18.8 101 0 TW Hya 8640 [0,1,2,3] [10.1, 10.1, 10.1, 10.1] [OBSERVE_TARGET#ON_SOURCE]
(nRows = Total number of rows per scan)

Fields: 1
ID Code Name RA Decl Epoch SrcId nRows
0 none TW Hya 11:01:51.844983 -34.42.17.16088 J2000 0 126900
Spectral Windows: (4 unique spectral windows and 1 unique polarization setups)
SpwID Name #Chans Frame Ch0(MHz) ChanWid(kHz) TotBW(kHz) CtrFreq(MHz) Corrs
0 3840 TOP0 356497.936 122.070 468750.0 356732.2500 XX YY
1 3840 TOP0 357734.314 122.070 468750.0 357968.6279 XX YY
2 3840 TOP0 346034.314 122.070 468750.0 345800.0000 XX YY
3 3840 TOP0 343955.936 122.070 468750.0 343721.6221 XX YY

Sources: 4
ID Name SpwID RestFreq(MHz) SysVel(km/s)
0 TW Hya - -
0 TW Hya 1 -
0 TW Hya 2 -
0 TW Hya 3 -

Antennas: 9:
ID Name Station Diam. Long. Lat. Offset from array center (m) ITRF Geocentric coordinates (m)
East North Elevation x y z
0 DV04 J505 12.0 m -067.45.18.0 -22.53.22.8 -7.2141 -541.3485 15.0178 2225061.036842 -5440128.036234 -2481534.422455
1 DV06 T704 12.0 m -067.45.16.2 -22.53.22.1 -42.8987 -520.1911 15.0694 2225110.551677 -5440116.726350 -2481514.951072
2 DV07 J510 12.0 m -067.45.17.8 -22.53.23.5 -0.3652 -563.8032 14.9605 2225064.049398 -5440117.310745 -2481555.086720
3 DV08 T703 12.0 m -067.45.16.2 -22.53.23.9 42.8798 -575.6928 14.6278 2225102.207484 -5440096.375809 -2481565.910698
4 DV09 N602 12.0 m -067.45.17.4 -22.53.22.3 8.8012 -527.8598 15.0513 2225077.857538 -5440126.858119 -2481522.008896
5 DV10 N606 12.0 m -067.45.17.1 -22.53.23.6 19.1981 -566.5667 14.9520 2225081.746122 -5440108.902762 -2481557.629365
6 PM01 T702 12.0 m -067.45.18.6 -22.53.24.1 -23.6269 -582.3103 14.9195 2225039.780230 -5440119.418780 -2481572.120530
7 PM02 T701 12.0 m -067.45.18.8 -22.53.22.2 -29.1268 -522.7917 15.0566 2225043.501617 -5440143.045000 -2481517.341976
8 PM03 J504 12.0 m -067.45.17.0 -22.53.23.0 22.2015 -550.2548 14.9948 2225086.942689 -5440113.674706 -2481542.618537
```

Fields: 1							
ID	Code	Name	RA	Decl	Epoch	SrcId	nRows
0	none	TW Hya	11:01:51.844983	-34.42.17.16088	J2000	0	126900
Spectral Windows: (4 unique spectral windows and 1 unique polarization setups)							
SpwID	Name	#Chans	Frame	Ch0(MHz)	ChanWid(kHz)	TotBW(kHz)	Corrs
0		3840	TOP0	356497.936	122.070	468750.0	XX YY
1		3840	TOP0	357734.314	122.070	468750.0	XX YY
2		3840	TOP0	346034.314	122.070	468750.0	XX YY
3		3840	TOP0	343955.936	122.070	468750.0	XX YY

Column	Meaning
SpwID	Zero-based spectral window identifier.
Name	Optional SPW label.
#Chans	Number of channels in the SPW.
Frame	Spectral reference frame of the listed frequencies.
Ch0(MHz)	Frequency of the first channel in MHz.
ChanWid(kHz)	Width of the channels in kHz.
TotBW(kHz)	Total bandwidth of the SPW in kHz.
CtrFreq(MHz)	Central frequency SPW in MHz.
Corrs	Recorded polarization-correlation products.

What is a Spectral Cube?

Tasks: listobs, plotms

Example: TWHydra_corrected.ms

```
CASA <3>: inp plotms
# plotms -- A plotter/interactive flagger for visibility data.
vis                = ''                # Input MS or CalTable (blank for none)
xaxis              = ''                # Plot x-axis (blank for default/current)
yaxis              = ''                # Plot y-axis (blank for default/current)
selectdata         = True              # Enable data selection parameters
  field            = ''                # Field names or ids (blank for all)
  spw               = ''                # Spectral windows:channels (blank for all)
  intent            = ''                # Observing intent (blank for all)
averagedata        = True              # Enable data averaging parameters
  avgchannel        = ''                # Average over channel (blank = False, otherwise value in channels)
  avgtime           = ''                # Average over time (blank = False, otherwise value in seconds)
  avgscan           = False            # Average over scans. Only valid with time averaging
  avgfield          = False            # Average over fields. Only valid with time averaging
  avgbaseline       = False            # Average over all baselines (mutually exclusive with avgantenna)
  avgantenna        = False            # Average per antenna (mutually exclusive with avgbaseline)
  avgspw            = False            # Average over all spectral windows
  scalar            = False            # Scalar averaging (False=vector averaging)
plotfile           = ''                # Name of plot file to save automatically
showgui            = True              # Show GUI
clearplots         = True              # Remove any existing plots so new ones can replace them.
```

```
CASA <4>: plotms(vis="TWHydra_corrected.ms", spw="2", xaxis="channel", yaxis="amp", avgtime="1e8", avgscan=True,
showgui=True)
```


What is a Spectral Cube?

Tasks: listobs, plotms

Example: TWHydra_corrected.ms

```
field="0"           # field ID 0
field="TW Hya"      # field name
field="0,2"         # fields 0 or 2
field="0~3"         # fields 0 through 3
field="TW*"         # names beginning with TW

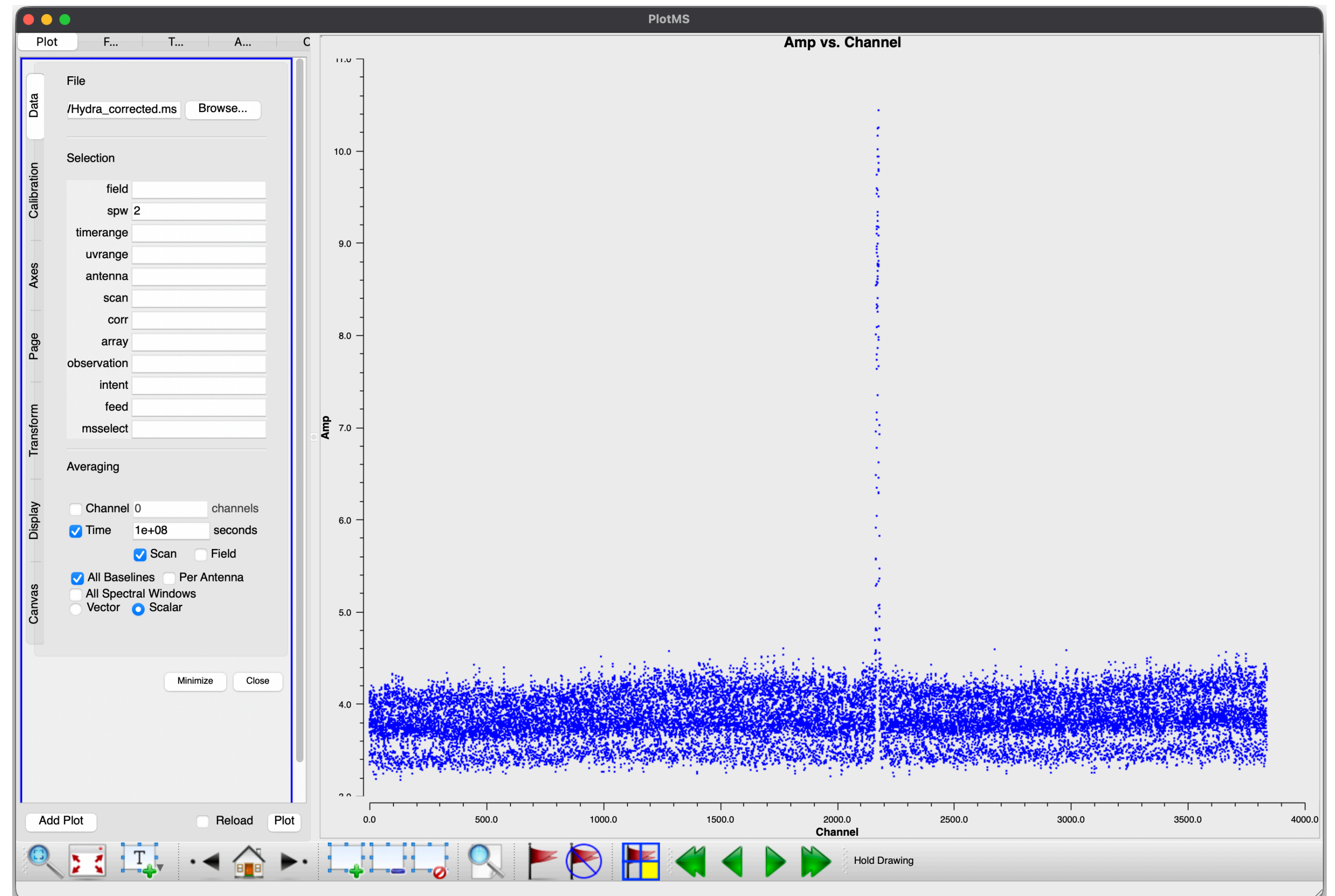
spw="2"             # all channels in SPW 2
spw="0,2"           # SPWs 0 and 2
spw="0~3"           # SPWs 0 through 3
spw="2:2100~2250"   # channels 2100~2250 in SPW 2
spw="2:0~2099;2251~3839" # two channel ranges in SPW 2
spw="0:100~200,2:2100~2250" # ranges in two SPWs
spw="*:0~100"       # channels 0~100 in every SPW

timerange="00:20:00~01:00:00" # date inferred from dataset
timerange=">03:00:00"        # after this time
timerange="<01:00:00"        # before this time
timerange="2011/04/22/00:20:00~2011/04/22/01:00:00"

uvrange="<30m"          # baselines shorter than 30m
uvrange="30~60m"        # 30~60m
uvrange=">30m"          # longer than 30m
uvrange="10~50klambda"  # in kilo-wavelengths

antenna="DV04"          # all baselines with DV04
antenna="DV04,DV06"     # baselines with DV04, DV06
antenna="DV04&DV06"     # the DV04-DV06 baseline
antenna="DV04&DV06;PM01&PM02" # two specific baselines
antenna="0&1"           # baseline using antenna IDs

intent="OBSERVE_TARGET#ON_SOURCE"
intent="*TARGET*"       # wildcard match
```



What is a Spectral Cube?

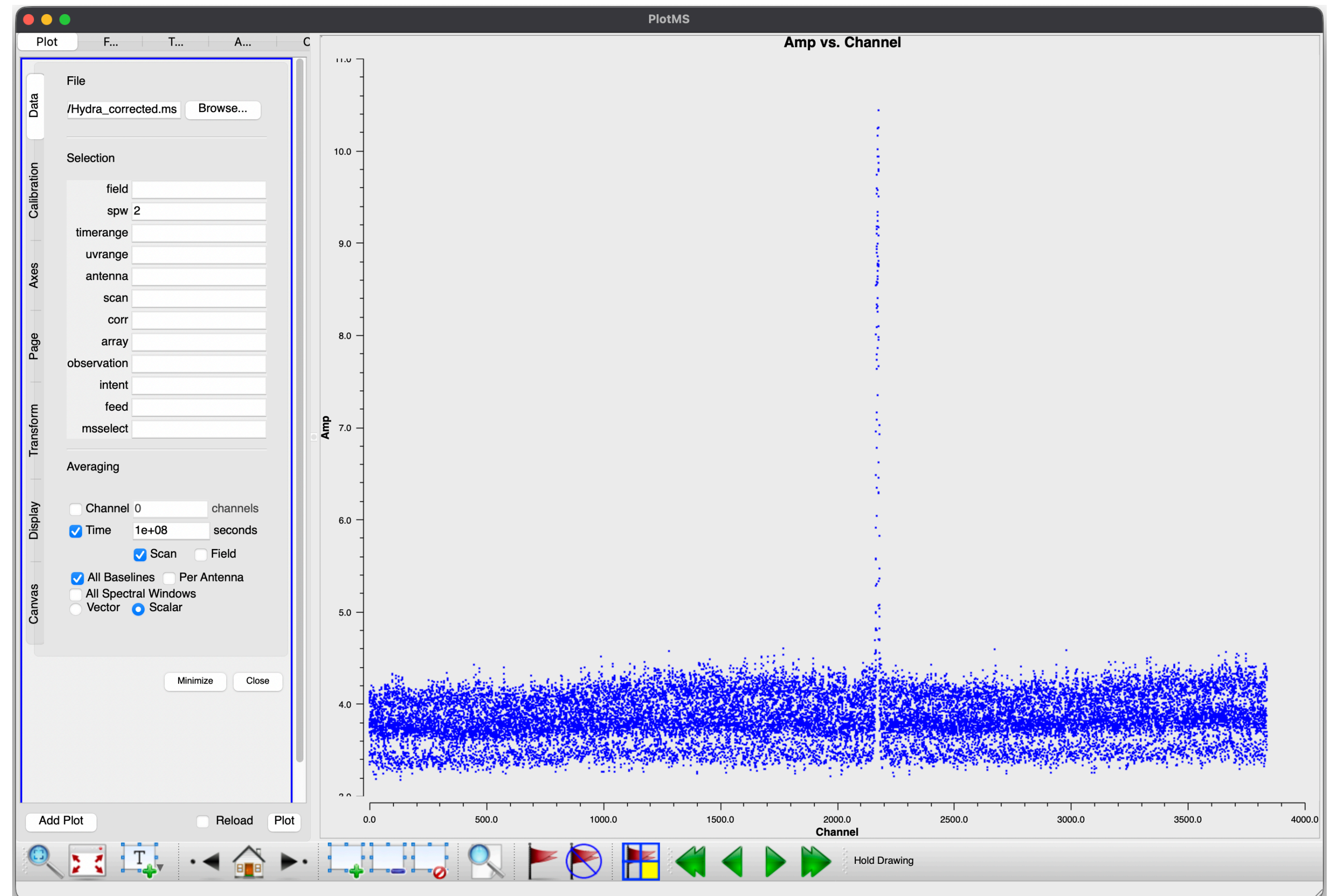
Tasks: listobs, plotms

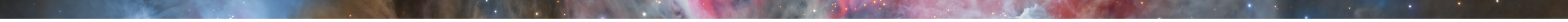
Example: TWHydra_corrected.ms

These averages produce a single, low-noise overview spectrum, making the CO line easier to identify:

- **avgtime='1e8'**: combines integrations over a very large time interval.
- **avgscan=True**: allows that time averaging to cross scan boundaries.
- **avgbaseline=True**: combines measurements from all antenna baselines.
- **scalar=True**: averages amplitude magnitudes rather than complex visibilities, avoiding phase cancellation.

Together, they reduce many thousands of visibility samples to approximately one amplitude value per channel.





Continuum Subtraction

Tasks: `split`, `uvcontsub`

Example: `TWHydra_corrected.ms`

Task 1: Identify the line-free channels.

Task 2: How would you set “spw”?

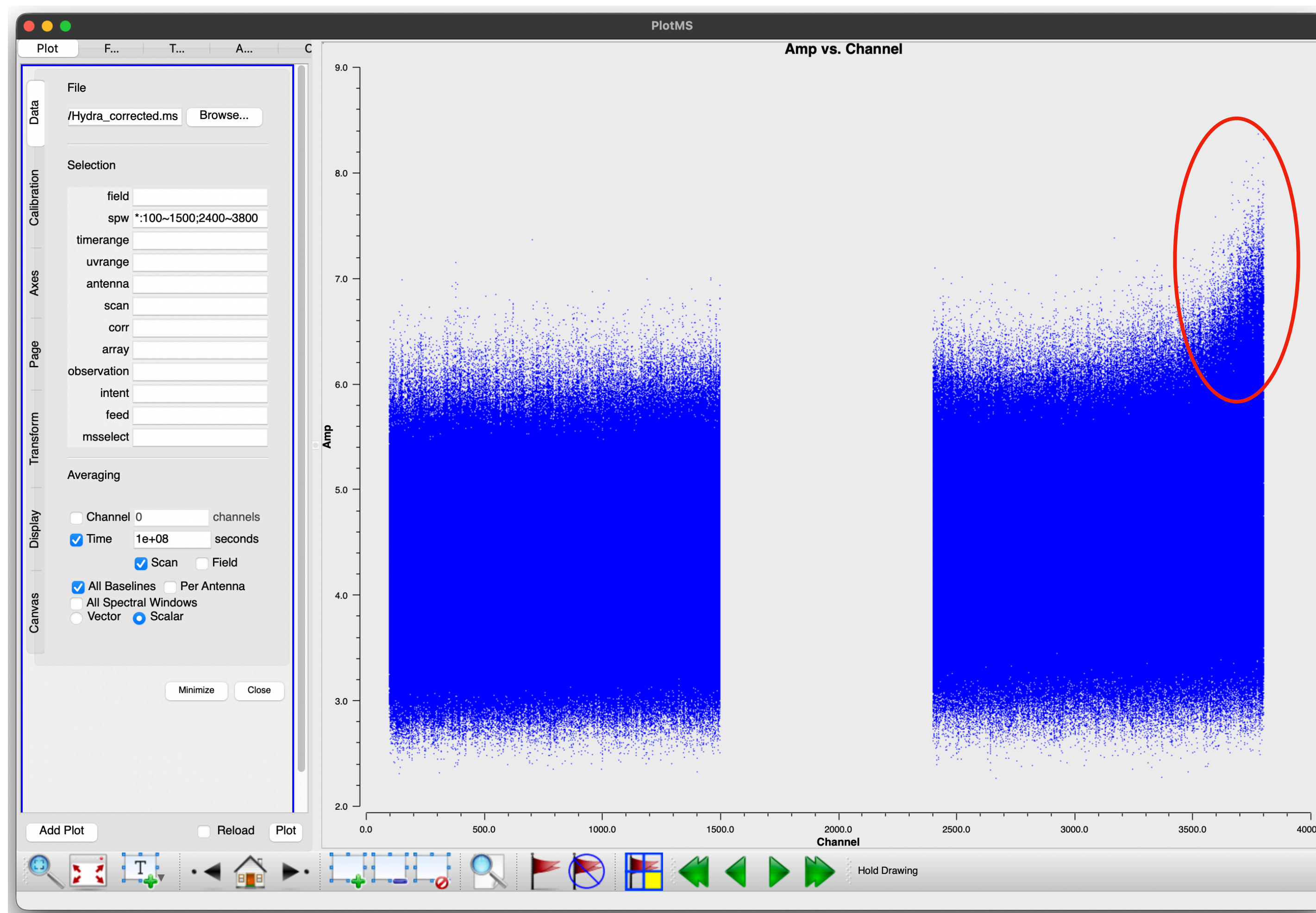
Continuum Subtraction

Tasks: split, uvcontsub

Example: TWHydra_corrected.ms

Task 1: Identify the line-free channels.
100–1800, 2500–3800

Task 2: How would you set “spw”?
spw = “*:100~1500;2400~3200”



Continuum Subtraction

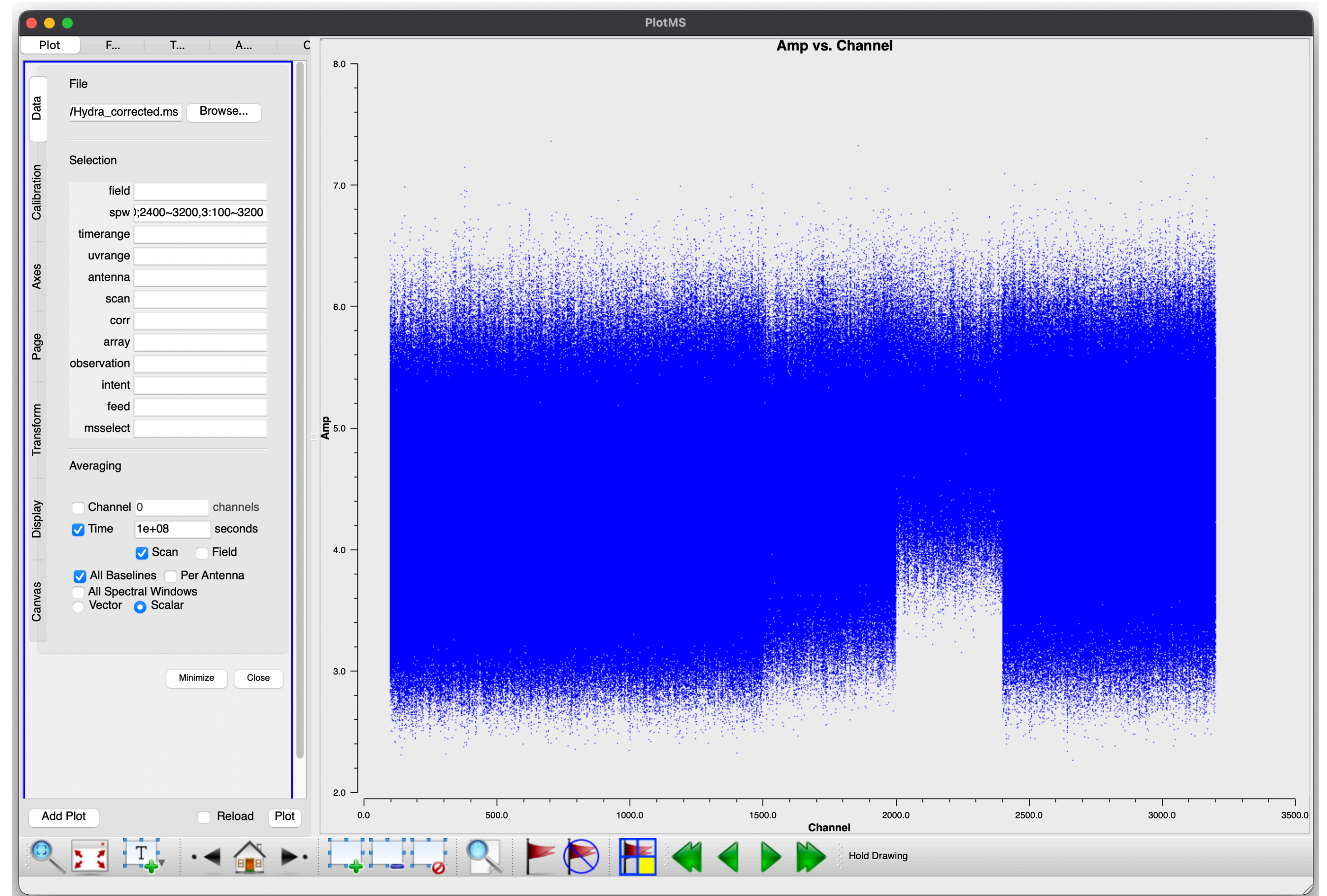
Tasks: split, uvcontsub

Example: TWHydra_corrected.ms

Task 1: Identify the line-free channels.
100–1800, 2500–3800

Task 2: How would you set “spw”?

spw =
"0:100~1500;2400~3200,1:100~3200,
2:100~1500;2400~3200,3:100~3200"



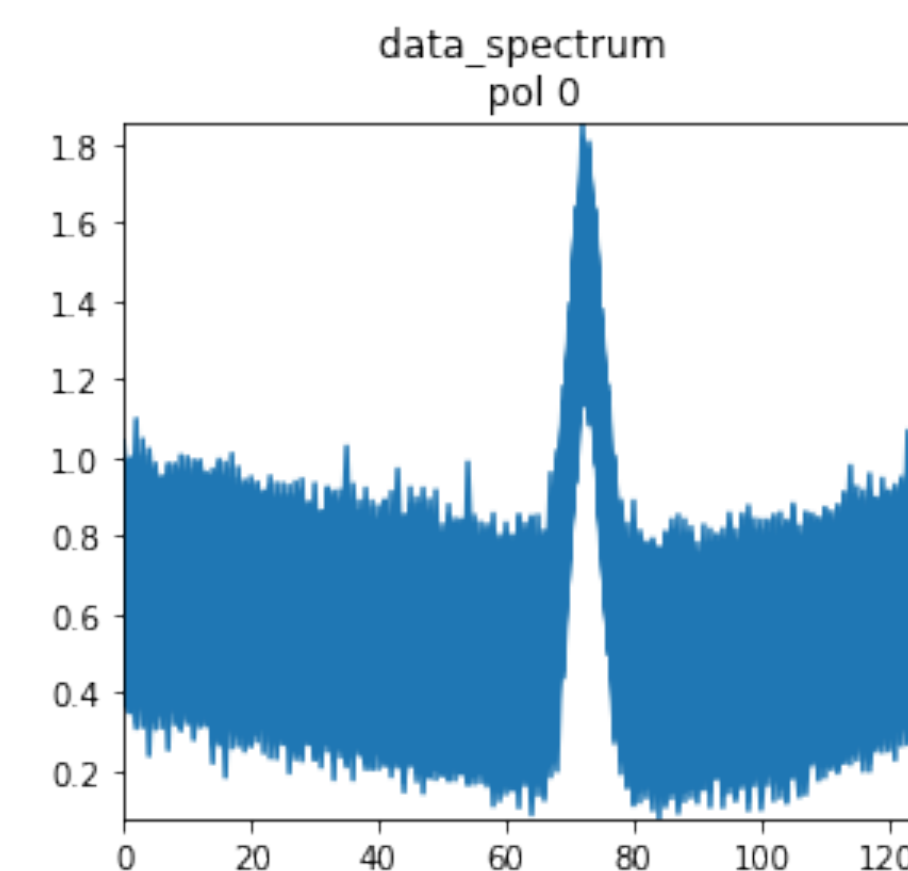
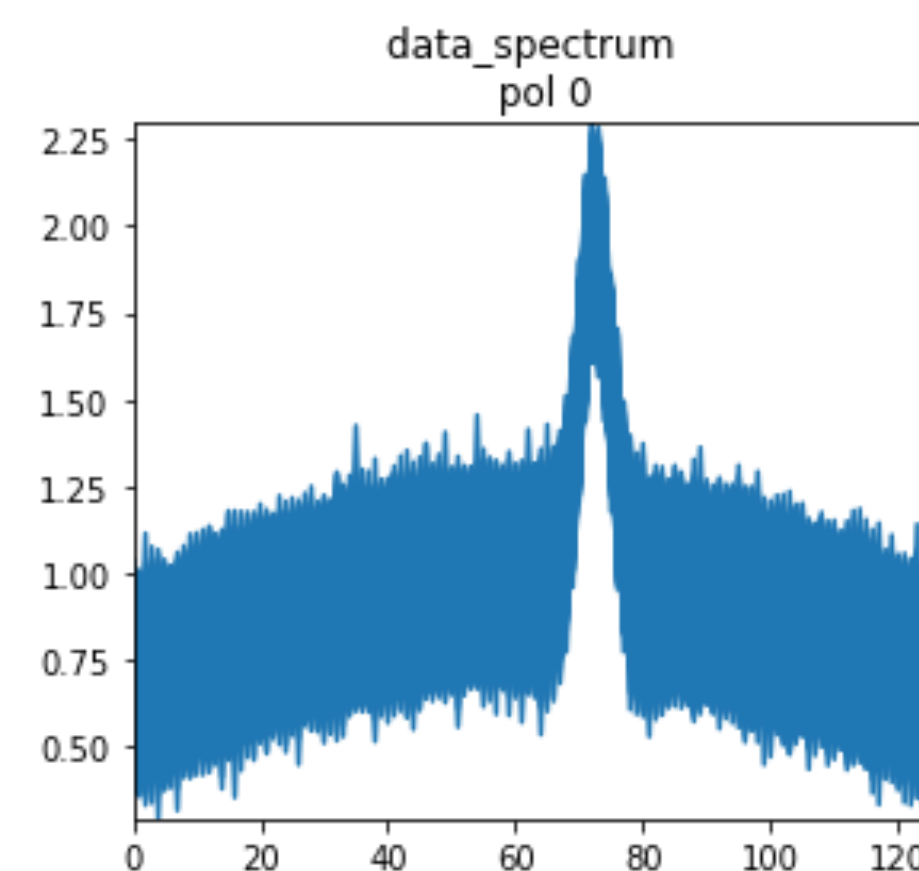
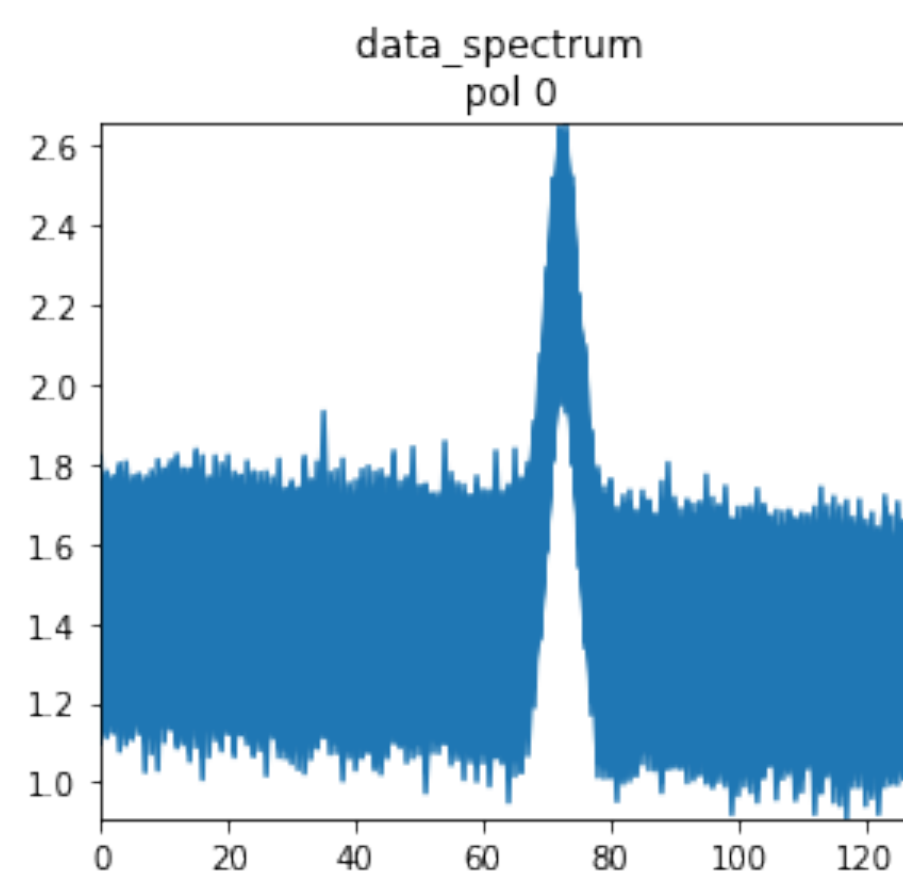
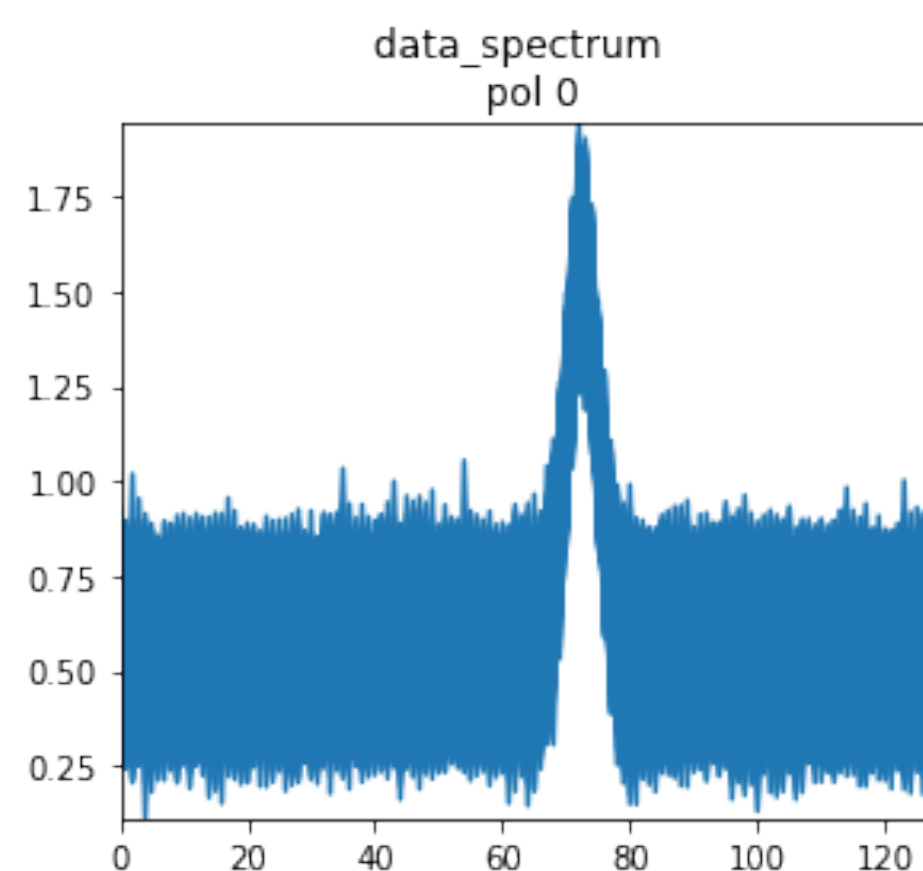
Continuum Subtraction

Tasks: `split`, `uvcontsub`

Example: `TWHydra_corrected.ms`

uvcontsub fits polynomials independently to the real and imaginary visibility components for each integration, baseline, and correlation. It fits only the channels listed in **fitspec**, predicts the continuum across the selected output channels, and subtracts that model.

Start with **fitorder=0** unless the visibility spectrum clearly requires a slope. A higher polynomial order can absorb broad or weak line emission.



Continuum Subtraction

Tasks: split, uvcontsub

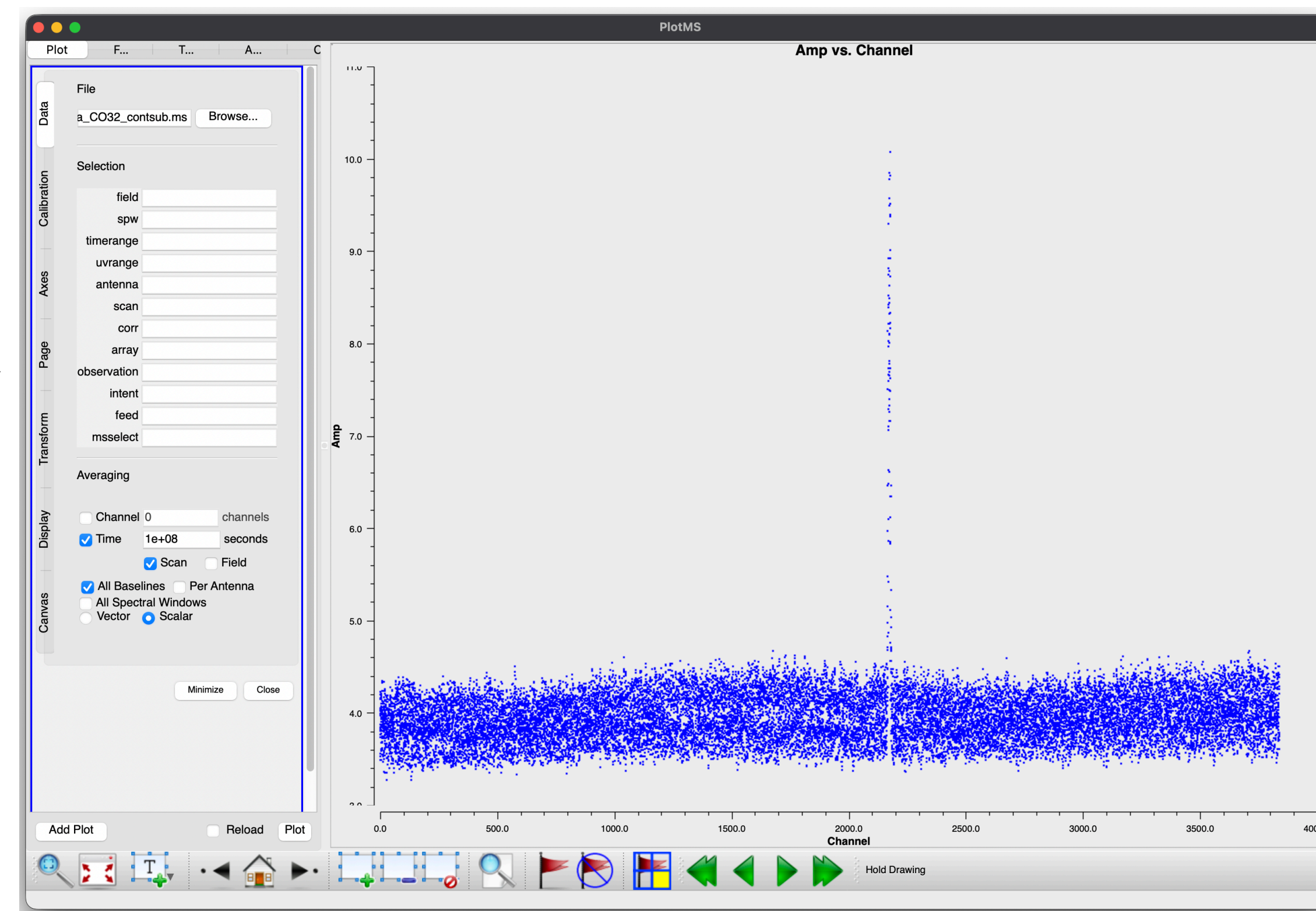
Example: TWHydra_corrected.ms

```
CASA <5>: split(vis="data/TWHydra_corrected.ms",  
              outputvis="data/TWHya_CO32.ms",  
              field="TW Hya",  
              spw="2",  
              datacolumn="data",  
              width=1) # retain native spectral resolution
```

```
CASA <6>: uvcontsub(vis="data/TWHya_CO32.ms",  
                  outputvis="TWHya_CO32_contsub.ms",  
                  datacolumn="data",  
                  fitspec="0:100~1500;2400~3200",  
                  fitorder=0)
```

OR

```
CASA <5>: uvcontsub(vis="data/TWHydra_corrected.ms",  
                  outputvis="TWHya_CO32_contsub.ms",  
                  spw="2",  
                  datacolumn="data",  
                  fitspec="0:100~1500;2400~3200,1:100~3200,2:100~1500;2400~3200,3:100~3200",  
                  fitorder=0)
```



Cleaning Parameters and Concepts

Tasks: tclean

Example: TWHydra_corrected.ms

For spectral-line cubes, focus on the parameters that define which spectral data are imaged and how the output spectral axis is constructed:

- **spw**: input spectral window and channel selection
- **specmode='cube'**: produce spectral cube
- **nchan**: number of output channels
- **start**: coordinate of the first output channel
- **width**: output-channel spacing and direction
- **outframe**: output spectral reference frame
- **veltype**: velocity convention
- **restfreq**: rest frequency of the molecular line transition

Example 1: Increasing velocity

```
nchan = 41  
start = "-2km/s"  
width = "0.25km/s"
```

The channel centers are:

-2.00, -1.75, -1.50, ..., +7.75, +8.00 km/s

Example 2: Decreasing velocity

```
nchan = 41  
start = "8km/s"  
width = "-0.25km/s"
```

The channel centers are:

+8.00, +7.75, +7.50, ..., -1.75, -2.00 km/s

Example 3: Frequency coordinates

```
nchan = 81  
start = "345.790GHz"  
width = "0.25MHz"
```

The channel centers are:

345.79000, 345.79025, 345.79050, ...,
345.80950, 345.80975, 345.81000 GHz

Cleaning Parameters and Concepts

Tasks: tclean

Example: TWHydra_corrected.ms

For spectral-line cubes, focus on the parameters that define which spectral data are imaged and how the output spectral axis is constructed:

- **spw**: input spectral window and channel selection
- **specmode= ' cube '**: produce spectral cube
- **nchan**: number of output channels
- **start**: coordinate of the first output channel
- **width**: output-channel spacing and direction
- **outframe**: output spectral reference frame
- **veltype**: velocity convention
- **restfreq**: rest frequency of the molecular line transition

outframe	Reference position	Typical use
"LSRK"	Kinematic Local Standard of Rest	Standard choice for Galactic molecular-line observations
"LSRD"	Dynamical Local Standard of Rest	Some Galactic-dynamics applications
"BARY"	Solar-system barycenter	Precision stellar and extragalactic spectroscopy
"GEO"	Center of Earth	Geocentric calculations
"TOPO"	Telescope location	Raw observed frequencies; varies during the observation
"GALACTO"	Galactic-center frame	Large-scale Galactic kinematics
"LGROUP"	Local Group reference frame	Nearby-galaxy studies
"CMB"	Cosmic microwave background rest frame	Cosmological and large-scale extragalactic work

Cleaning Parameters and Concepts

Tasks: tclean

Example: TWHydra_corrected.ms

For spectral-line cubes, focus on the parameters that define which spectral data are imaged and how the output spectral axis is constructed:

- **spw**: input spectral window and channel selection
- **specmode='cube'**: produce spectral cube
- **nchan**: number of output channels
- **start**: coordinate of the first output channel
- **width**: output-channel spacing and direction
- **outframe**: output spectral reference frame
- **veltype**: velocity convention
- **restfreq**: rest frequency of the molecular line transition

Which mathematical convention converts frequency into velocity?

ν_{obs} : observed frequency in the selected outframe

ν_0 : rest frequency of the transition

c : speed of light

Radio convention:

$$v_{\text{radio}} = c \left(1 - \frac{\nu_{\text{obs}}}{\nu_0} \right)$$

Optical convention:

$$v_{\text{optical}} = c \left(\frac{\nu_0}{\nu_{\text{obs}}} - 1 \right) = c \left(\frac{\lambda_{\text{obs}}}{\lambda_0} - 1 \right)$$

Redshift convention:

$$v_{\text{optical},z} = cz, \text{ where } z = \frac{\nu_0}{\nu_{\text{obs}}} - 1 = \frac{\lambda_{\text{obs}}}{\lambda_0} - 1$$

Cleaning Parameters and Concepts

Tasks: tclean

Example: TWHydra_corrected.ms

For spectral-line cubes, focus on the parameters that define which spectral data are imaged and how the output spectral axis is constructed:

- **spw**: input spectral window and channel selection
- **specmode='cube'**: produce spectral cube
- **nchan**: number of output channels
- **start**: coordinate of the first output channel
- **width**: output-channel spacing and direction
- **outframe**: output spectral reference frame
- **veltype**: velocity convention
- **restfreq**: rest frequency of the molecular line transition

The screenshot displays the 'Basic Splatalogue' web interface. At the top, there's a header with the title 'Basic Splatalogue' and a photo credit 'Photo Credit: NRAO/AUI/NSF'. Below the header is a 'Search Species' section with a text input field containing 'e.g. ammonia, carbon, methanol, CO'. The main section is titled 'Search Parameters' and is divided into four columns: 'Quick Picker', 'Query Filters', 'Astronomical Filters', and 'Export Data'. The 'Quick Picker' column contains a grid of molecular species with checkboxes for their rest frequencies (e.g., CO v=0, C¹⁷O, CH₃OH v₁=0, HCN v=0, H¹³CN v=0, DCN v=0, CS, NH₃, C II, O III, H₂O v=0, SiO v=0, ¹³CO v=0, C¹⁸O, H₂CO, HNC v=0, HC¹⁵N v=0, HCO⁺ v=0, H¹³CO⁺, C I, O I, N II, HDO). The 'Query Filters' column includes 'Telescope Bands' (Any, ALMA Band 1 (35-50 GHz), ALMA Band 3 (84-116 GHz), ALMA Band 4 (125-163 GHz)), 'Energy Range' (From, To, E_L (K), E_U (K)), 'Specify a Frequency Range' (From, To, Frequency Unit: GHz), and 'Load Freqs From listobs' (Choose Files, no files selected). The 'Astronomical Filters' column has a list of filters (None, Top 20 list, Planetary Atmosphere, Hot Cores, Dark Clouds, Diffuse Clouds, Comets, AGB/PPN/PN, Extragalactic) with radio buttons. The 'Export Data' column includes 'Export Fields' (Export current fields, Export current fields without Resolved QNs, Export CASA fields), 'Field Separator' (Tab, Colon), 'Range' (All Records, Current Page), and an 'Export' button.

Hands-On Exercise

Data: Day2_spectral_line_data/TWHydra/TWHydra_corrected.ms

Use Splatalogue to figure out which spectral line is covered by the first spectral window (SPW 0).

Cleaning Parameters and Concepts

Tasks: tclean

Example: TWHydra_corrected.ms

For spectral-line cubes, focus on the parameters that define which spectral data are imaged and how the output spectral axis is constructed:

- **spw**: input spectral window and channel selection
- **specmode='cube'**: produce spectral cube
- **nchan**: number of output channels
- **start**: coordinate of the first output channel
- **width**: output-channel spacing and direction
- **outframe**: output spectral reference frame
- **veltype**: velocity convention
- **restfreq**: rest frequency of the molecular line transition

TCLEAN Dirty Map Example:

```
tclean(vis="data/TWHydra_corrected.ms",  
       imagename=TWHydra_CO32_dirty.ms,  
       datacolumn='data',  
       specmode='cube',  
       restfreq='345.7959899GHz',  
       outframe='LSRK',  
       veltype='radio',  
       start='-2km/s',  
       width='0.25km/s',  
       nchan=41,  
       imsize=[128, 128],  
       cell='0.3arcsec',  
       gridder='standard',  
       weighting='briggs',  
       robust=0.5,  
       deconvolver='hogbom',  
       restoringbeam='common',  
       niter=0,  
       pblimit=0.2,  
       pbcor=False,  
       interactive=False)
```


Cleaning Parameters and Concepts

Tasks: tclean

Example: TWHydra_corrected.ms

For spectral-line cubes, focus on the parameters that define which spectral data are imaged and how the output spectral axis is constructed:

- **spw**: input spectral window and channel selection
- **specmode='cube'**: produce spectral cube
- **nchan**: number of output channels
- **start**: coordinate of the first output channel
- **width**: output-channel spacing and direction
- **outframe**: output spectral reference frame
- **veltype**: velocity convention
- **restfreq**: rest frequency of the molecular line transition

TCLEAN Dirty Map Example:

```
tclean(vis="data/TWHydra_corrected.ms",  
       imagename=TWHydra_CO32_dirty.ms,  
       datacolumn='data',  
       specmode='cube',  
       restfreq='345.7959899GHz',  
       outframe='LSRK',  
       veltype='radio',  
       start='-2km/s',  
       width='0.25km/s',  
       nchan=41,  
       imsize=[128, 128],  
       cell='0.3arcsec',  
       gridder='standard',  
       weighting='briggs',  
       robust=0.5,  
       deconvolver='hogbom',  
       restoringbeam='common',  
       niter=0,  
       pblimit=0.2,  
       pbcor=False,  
       interactive=False)
```


Cleaning Parameters and Concepts

Tasks: tclean

Example: TWHydra_corrected.ms

For spectral-line cubes, focus on the parameters that define which spectral data are imaged and how the output spectral axis is constructed:

- **spw**: input spectral window and channel selection
- **specmode='cube'**: produce spectral cube
- **nchan**: number of output channels
- **start**: coordinate of the first output channel
- **width**: output-channel spacing and direction
- **outframe**: output spectral reference frame
- **veltype**: velocity convention
- **restfreq**: rest frequency of the molecular line transition

TCLEAN Clean Map Example:

```
tclean(vis="data/TWHydra_corrected.ms",  
       imagename=TWHydra_CO32_dirty.ms,  
       datacolumn='data',  
       specmode='cube',  
       restfreq='345.7959899GHz',  
       outframe='LSRK',  
       veltype='radio',  
       start='-2km/s',  
       width='0.25km/s',  
       nchan=41,  
       imsize=[128, 128],  
       cell='0.3arcsec',  
       gridder='standard',  
       weighting='briggs',  
       robust=0.5,  
       deconvolver='hogbom',  
       restoringbeam='common',  
       threshold='0.02mJy',  
       niter=1000000000,  
       pblimit=0.2,  
       pbcor=False,  
       interactive=False)
```


Common
Astronomy
Software
Applications
Documentation:



Part 2:

Spectral Cube Manipulation



Headers

Tasks: imhead

Example: TWHydra_CO32.image

Mode	Purpose	Example
summary	Display an overall image summary and return metadata as a dictionary	<code>imhead("cube.image", mode="summary")</code>
list	Display supported header keywords and their current values	<code>imhead("cube.image", mode="list")</code>
get	Return one header value	<code>imhead("cube.image", mode="get", hdkey="restfreq")</code>
put	Replace an existing header value	<code>imhead("cube.image", mode="put", hdkey="restfreq", hdvalue="345.7959899GHz")</code>
add	Add a header keyword that does not already exist	<code>imhead("cube.image", mode="add", hdkey="observer", hdvalue="ALMA Workshop")</code>
del	Delete a keyword or reset its value when deletion is supported	<code>imhead("cube.image", mode="del", hdkey="observer")</code>
history	Print the image-processing history to the CASA logger	<code>imhead("cube.image", mode="history")</code>

Headers

Tasks: imhead

Example: TWHydra_CO32.image

Mode	Purpose
summary	Display an overall image summary and return metadata as a dictionary
list	Display supported header keywords and their current values
get	Return one header value
put	Replace an existing header value
add	Add a header keyword that does not already exist
del	Delete a keyword or reset its value when deletion is supported
history	Print the image-processing history to the CASA logger

CASA Example:

```
summary = imhead(imagename="TWHydra_CO32.fits",
mode='summary')

print('Axis names:', summary['axisnames'])
print('Shape:', summary['shape'])
print('Axis increments:', summary['incr'])
print('Brightness unit:', summary['unit'])

rest_frequency =
imhead(imagename="TWHydra_CO32.fits", mode='get',
hdkey='restfreq')
reference_frame = imhead(
imagename=str(CUBE_IMAGE), mode='get',
hdkey='reffreqtype')

print('Rest frequency:', rest_frequency)
print('Spectral frame:', reference_frame)
```

Output:

```
Axis names: ['Right Ascension' 'Declination'
'Frequency' 'Stokes']
Shape: [100 100 118 1]
Axis increments: [-1.45444104e-06 1.45444104e-06
-1.38414152e+05 1.00000000e+00]
Brightness unit: Jy/beam
Rest frequency: {'unit': 'Hz', 'value':
345795990000.0}
Spectral frame: LSRK
```


Regridding

Tasks: imregrid

Example: TWHydra_CO32.image

imregrid interpolates an image onto the coordinate system and shape of a template. Typical uses include matching cubes from different transitions or aligning an ancillary image with an ALMA cube.

Important cautions:

- Regridding is interpolation, not averaging.
- Specify `axes` so unaffected axes remain untouched.
- Check axis order with `imhead` first.
- Spectral regridding with per-plane beams requires a common beam first.
- Regridding a noisy image creates correlated noise.
- A PV image is not a supported regridding target; regrid the parent cube before making the PV slice.

CASA Example:

```
imregrid(imagename="TWHydra_CO32.image",
         template=template_cube.image,
         output="TWHydra_CO32_regrid.image",
         axes=[0, 1],
         interpolation="linear",
         overwrite=True)

summary =
imhead(imagename="TWHydra_CO32_regrid.image",
       mode='summary')

print('Axis names:', summary['axisnames'])
print('Shape:', summary['shape'])
print('Axis increments:', summary['incr'])
```


Rebinning

Tasks: imrebin

Example: TWHydra_CO32.image

imrebin combines adjacent pixels using integer factors. It can improve per-pixel signal-to-noise and reduce file size, but it lowers sampling and cannot recover information later.

The `factor` list follows the current image axis order. A standard `tclean` cube is often ordered RA, Dec, Frequency, Stokes.

But never assume this, always check `imhead`.

For Example:

`[2, 2, 1, 1]`: average 2×2 spatial pixels,
`[1, 1, 3, 1]`: average 3 spectral channels.

Do not confuse rebinning with convolution to a larger synthesized beam. Spatial rebinning changes pixel sampling; smoothing changes angular resolution.

CASA Example:

```
imrebin(imagename="imagename=TWHydra_CO32.image",  
        outfile=TWHydra_CO32_rebin.image  
        factor=[2, 2, 1],  
        overwrite=True)
```

```
summary =  
imhead(imagename="TWHydra_CO32_rebin.image",  
        mode='summary')
```

```
print('Axis names:', summary['axisnames'])  
print('Shape:', summary['shape'])  
print('Axis increments:', summary['incr'])
```


Transposing

Tasks: imtrans

Example: TWHydra_CO32.image

imtrans transposes both pixel values and the coordinate system. This is useful when another program expects a different axis order or when an image must match a CASA template's coordinate structure.

The supplied FITS cube is ordered:

`[RA, Dec, Frequency, Stokes]`

Setting `order='0132'` produces:

`[RA, Dec, Stokes, Frequency]`.

Axis indices are zero-based. String orders are safer than integers because a leading zero in an integer can be ambiguous.

A minus sign can reverse an axis, but verify the resulting coordinate increment and channel direction afterward.

CASA Example:

```
imtrans(imagename="TWHydra_CO32.image",
        outfile="TWHydra_CO32_transpose.image",
        order=["frequency", "declination",
               "right ascension"])

summary =
imhead(imagename="TWHydra_CO32_transpose.image",
       mode='summary')

print('Axis names:', summary['axisnames'])
print('Shape:', summary['shape'])
print('Axis increments:', summary['incr'])
```


Reframing

Tasks: imreframe

Example: TWHydra_CO32.image

The same measured frequency corresponds to different velocities in TOPO, BARY, LSRK, and other frames.

imreframe changes the frame in which an image reports spectral coordinates. It does not create a new regularly sampled velocity grid. Use `imregrid` when resampling onto a specified spectral grid is required.

Always supply or verify the rest frequency when interpreting an axis as velocity. Writing to a new `output` is safer than modifying an input image in place.

Our tclean example already requested `outframe='LSRK'`.

CASA Example:

```
imreframe(imagename="TWHydra_CO32.image",
          outfile="TWHydra_CO32_reframe.image",
          outframe="GEO",
          restfreq="345.7959899GHz",
          veltype="radio")

summary =
imhead(imagename="TWHydra_CO32_transpose.image",
       mode='summary')

print("Rest frequency:", summary['restfreq'])
print("Reference frame:", summary['reffreq'])
print("Velocity type:", summary['veltype'])
```


Common
Astronomy
Software
Applications
Documentation:



Part 3: Spectral Cube Analysis



Moment Maps

Tasks: imcollapse, immoments

Example: TWHydra_CO32.image

For intensity I_i at spectral coordinate ν_i and channel width $\Delta\nu$:

$$\text{Moment 0: } M_0(x, y) = \sum_i I_i(x, y) \Delta\nu_i$$

Integrated-Intensity [units: Jy beam⁻¹ km s⁻¹]

$$\text{Moment 1: } M_1(x, y) = \frac{\sum_i \nu_i I_i(x, y) \Delta\nu_i}{\sum_i I_i(x, y) \Delta\nu_i}$$

Intensity-Weighted Mean Velocity [units: km s⁻¹]

$$\text{Moment 2: } M_2(x, y) = \left[\frac{\sum_i (\nu_i - M_1(x, y))^2 I_i(x, y) \Delta\nu_i}{\sum_i I_i(x, y) \Delta\nu_i} \right]^{1/2}$$

Intensity-Weighted Velocity Dispersion [units: km s⁻¹]

CASA Example:

```
immoments(imagename="TWHydra_CO32.image",  
           moments=[0],  
           axis="spectral",  
           chans="18~32",  
           outfile="TWHydra_CO32_M0_nomask.image")
```

```
immoments(imagename="TWHydra_CO32.image",  
           moments=[0,1,2],  
           axis="spectral",  
           chans="18~32",  
           mask=f'"TWHydra_CO32_M0_nomask.image">0.028',  
           stretch=True,  
           excludepix=[-1000000, 22]  
           outfile="TWHydra_CO32_M0_3sigma.image")
```


Moment Maps

Tasks: imcollapse, immoments

Example: TWHydra_CO32.image

For intensity I_i at spectral coordinate ν_i and channel width $\Delta\nu$:

Moment 8: $M_8(x, y) = \max_{\nu \in \mathcal{V}} I(x, y, \nu)$

Peak Intensity [**units:** Jy beam⁻¹]

Moment 9: $M_9(x, y) = \nu_{i_{\text{peak}}(x, y)}$

Velocity at the Peak Intensity [**units:** km s⁻¹]

Summary of Moments:

- Moment -1: Mean value of the spectrum.
- Moment 3: Median intensity of the spectrum.
- Moment 4: Median spectral coordinate.
- Moment 5: Standard deviation of the intensity values about their mean.
- Moment 6: Root mean square (RMS) intensity.
- Moment 7: Mean absolute deviation of the intensity values.
- Moment 10: Minimum intensity of the spectrum.
- Moment 11: Spectral coordinate of the minimum intensity.

CASA Example:

```
immoments(imagename="TWHydra_CO32.image",  
           moments=[0],  
           axis="spectral",  
           chans="18~32",  
           outfile="TWHydra_CO32_M0_nomask.image")
```

```
immoments(imagename="TWHydra_CO32.image",  
           moments=[0,1,2],  
           axis="spectral",  
           chans="18~32",  
           mask=f'"TWHydra_CO32_M0_nomask.image">0.028',  
           stretch=True,  
           excludepix=[-1000000, 22]  
           outfile="TWHydra_CO32_M0_3sigma.image")
```


Estimating the Cube and Moment 0 RMS

Cube RMS

- Use emission-free channels
- Use an emission-free spatial region around the source
- Region should be similar size to your emission of interest
- Use several channels to get a per-channel average

CASA Example:

```
shape = imhead("target.image", mode="list")["shape"]
nx, ny, nchan = shape[0], shape[1], shape[2]

region = f"circle[{(nx-1)/2}pix,{(ny-1)/2}pix],2arcsec"

chans = f"0~9;{nchan-10}~{nchan-1}"

stats = imstat(imagename=cube,
               region= region,
               chans=chans)

print("RMS:", stats["rms"])
```

Moment 0 RMS

For N independent channels of width Δv and equal per-channel RMS:

$$\sigma(M_0) \approx \sigma_{\text{cube}} \Delta v \sqrt{N}$$

Example: $\sigma = 22 \text{ mJy beam}^{-1}$, $\Delta v = 0.25 \text{ km s}^{-1}$, and $N = 25$ gives $\sigma(M_0) \approx 0.028 \text{ Jy beam}^{-1} \text{ km s}^{-1}$.

Otherwise, use an emission-free spatial region off-center from the source, also with similar size to your emission.

Moment Maps

Tasks: imcollapse, immoments

Example: TWHydra_CO32.image

imcollapse applies an aggregate such as mean, sum, rms, median, or max along one or more axes. It is appropriate when the desired output is a generic statistic rather than a coordinate-weighted spectral moment.

The example makes an rms map from illustrative line-free channels. This reveals spatial noise variations and residual calibration structure.

CASA Example:

```
imcollapse(imagename="TWHydra_CO32.image",
           function="rms",
           axes="spectral",
           chans="0~10",
           outfile="TWHydra_CO32_rms.image",
           overwrite=True)

shape = imhead("TWHydra_CO32_rms.image",
               mode="list")["shape"]
nx, ny, nchan = shape[0], shape[1], shape[2]

region = f"circle[{(nx-1)/2}pix,{(ny-1)/2}
           pix],2arcsec]"

stats = imstat(imagename=TWHydra_CO32_rms.image,
               region= region)

print("RMS:", stats["rms"])
```


Position-Velocity Diagrams

Tasks: `impv`

Example: `TWHydra_CO32.image`

`impv` makes a position-velocity (PV) diagram extracts intensity along a spatial cut and retains the spectral axis. For a rotating disk, a cut near the projected major axis often shows the strongest velocity gradient.

The main parameters are:

- center of the cut,
- position angle, measured north through east,
- length of the cut,
- averaging width perpendicular to the cut
- spectral interval.

A wider slice improves sensitivity but mixes emission from different positions. Beam convolution means adjacent offsets are not statistically independent.

CASA Example:

```
impv(imagename="TWHydra_CO32.image",  
      outfile="TWHydra_CO32_pv.image",  
      mode="length",  
      center="J2000 17h56m21.277s -21d57m22.64s",  
      length="12arcsec",  
      pa="133deg",  
      width="0.3arcsec",  
      unit="arcsec",  
      overwrite=True)
```


Cube Analysis
and Rendering
Tool for
Astronomy
Documentation:



Part 4:

Spectral Cube Analysis in CARTA



Hands-On
Demo