ADMIT
ALMA Data Mining Toolkit

Peter Teuben
for:
the ADMIT team

UMD: Lee Mundy (PI), Peter Teuben, Marc Pound, Kevin Rauch
UIUC: Leslie Looney, Doug Friedel, Robert Harris, Lisa Xu
NRAO: Jeff Kern, Tony Remijan, John Hibbard, Mark Lacey

ALMA Future Science Development Program Workshop
Charlottesville, August 25, 2016

http://admit.astro.umd.edu/admit/
ADMIT:
Extract and add interesting science data products to the ALMA archive

• ALMA data delivered as FITS data cubes
  – Typical sizes: 1000 x 1000 x 4000
  – Several spectral windows, polarization

• CASA as base for ADMIT software (cf. ALMA pipeline)
  – Python toolkit (generic)
  – “flow” with reflow (ala “make”) and dependencies
    • AT = ADMIT Task
    • BDP = Basic Data Product (xml wrapper)
A simple ADMIT flow:
cube statistics, spectra, line-id, line-cubes
A simple ADMIT flow: with added continuum subtraction
import admit

# Master project.
p = admit.Project('test0.admit', commit=False)

# Flow tasks.
t0  = p.addtask(admit.Ingest_AT(basename='x', file='test0.fits'))
t1  = p.addtask(admit.CubeStats_AT((ppp=True), [t0]))
t2  = p.addtask(admit.CubeSum_AT(numsigma=4.0, sigma=99.0), [t0, t1])
t3  = p.addtask(admit.Moment_AT(mom0clip=2.0, numsigma=[3.0]), [t0, t1])
t4  = p.addtask(admit.SFind2D_AT(alias='csm', sigma=2.573442400847707), [t2])
t5  = p.addtask(admit.PVSlice_AT(clip=0.3, pvsmooth=[10, 10], width=5), [t0, t2])
t6  = p.addtask(admit.CubeSpectrum_AT(sources=[0, 1]), [t0, t1, t2, 'csm'])
t7  = p.addtask(admit.PVCorr_AT(), [t5, t1])
t8  = p.addtask(admit.LineSegment_AT(csub=[0, 0], minchan=4, numsigma=5.0), [t6, t1])
t9  = p.addtask(admit.LineID_AT(csub=[0, 0], references='etc/co_lines.list'), [t6,t1,t7])
t10 = p.addtask(admit.LineCube_AT(), [t0, t9])
t11 = p.addtask(admit.Moment_AT(mom0clip=2.0, moments=[0, 1, 2]), [t10, t1])
t12 = p.addtask(admit.CubeSpectrum_AT(), [t10, t11])

# Update project.
p.run()
ADMIT in theory
(Pipeline/Archive mode)

Watch this space
ADMIT in practice
(USER/DESKTOP mode)

% runa1 test0.fits
% casa

In a terminal:
...
> import admit
> a = admit.Project('test0.admit', dataserver=True)

Watch your default browser:
ADMIT in practice
(USER mode)

Watch your default browser:

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task Name</th>
<th>Parameters</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Ingest_AT</td>
<td>file=x.fits, mask=True</td>
<td>✔</td>
</tr>
<tr>
<td>1</td>
<td>CubeStats_AT</td>
<td>robust=medabsdevmed, ppp=True</td>
<td>✔</td>
</tr>
<tr>
<td>2</td>
<td>CubeSum_AT</td>
<td>numsigma=4.0, sigma=0.0013133, smooth=[]</td>
<td>✔</td>
</tr>
<tr>
<td>3</td>
<td>SFind2D_AT</td>
<td>nsigma=6.0, sigma=2.57344, region=robust=[hin'-15], snmax=35.0</td>
<td>✔</td>
</tr>
</tbody>
</table>
| 4       | CubeSpectrum_AT    | pos=[(68, 63), (69, 63), (10h27m51.227s, '-43d54m18.444s'), (10h27m50.243s, '-43d54m24.366s)] | x.im | ✔
| 5       | LineSegment_AT     | numsigma=5.0, minchan=4, maxgap=3, segment=ADMIT, smooth=[] | ✔      |
| 6       | PVSlice_AT         | slice=['6.00', '51.28', '121.00', '73.02'], width=5 | ✔      |
| 7       | PVCorr_AT          | numsigma=3.0, range=['16', '32'] | ✔      |
| 8       | LineID_AT          | numsigma=5.0, minchan=4, maxgap=3, recarb=shallow, smooth=[], tier1width=0.0, csub=[0], iterate=True | ✔      |
| 9       | LineCube_AT        | pad=5, equalize=False | ✔      |
| 10      | Moment_AT          | moments=[0, 1, 2], numsigma=[2.0], mom0=clip=2, chans=all, x:CO_115.27120 | ✔      |
| 11      | CubeSpectrum_AT    | pos=[(68, 63)] | x.CO_115.27120 | ✔      |
| 12      | Moment_AT          | moments=[0], numsigma=[3.0], mom0=clip=2, chans=all, x@1.mom | ✔      |
ADMIT in practice
(USER mode)

CubeStats_AT computes image-plane robust statistics on datacubes. These statistics are particularly useful for identifying spectral lines in images where the noise varies as a function of frequency.

<table>
<thead>
<tr>
<th>CASA image</th>
<th>x.im</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS method</td>
<td>medabsdevmed</td>
</tr>
<tr>
<td>RMS value</td>
<td>1.131E-03</td>
</tr>
<tr>
<td>Dynamic range</td>
<td>1.361E+03</td>
</tr>
<tr>
<td>Data mean</td>
<td>8.325E-04</td>
</tr>
</tbody>
</table>

Emission characteristics as a function of channel, as derived by CubeStats_AT (cyan: global rms, green: noise per channel, blue: peak value per channel, red: peak/noise per channel).

Peak point plot: Locations of per-channel peaks in the image cube x.im
ADMIT in practice
(USER mode)
ADMIT in practice
(USER mode)

ADMIT Task Inputs for test0.admit

### Parameters

#### Input BDPs

<table>
<thead>
<tr>
<th>Index</th>
<th>Type</th>
<th>File name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CubeSpectrum_BDP</td>
<td>x.csp</td>
</tr>
<tr>
<td>1</td>
<td>CubeStats_BDP</td>
<td>x.cat</td>
</tr>
<tr>
<td>2</td>
<td>PVCorr_BDP</td>
<td>x.pvc</td>
</tr>
</tbody>
</table>

#### Output BDPs

<table>
<thead>
<tr>
<th>Index</th>
<th>Type</th>
<th>File name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LineList_BDP</td>
<td>x.l</td>
</tr>
</tbody>
</table>

### Keywords

- **numsigma**: 5.0
- **force**: 0
- **tier1width**: 0.0
- **online**: False
### ADMIT in practice

**USER mode**

#### ADMIT Task Inputs for test0.admit

<table>
<thead>
<tr>
<th>muncnan</th>
<th>pattern</th>
<th>smooth</th>
<th>recalcnoise</th>
<th>vlsr</th>
<th>maxgap</th>
<th>reject</th>
<th>method</th>
<th>mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>AUTO</td>
<td></td>
<td>False</td>
<td>-999999.99</td>
<td>3</td>
<td>[]</td>
<td>('PeakFinder': {'thresh': 0.0})</td>
<td>ONE</td>
</tr>
</tbody>
</table>

**Task IDs**

- LineCube_AT (taskid=9)
- Moment_AT (taskid=10)
- CubeSpectrum_AT (taskid=11)
- Moment_AT (taskid=12)

[Update ADMIT flow state (dry run)]

[Re-run ADMIT flow]
ADMIT in practice (USER mode)

**Moment_AT** creates moment maps using custom clip levels.

Moment_AT output for x.CO_115.27120/lc.im

- [Carbon Monoxide Moment 0 map of Source NGC3256](View in CASA)
- [Carbon Monoxide Moment 1 map of Source NGC3256](View in CASA)
ADMIT in practice – mode 2
(USER mode using file browser)
Random Notes

- Lack of VLSR in FITS header (where useful)
- Continuum subtraction not always done
- lustre FS unstable for nautilus (dolphin ok)
- Firefox threading/signal bug (no chrome @ NRAO)
- Manual pipeline line-free channels shortage
- New CASA code (robust stats)
- Old CASA shakedown
LineID without any priors?

- Missing priors:
  - VLSR
    - Simbad/NED(object) [vlsr.tab cheat]
    - VLSRc = RESTFREQ vs. centerfreq
    - VLSRf = RESTFREQ vs. linefreq (high-z cheaters)
  - Object Type (e.g. galaxy, high-z, hot core, outflow)
    - cf. splatalogue’s “Astronomical Filters”
CubeStats (AT and BDP)
Feedback to imaging?

- **CubeStats_AT**: \( \text{rms(channel)} \) plot
  - Aliasing/periodic tick (up/down) in RMS (v-grid)
  - Periodic variations
  - Edge channel (begin or end?) - v-grid?
  - Crazy variations
CubeStats (AT and BDP)
CubeStats (AT and BDP)
CubeStats (AT and BDP)
CubeStats (AT and BDP)
CubeStats (AT and BDP)
CubeStats (AT and BDP)
CubeStats (AT and BDP)
CubeStats (AT and BDP)
LineID without any priors?

• Missing priors:
  
  – VLSR
    • Simbad/NED(object) [vlsr.tab cheat]
    • VLSRc = RESTFREQ vs. centerfreq
    • VLSRf = RESTFREQ vs. linefreq
  
  – Object Type (e.g. galaxy, high-z, hot core, outflow)
    • cf. Splatalogue’s “Astronomical Filters”
Timeline

• 2013-2014  ALMA Development Study (astute)
• May 1, 2016:  ADMIT 1.0 (UMD/UIUC)
• Oct 1, 2016:  ADMIT 1.1 (UMD/NRAO)
• ??? :  ADMIT 2.0

http://admit.astro.umd.edu/admit/