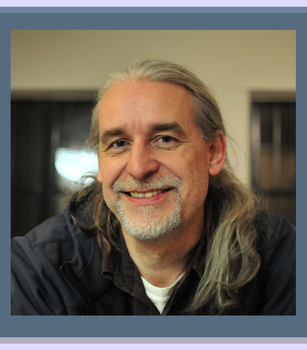
The ALMA Data Mining Toolkit



Peter Teuben[1], Marc Pound[1], Kevin Rauch[1], Lee Mundy[1],
Leslie Looney[2], Douglas Friedel[2], Lisa Xu[2], Jeff Kern[3]

[1]University of Maryland [2]University of Illinois [3]NRAO

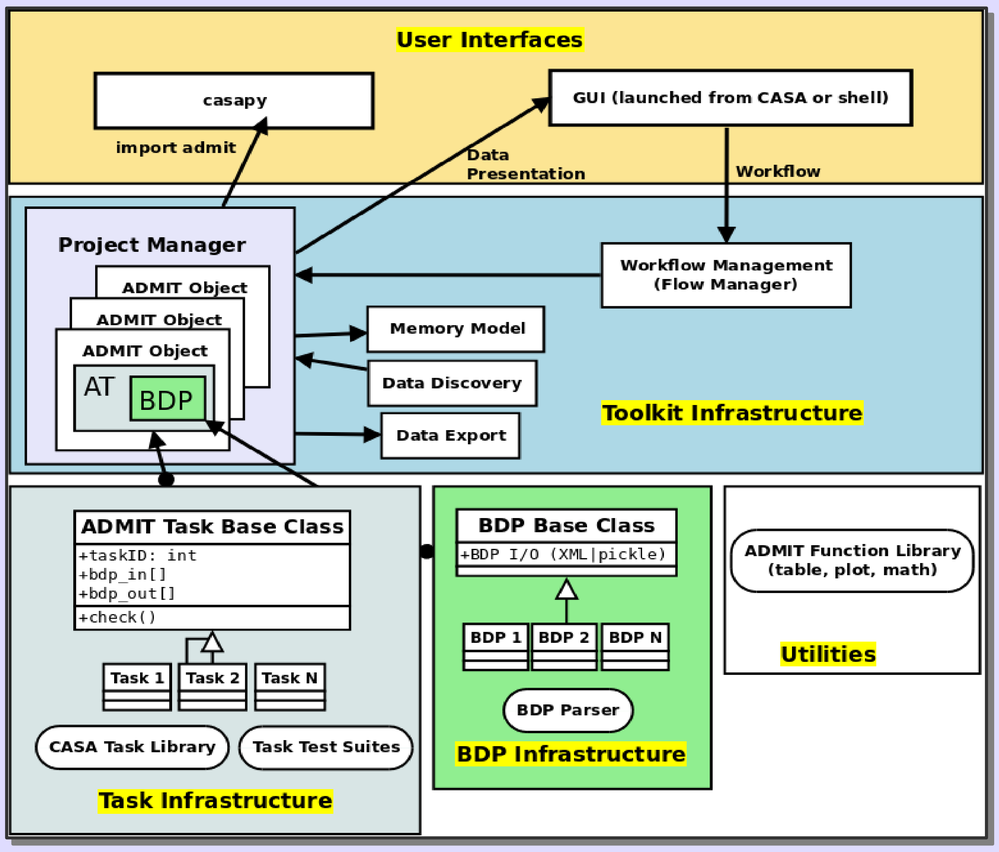
Abstract

We report on a toolkit **ADMIT** (**A**LMA **D**ata **Mi**ning **T**oolkit) that works within the python based CASA environment. It will produce detailed science products from ALMA image data cubes and make them available within the ALMA archive, as well allow users to reproduce and further enhance these data on their workstation.

At the archive:
- Ingest image data cube (FITS file)
- Process data: Line ID, Line Cubes, Feature Detection, etc.
- Save state in a small (<20MB) **admit**.zip file
- Enrich archive with descriptor vector of discovered data for later data mining

At the workstation:
- Query archive for interesting datasets, download **admit**.zip's
- Preview pipeline results in an ADMIT GUI
- Fine tune pipeline to extract more data or verify results
- Add your own ADMIT analysis tasks
- Extract data and move to other python or non-python environments

ADMIT Architecture



Take Home Points: ADMIT ...

- ...is an ALMA development study, delivering a functional toolkit after 2 years (1.5 years from now)
- ...give users easy access to science data products derived from ALMA pipeline data image cubes
- ...is a platform for developing more sophisticated data mining algorithms
- ...can be directed from python scripts as well as from a GUI
- ...can do effective data mining via its python interfaces, and integrates well with community software.
- ...could be extended to also work on other science data cubes (ATCA, CARMA, SMA, VLA, WSRT etc.)
- ...executes basic CASA tasks, but could also extended to use other packages, such as MIRIAD, or radio_astro_tools

Initializing ADMIT projects

```
import admit

ff = admit.ADMIT.discover(mode='ALMA',rootdir='.')

for f in ff:                              # loop over all fits files
   a = admit.ADMIT(f)                     # new admit object
   a.plotmode(0,'png')                    # set a plotmode
   a.pushd()                              # change directory here
   if a.new():                            # a new admit?
      a.pipeset('pickle=1')               # global parameter
      a.pipeset('maxline=0')              # global parameter
      a.pipeset('cubestats/robust=1')     # module parameter
      a.pipeset('moments/sigmaclip=3')    # module parameter
      a.pipeload('cubestats,moments')     # construct a pipeline
   a.piperun()                            # run it
   a.save()                               # save it (XML/BDP)
   a.popd()                               # pop back the directory
```

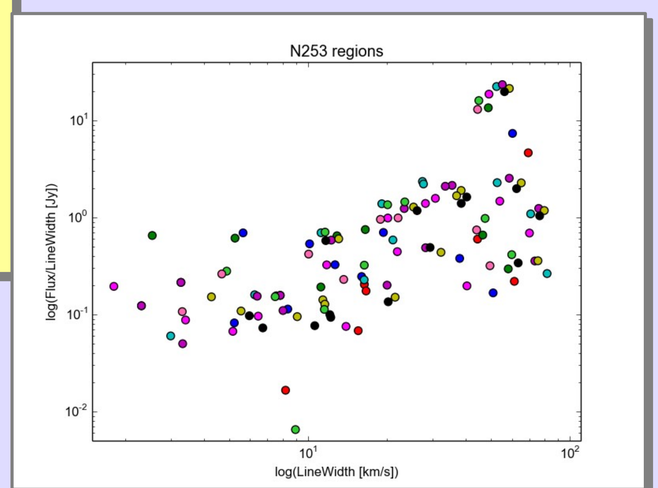
Virtual ADMIT projects + Data Mining

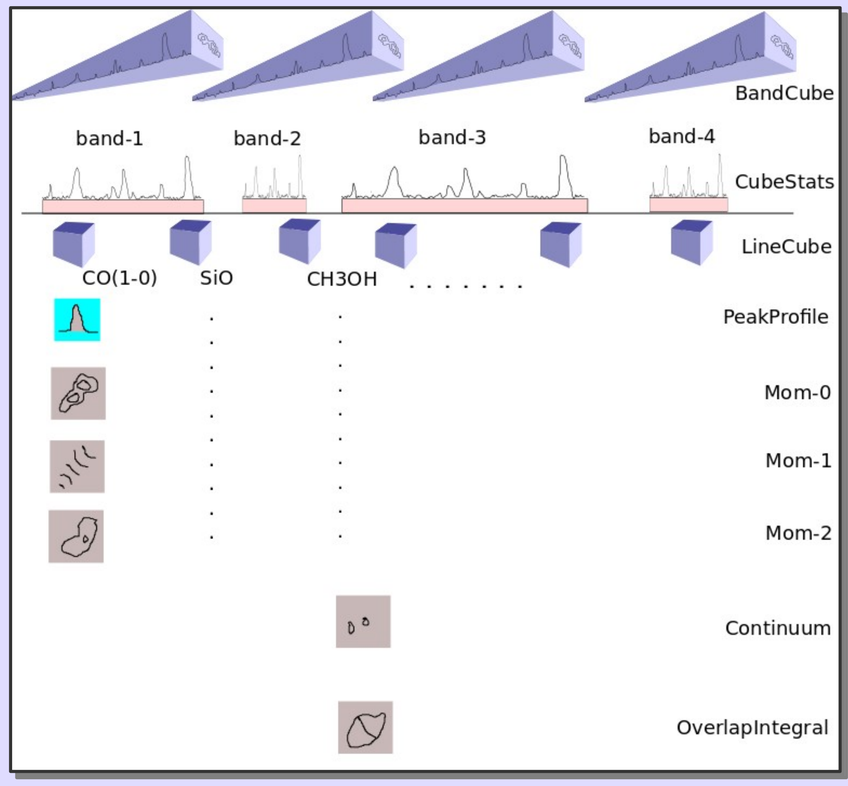
```
import admit

# create a new project for a specific question
q1 = admit.ADMIT('q1')
p1 = a1.discover(rootdir=["/data/alma"],mode="ALMA")
     q1.add_project(p1)
p2 = a1.discover(rootdir=["/data/carma"])
     q1.add_project(p2)
bdp = q1.query_bdp(type="linecube,mom0")
bdp.sort('sum')
cartaviewer(bdp, animate=True, channel=1)

# add new ADMIT task
a1 = AT_overlap('q1-top8')
i1 = q1.add(a1, bdp[:10])
a1.run()
cartaviewer('q1-top8', channel=2)

# from here on, some manual python labor
# correlate intensity to linewidth for different regions
regions = ['a1.rng', 'a2.rng', 'a3.rng', 'a4.rng']
for b1 in bdp:
   b2 = bdp_sibling(b1,'mom2')
   for r in regions:
      h1 = imstat(bdp_name(b1),region=r)
      h2 = imstat(bdp_name(b2),region=r)
      print r,b1.restfreq,h1['flux'],h2['mean']
```

Concepts:

- AT (ADMIT Task)
  - Wraps tools and tasks, but in a package agnostic way
  - Easy to write your own AT's
- BDP (Basic Data Product)
  - Wraps data in XML (most big data via XLINK)
  - BDP's are owned by an AT (see flow diagram below)
  - Easy to write your own BDP's
- ADMIT
  - Contains state of the project (AT + BDP + flow)
  - Multiple ADMIT's combine into a virtual project

Flow diagram showing line identification and cutting line cubes for further analysis]





Overview of some of the ADMIT data products: starting with Band Cubes and line identification, Line Cubes can be cut, with derived products such as Peak Profile, Moment Maps, Feature Extraction, Overlap Integrals, various tables, and a suite of Description Vectors derived from these.

Future Development:

- GUI
  - Python command line only access now
- Interfaces to external (python) world
  - GLUEviz
  - AstroML
- ADMIT service on archive servers?
  - Closer connection ADMIT products to original ALMA data cubes