LMT Spectral Line Pipeline

A pragmatic overview (December 2021, LMTOY version 0.2)

Talk Overview

- 1. Introduction of concepts
- 2. Pipeline and Data: overview of the current "load and go"
- 3. Instrument Specifics
 - a. RSR processing (+ live demo)
 - b. SEQ processing
 - c. 1MM processing (for later?)
- 4. SRDP and ADMIT post-processing
- 5. Future work

Concepts

- Observations are organized in **ProjectId** and **ObsNum**
- ObsNum's can be stacked (spectra or cubes)
- Data can be written in a simplified form:
 - Single obsnum: data(time, beam, pol, band, chan)
 - Wider view: data(projectid; obsnum; time, beam, pol, band, chan)
- Current pipeline is "load and go"
- Data are made available as: RAW, TAP, SRDP
 - **RAW** = several NetCDF files in a specific directory hierarchy
 - **TAP** = Timely Analysis Products few text and figures with html
 - **SRDP** = Science Ready Data Products all files

LMTOY: exploring a user level unified SL pipeline for LMT

- LMTOY = toolbox of existing packages (Imtslr, dreampy3, ...)
- How do you run the pipeline (from your SHELL)
 - 1. Setup: source Imtoy_start.sh
 - 2. One obsnum: SLpipeline.sh obsnum=12345
 - 3. Stacking: SLpipeline.sh obsnums=12345,12347,12349
- How do you find your obsnums: Shell: Intinfo.py grep \$source
- Where are the data:
 - One obsnum:\$WORK_LMT/ProjectId/obsnumStacking:\$WORK_LMT/ProjectId/obsnum0_obsnum1
- Pipeline will also run ADMIT on the spectrum/cube

Overview of Data Flow

- Observations are done per **ObsNum** and produce **RAW** data
 - \circ Copied to UM within 10 mins 10 days
- Pipeline at LMT produces small Timely Analysis Products at LMT (TAP)
 - Can be viewed at LMT with a web browser
 - Only single ObsNum's will be produced, no stacking
 - TAP is copied to UM within 1–10 mins of observation and can be viewed with web browser
- Pipeline at UM produces Scientific Ready Data Products (SRDP)
 - Can be downloaded from web browser
 - RAW data can be requested, and PI is responsible to set up their own LMTOY
- Pipeline at UM could detect same object in **ProjectId** and will stack
 - Same TAP and SRDP data will be produced, looking similar to single ObsNum pipeline

Currently installed LMTOY:

Setup: source \$LMTOY/Imtoy_start.sh - the command **Imtoy** displays these settings

Machine	\$LMTOY	\$DATA_LMT	\$WORK_LMT
Umass: cln	~teuben/lmtoy	~teuben/LMT/data_lmt	
Umass: unity	~lmtslr_umass_edu/lmtoy	\$LMTOY/data_lmt	
UMD: Ima	/lma1/teuben/lmtoy	/lma1/lmt/data_lmt/	/lma1/teuben/LMT/work_lmt/
LMT: malt(slrtac)	~lmtslr/lmtoy	~lmtslr/data_lmt	~lmtslr/work_lmt
generic	~/lmtoy	\$LMTOY/data_lmt	\$LMTOY/work_lmt

https://www.astro.umd.edu/~teuben/LMT/live/ No username/password needed. https://www.astro.umd.edu/~teuben/work_Imt Username: Imtoy Password: yotml

RSR (Redshift Receiver)

• Live demo will take ~1 minute

Sequoia (WARES)

• Looking at existing results (pipeline takes about 5-10 mins)

1MM (WARES)

• Not covered yet by SLpipeline, though an example exists

Advanced Concepts

- Combining ObsNum's
 - For SEQ there is a good parallel mode using GNU parallel
- Editing parameter files (Imtoy_OBSNUM.rc)
 - RSR: rfile, blanking, badlags
 - SEQ: no others yet

Remaining work?

- 1. Manual
- 2. Feed PI parameters into SLpipeline
- 3. ADMIT for CASA6
- 4. Consistent CLI in various scripts
- 5. Unified RSR processing (2 scripts now)
- 6. Better workflow for those in a pure python environment (e.g. jupyter)
- 7. SRDP most likely will need editing before truly SRDP
- 8. Using SDFITS as the storage format for (calibrated) spectra
- 9. Interactive pipeline re-processing via a web interface
- 10. Archive with SRDP indexed
- 11. Multi-band OTF (Omaya, SEQ++)

What workflow are we using for the DA?

- Incoming data is automatically detected and pipeline will be run
 a. If not, the DA should be able to see (and fix?) this
- 2. New TAP data can be reviewed, and some QA is done
 - a. DA comments would be useful to gather
 - b. SRDP data is also available for the PI to be downloaded
- 3. Policy on combining data to be reviewed
 - a. Auto-combined based on source name can be a PI parameter (or always)
 - b. ...
- 4. ..

Installing your own LMTOY?

- See <u>https://github.com/astroumd/Imtoy</u>
- But essentially:
 - 1. wget https://astroumd.github.io/Imtoy/install_Imtoy
 - 2. bash install_Imtoy
 - 3. source Imtoy/Imtoy_start.sh
- Resources:
 - 12 minutes CPU
 - \circ 9 GB disk space, including a regression test, admit and anaconda3
- Problems:
 - <u>https://github.com/astroumd/Imtoy/issues</u>
 - teuben@umd.edu
 - 0