The ALMA (and JVLA) Pipeline

Andy Biggs and Liz Humphreys ALMA Regional Centre, European Southern Observatory







The need for pipelines

- ALMA datasets can be HUGE
 - Up to 64 antennas (>2000 baselines!)
 - Up to 8 x 8192 channels per integration
 - Targets observed for tens of hours
 - Data sets with TB of data are not unusual
- Processing times are similarly now very long
 - Several days can be required in some cases
- Flagging, calibration and imaging by a human is often simply not possible
- Automatic re-processing of data becomes possible
 - Data products in archive can be updated as Pipeline improves

The ALMA archive

Aims to provide science-ready data products

🔍 🔍 🔍 🦹 Cycle 6 Pha	A News 🥏 Lyman alpha <u>द</u> ReleasesPla	(ICT-2499)	g । 😤 [SCIREQ-110 💶 F2FOctober2 🌿 kennzeichen 🌔 ERIS2017: Eu 👻 [ICT-11157] 🖉 🌉 Alma Scie 🗙 🕂	-
(i) almascience.eso.org/aq/			C 🔍 alma pipeline users guide 🔶 🟠 💼 🖡 🎓 😰 💥 🔳	Ξ
Most Visited News Wiki ALMA ESO Lan	nguages APOD ADS Astro-ph YouTube eBa	y Das Wetter M94.5 MVV Metrono	ime	
ALMA Science Archi	ive Query			
Query Form Results Table			Quartila	
Search Reset			<u>Query netp</u>	
Position	Energy	Time	Polarisation	
Source name (Resolver) HL TAU Source name (ALMA) RA Dec Galactic Target list Angular resolution Largest angular scale Field of view Observation Line sensitivity (10 km/s) Continuum sensitivity Water vapour	Source name (Resolver) Case-insensitive search for source name, to be resolved with Sesame. Wildcard matching is disabled. Search is performed within a radius of 10 arcminutes. A search radius in degrees can be added to the end separated by a comma. Description. Use Sesame (via. NED, Simbad and VizieR) to parse names commonly found throughout literature. A green tick indicates a successful search, otherwise, a red cross is returned. Example Cen A NGC3375 ARP220, 20	Durce 'HL Tau bordinates (RA Dec) :31:38.43 +18:13:57.6 bject type I* (T Tau-type Star) esolver assame using <u>Simbad</u>	Polarisation type Options View: • observation • project • publication • publication • publication • publication • publication only • science observations only	
			<pre>https://almascience.eso.org/aq</pre>	

CASA pipeline

- Used by both ALMA and the JVLA
- Dedicated pipeline tasks within **CASA** are used
 - E.g. hif_flagdata, hif_setjy, hif_bandpass, hif_applycal, etc.
- Prefixes are actually meaningful
 - hif: interferometry (ALMA and VLA)
 - hifa: interferometry (ALMA-only)
 - hifv: interferometry (VLA-only)
 - hfs: single dish
- Pipeline is data-driven
 - In principle, no user input is required
 - Heuristics have been developed to help Pipeline decision making

The ALMA Pipeline

- All ALMA data is calibrated and imaged by the Observatory
 - Data not released to PI until it has undergone 'Quality Assurance'
- Huge progress in developing the Pipeline has been made
 - The calibration pipeline was first used in Cycle 2
 - 75% of data is now calibrated without human intervention
 - First imaging deliveries in Cycle 4
 - Imaging in Cycle 5
 - Expect that only 30% of projects will require human intervention
 - 50% of data will be calibrated and imaged completely automatically
- Not all observing modes can be pipelined
 - e.g. polarization, bandwidth-switching (for narrow bandwidths)
 - "Non-standard modes" restricted to 20% of ALMA observing time

Pipeline User's Guide

Doc 4.13, ver. 1 | October, 2016

ALMA Science Pipeline User's Guide for CASA 4.7.0

Interferometric and Single-Dish Data



Available from the ALMA Science Portal:

https://almascience.eso.org

A CASA Guide also exists:

https://casaguides.nrao.edu/index.php/ALMA_Imaging_Pipeline_Reprocessing

Running the Pipeline

- Must use a version of CASA which includes the Pipeline
 - Not every CASA release includes the Pipeline
 - The last release with a Pipeline was 4.7.2
 - The Cycle-5 Pipeline will appear in CASA 5.1
 - Current release does not include the Pipeline
 - This will be updated at some point
- Must start CASA in a special mode
 - >casa --pipeline (must be done from the command line)
- Data download will contain various Python scripts e.g.
 - casa_pipescript.py (imaging and/or calibration)
 - scriptForImaging.py (if imaging was done manually)
 - Run in usual way e.g. >execfile('casa_pipescript.py')

Why re-run the Pipeline?

- Remove datasets from the calibration/imaging
 - ALMA observes in approx. 1-hour sessions called Execution Blocks
 - Each is delivered to the user as an 'ASDM' (ALMA Science Data Model)
- Introducing additional flagging
 - Edit the flagtemplate.txt file
- Setting more accurate calibrator flux densities
 - More reliable values may now be available
- Modifying the imaging parameters
 - Weighting
 - Tapering
 - Redoing continuum subtraction
 - And many, many more ...

Example script

<pre>File Edit Options Buffers Tools Python Help rethrow_casa_exceptions = True h_init() #pipelinemode='automatic' #context=h_init() #context.project_summary.proposal_code = '2016.1.*****.5' #context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' try: hifa_importdata(dbservice=False, vis=['/rawdata/uidA002_X72e960_****']) execfile('fixIF.casa.py') h_save() h_init() context.project_summary.proposal_code = '2016.1.*****.5' context.project_summary.proposal_code = '2016.1.*****.5' context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' hifa_importdata(dbservice=False, vis=['uidA002_X72c4aa_****']) hifa_flagdata(pipelinemode="automatic") hifa_fluxcaflag(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_spuphaseu(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_styscl(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_bandpas(pipelinemode="automatic") hifa_bandpas(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_bandpas(pipelinemode="automatic") hifa_fluxscale(pipelinemode="automatic") hifa_fluxscale(</pre>	<pre>File Edit Options Buffers Tools Python Helprethrow_casa_exceptions = True h_init() #pipelinemode='automatic' #context=h_init() #context.project_summary.proposal_code = '2016.1.*****.S' #context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' try.</pre>
<pre>rethrow_casa_exceptions = True h_init() #pipelinemode='automatic' #context=h_init() #context.project_summary.proposal_code = '2016.1.*****.5' #context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' try: hifa_importdata(dbservice=False, vis=['/rawdata/uidA002_X72e960_*****']) execfile('fixIF.casa.py') # h_save() # h_init() # context.project_summary.proposal_code = '2016.1.*****.S' # context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' # hifa_importdata(dbservice=False, vis=['uidA002_X72c4aa_****']) hifa_flagdata(pipelinemode="automatic") hifa_flagdata(pipelinemode="automatic") hifa_fstyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_usyscal(pipelinemode="automatic") hifa_suproglipelinemode="automatic") hifa_antpos(pipelinemode="automatic") hifa_antpos(pipelinemode="automatic") hifa_gipipelinemode="automatic") hifa_gipipelinemode="automatic") hifa_gipipelinemode="automatic") hifa_fig_gipipelinemode="automatic") hifa_gipluscal(pipelinemode="automatic") hifa_gipluscal(pipelinemode="automatic") hifa_gipluscal(pipelinemode="automatic") hifa_gipluscal(pipelinemode="automatic") hifa_gipluscal(pipelinemode="automatic") hifa_gipluscal(pipelinemode="automatic") hifa_finemisin(intent='PHASE, BANDPASS, CHECK') hif_makeimist(strent='PHASE, BANDPASS, CHECK') hif_makeimist(sperimeode="automatic") hif_flagtargets(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_functon(pipelinemode="automatic") hif_functon(pipel</pre>	<pre>rethrow_casa_exceptions = True h_init() #pipelinemode='automatic' #context=h_init() #context.project_summary.proposal_code = '2016.1.*****.S' #context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' try'.</pre>
<pre>h_nint() #pipelinemode='automatic' #context-project_summary.proposal_code = '2016.1.*****.5' #context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' try: hifa_importdata(dbservice=False, vis=['/rawdata/uidA002_X72e960_*****']) execfile('fixIF.casa.py') # h_save() # h_init() # context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' # hifa_importdata(dbservice=False, vis=['/rawdata/uidA001_****_X2e7' # hifa_importdata(dpservice=False, vis=['/rawdata/uidA001_****_X2e7' # hifa_importdata(dpservice=False, vis=['/rawdata/uidA001_****_X2e7' # hifa_importdata(dpservice=False, vis=['/rawdata/uidA001_****_X2e7' # hifa_importdata(dpipelinemode="automatic") hifa_flagdata(pipelinemode="automatic") hifa_fusycal(pipelinemode="automatic") hifa_tsysta(pipelinemode="automatic") hifa_usycal(pipelinemode="automatic") hifa_sety(pipelinemode="automatic") hifa_timegatical(pipelinemode="automatic") hifa_timegatical(pipelinemode="automatic") hifa_timegatical(pipelinemode="automatic") hifa_timegatical(pipelinemode="automatic") hifa_timegatical(pipelinemode="automatic") hifa_timegatical(pipelinemode="automatic") hifa_timegatical(pipelinemode="automatic") hifa_timegatical(pipelinemode="automatic") hifa_timegatical(pipelinemode="automatic</pre>	<pre>h_init() #pipelinemode='automatic' #context=h_init() #context.project_summary.proposal_code = '2016.1.*****.S' #context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' try'</pre>
<pre>#ppelinemode='automatic' #context.project_summary.proposal_code = '2016.1.*****_S' #context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' try: hifa_importdata(dbservice=False, vis=['/rawdata/uidA002_X72e960_*****]) execfile('fixIF.casa.py') h_save() h_init() t context.project_summary.proposal_code = '2016.1.*****.S' context.project_summary.proposal_code = '2016.1.*****.S' context.project_summary.proposal_code = '2016.1.*****.S' context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' hifa_importdata(dbservice=False, vis=['uidA002_X72c4aa_****']) hifa_fluagdata(pipelinemode="automatic") hifa_fluxcalflag(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_dsyrgaiflag(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_styphaseup(pipelinemode="automatic") hifa_folg(pipelinemode="automatic") hifa_folg(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_imakeinist(intent="PHASE, BANDPASS,CHECK') hif_makeinist(specmode="automatic") hif_indcont(pipelinemode="automatic") hif_indcont(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtar</pre>	<pre>#pipelinemode='automatic' #context=h_init() #context.project_summary.proposal_code = '2016.1.*****.S' #context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' try</pre>
<pre>#context=m_init() #context.project_summary.proposal_code = '2016.1.*****.S' #context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' try: hifa_importdata(dbservice=False, vis=['/rawdata/uidA002_X72e960_****']) execfile('fixIF.casa.py') # h_save() # h_init() # context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' # hifa_importdata(dbservice=False, vis=['uidA002_X72e960_****']) hifa_flagdata(pipel.structure.ousstatus_entity_id = 'uidA001_****_X2e7' # hifa_importdata(dbservice=False, vis=['uidA002_X72c4aa_****']) hifa_flagdata(pipelinemode="automatic") hifa_flagdata(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_dag(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_spwphaseu(pipelinemode="automatic") hifa_spwphaseu(pipelinemode="automatic") hifa_spwphaseu(pipelinemode="automatic") hifa_spwphaseu(pipelinemode="automatic") hifa_spwphaseu(pipelinemode="automatic") hifa_spwphaseu(pipelinemode="automatic") hifa_spwphaseu(pipelinemode="automatic") hifa_spwphaseu(pipelinemode="automatic") hifa_spwphaseu(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_atimegaincal(pipelinemode="automatic") hif_atimegaincal(pipelinemode="automatic") hif_imakeimist(intent='PHASE, BANDPASS,CHECK') hif_makeimist(specmode="automatic") hif_indcont(pipelinemode="automatic") hif_indcont(pipelinemode="automatic") hif_indcont(pipelinemode="automatic") hif_indcont(pipelinemod</pre>	<pre>#context=n_init() #context.project_summary.proposal_code = '2016.1.*****.S' #context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' trv</pre>
<pre>#context.project_summary.proposal_code = '2016.1.*****.5' #context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' try: hifa_importdata(dbservice=False, vis=['/rawdata/uidA002_X72e960_****']) execfile('fixIF.casa.py') # h_save() # h_init() # context.project_summary.proposal_code = '2016.1.*****.5' # context.project_summary.proposal_code = '2016.1.*****.5' # context.project_summary.proposal_code = '2016.1.*****.5' # context.project_structure.ousstatus_entity_id = 'uidA001_*****_X2e7' # hifa_importdata(dbservice=False, vis=['uidA002_X72c4aa_****']) hifa_flagdata[pipelinemode="automatic"] hifa_flagdata[pipelinemode="automatic"] hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_antpos(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_sety(pipelinemode="automatic") hifa_sety(pipelinemode="automatic") hifa_styscal(pipelinemode="automatic") hifa_tifa_supphaseup(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_styscal(pipelinemode="automatic") hifa_stifa_flag(pipelinemode="automatic") hifa_stifa_flag(pipelinemode="automatic") hifa_tifa_supphaseup(pipelinemode="automatic") hifa_tifa_spuphaseup(pipelinemode="automatic") hifa_tifa_spuphaseup(p</pre>	<pre>#context.project_summary.proposal_code = '2016.1.*****.5' #context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' trv.</pre>
<pre>#context.project_structure.ousstatus_entity_ld = 'uldA001_*****_X2e'' try: hifa_importdata(dbservice=False, vis=['/rawdata/uidA002_X72e960_*****']) execfile('fixIF.casa.py') # h_save() # h_init() # context.project_summary.proposal_code = '2016.1.*****.5' # context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' # hifa_importdata(dbservice=False, vis=['uidA002_X72c4aa_****']) hifa_flagdata(pipelinemode="automatic") hifa_flagdata(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_antpos(pipelinemode="automatic") hifa_swpraseu(pipelinemode="automatic") hifa_supraseu(pipelinemode="automatic") hifa_supraseu(pipelinemode="automatic") hifa_spuphaseu(pipelinemode="automatic") hifa_tsysflag(pipelinemode="automatic") hif_stell(pipelinemode="automatic") hifa_spuphaseu(pipelinemode="automatic") hifa_spuphaseu(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_tate=tryNASE_SANDPASS_CHECK') hif_makeimiist(intent='PHASE_SANDPASS_CHECK') hif_imakeimiist(specmode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_nakeimiist(specmode="automatic") hif_indcont(pipelinemode="automatic") hifa_nakeimiist(specmode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_nakeimiist(specmode="automatic")</pre>	<pre>#context.project_structure.ousstatus_entity_1d = 'u1dA001_****_X2e/' try.</pre>
<pre>try: hifa_importdata(dbservice=False, vis=['/rawdata/uidA002_X72e960_*****']) execfile('fixIF.casa.py') # h_save() # h_init() # context.project_summary.proposal_code = '2016.1.*****.S' # context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' # hifa_importdata(dbservice=False, vis=['uidA002_X72c4aa_****']) hifa_flagdata[pipelinemode="automatic"] hifa_flagdata[pipelinemode="automatic"] hifa_tsyscl(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_tsysflag(pipelinemode="automatic") hifa_tsysflag(pipelinemode="automatic") hifa_tsysflag(pipelinemode="automatic") hifa_tsysflag(pipelinemode="automatic") hifa_tsysflag(pipelinemode="automatic") hifa_bantpos(pipelinemode="automatic") hifa_bantpos(pipelinemode="automatic") hifa_sety(pipelinemode="automatic") hifa_sety(pipelinemode="automatic") hifa_spuphaseup(pipelinemode="automatic") hifa_spuphaseup(pipelinemode="automatic") hifa_flawtscale(pipelinemode="automatic") hifa_flawtscale(pipelinemode="automatic") hifa_flawtscale(pipelinemode="automatic") hifa_afluxscale(pipelinemode="automatic") hifa_afluxscale(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_atimegaincal(pipelinemode="automatic") hif_atimegaincal(pipelinemode="automatic") hif_atapipycal(pipelinemode="automatic") hif_atapipycal(pipelinemode="automatic") hif_atimegaincal(pipelinemode="automatic") hif_atapipycal(pipelinemode="automatic") hif_atapipycal(pipelinemode="automatic") hif_tcheckproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_makeimist(specmode="automatic") hif_fimakeimist(specmode="automatic") hif_imakeimist(specmode="automatic") hif_if_imaterist(specmode="automatic") hif_imakeimist(specmode="automatic") hif_imakeimist(specmode="automatic") hif_imakeimist(specmode="automatic") hif_ifucont(pipelinemode="automatic") hif_uvcontit(pipelinemode="automatic")</pre>	
<pre>hlia_lmportata(dbservice=ratse, vis=['/rawdata/didA002_A72e906_*****']) execfile('fixIF.casa.py') # h_save() # h_init() # context.project_summary.proposal_code = '2016.1.*****.S' # context.project_structure.ousstatus_entity_id = 'uidA001_*****_X2e7' # hifa_inportdata(dbservice=False, vis=['uidA002_X72c4aa_****']) hifa_fluxcalflag(pipelinemode="automatic") hif_rawflagchans(pipelinemode="automatic") hif_refant(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_astpscal(pipelinemode="automatic") hifa_astpscal(pipelinemode="automatic") hifa_astpscal(pipelinemode="automatic") hifa_astpscal(pipelinemode="automatic") hif_astpscal(pipelinemode="automatic") hif_astpscalflag(pipelinemode="automatic") hif_astpscalflag(pipelinemode="automatic") hif_astpscalflag(pipelinemode="automatic") hif_astpscalflag(pipelinemode="automatic") hif_astpy(pipelinemode="automatic") hif_astpy(pipelinemode="automatic") hif_astpy(pipelinemode="automatic") hifa_bandpas(pipelinemode="automatic") hifa_fistpy(pipelinemode="automatic") hifa_fistpy(pipelinemode="automatic") hifa_fistpy(pipelinemode="automatic") hifa_bandpas(pipelinemode="automatic") hifa_fistpy(pipelinemode="automatic") hifa_stpy(pipelinemode="automatic") hifa_fistpy(pipelinemode="automatic") hifa_fistpy(pipelinemode="automatic") hifa_fistpy(pipelinemode="automatic") hifa_fistpy(pipelinemode="automatic") hifa_fistpy(pipelinemode="automatic") hifa_fistpy(pipelinemode="automatic") hifa_fistpy(pipelinemode="automatic") hifa_fistpy(pipelinemode="automatic") hifa_timegincal(pipelinemode="automatic") hif_astproductsize(pipelinemode="automatic") hif_astproductsize(pipelinemode="automatic") hif_astermode="automatic") hif_fistermode="automatic") hif_fist</pre>	hife importants (dheerwise_False wise[1 (revidets (wid A002 V72e060 ****1])
<pre>bdet() file(file(</pre>	exectile light case out
<pre># h_init() # context.project_summary.proposal_code = '2016.1.*****.S' context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' # hifa_importdata(dbservice=False, vis=['uidA002_X72c4aa_****']) hifa_fluxcalflag(pipelinemode="automatic") hif_rawflagchans(pipelinemode="automatic") hif_refant(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsysflag(pipelinemode="automatic") hif_awrgcalflag(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_styscal(pipelinemode="automatic") hifa_styscal(pipelinemode="automatic") hifa_styple_linemode="automatic") hifa_tifa_bandpass(pipelinemode="automatic") hifa_styple_linemode="automatic") hifa_tifa_bandpass(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_anakeimist(intent='PHASE,BANDPASS,CHECK') hif_makeimist(intent='PHASE,BANDPASS,CHECK') hif_makeimist(intent='PHASE,BANDPASS,CHECK') hif_makeimist(intent='PHASE,BANDPASS,CHECK') hif_indcont(pipelinemode="automatic") hif_tcheckproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hif_indcont(pipelinemode="automatic") hif_indcont(pipelinemode="automatic") hif_indcont(pipelinemode="automatic") hif_indcont(pipelinemode="automatic") hif_indcont(pipelinemode="automatic") hif</pre>	# b sayo()
<pre># context.project_summary.proposal_code = '2016.1.*****.S' # context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' # hifa_flagdat(pipelinemode="automatic") hifa_flagdat(pipelinemode="automatic") hif_refant(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_wrgcalflag(pipelinemode="automatic") hif_gainflag(pipelinemode="automatic") hif_gainflag(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_tsyscl(pipelinemode="automatic") hifa_tsetjy(pipelinemode="automatic") hifa_setjy(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_aplycal(pipelinemode="automatic") hif_aplycal(pipelinemode="automatic") hif_makeimist(intent='PHASE,BANDPASS,CHECK') hif_makeimage(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_fatagtargets(pipelinemode="automatic") hif_fatagtargets(pipelinemode="automatic") hif_finakcinteripelinemode="automatic") hif_finakcinteripelinemode="automatic") hif_finakcinteripelinemode="automatic") hif_finakcinteripelinemode="automatic") hif_finakcinteripelinemode="automatic") hif_finakcinteripelinemode="automatic") hif_finakcinteripelinemode="automatic") hif_finakcinteripelinemode="automatic") hif_finakcinteripelinemode="automatic") hif_finakcinteripelinemode="automatic") hif_finakcinteripelinemode="automatic") hif_finakcint(pipelinemode="automatic") hif_finakcint(pipelinemode="automatic") hif_finakcint(pipelinemode="automatic") hif_finakcint(pipelinemode="automatic") hif_ifunctif(pipelinemode="automatic")</pre>	# h init()
<pre># context.project_structure.ousstatus_entity_id = 'uidA001_****_X2e7' # hifa_importdata(dbservice=False, vis=['uidA002_X72c4aa_****']) hifa_flagdata(pipelinemode="automatic") hifa_fluxcalflag(pipelinemode="automatic") hif_refant(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_wrygcalflag(pipelinemode="automatic") hif_owgainflag(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hifa_setjy(pipelinemode="automatic") hifa_setjy(pipelinemode="automatic") hifa_setjy(pipelinemode="automatic") hifa_setjy(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_spupelinemode="automatic") hifa_spupelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_anakeimist(intent='PHASE, BANDPASS, CHECK') hif_makeimist(intent='PHASE, BANDPASS, CHECK') hif_makeimist(intent='PHASE, BANDPASS, CHECK') hif_makeimist(intent='PHASE, BANDPASS, CHECK') hif_makeimist(intent='PHASE, BANDPASS, CHECK') hif_makeimist(specmode="automatic") hifa_flagtarget(pipelinemode="automatic") hifa_flagtarget(pipelinemode="automatic") hifa_flagtarget(pipelinemode="automatic") hifa_timede="automatic") hifa_nakeimise(pipelinemode="automatic") hifa_timede="automatic") hifa_indeont(pipelinemode="au</pre>	# context project summary proposal code = '2016 1 ***** S'
<pre># hifa_importdata(dbservice=False, vis=['uidA002_X72c4aa_****']) hifa_flagdata(pipelinemode="automatic") hifa_fluxcalflag(pipelinemode="automatic") hif_rawflagchas(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_wryccalflag(pipelinemode="automatic") hif_owryccalflag(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_spupcal(pipelinemode="automatic") hifa_spupcal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_aplycal(pipelinemode="automatic") hif_aplycal(pipelinemode="automatic") hif_atimegaincal(pipelinemode="automatic") hif_anakeimist(intent='PHASE, BANDPASS, CHECK') hif_makeimist(intent='PHASE, BANDPASS, CHECK') hif_makeimist(intent='Pinze') hif_interchece=</pre>	# context project structure ousstatus entity id = 'uid A001 **** X2e7'
<pre>hifa_flagdata(pipelinemode="automatic") hifa_fluxcalflag(pipelinemode="automatic") hif_rawflagchans(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_antpos(pipelinemode="automatic") hifa_wvrgcalflag(pipelinemode="automatic") hifa_wvrgcalflag(pipelinemode="automatic") hif_gainflag(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_gfluxscale(pipelinemode="automatic") hifa_gfluxscale(pipelinemode="automatic") hifa_gfluxscale(pipelinemode="automatic") hifa_naplycal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_atimegaincal(pipelinemode="automatic") hif_atimegaincal(pipelinemode="automatic") hif_atimegaincal(pipelinemode="automatic") hif_atimegaincal(pipelinemode="automatic") hif_atimegaincal(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimage(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_findcont(pipelinemode="automatic") hif_imakeimlist(specmode="automatic") hif_imakeimlist(specmode="automatic") hif_imakeimlist(specmode="automatic") hif_imakeimlist(specmode="automatic") hif_imakeimlist(specmode="automatic") hif_inuccontfit(pipelinemode="automatic")</pre>	# bifa importdata(dbservice=Ealse, vis=['uid A002 X72c4aa ****'])
<pre>hifa_fluxcalflag(pipelinemode="automatic") hif_rawflagchans(pipelinemode="automatic") hif_refant(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_antpos(pipelinemode="automatic") hifa_wvrgcalflag(pipelinemode="automatic") hif_lowgainflag(pipelinemode="automatic") hif_gainflag(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hifa_shdpass(pipelinemode="automatic") hifa_shdpass(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_gfluxscale(pipelinemode="automatic") hifa_gfluxscale(pipelinemode="automatic") hifa_fluxscale(pipelinemode="automatic") hifa_fluxscale(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_tocont(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hif_makeimist(specmode="matomatic") hif_functont(pipelinemode="automatic") hif_functont(pipelinemode="automatic") hif_functont(pipelinemode="automatic") hif_functont(pipelinemode="automatic")</pre>	hifa flagdata(pipelinemode="automatic")
<pre>hif_rawflagchans(pipelinemode="automatic") hif_refant(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsysflag(pipelinemode="automatic") hifa_awrgcalflag(pipelinemode="automatic") hif_lowgainflag(pipelinemode="automatic") hif_gainflag(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_fa_pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_flagtargets(pipelinemode="automatic") hif_makeimlist(specmode="automatic") hif_flagtargets(pipelinemode="automatic") hif_makeimlist(specmode="automatic") hif_makeimlist(specmode="automatic") hif_makeimlist(specmode="automatic") hif_findcont(pipelinemode="automatic") hif_findcont(pipelinemode="automatic")</pre>	hifa fluxcalflad(pipelinemode="automatic")
<pre>hif_refant(pipelinemode="automatic") hifa_tsyscal(pipelinemode="automatic") hifa_tsysflag(pipelinemode="automatic") # hifa_antpos(pipelinemode="automatic") hifa_wvrgcalflag(pipelinemode="automatic") hif_lowgainflag(pipelinemode="automatic") hif_gainflag(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_flag(pipelinemode="automatic") hifa_gfluxscale(pipelinemode="automatic") hifa_aplycal(pipelinemode="automatic") hif_aplycal(pipelinemode="automatic") hif_abandpass(pipelinemode="automatic") hif_aplycal(pipelinemode="automatic") hif_atimegaincal(pipelinemode="automatic") hif_atimegaincal(pipelinemode="automatic") hif_atimegais(pipelinemode="automatic") hif_inakeimages(pipelinemode="automatic") hif_inakeimages(pipelinemode="automatic") hif_inakeimages(pipelinemode="automatic") hif_inakeimages(pipelinemode="automatic") hif_inakeimages(pipelinemode="automatic") hif_inakeimages(pipelinemode="automatic") hif_inakeimages(pipelinemode="automatic") hif_inakeimages(pipelinemode="automatic") hif_inakeimalist(specmode="automatic") hif_inakeimalist(specmode="automatic") hif_inakeimalist(specmode="automatic") hif_inakeimalist(specmode="automatic") hif_indcont(pipelinemode="automatic") hif_indcont(pipelinemode="automatic") hif_invcontfit(pipelinemode="automatic")</pre>	hif rawflagchans(pipelinemode="automatic")
<pre>hifa_tsyscal(pipelinemode="automatic") hifa_tsysflag(pipelinemode="automatic") # hifa_antpos(pipelinemode="automatic") hifa_wvrgcalflag(pipelinemode="automatic") hif_lowgainflag(pipelinemode="automatic") hif_gainflag(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_smyhaseup(pipelinemode="automatic") hifa_stimegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hif_imakeimlist(specmode="automatic") hif_imakeimlist(specmode="automatic</pre>	hif refant(pipelinemode="automatic")
<pre>hifa_tsysflag(pipelinemode="automatic") # hifa_antpos(pipelinemode="automatic") hifa_wvrgcalflag(pipelinemode="automatic") hif_lowgainflag(pipelinemode="automatic") hif_gainflag(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_smyhaseup(pipelinemode="automatic") hifa_smyhaseup(pipelinemode="automatic") hifa_gfluxscale(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hif_aflagtargets(pipelinemode="automatic") hif_noductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hif_noductsize(pipelinemode="automatic") hif_noductsize(pipelinemode="automatic") hif_lowgain(pipelinemode="automatic") hif_noductsize(pipelinemode="automatic") hif_note(pipelinemode="automatic") hif_note(p</pre>	hifa tsyscal(pipelinemode="automatic")
<pre># hifa_antpos(pipelinemode="automatic") hifa_wvrgcalflag(pipelinemode="automatic") hif_lowgainflag(pipelinemode="automatic") hif_gainflag(pipelinemode="automatic") hifa_setjy(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_smyhaseup(pipelinemode="automatic") hifa_gfluxscale(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_stransform(pipelinemode="automatic") hif_alagtargets(pipelinemode="automatic") hif_makeimages(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_fimakeimages(pipelinemode="automatic") hif_fimakeimages(pipelinemode="automatic") hif_fimakeimages(pipelinemode="automatic") hif_fingtargets(pipelinemode="automatic") hif_fingtargets(pipelinemode="automatic") hif_fingtargets(pipelinemode="automatic") hif_fingtargets(pipelinemode="automatic") hif_fingtargets(pipelinemode="automatic") hif_not(pipelinemode="automatic") hif_fingtargets(pipelinemode="automatic") hif_not(pipelinemode="automatic") hif_findcont(pipelinemode="automatic") hif_lowstargets(pipelinemode="automatic") hif_lowstargets(pipelinemode="automatic") hif_findcont(pipelinemode="automatic") hif_lowstargets(pipelinemode="automatic") hif_lowstargets(pipelinemode="automatic") hif_lowstargets(pipelinemode="automatic") hif_lowstargets(pipelinemode="automatic") hif_lowstargets(pipelinemode="automatic") hif_lowstargets(pipelinemode="automatic") hif_lowstargets(pipelinemode="automatic") hif_lowstargets(pipeline</pre>	hifa_tsysflag(pipelinemode="automatic")
<pre>hifa_wvrgcalflag(pipelinemode="automatic") hif_lowgainflag(pipelinemode="automatic") hif_gainflag(pipelinemode="automatic") hifa_setjy(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_fluxscale(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_fatestrages(pipelinemode="automatic") hif_makeimages(pipelinemode="automatic") hif_makeimages(pipelinemode="automatic") hif_fatestrages(pipelinemode="automatic") hif_fatestrages(pipelinemode="automatic") hif_fatestragets(pipelinemode="automatic") hif_fatestragets(pipelinemode="automatic") hif_fatestragets(pipelinemode="automatic") hif_findcont(pipelinemode="automatic") hif_findcont(pipelinemode="automatic")</pre>	<pre># hifa_antpos(pipelinemode="automatic")</pre>
<pre>hif_lowgainflag(pipelinemode="automatic") hif_gainflag(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="automatic") hifa_fluxscale(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hif_aflagtargets(pipelinemode="automatic") hif_fimakeimlist(specmode="automatic") hif_fimakeimlist(specmode="automatic") hif_findcont(pipelinemode="automatic") hif_findcont(pipelinemode="automatic")</pre>	hifa_wvrgcalflag(pipelinemode="automatic")
<pre>hif_gainflag(pipelinemode="automatic") hif_setjy(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="interactive",hm_spwmapmode='simple') hifa_gfluxscale(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hif_fmakeimlist(specmode="automatic") hif_findcont(pipelinemode="automatic") hif_findcont(pipelinemode="automatic")</pre>	hif_lowgainflag(pipelinemode="automatic")
<pre>hif_setjy(pipelinemode="automatic") hifa_bandpass(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="interactive",hm_spwmapmode='simple') hifa_gfluxscale(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hif_fmakeimlist(specmode="matomatic") hif_indection(pipelinemode="automatic") hif_netion(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hif_undection(pipelinemode="automatic") hif_indection(pipelinemode="matomatic") hif_undection(pipelinemode="matomatic") hif_undection(pipelinem</pre>	hif_gainflag(pipelinemode="automatic")
<pre>hifa_bandpass(pipelinemode="automatic") hifa_spwphaseup(pipelinemode="interactive",hm_spwmapmode='simple') hifa_gfluxscale(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hif_fmakeimlist(specmode="matomatic") hif_findcont(pipelinemode="automatic") hif_findcont(pipelinemode="automatic") hif_functont(pipelinemode="automatic")</pre>	hif_setjy(pipelinemode="automatic")
<pre>hifa_spwphaseup(pipelinemode="interactive",hm_spwmapmode='simple') hifa_gfluxscale(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hif_fmakeimlist(specmode="matomatic") hif_fmakeimlist(specmode="matomatic") hif_findcont(pipelinemode="automatic") hif_uccontfit(pipelinemode="automatic")</pre>	hifa_bandpass(pipelinemode="automatic")
<pre>hifa_gfluxscale(pipelinemode="automatic") hifa_timegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimlist(intent='PHASE, BANDPASS, CHECK') hif_makeimages(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hif_makeimlist(specmode="matomatic") hif_findcont(pipelinemode="automatic") hif_findcont(pipelinemode="automatic") hif_uvcontfit(pipelinemode="automatic")</pre>	hifa_spwphaseup(pipelinemode="interactive",hm_spwmapmode='simple')
<pre>hita_timegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hif_makeimlist(specmode='matomatic') hif_findcont(pipelinemode="automatic") hif_findcont(pipelinemode="automatic") hif_functiont(pipelinemode="automatic")</pre>	hita_gtluxscale(pipelinemode="automatic")
<pre>hit_apptycat(pipetinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_makeimlist(specmode='mautomatic') hif_findcont(pipelinemode="automatic") hif_findcont(pipelinemode="automatic") hif_uvcontfit(pipelinemode="automatic")</pre>	hita_timegaincal(pipelinemode="automatic")
<pre>hir_makeIntist(intent='PHASE, BANDPASS, CHECK') hif_makeimages(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hif_makeimlist(specmode='matomatic') hif_findcont(pipelinemode="automatic") hif_findcont(pipelinemode="automatic") hif_uvcontfit(pipelinemode="automatic")</pre>	hit_applycal(pipelinemode="automatic")
<pre>hil_makeImages(piperinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hif_makeimlist(specmode='mfs') hif_findcont(pipelinemode="automatic") hif_uvcontfit(pipelinemode="automatic")</pre>	hit_makelmlist(intent='PHASE, BANDPASS, CHECK')
hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hif_makeimlist(specmode='mfs') hif_findcont(pipelinemode="automatic") hif_uvcontfit(pipelinemode="automatic")	hit checkraducteize (navareducteize-200 0, maxeuhecize-20 0)
hif_mstranstrans(pipelinemode="automatic") hif_makeimlist(specmode="mfs') hif_findcont(pipelinemode="automatic") hif_uvcontfit(pipelinemode="automatic")	hil_checkproductsize(maxproductsize=400.0, maxcubesize=30.0)
hif_makeimlist(specinemode='mfs') hif_findcont(pipelinemode="automatic") hif_uvcontfit(pipelinemode="automatic")	hit_mstransform(pipelinemode_wautomatic")
hif_findcont(pipelinemode="automatic") hif_uvcontfit(pipelinemode="automatic")	hit makamist (chermode-infe)
hif_uvcontfit(pipelinemode="automatic")	hif findcont(pipelinemode="automatic")
,	hif uvcontfit(pipelinemode="automatic")
hif uvcontsub(pipelinemode="automatic")	hif uvcontsub(pipelinemode="automatic")
U: imagingScript.py Top (14,67) (Python)	U: imagingScript.py Top (14,67) (Python)
Wrote /Users/abiggs/eris/eris 2017/imagingScript.pv	Wrote /Users/abiggs/eris/eris_2017/imagingScript.py

Example script

File Edit Options Buffers Tools Python Help

File Edit Options Constant, Python Help

File Edit Options Buffers Tools Python Help

File Edit Options Bu

hita_bandpass(pipelinemoinceractive moode - user can override the d hifa_spwphaseup(pipelinemode='nteractive',hm_spwmapmode='simple') hifa_timegaincal(pipelinemode="automatic") hif_applycal(pipelinemode="automatic") hif_makeimlist(intent='PHASE,BANDPASS,CHECK') hif_makeimages(pipelinemode="automatic") hif_checkproductsize(maxproductsize=400.0, maxcubesize=30.0) hif_mstransform(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hifa_flagtargets(pipelinemode="automatic") hif_imstransform(pipelinemode="automatic") hif_findcont(pipelinemode="automatic") hif_findcont(pipelinemode="automatic") hif_findcont(pipelinemode="automatic") hif_stransform(pipelinemode="automatic") hif_stransform(pipelinemode="automatic") hif_uvcontfit(pipelinemode="automatic") - imagingScript.py Top (14,67) (Python) rote /Users/abiggs/eris/eris 2017/imagingScript.py

The Weblog

- Results of the Pipeline are reported in the Weblog
 - View 'index.html' file in your browser
- Shows clickable menu of all tasks that were run
- Clicking on a task will show results of that task
 - Tasks will report warnings, useful messages
 - Many plots are included

			file://	/Users/ehumphre/pip	elineC5_r40783weblog/t2-4m.html?sidebar=sidebar_s	stage1 Č	0	ð ð
Mail	- Elizabeth Hu	imphreys - Out	tlook	127	3635062492792877.png 757×573 pixels	12. h	ifa_bandpassflag	
Home	Ву Торіс	By Task					2016.1	.00496.S
Tasks in execution order 1. hifa_importdata 2. hifa_flagdata 3. hifa_fluxcalflag 4. hif_rawflagchans 5. hif_refant	r 9	12.	Bandpass	s Calibra	tion and Flagging			BACK
6. h_tsyscal 7. hifa_tsysflag	9	Warnir	ng! Evaluation of uidA0	02_Xb9cc97_X3ce7.n	ns raised 1 flagging command(s)			
 8. hifa_antpos 9. hifa_wvrgcalflag 10. hif_lowgainflag 11. hif_setmodels 12. hifa_bandpassflag 13. hifa_spwphaseup 14. hifa_gfluxscaleflag 15. hifa_gfluxscale 16. hifa_timegaincal 17. hif_applycal 	•	This tas statistic generate In furthe solution caltable have be data flag	k performs a preliminary b ally examining the scalar of ed). The philosophy is that er detail, the workflow is an is solved and applied, the . Plots are generated at th en run and applied. If no p gging score (depending or Jging	pandpass solution and difference of the corro t only outlier data poin s follows: an a priori o flagging heuristics a ree points in this worl oints were flagged, th n the fraction of data t	d applies it, then computes the flagging heuristics by c acted amplitude minus model amplitudes, flags those ints that have remained outliers after calibration will be calibration is applied using pre-existing caltables in the re run and any outliers are flagged, a final bandpass so kflow: after a priori calibration, after bandpass calibrat ie "after" plots are not generated or displayed. The sco flagged) and the score for the bandpass solution.	calling hif_correctedampflag whic outliers, then derives a final band flagged. Note that the phase of t e calibration state, a preliminary b olution is solved (if necessary) an ion but before flagging heuristics re for this stage is a simple comb	ch looks for outlier visibility points dpass solution (if any flags were the data is not assessed. eandpass solution and amplitude ad the name "final" is appended to s are run, and after flagging heuris bination (multiplication) of the sta	s by gaincal o this stics andard
18. hifa_imageprecheck 19. hif_makeimlist	0	Measu	rement Set		Flagging Commands		Number of Statements	
20. hif_makeimages 21. hif_checkproductsiz	e	uid Report F	A002_Xb9cc97_X3ce7.ms Files 10ed data su	mmarv	uidA002_Xb9cc97_X3ce7.ms-flag_commands.txt	t	1	

Measurement Set: uid___A002_Xb9cc97_X3ce7.ms

Data Selection	flagged before	flagged after

IVICI	il - Elizabeth Hur	mohrevs - Outlook	1273635062492792877 ppg 757×573 pixels	12 hifa bandnassflag
	Du Tania			
ALMA	ву горіс	Вутаѕк		2016.1.00498
asks in execution orde	er			
hifa_importdata	θ	12 Dandrage O	Alibration and Elegaina	
hifa_flagdata	- 1 1	TZ. Bandpass C	alloration and Flagging	BACK
hifa_fluxcalflag	- 1 1			
hif_rawflagchans	- 1 1			
nif_refant	- 1 1	Task notifications		
h_tsyscal	- 1 1	Warning! Evaluation of uidA002_Xb	b9cc97_X3ce7.ms raised 1 flagging command(s)	
nifa_tsysflag	θ			
nifa_antpos	θ	This task performs a preliminary bandp	bass solution and applies it, then computes the flagging heuristics by calling hif_c	correctedampflag which looks for outlier visibility points by
nifa_wvrgcalflag	θ	statistically examining the scalar different	ence of the corrected amplitude minus model amplitudes, flags those outliers, the	en derives a final bandpass solution (if any flags were
. hif_lowgainflag	- 1 1	generated). The philosophy is that only	outlier data points that have remained outliers after calibration will be flagged. N	lote that the phase of the data is not assessed.
. hif_setmodels		In further detail, the workflow is as follo	we: an a priori calibration is applied using pre-existing caltables in the calibration	
hifa_bandpassflag	•		ws. an a phone calibration is applied using pre-existing calibration	n state, a preliminary bandpass solution and amplitude gainc
		solution is solved and applied, the flagg	ging heuristics are run and any outliers are flagged, a final bandpass solution is so	n state, a preliminary bandpass solution and amplitude gain olved (if necessary) and the name "final" is appended to this
8. hifa_spwphaseup		solution is solved and applied, the flagg caltable. Plots are generated at three po	ging heuristics are run and any outliers are flagged, a final bandpass solution is so oints in this workflow: after a priori calibration, after bandpass calibration but bef	n state, a preliminary bandpass solution and amplitude gain olved (if necessary) and the name "final" is appended to this ore flagging heuristics are run, and after flagging heuristics
. hifa_spwphaseup . hifa_gfluxscaleflag		solution is solved and applied, the flagg caltable. Plots are generated at three po have been run and applied. If no points	ging heuristics are run and any outliers are flagged, a final bandpass solution is so oints in this workflow: after a priori calibration, after bandpass calibration but bef were flagged, the "after" plots are not generated or displayed. The score for this s	n state, a preliminary bandpass solution and amplitude gains olved (if necessary) and the name "final" is appended to this fore flagging heuristics are run, and after flagging heuristics stage is a simple combination (multiplication) of the standa
3. hifa_spwphaseup 4. hifa_gfluxscaleflag 5. hifa_gfluxscale		solution is solved and applied, the flagg caltable. Plots are generated at three po have been run and applied. If no points data flagging score (depending on the f	ging heuristics are run and any outliers are flagged, a final bandpass solution is so oints in this workflow: after a priori calibration, after bandpass calibration but bef were flagged, the "after" plots are not generated or displayed. The score for this s fraction of data flagged) and the score for the bandpass solution.	n state, a preliminary bandpass solution and amplitude gains olved (if necessary) and the name "final" is appended to this ore flagging heuristics are run, and after flagging heuristics stage is a simple combination (multiplication) of the standar
3. hifa_spwphaseup 4. hifa_gfluxscaleflag 5. hifa_gfluxscale 6. hifa_timegaincal	o	solution is solved and applied, the flagg caltable. Plots are generated at three po have been run and applied. If no points data flagging score (depending on the f	ging heuristics are run and any outliers are flagged, a final bandpass solution is so oints in this workflow: after a priori calibration, after bandpass calibration but bef were flagged, the "after" plots are not generated or displayed. The score for this s fraction of data flagged) and the score for the bandpass solution.	I state, a preliminary bandpass solution and amplitude gains olved (if necessary) and the name "final" is appended to this ore flagging heuristics are run, and after flagging heuristics stage is a simple combination (multiplication) of the standar
. hifa_spwphaseup . hifa_gfluxscaleflag . hifa_gfluxscale . hifa_timegaincal . hif_applycal	0	solution is solved and applied, the flagg caltable. Plots are generated at three po have been run and applied. If no points data flagging score (depending on the f Flagging	ging heuristics are run and any outliers are flagged, a final bandpass solution is so oints in this workflow: after a priori calibration, after bandpass calibration but bef were flagged, the "after" plots are not generated or displayed. The score for this s fraction of data flagged) and the score for the bandpass solution.	n state, a preliminary bandpass solution and amplitude gaind olved (if necessary) and the name "final" is appended to this fore flagging heuristics are run, and after flagging heuristics stage is a simple combination (multiplication) of the standar
hifa_spwphaseup hifa_gfluxscaleflag hifa_gfluxscale hifa_timegaincal hif_applycal hifa_imageprecheck	Co k D	solution is solved and applied, the flagg caltable. Plots are generated at three po have been run and applied. If no points data flagging score (depending on the f Flagging Measurement Set	ging heuristics are run and any outliers are flagged, a final bandpass solution is so oints in this workflow: after a priori calibration, after bandpass calibration but bef were flagged, the "after" plots are not generated or displayed. The score for this s fraction of data flagged) and the score for the bandpass solution.	n state, a preliminary bandpass solution and amplitude gains olved (if necessary) and the name "final" is appended to this fore flagging heuristics are run, and after flagging heuristics stage is a simple combination (multiplication) of the standar
 bifa_spwphaseup bifa_gfluxscaleflag bifa_gfluxscale bifa_timegaincal bifa_applycal bifa_imageprecheck bifa_makeimlist 	© k 0	solution is solved and applied, the flagg caltable. Plots are generated at three po have been run and applied. If no points data flagging score (depending on the f Flagging Measurement Set	wish a prior calibration is applied using pre-existing databases in the calibration ging heuristics are run and any outliers are flagged, a final bandpass solution is so oints in this workflow: after a priori calibration, after bandpass calibration but bef were flagged, the "after" plots are not generated or displayed. The score for this s fraction of data flagged) and the score for the bandpass solution. Flagging Commands uid uid A002 Xb9cc97 X3ce7 ms-flag commands txt	state, a preliminary bandpass solution and amplitude gains olved (if necessary) and the name "final" is appended to this fore flagging heuristics are run, and after flagging heuristics stage is a simple combination (multiplication) of the standar Number of Statements

Continuum Imaging



Moment-8 maps



Continuum-subtracted

<u>cubes</u> Moment-8 maps show brightest channel at each spatial position



Recent improvements

- For Cycle 5 there have been a number of improvements
 - Attempt to reduce the runtime if data products are too large
 - i.e. any cube > 30 GB or total cubes > 400 GB
 - Will channel average, reduce image size, number of sources imaged, etc.
 - Automatic flagging of calibrated calibrator data
 - Reduces the amount of manual flagging required
 - Note: Calibrated target data is not yet flagged
 - Auto-masking (auto-boxing) during cleaning
 - Better imaging and allows a deeper clean
 - Plus many others!

Auto-flagging of calibrators (1)







Auto-flagging of calibrators (2)



Auto-masking

Will allow deeper cleaning



Cycle 5 pipeline with automasking

Cycle 4 pipeline with primary beam mask

Auto-masking



Cycle 4 pipeline with primary beam mask Cycle 5 pipeline with automasking

Summary

- The CASA pipeline is used to calibrate ALMA and JVLA data
 - Including ALMA Total Power array (single dish)
- Fraction of pipelined ALMA data continually increasing
 - Expect about 70% of projects to be pipeline-imaged in Cycle 5
- Cycle-5 Pipeline will be released in an update of CASA 5.1
- Parallelization of tclean^{*} will appear in CASA 5.2

*Refactor of clean – will become the standard version of 'clean'



RadioNet has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730562