



EUROPEAN ARC
ALMA Regional Centre

I-TRAIN with the European ARC Network

#1. Imaging with the ALMA Science Pipeline

Daniel Tafoya - Nordic ARC

L. T. Maud - ESO

Emily Moravec - Czech ARC

MC Toribio- Nordic ARC

CHALMERS



Astronomical
Institute
of the Czech Academy
of Sciences

This tutorial assumes that you followed the instructions indicated in the following 'INSTRUCTIONS' document:



I-TRAIN with the European ARC Network

#1. Imaging with the ALMA Science Pipeline

D Tafoya - Nordic ARC
L T Maud - ESO
E Moravec - Czech ARC
MC Toribio- Nordic ARC



CHALMERS



You can get the document from [here](https://almascience.org/tools/eu-arc-network/i-train):
<https://almascience.org/tools/eu-arc-network/i-train>

Overview:

- The data and the working directory
- Starting CASA with the pipeline
- List of steps to be run
- Hands-on tutorial
- Final remarks

These steps were already done:



Data download from the ALMA Science Archive (ASA)
(including raw data and calibration tables)



scriptForPI.py has been run
`execfile('scriptForPI.py')`




calibrated



products



rawdata



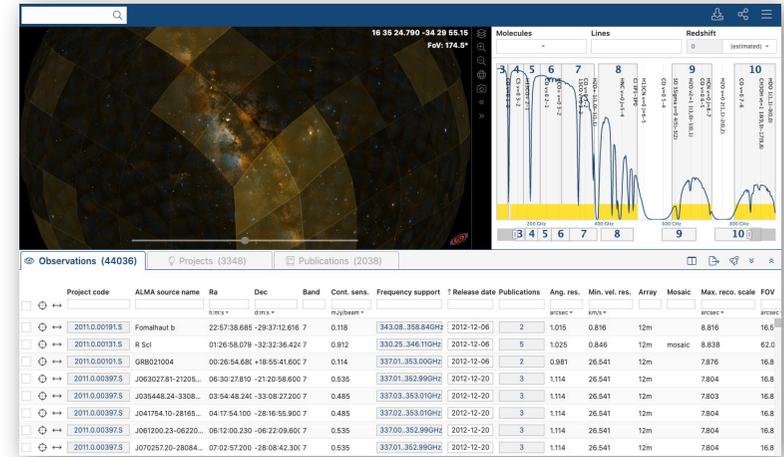
uid__A002_Xd2
b681_Xa1c2.ms



uid__A002_Xd2
b681_Xb99d.ms



working



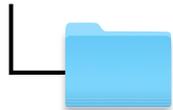
```
C++ shared library loaded successfully
QAC: Root /Users/dtafoya/.casa/QAC
QAC: Load src/qac.py
QAC: Load src/ssc.py
QAC: Load src/plot.py
QAC: Skip distribute/tp2vis.py
QAC: Skip tp2vis/tp2vis.py
QAC: qac: version 19-sep-2019
qac_root: /Users/dtafoya/.casa/QAC
casa: 5.6.1-8
data:/Applications/CASA.app/Contents/data
None
QAC: Root /Users/dtafoya/.casa/QAC
QAC: Load src/qac.py
QAC: Load src/ssc.py
QAC: Load src/plot.py
QAC: Load contrib/tp2vis.py
QAC: Skip distribute/tp2vis.py
QAC: Skip tp2vis/tp2vis.py
QAC: Load contrib/sdint_helper.py
QAC: Load contrib/sdint_imager.py
QAC: qac: version 19-sep-2019
qac_root: /Users/dtafoya/.casa/QAC
casa: 5.6.1-8
data:/Applications/CASA.app/Contents/data
None
this is working!
--> CrashReporter initialized.
Enter help for help getting started with CASA...
Using matplotlib backend: TkAgg
[CASA ~1]: execfile('scriptForPI.py')
```



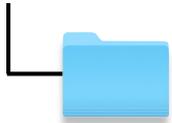
2018.1.01201.S.PLtutorial.tar.gz



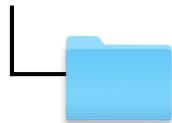
2018.1.01201.S



science_goal.uid___A001_X133d_X2c85



group.uid___A001_X133d_X2c86



member.uid___A001_X133d_X2c8b



calibrated



calibration



log



product



qa



script



member.uid___A001_X133d_X2c8b.README.txt



products



rawdata



uid___A002_Xd2b681_Xa1c2.ms



uid___A002_Xd2b681_Xb99d.ms



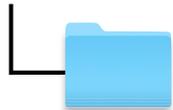
working



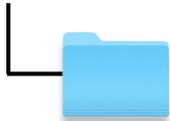
2018.1.01201.S.PLtutorial.tar.gz



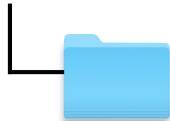
2018.1.01201.S



science_goal.uid__A001_X133d_X2c85



group.uid__A001_X133d_X2c86



member.uid__A001_X133d_X2c8b



calibrated



calibration



log



product



qa



script



member.uid__A001_X133d_X2c8b.README.txt



products



rawdata



uid__A002_Xd2b681_Xa1c2.ms



uid__A002_Xd2b681_Xb99d.ms



working

This is the directory tree that you will have after running scriptForPI.py when downloading data from the ASA.





2018.1.01201.S.PLtutorial.tar.gz



2018.1.01201.S



science_goal.uid__A001_X133d_X2c85



group.uid__A001_X133d_X2c86



member.uid__A001_X133d_X2c8b



calibrated



calibration



log



product



qa



script



products



rawdata



uid__A002_Xd2b681_Xa1c2.ms



uid__A002_Xd2b681_Xb99d.ms

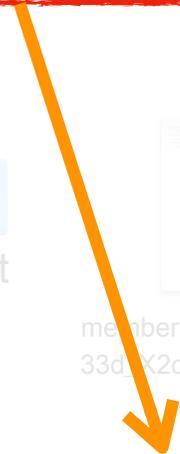


member.uid__A001_X133d_X2c8b.README.txt

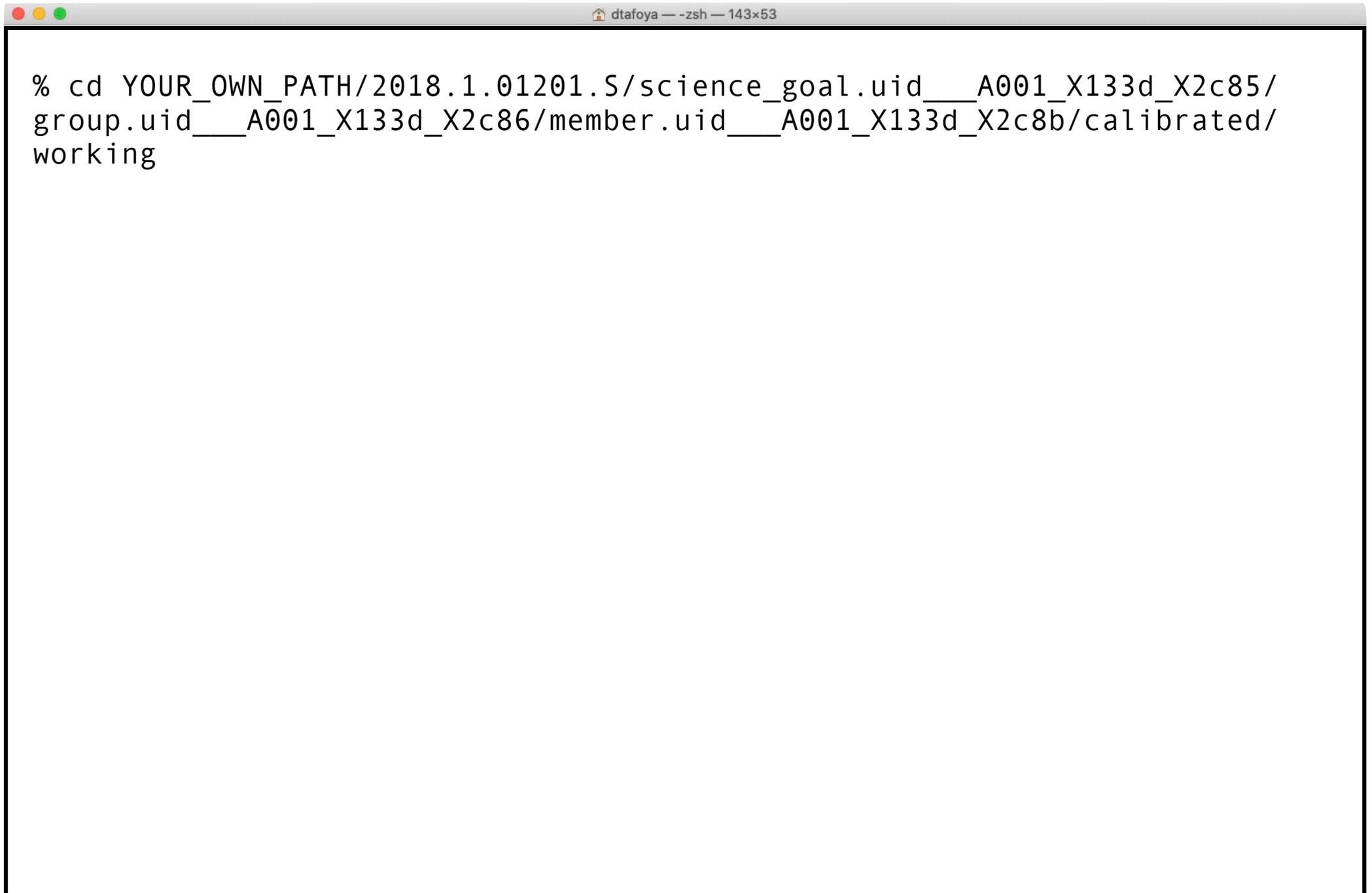


working

We will work in this folder



In a terminal, move to the *working* directory:

A terminal window with a title bar that reads "dtafoya — -zsh — 143x53". The terminal content shows a command to change the current directory to a specific path. The command is: % cd YOUR_OWN_PATH/2018.1.01201.S/science_goal.uid___A001_X133d_X2c85/group.uid___A001_X133d_X2c86/member.uid___A001_X133d_X2c8b/calibrated/working. The terminal is otherwise empty.

```
% cd YOUR_OWN_PATH/2018.1.01201.S/science_goal.uid___A001_X133d_X2c85/  
group.uid___A001_X133d_X2c86/member.uid___A001_X133d_X2c8b/calibrated/  
working
```

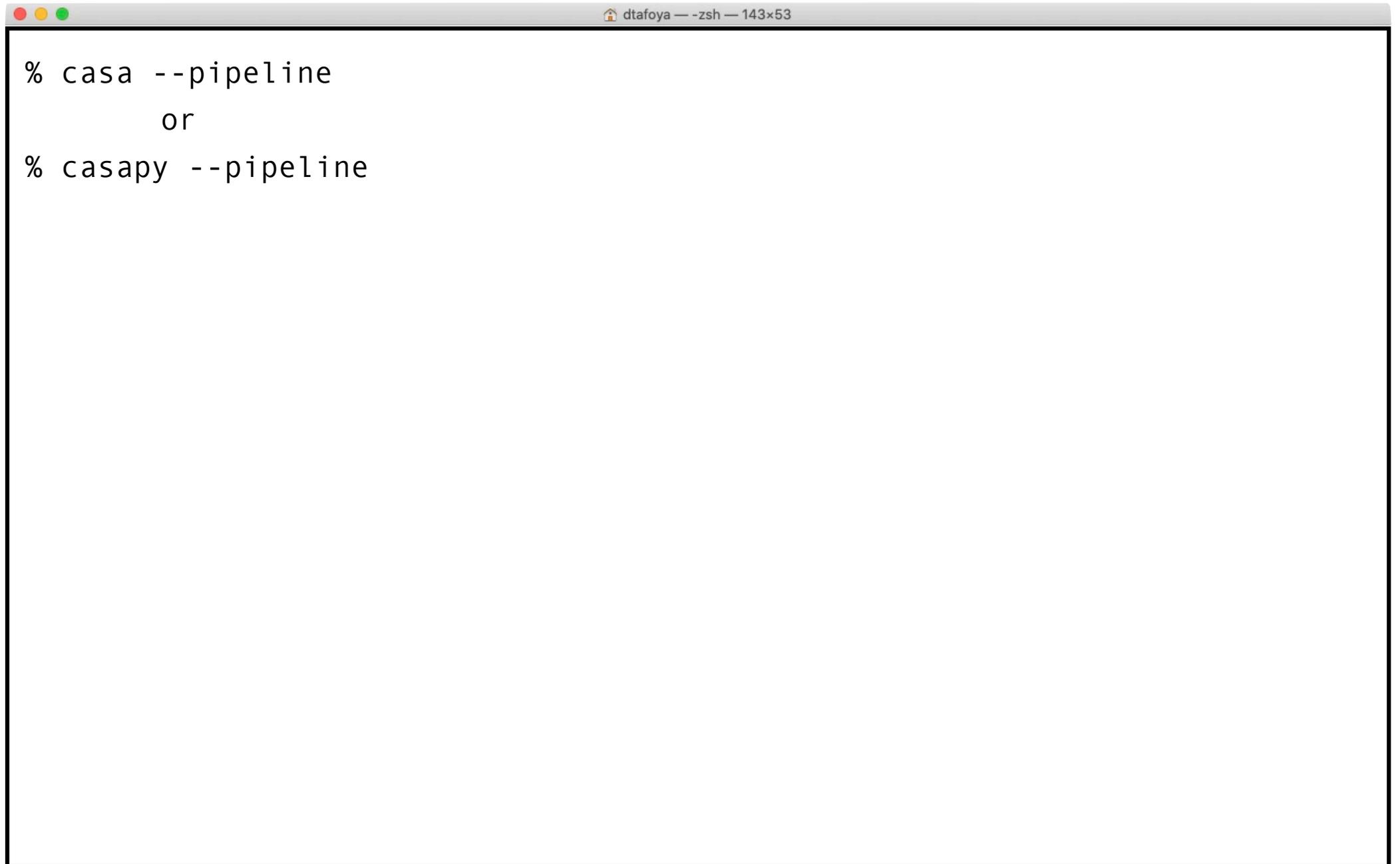
Then, check the content of the *working* directory:

```
dtafoya — -zsh — 143x53

% cd YOUR_OWN_PATH/2018.1.01201.S/science_goal.uid___A001_X133d_X2c85/
group.uid___A001_X133d_X2c86/member.uid___A001_X133d_X2c8b/calibrated/
working

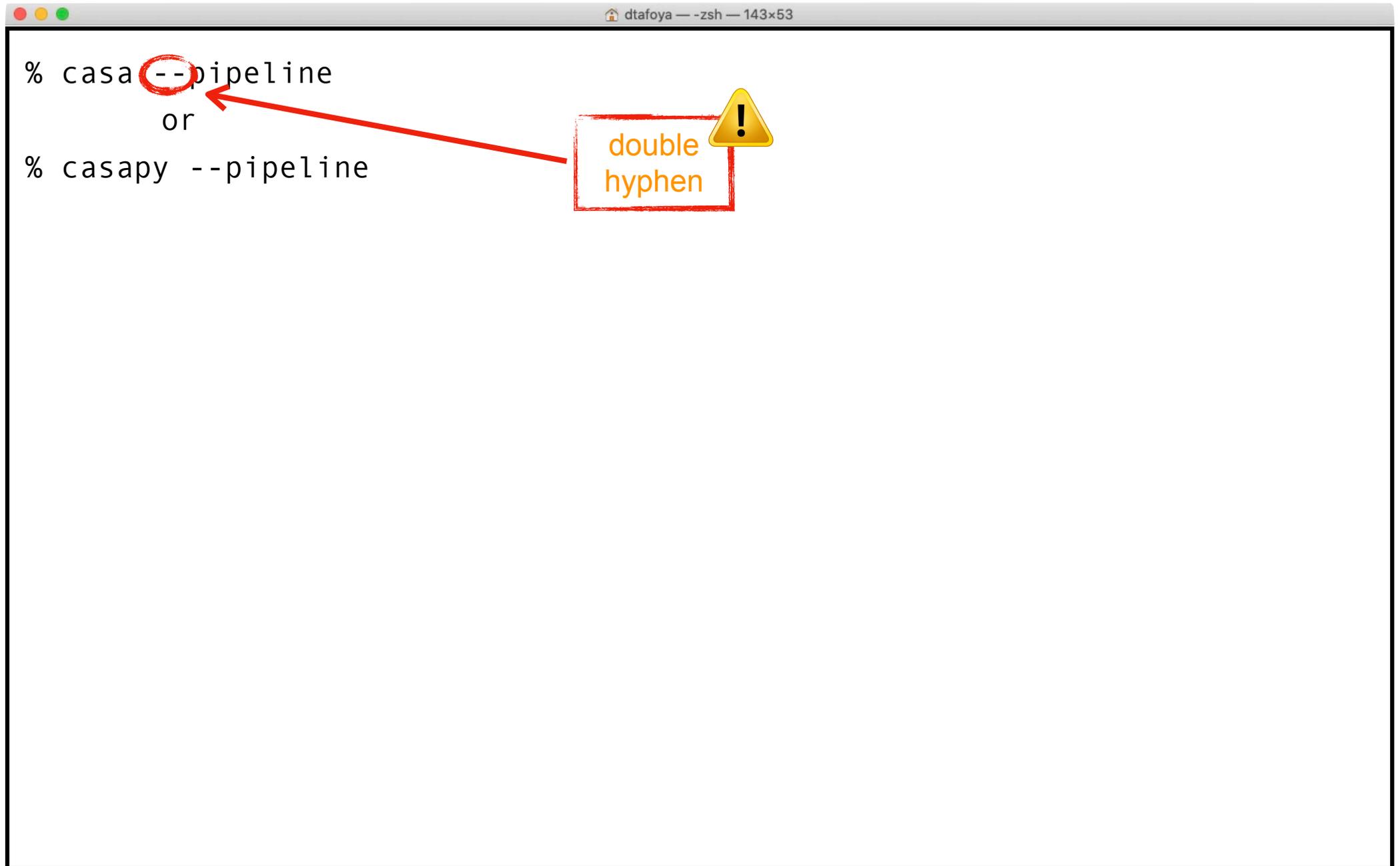
working% ls
flagmanager.last
flux.csv
h_init.last
h_save.last
hifa_restoredata.last
importasdm.last
listobs.last
pipeline-20201105T143636
pipeline-20201105T143636.context
uid___A002_Xd2b681_Xa1c2.ms
uid___A002_Xd2b681_Xa1c2.ms.flagversions
uid___A002_Xd2b681_Xa1c2.ms.h_tsyscal.s6_1.tsyscal.tbl
uid___A002_Xd2b681_Xa1c2.ms.hifa_bandpassflag.s12_7.spw16_18_20_22_24_26.channel.solintinf.bcal.final.tbl
uid___A002_Xd2b681_Xa1c2.ms.hifa_spwphaseup.s13_3.spw16_18_20_22_24_26.solintinf.gpcal.tbl
uid___A002_Xd2b681_Xa1c2.ms.hifa_timegaincal.s16_3.spw16_18_20_22_24_26.solintinf.gpcal.tbl
uid___A002_Xd2b681_Xa1c2.ms.hifa_timegaincal.s16_4.spw16_18_20_22_24_26.solintint.gpcal.tbl
uid___A002_Xd2b681_Xa1c2.ms.hifa_timegaincal.s16_6.spw16_18_20_22_24_26.solintinf.gacal.tbl
uid___A002_Xd2b681_Xb99d.ms
uid___A002_Xd2b681_Xb99d.ms.flagversions
uid___A002_Xd2b681_Xb99d.ms.h_tsyscal.s6_2.tsyscal.tbl
uid___A002_Xd2b681_Xb99d.ms.hifa_bandpassflag.s12_8.spw16_18_20_22_24_26.channel.solintinf.bcal.final.tbl
uid___A002_Xd2b681_Xb99d.ms.hifa_spwphaseup.s13_4.spw16_18_20_22_24_26.solintinf.gpcal.tbl
uid___A002_Xd2b681_Xb99d.ms.hifa_timegaincal.s16_4.spw16_18_20_22_24_26.solintinf.gpcal.tbl
uid___A002_Xd2b681_Xb99d.ms.hifa_timegaincal.s16_5.spw16_18_20_22_24_26.solintint.gpcal.tbl
uid___A002_Xd2b681_Xb99d.ms.hifa_timegaincal.s16_7.spw16_18_20_22_24_26.solintinf.gacal.tbl
```

In a terminal, run CASA with the pipeline:

A terminal window with a title bar containing a home icon, the text 'dtafoya', a separator '—', 'zsh', another separator '—', and '143x53'. The terminal content shows three lines of text: '% casa --pipeline', 'or', and '% casapy --pipeline'.

```
dtafoya — zsh — 143x53  
% casa --pipeline  
or  
% casapy --pipeline
```

In a terminal, run CASA with the pipeline:



The image shows a terminal window with a title bar that reads "dtafoya --zsh-- 143x53". Inside the terminal, the following text is displayed:

```
% casa --pipeline  
or  
% casapy --pipeline
```

A red circle highlights the "--" in the first command. A red arrow points from a yellow warning triangle (containing an exclamation mark) to this circle. The warning triangle is positioned above a red-bordered box containing the text "double hyphen".

In a terminal, run CASA with the pipeline:

```
% casa --pipeline
```

```
or
```

```
% casapy --pipeline
```

```
IPython 5.4.0 -- An enhanced Interactive Python.
```

```
PIPELINE CASA 5.6.1-8 -- Common Astronomy Software Applications
```

```
Found an existing telemetry logfile: /Users/dtafoya/.casa/casastats-561-8-21c3d94ba1ca45a13d-20201116-201536-PIPELINE.log
```

```
Telemetry initialized. Telemetry will send anonymized usage statistics to NRAO.
```

```
You can disable telemetry by adding the following line to your ~/.casarc file:
```

```
EnableTelemetry: False
```

```
2020-11-17 21:51:55 INFO: Environment is not MPI enabled. Pipeline operating in single host mode
```

```
2020-11-17 21:51:57 INFO: Environment variable FLUX_SERVICE_URL not defined. Switching to backup url.
```

```
2020-11-17 21:51:57 INFO: Environment variable FLUX_SERVICE_URL_BACKUP not defined.
```

```
2020-11-17 21:51:58 INFO: Pipeline version 42866 (Pipeline-CASA56-P1-B) running on Daniel-no-MacBook-Pro.local
```

```
2020-11-17 21:51:58 INFO: Host environment: 16.0 GB memory, 4 x Intel(R) Core(TM) i5-7360U CPU @ 2.30GHz running MacOS 10.15.7
```

```
2020-11-17 21:51:58 INFO: Initializing cli...
```

```
2020-11-17 21:51:58 INFO: Loaded CASA tasks from package: h
```

```
2020-11-17 21:51:58 INFO: Loaded CASA tasks from package: hif
```

```
2020-11-17 21:51:58 INFO: Loaded CASA tasks from package: hifa
```

```
2020-11-17 21:51:58 INFO: Loaded CASA tasks from package: hifv
```

```
2020-11-17 21:51:58 INFO: Loaded CASA tasks from package: hsd
```

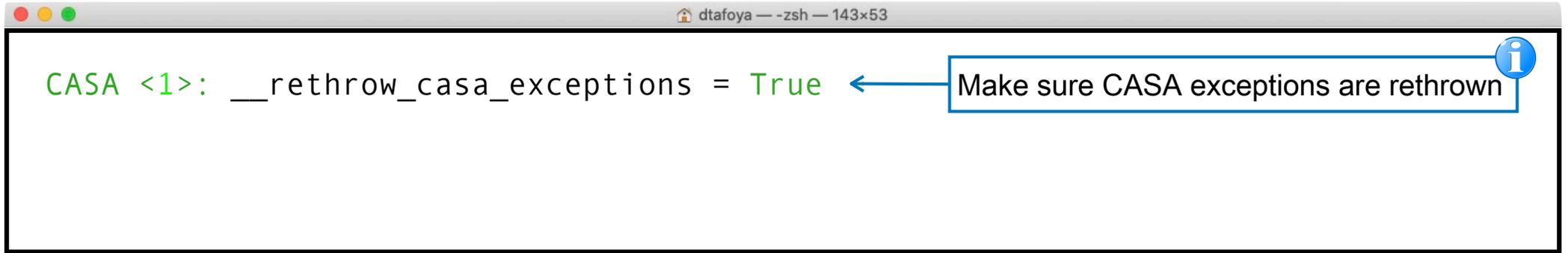
```
2020-11-17 21:51:58 INFO: Loaded CASA tasks from package: hsdn
```



indicates the pipeline tasks have been loaded

```
CASA <1>:
```

Create the pipeline context:



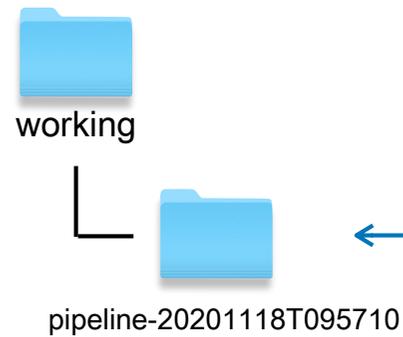
A terminal window with a title bar that reads "dtafoya --zsh-- 143x53". The terminal content shows the command "CASA <1>: __rethrow_casa_exceptions = True". A blue callout box with a white arrow points from the text "Make sure CASA exceptions are rethrown" to the "True" value in the command. A small blue information icon is located in the top right corner of the terminal window.

```
CASA <1>: __rethrow_casa_exceptions = True
```

Make sure CASA exceptions are rethrown

Create the pipeline context:

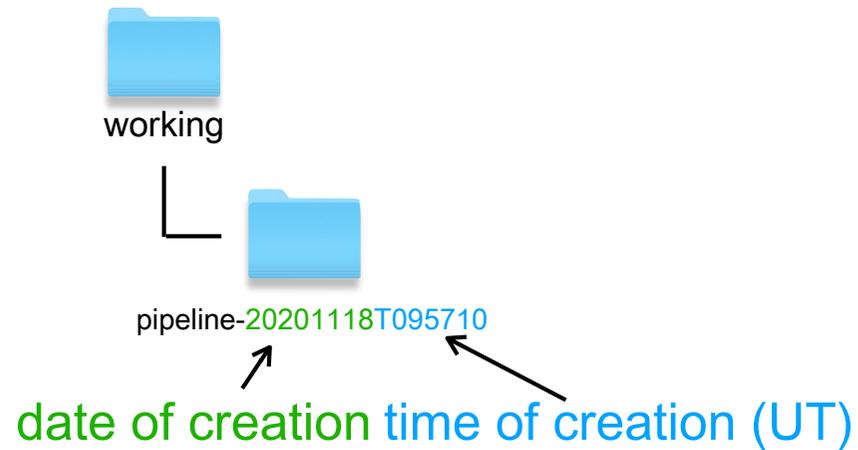
```
dtafoya — zsh — 143x53  
CASA <1>: __rethrow_casa_exceptions = True  
CASA <2>: h_init() ← Initiates the context  
Out[2]: <Context(name='pipeline-20201118T095710')>
```



a pipeline folder is created when `h_init` is run

Create the pipeline context:

```
dtafoya --zsh-- 143x53  
CASA <1>: __rethrow_casa_exceptions = True  
CASA <2>: h_init()  
Out[2]: <Context(name='pipeline-20201118T095710')>
```



Define global variable for the measurement sets:

```
dtafoya — -zsh — 143x53  
  
CASA <3>:import glob  
CASA <4>:MyVis=glob.glob('*ms') ← define a global variable that includes all the calibrated ms  
CASA <5>: MyVis  
Out[4]: ['uid___A002_Xd2b681_Xa1c2.ms', 'uid___A002_Xd2b681_Xb99d.ms']
```

Steps of this tutorial:

```
1 hifa_importdata(vis=MyVis,dbservice=False)
2 hif_mstransform(pipeline="automatic")
3 hifa_flagtargets(pipeline="automatic")
4 hifa_imageprecheck(pipeline="automatic")
5 hif_makeimlist(specmode='mfs')
6 hif_findcont(pipeline="automatic")
7 hif_uvcontfit(pipeline="automatic")
8 hif_uvcontsub(pipeline="automatic")
9 hif_makeimages(pipeline="automatic")
10 hif_makeimlist(specmode='cont')
11 hif_makeimages(pipeline="automatic")
12 hif_makeimlist(specmode='cube')
13 hif_makeimages(pipeline="automatic")
14 hifa_exportdata(imaging_products_only=True)
```

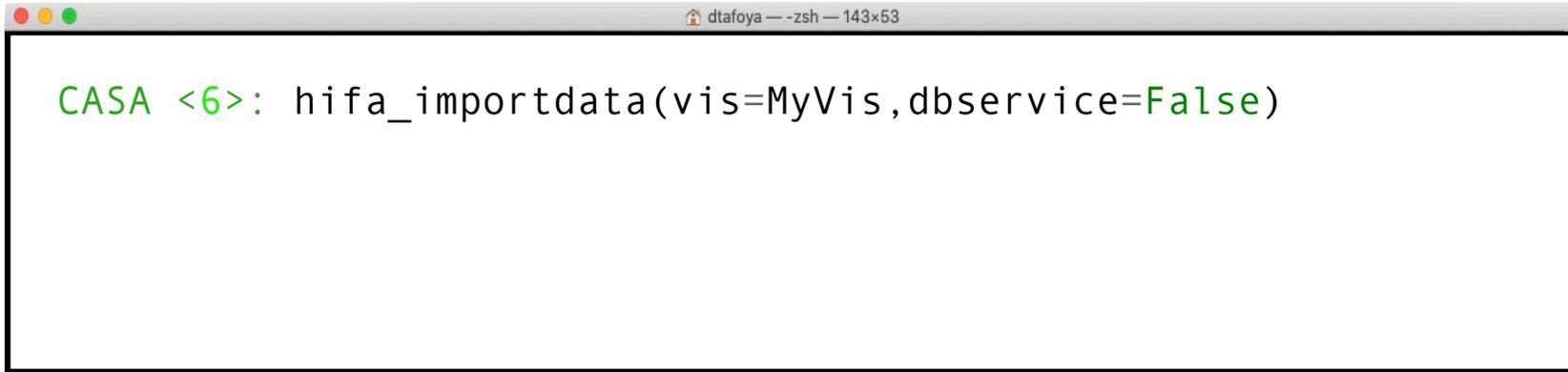
Steps of this tutorial:

| | |
|--|------------------------------------|
| <pre>1 hifa_importdata(vis=MyVis,dbservice=False) 2 hif_mstransform(pipeline="automatic") 3 hifa_flagtargets(pipeline="automatic") 4 hifa_imageprecheck(pipeline="automatic") 5 hif_makeimlist(specmode='mfs')</pre> | import and prepare the data |
| <pre>6 hif_findcont(pipeline="automatic") 7 hif_uvcontfit(pipeline="automatic") 8 hif_uvcontsub(pipeline="automatic")</pre> | find and subtract the continuum |
| <pre>9 hif_makeimages(pipeline="automatic") 10 hif_makeimlist(specmode='cont') 11 hif_makeimages(pipeline="automatic") 12 hif_makeimlist(specmode='cube') 13 hif_makeimages(pipeline="automatic")</pre> | make images and cubes |
| <pre>14 hifa_exportdata(imaging_products_only=True)</pre> | exports to fits files |

Steps of this tutorial:

```
1 hifa_importdata(vis=MyVis,dbservice=False)
2 hif_mstransform(pipelinemode="automatic")
3 hifa_flagtargets(pipelinemode="automatic")
4 hifa_imageprecheck(pipelinemode="automatic")
5 hif_makeimlist(specmode='mfs')
6 hif_findcont(pipelinemode="automatic")
7 hif_uvcontfit(pipelinemode="automatic")
8 hif_uvcontsub(pipelinemode="automatic")
9 hif_makeimages(pipelinemode="automatic")
10 hif_makeimlist(specmode='cont')
11 hif_makeimages(pipelinemode="automatic")
12 hif_makeimlist(specmode='cube')
13 hif_makeimages(pipelinemode="automatic")
14 hifa_exportdata(imaging_products_only=True)
```

Getting started: Import the data



A terminal window with a title bar that reads "dtafoya --zsh -- 143x53". The terminal content shows the command: `CASA <6>: hifa_importdata(vis=MyVis,dbservice=False)`. The prompt and the function name are in green, while the arguments are in black.

- 1 hifa_importdata
- 2 hif_mstransform
- 3 hifa_flagtargets
- 4 hifa_imageprecheck
- 5 hif_makeimlist
- 6 hif_findcont
- 7 hif_uvcontfit
- 8 hif_uvcontsub
- 9 hif_makeimages
- 10 hif_makeimlist
- 11 hif_makeimages
- 12 hif_makeimlist
- 13 hif_makeimages
- 14 hifa_exportdata

Getting started: Import the data

```
CASA <6>: hifa_importdata(vis=MyVis,dbservice=False)
```

do not use the online flux catalog on import

- 1 hifa_importdata
- 2 hif_mstransform
- 3 hifa_flagtargets
- 4 hifa_imageprecheck
- 5 hif_makeimlist
- 6 hif_findcont
- 7 hif_uvcontfit
- 8 hif_uvcontsub
- 9 hif_makeimages
- 10 hif_makeimlist
- 11 hif_makeimages
- 12 hif_makeimlist
- 13 hif_makeimages
- 14 hifa_exportdata

Getting started: Import the data

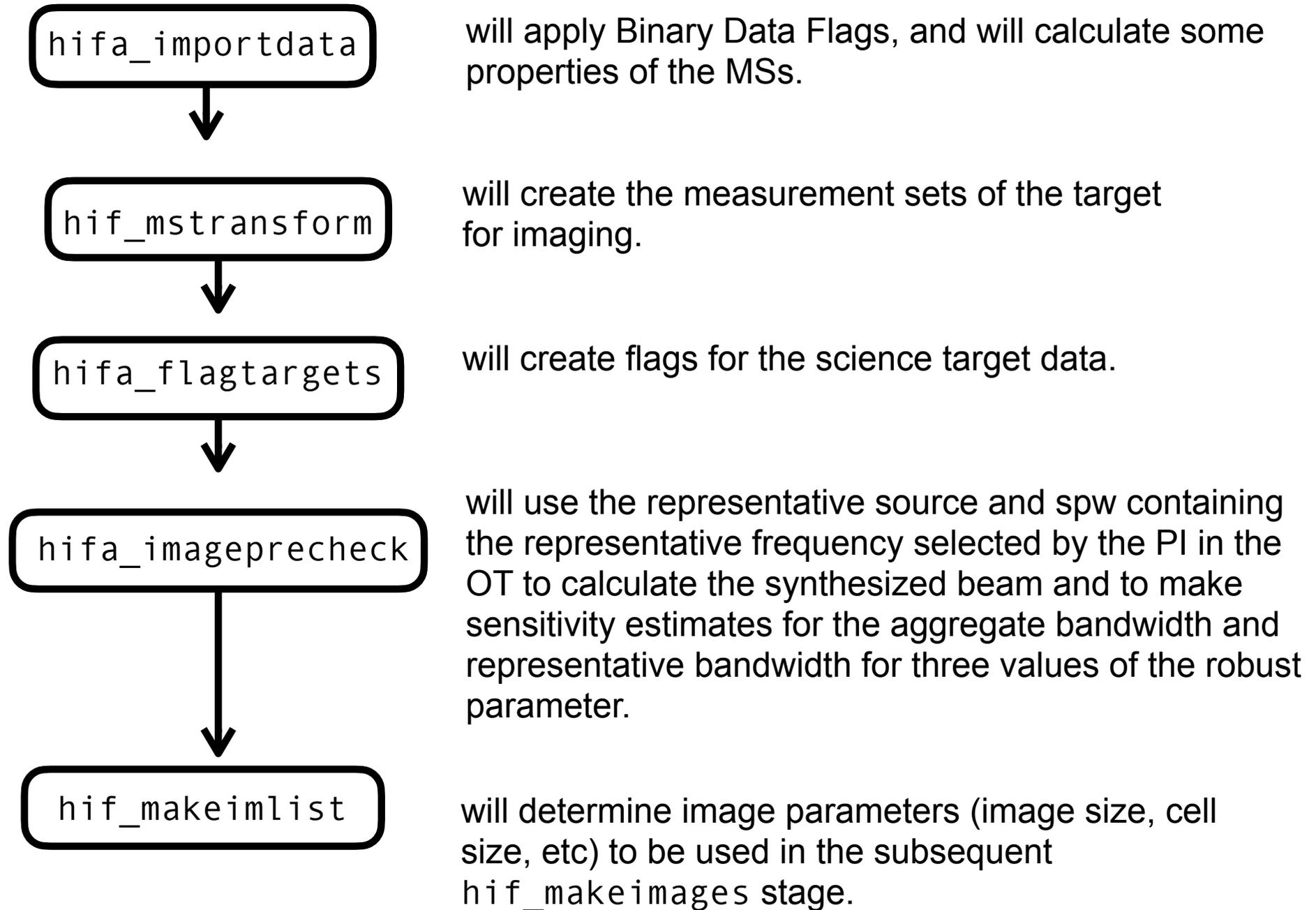
```
CASA <6>: hifa_importdata(vis=MyVis,dbservice=False)
```

do not use the online flux catalog on import

- Several messages will appear in the terminal and logger.
- It may take around 10 min to finish.

```
1 hifa_importdata
2 hif_mstransform
3 hifa_flagtargets
4 hifa_imageprecheck
5 hif_makeimlist
6 hif_findcont
7 hif_uvcontfit
8 hif_uvcontsub
9 hif_makeimages
10 hif_makeimlist
11 hif_makeimages
12 hif_makeimlist
13 hif_makeimages
14 hifa_exportdata
```

Steps before continuum subtraction:



While waiting for `hifa_importdata` to finish, a few comments on the pipeline tasks:

-What does `hifa` and `hif` stand for?

- `hif`: interferometry (ALMA & VLA)
- `hifa`: interferometry (ALMA-only)
- `hifv`: interferometry (VLA-only)
- `hsd`: single-dish

A complete list of tasks used by the ALMA (and VLA) pipeline may be found by typing `tasklist` at the CASA prompt and looking under the "User defined tasks" for all of the `hifv_*`, `hifa_*`, `hsd_*`, `hif_*`, and `h_*` tasks



The lists of tasks can also be put in a python script:

```
example.py

import glob
import os

__rethrow_casa_exceptions = True
context = h_init()

os.system('rm -rf uid*_target.ms')
os.system('rm -rf uid*_target.ms.flagversions')

MyVis=glob.glob('*ms')

try:
    hifa_importdata(vis=MyVis,dbservice=False)
    hif_mstransform(pipelinemode="automatic")
    hifa_flagtargets(pipelinemode="automatic")
    hifa_imageprecheck(pipelinemode="automatic")
    hif_makeimlist(specmode='mfs')
    hif_findcont(pipelinemode="automatic")
    hif_uvcontfit(pipelinemode="automatic")
    hif_uvcontsub(pipelinemode="automatic")
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cont')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cube')
    hif_makeimages(pipelinemode="automatic")
    hifa_exportdata(imaging_products_only=True)
finally:
    h_save()

U:***- example.py All L5 (Python EIDoc)
```

The pipeline script is executed in the same way as other python scripts:

```
CASA <7>: execfile('my_pipeline_script.py')
```

```
import glob
import os

__rethrow_casa_exceptions = True
context = h_init()

os.system('rm -rf uid*_target.ms')
os.system('rm -rf uid*_target.ms.flagversions')

MyVis=glob.glob('*ms')

try:
    hifa_importdata(vis=MyVis,dbservice=False)
    hif_mstransform(pipelinemode="automatic")
    hifa_flagtargets(pipelinemode="automatic")
    hifa_imageprecheck(pipelinemode="automatic")
    hif_makeimlist(specmode='mfs')
    hif_findcont(pipelinemode="automatic")
    hif_uvcontfit(pipelinemode="automatic")
    hif_uvcontsub(pipelinemode="automatic")
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cont')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cube')
    hif_makeimages(pipelinemode="automatic")
    hifa_exportdata(imaging_products_only=True)
finally:
    h_save()
```

Handling the pipeline **context**:

Handling the pipeline **context**:



Handling the context:

- The pipeline always runs with a context:
`h_init()`
- To resume at any specific stage, save context before closing CASA:
`h_save('my_context.context')`
- To resume from a specific task:
`h_resume('my_context.context')`

Handling the pipeline **context**:

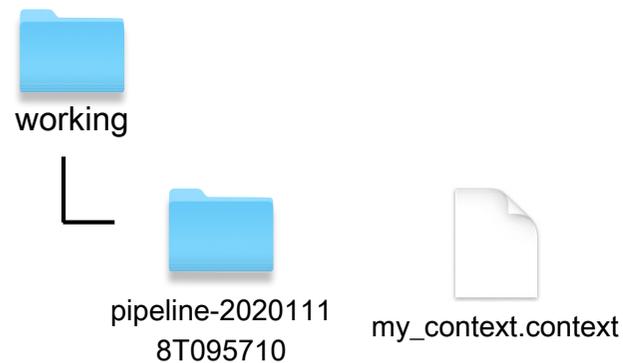


Handling the context:

- The pipeline always runs with a context:
`h_init()`
- To resume at any specific stage, save context before closing CASA:
`h_save('my_context.context')`
- To resume from a specific task:
`h_resume('my_context.context')`

A screenshot of a terminal window with a grey title bar. The title bar contains a home icon, the text 'dtafoya', a double-dash icon, '-zsh', and '143x53'. The terminal area has a black border and shows the command 'CASA <7>: h_save('my_context.context')' in green text.

```
CASA <7>: h_save('my_context.context')
```



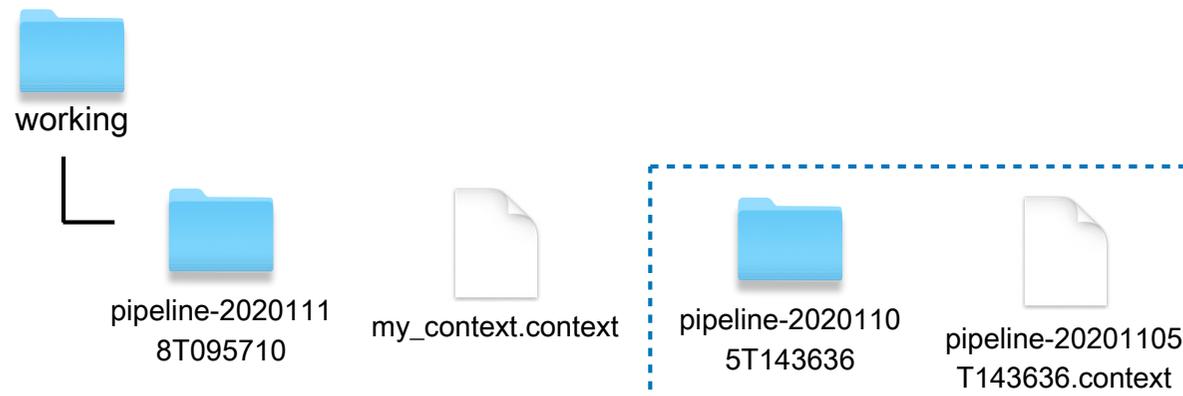
Handling the pipeline context:



Handling the context:

- The pipeline always runs with a context:
`h_init()`
- To resume at any specific stage, save context before closing CASA:
`h_save('my_context.context')`
- To resume from a specific task:
`h_resume('my_context.context')`

```
dtafoya --zsh -- 143x53  
CASA <7>: h_save('my_context.context')
```



a pipeline folder and a context file already exist



Handling the pipeline context:



Handling the context:

- The pipeline always runs with a context:
`h_init()`
- To resume at any specific stage, save context before closing CASA:
`h_save('my_context.context')`
- To resume from a specific task:
`h_resume('my_context.context')`

```
dtafoya --zsh -- 143x53  
CASA <7>: h_save('my_context.context')
```



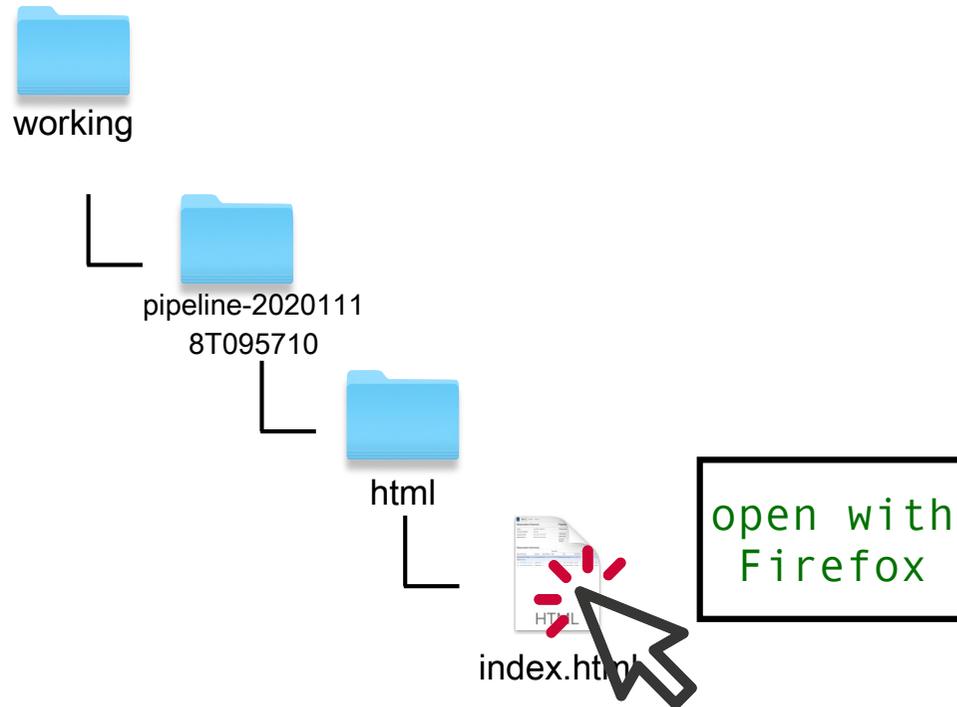
The pipeline weblog:

The pipeline weblog:

-Where is my weblog?



- A html weblog will be created in the working folder:



In Firefox:

Home

file:///Users/dtafoya/Documents/Work/Nordic_ARC_node/Pipeline_tutorial/2018.1.0120 ... 検索

ALMA Home By Topic By Task Project Code N/A

Observation Overview

| | |
|------------------------|-------------------------|
| Project | uid://A001/X12ed/X103 |
| Principal Investigator | janehuang |
| Observation Start | 2018-10-02 17:33:17 UTC |
| Observation End | 2018-10-02 21:08:47 UTC |

Pipeline Summary

| | |
|--------------------|--|
| Pipeline Version | 42866 (Pipeline-CASA56-P1-B) (documentation) |
| CASA Version | 5.6.1-8 (environment) |
| Pipeline Start | 2020-11-18 08:10:43 UTC |
| Execution Duration | 0:03:55 |

Observation Summary

| Measurement Set | Receivers | Num Antennas | Time (UTC) | | | Baseline Length | | | Size |
|--|-------------|--------------|---------------------|---------------------|-----------|-----------------|--------|--------|--------|
| | | | Start | End | On Source | Min | Max | RMS | |
| Observing Unit Set Status: unknown Scheduling Block ID: uid://A001/X133d/X2c83 Scheduling Block Name: RU_Lup_a_06_7M | | | | | | | | | |
| Session: session_1 | | | | | | | | | |
| uid__A002_Xd2b681_Xa1c2.ms | ALMA Band 6 | 11 | 2018-10-02 17:33:17 | 2018-10-02 18:40:03 | 0:39:51 | 8.9 m | 48.0 m | 25.8 m | 5.8 GB |
| uid__A002_Xd2b681_Xb99d.ms | ALMA Band 6 | 12 | 2018-10-02 20:25:54 | 2018-10-02 21:08:46 | 0:16:09 | 8.9 m | 48.9 m | 26.7 m | 3.9 GB |

In Firefox:

The screenshot shows a Firefox browser window with the following elements:

- Address bar: `file:///Users/dtafoya/Documents/Work/Nordic_ARC_node/Pipeline_tutorial/2018.1.0120...`
- Navigation tabs: Home, By Topic, **By Task** (highlighted with a mouse cursor), Project Code N/A
- Section: **Observation Overview**
- Table 1: Observation Overview
- Section: **Pipeline Summary**
- Table 2: Pipeline Summary
- Section: **Observation Summary**
- Table 3: Observation Summary (Table with columns: Measurement Set, Receivers, Num Antennas, Time (UTC), Baseline Length, Size)

| Project | uid://A001/X12ed/X103 |
|-------------------|-------------------------|
| | Principal Investigator |
| Observation Start | 2018-10-02 17:33:17 UTC |
| Observation End | 2018-10-02 21:08:47 UTC |

| Pipeline Version | 42866 (Pipeline-CASA56-P1-B) (documentation) |
|--------------------|--|
| CASA Version | 5.6.1-8 (environment) |
| Pipeline Start | 2020-11-18 08:10:43 UTC |
| Execution Duration | 0:03:55 |

| Measurement Set | Receivers | Num Antennas | Time (UTC) | | | Baseline Length | | | Size |
|---|-------------|--------------|---------------------|---------------------|-----------|-----------------|--------|--------|--------|
| | | | Start | End | On Source | Min | Max | RMS | |
| Observing Unit Set Status: unknown Scheduling Block ID: uid://A001/X133d/X2c83 Scheduling Block Name: RU_Lup_a_06_7M | | | | | | | | | |
| Session: session_1 | | | | | | | | | |
| uid__A002_Xd2b681_Xa1c2.ms | ALMA Band 6 | 11 | 2018-10-02 17:33:17 | 2018-10-02 18:40:03 | 0:39:51 | 8.9 m | 48.0 m | 25.8 m | 5.8 GB |
| uid__A002_Xd2b681_Xb99d.ms | ALMA Band 6 | 12 | 2018-10-02 20:25:54 | 2018-10-02 21:08:46 | 0:16:09 | 8.9 m | 48.9 m | 26.7 m | 3.9 GB |

In Firefox:

Task Summaries

Project Code N/A

| Task | QA Score | Duration |
|--|---|----------|
| 1. hifa_importdata : Register measurement sets with the pipeline | 1/1 have HISTORY  0.50 | 0:10:04 |

CASA logs and scripts

- [View, view in new tab](#) or [download](#) casa-20201118-102913.log (257.1 KB)
- [View, view in new tab](#) or [download](#) casa_commands.log (1.0 KB)
- [View, view in new tab](#) or [download](#) casa_pipescript.py (414 bytes)
- [View, view in new tab](#) or [download](#) (608 bytes)

back to CASA...

Next step is running `mstransform`:

```
dtfoya — -zsh — 143x53  
CASA <8>: hif_mstransform(pipeline="automatic")
```

- It may take around 1.5 min to finish.



```
1 hifa_importdata  
2 hif_mstransform  
3 hifa_flagtargets  
4 hifa_imageprecheck  
5 hif_makeimlist  
6 hif_findcont  
7 hif_uvcontfit  
8 hif_uvcontsub  
9 hif_makeimages  
10 hif_makeimlist  
11 hif_makeimages  
12 hif_makeimlist  
13 hif_makeimages  
14 hifa_exportdata
```

Next step is running `mstransform`:

```
dtafoya — -zsh — 143x53  
CASA <8>: hif_mstransform(pipelinemode="automatic")
```

```
1 hifa_importdata  
2 hif_mstransform  
3 hifa_flagtargets  
4 hifa_imageprecheck  
5 hif_makeimlist  
6 hif_findcont  
7 hif_uvcontfit  
8 hif_uvcontsub  
9 hif_makeimages  
10 hif_makeimlist  
11 hif_makeimages  
12 hif_makeimlist  
13 hif_makeimages  
14 hifa_exportdata
```



- It may take around 1.5 min to finish.
- `mstransform` will create target measurement sets for imaging.



working



uid__A002_Xd2b681
_Xa1c2_target.ms



uid__A002_Xd2b681
_Xb99d_target.ms

Flagging data of the targets:

```
dtafoya — -zsh — 143x53  
CASA <9>: hifa_flagtargets(pipeline="automatic")
```

- It may take around 1.5 min to finish.
- Usually no flags for the target unless problematic data - you can re-run this stage and input values in the flag templates if you want.

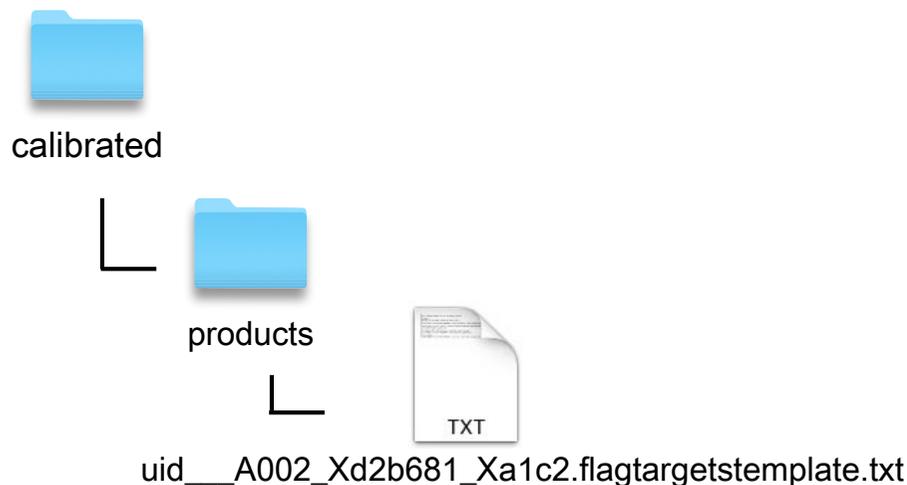


```
1 hifa_importdata  
2 hif_mstransform  
3 hifa_flagtargets  
4 hifa_imageprecheck  
5 hif_makeimlist  
6 hif_findcont  
7 hif_uvcontfit  
8 hif_uvcontsub  
9 hif_makeimages  
10 hif_makeimlist  
11 hif_makeimages  
12 hif_makeimlist  
13 hif_makeimages  
14 hifa_exportdata
```

Flagging data of the targets:

```
CASA <9>: hifa_flagtargets(pipelineMode="automatic")
```

- It may take around 1.5 min to finish.
- Usually no flags for the target unless problematic data - you can re-run this stage and input values in the flag templates if you want.
- Flagging commands can be added to the *uid*flagtargetstemplate.txt* files that are provided with the archived pipeline products. There should be one file for every MS that needs additional flagging, with a name matching the MS uid.
- If these files are found in the directory where the pipeline is run (*working*), they will be picked up by the `hifa_flagtargets` task and applied to the data before science target imaging.



```
1 hifa_importdata
2 hif_mstransform
3 hifa_flagtargets
4 hifa_imageprecheck
5 hif_makeimlist
6 hif_findcont
7 hif_uvcontfit
8 hif_uvcontsub
9 hif_makeimages
10 hif_makeimlist
11 hif_makeimages
12 hif_makeimlist
13 hif_makeimages
14 hifa_exportdata
```

Pre-check the imaging parameters:

```
CASA <10>: hifa_imageprecheck(pipeline="automatic")
```

- It may take around 7 min to finish.
- Useful if you want pipeline to figure out the best image parameters for you in order to match the resolution of the original science goal.
- if you don't run this, the default robust is 0.5.



```
1 hifa_importdata
2 hif_mstransform
3 hifa_flagtargets
4 hifa_imageprecheck
5 hif_makeimlist
6 hif_findcont
7 hif_uvcontfit
8 hif_uvcontsub
9 hif_makeimages
10 hif_makeimlist
11 hif_makeimages
12 hif_makeimlist
13 hif_makeimages
14 hifa_exportdata
```

Pre-check the imaging parameters:

```
dtafoya — -zsh — 143x53  
CASA <10>: hifa_imageprecheck(pipelinemode="automatic")
```

```
1 hifa_importdata  
2 hif_mstransform  
3 hifa_flagtargets  
4 hifa_imageprecheck  
5 hif_makeimlist  
6 hif_findcont  
7 hif_uvcontfit  
8 hif_uvcontsub  
9 hif_makeimages  
10 hif_makeimlist  
11 hif_makeimages  
12 hif_makeimlist  
13 hif_makeimages  
14 hifa_exportdata
```

- It may take around 7 min to finish.
- Useful if you want pipeline to figure out the best image parameters for you in order to match the resolution of the original science goal.
- if you don't run this, the default robust is 0.5.



Remember

- keep checking your weblog
- refresh and see the steps appear
- run `h_save('my_context.context')` regularly



In the weblog:

Task Summaries

Project Code N/A

| Task | QA Score | Duration |
|---|---|----------|
| 1. hifa_importdata: Register measurement sets with the pipeline | 1/1 have HISTORY  0.50 | 0:21:59 |
| 2. hif_mstransform: Create science target MS |  1.00 | 1:24:08 |
| 3. hifa_flagtargets: ALMA Target flagging |  1.00 | 0:09:36 |
| 4. hifa_imageprecheck: ImagePreCheck |  1.00 | 0:06:47 |

CASA logs and scripts

- [View, view in new tab](#) or [download](#) casa-20201118-102913.log (670.0 KB)
- [View, view in new tab](#) or [download](#) casa_commands.log (2.3 KB)
- [View, view in new tab](#) or [download](#) casa_pipescript.py (572 bytes)
- [View, view in new tab](#) or [download](#) (736 bytes)

In the weblog:

The screenshot shows a web browser window with the following elements:

- Browser Address Bar:** file:///Users/dtafoya/Documents/Work/Nordic_ARC_node/Pipeline_tutorial/2018.1.01201.S/science_goal ...
- Navigation:** Home, By Topic, By Task (selected), Project Code N/A
- Left Sidebar:** Tasks in execution order: 1. hifa_importdata, 2. hif_mstransform, 3. hifa_flagtargets, 4. hifa_imageprecheck (highlighted)
- Main Content:**
 - ## 4. Image Pre-Check
 - Goals From OT:**
 - Representative Target: RU_Lup
 - Representative Frequency: 230.5345 GHz (SPW 16)
 - Bandwidth for Sensitivity: 0.1922 MHz (rounded to nearest integer #channels (3), repBW = 0.1831 MHz)
 - Min / Max Acceptable Resolution: Not available
 - Maximum expected beam axial ratio (from OT): Not available
 - Goal PI sensitivity: 32.9 mJy
 - Single Continuum: False
 - Estimated Synthesized Beam and Sensitivities for the Representative Target/Frequency**

Estimates are given for four possible values of the tclean robust weighting parameter: robust = 0.0, +0.5 (default), +1.0, and +2.0. **If the "Min / Max Acceptable Resolution" is available (>=Cycle 5 12-m Array data),** the robust value closest to the default (+0.5) that predicts a beam area (defined as simply major x minor) that is in the range of the PI requested beam areas according to the table row for repBW (Bandwidth for Sensitivity) is chosen. If none of these robust values predict a beam area that is in range, robust=+2.0 is chosen if the predicted beam area is too small, and robust=0.0 is chosen if the predicted beam area is too large. The chosen robust value is highlighted in green and used for all science target imaging. In addition to an estimate for the repBW, an estimate for the aggregate continuum bandwidth (aggBW) is also given assuming NO line contamination but accounting for spw frequency overlap. If the Bandwidth for Sensitivity (repBW) is > the bandwidth of the spw containing the representative frequency (repSPW), then the beam is predicted using all spws, otherwise the beam is predicted for the repSPW alone. A message appears on the "By Task" view if a non-default value of robust (i.e., not +0.5) is chosen. Additionally, if the predicted beam is not within the PI requested range using one of the four robust values, Warning messages appear on this page.

These estimates should always be considered as the BEST CASE SCENARIO. These estimates account for Tsys, the observed uv-coverage, and prior flagging. The estimates DO NOT account for (1) subsequent science target flagging; (2) loss of continuum bandwidth due to the hif_findcont process (i.e. removal of lines and other spectral features from the data used to image the continuum); (3) Issues that affect the image quality like (a) poor match of uv-coverage to image complexity; (b) dynamic range effects; (c) calibration deficiencies (poor phase transfer, residual baseline based effects, residual antenna position errors, etc.). *It is also important to note that both the repBW and aggBW beam calculations are intrinsically multi-frequency synthesis continuum calculations, using the relevant spws as described above. The synthesized beam for a single channel in a cube will typically be larger and can be significantly larger depending on the details of uv-coverage and channel width.*
 - | robust | uvtaper | Synthesized Beam | Cell | Beam Ratio | Bandwidth | BW Mode | Effective Sensitivity |
|--------|---------|--------------------------------|--------------------|------------|------------|---------|-----------------------|
| 0.0 | [] | 6.62 x 4.19 arcsec @ -86.0 deg | 0.84 x 0.84 arcsec | 1.58 | 0.1831 MHz | repBW | 0.028 Jy/beam |
| 0.0 | [] | 6.54 x 4.20 arcsec @ -88.2 deg | 0.84 x 0.84 arcsec | 1.58 | 2375 MHz | aggBW | 0.000233 Jy/beam |
| 0.5 | [] | 6.85 x 4.28 arcsec @ -86.6 deg | 0.86 x 0.86 arcsec | 1.60 | 0.1831 MHz | repBW | 0.0237 Jy/beam |
| 0.5 | [] | 6.76 x 4.30 arcsec @ -87.9 deg | 0.86 x 0.86 arcsec | 1.60 | 2375 MHz | aggBW | 0.000189 Jy/beam |
| 1.0 | [] | 7.10 x 4.48 arcsec @ -86.6 deg | 0.9 x 0.9 arcsec | 1.58 | 0.1831 MHz | repBW | 0.0222 Jy/beam |
| 1.0 | [] | 7.04 x 4.51 arcsec @ -86.6 deg | 0.9 x 0.9 arcsec | 1.58 | 2375 MHz | aggBW | 0.000178 Jy/beam |

open a Firefox browser
and do a demonstration

Set the imaging parameters:

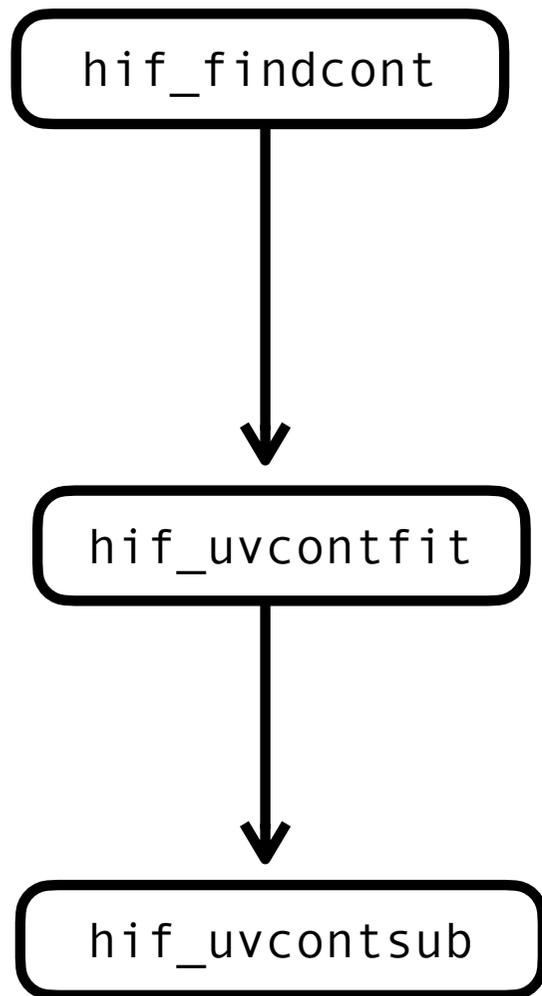
```
dtafoya --zsh -- 143x53  
CASA <11>: hif_makeimlist(specmode='mfs')
```

- this will create all image parameters; specmode = 'mfs' will do aggregated continuum for each SPW.
- the parameters defined here will be used when making images in step 9.



```
1 hifa_importdata  
2 hif_mstransform  
3 hifa_flagtargets  
4 hifa_imageprecheck  
5 hif_makeimlist  
6 hif_findcont  
7 hif_uvcontfit  
8 hif_uvcontsub  
9 hif_makeimages  
10 hif_makeimlist  
11 hif_makeimages  
12 hif_makeimlist  
13 hif_makeimages  
14 hifa_exportdata
```

Finding and subtracting the continuum



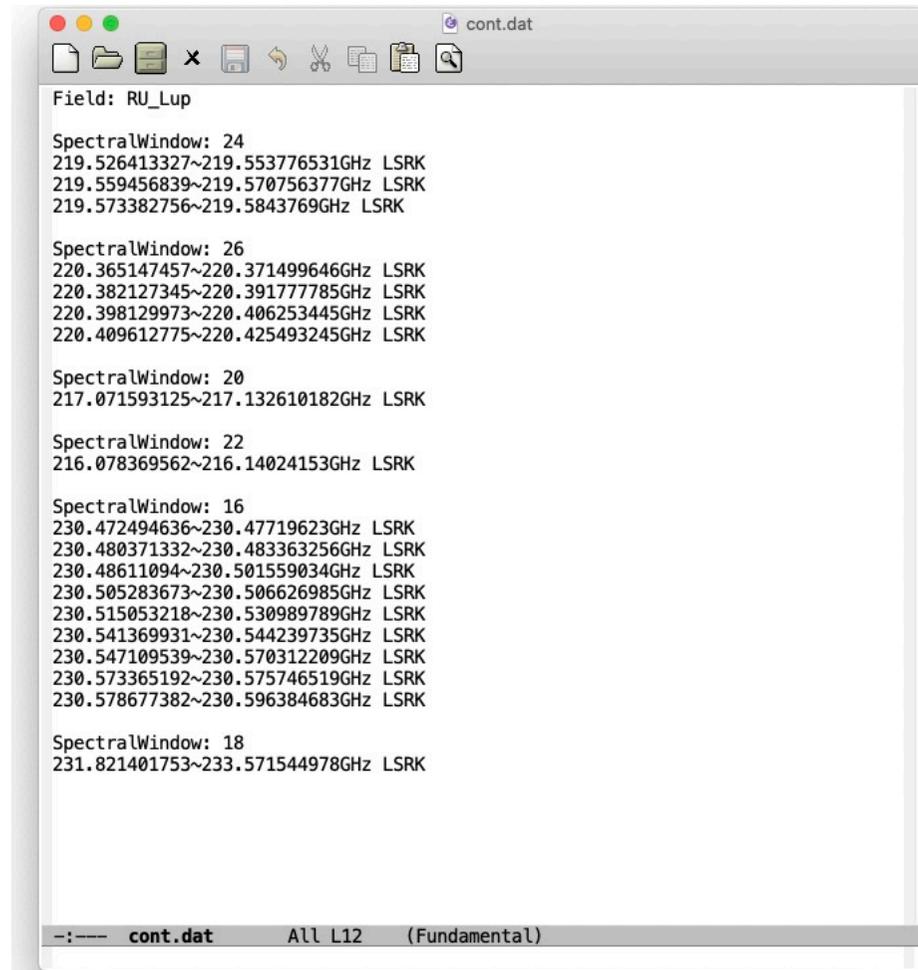
will make dirty image cubes for each spectral window of each science target. It will generate and evaluate the mean spectrum of a masked region of the dirty line+cont image constructed from *moment0* and *moment8* (peak) images. It will calculate frequency ranges that are the least likely to contain any line emission or absorption.

This task will use these frequency ranges to fit and subtract the continuum. Subsequent continuum images will include only these frequency ranges, and the line cubes will be made from the continuum subtracted data.

The `hif_uvcontfit` calibration table is applied to the data. After this step, the original continuum + line emission is contained in the **DATA** column of the MS, while the continuum subtracted data are written to the **CORRECTED** column.

The *cont.dat* file:

- A file named *cont.dat* is delivered with the pipeline products.
- It lists the LSRK frequency ranges that are used to make the per-spw and aggregate continuum images, and for fitting and subtracting the continuum for the image cubes.



```
Field: RU_Lup

SpectralWindow: 24
219.526413327~219.553776531GHz LSRK
219.559456839~219.570756377GHz LSRK
219.573382756~219.5843769GHz LSRK

SpectralWindow: 26
220.365147457~220.371499646GHz LSRK
220.382127345~220.391777785GHz LSRK
220.398129973~220.406253445GHz LSRK
220.409612775~220.425493245GHz LSRK

SpectralWindow: 20
217.071593125~217.132610182GHz LSRK

SpectralWindow: 22
216.078369562~216.14024153GHz LSRK

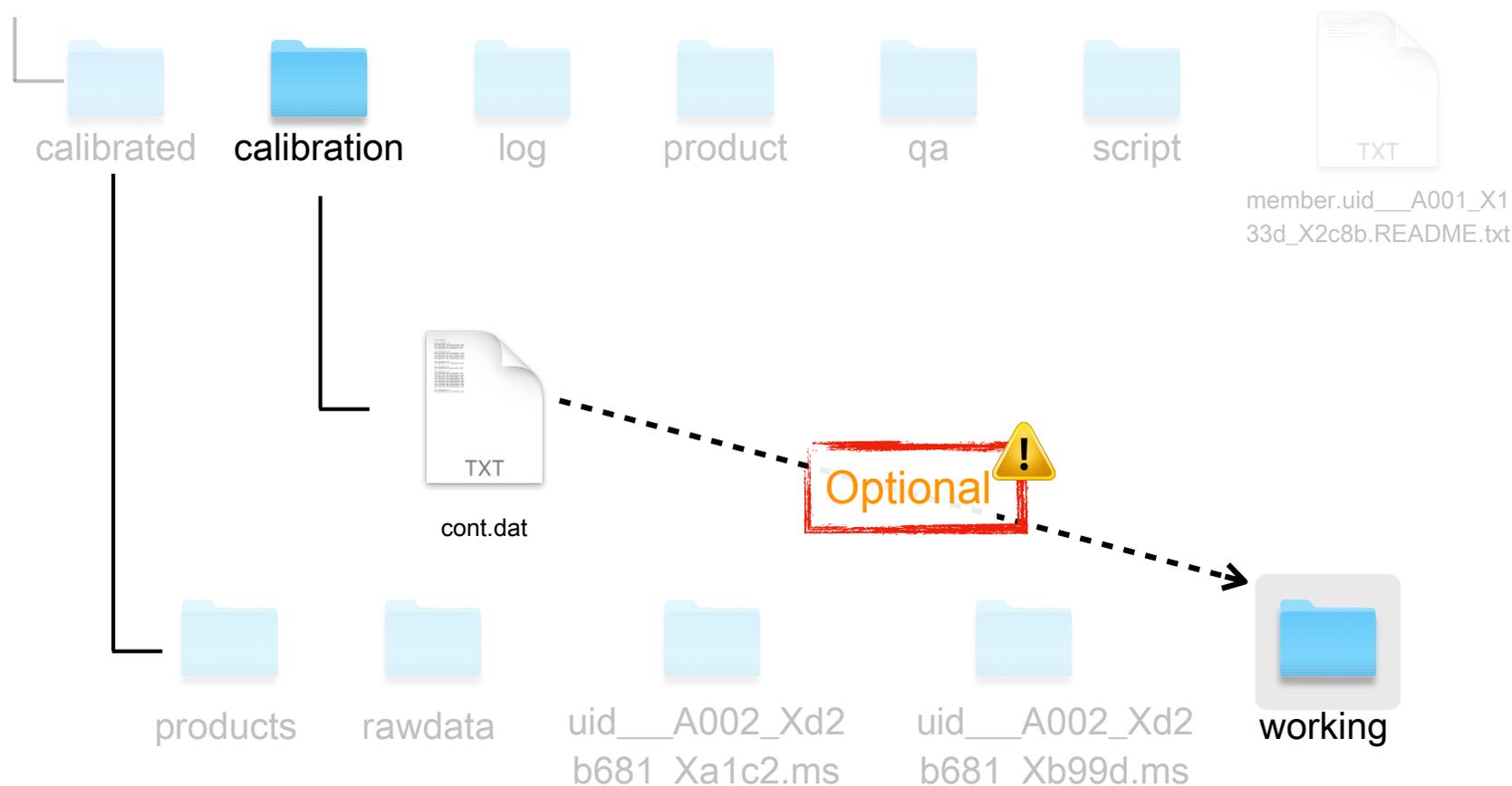
SpectralWindow: 16
230.472494636~230.47719623GHz LSRK
230.480371332~230.483363256GHz LSRK
230.48611094~230.501559034GHz LSRK
230.505283673~230.506626985GHz LSRK
230.515053218~230.530989789GHz LSRK
230.541369931~230.544239735GHz LSRK
230.547109539~230.570312209GHz LSRK
230.573365192~230.575746519GHz LSRK
230.578677382~230.596384683GHz LSRK

SpectralWindow: 18
231.821401753~233.571544978GHz LSRK
```

--:--- cont.dat All L12 (Fundamental)

The *cont.dat* file:

- A file named *cont.dat* is delivered with the pipeline products.
- It lists the LSRK frequency ranges that are used to make the per-spw and aggregate continuum images, and for fitting and subtracting the continuum for the image cubes.



Find the continuum channels:

```
dtafoya — -zsh — 143x53  
CASA <12>: hif_findcont(pipeline="automatic")
```

- this will take around 7 min for the ACA data - as it makes some cubes and will show a spectrum in your weblog.
- if the pre-existing *cont.dat* file is used, the information in the weblog will be shown, but no spectrum will be shown.



```
1 hifa_importdata  
2 hif_mstransform  
3 hifa_flagtargets  
4 hifa_imageprecheck  
5 hif_makeimlist  
6 hif_findcont  
7 hif_uvcontfit  
8 hif_uvcontsub  
9 hif_makeimages  
10 hif_makeimlist  
11 hif_makeimages  
12 hif_makeimlist  
13 hif_makeimages  
14 hifa_exportdata
```

In the weblog (if `hif_findcont` was run without `cont.dat`):

The screenshot shows a web browser window with the title 'Task Summaries'. The address bar shows a file path: `file:///Users/dtafoya/Documents/Work/Nordic_ARC_node/Pipeline_tutorial/2018.1.01201.S/science_goal`. The browser has tabs for 'Home', 'By Topic', and 'By Task'. The ALMA logo is visible in the top left. The main content area is titled 'Task Summaries' and contains a table of tasks. The first task, '1. hifa_importdata: Register measurement sets with the pipeline', has a QA score of 0.50 and a duration of 0:21:59. The other five tasks have a QA score of 1.00. A mouse cursor is pointing to the sixth task, '6. hif_findcont: Detect continuum frequency ranges'. Below the table is a section titled 'CASA logs and scripts' with four links to view or download log files.

| Task | QA Score | Duration |
|--|--------------------------|----------|
| 1. hifa_importdata : Register measurement sets with the pipeline | 1/1 have HISTORY 0.50 | 0:21:59 |
| 2. hif_mstransform : Create science target MS | 1.00 | 1:24:08 |
| 3. hifa_flagtargets : ALMA Target flagging | 1.00 | 0:09:36 |
| 4. hifa_imageprecheck : ImagePreCheck | 1.00 | 0:16:27 |
| 5. hif_makeimlist : Set-up parameters for target per-spw continuum imaging | 1.00 | 0:08:28 |
| 6. hif_findcont : Detect continuum frequency ranges | 1.00 | 0:06:39 |

CASA logs and scripts

- [View, view in new tab](#) or [download](#) casa-20201118-102913.log (1.2 MB)
- [View, view in new tab](#) or [download](#) casa_commands.log (8.0 KB)
- [View, view in new tab](#) or [download](#) casa_pipescript.py (816 bytes)
- [View, view in new tab](#) or [download](#) (800 bytes)

In the weblog (if hif_findcont was run without *cont.dat*):

Task Details

file:///Users/dtafoya/Documents/Work/Nordic_ARC_node/Pipeline_tutorial/2018.1.01201.S/science_goal ... 検索

ALMA Home By Topic By Task Project Code N/A

Tasks in execution order

1. hifa_importdata
2. hif_mstransform
3. hifa_flagtargets
4. hifa_imageprecheck
5. hif_makeimlist (mfs)
6. hif_findcont

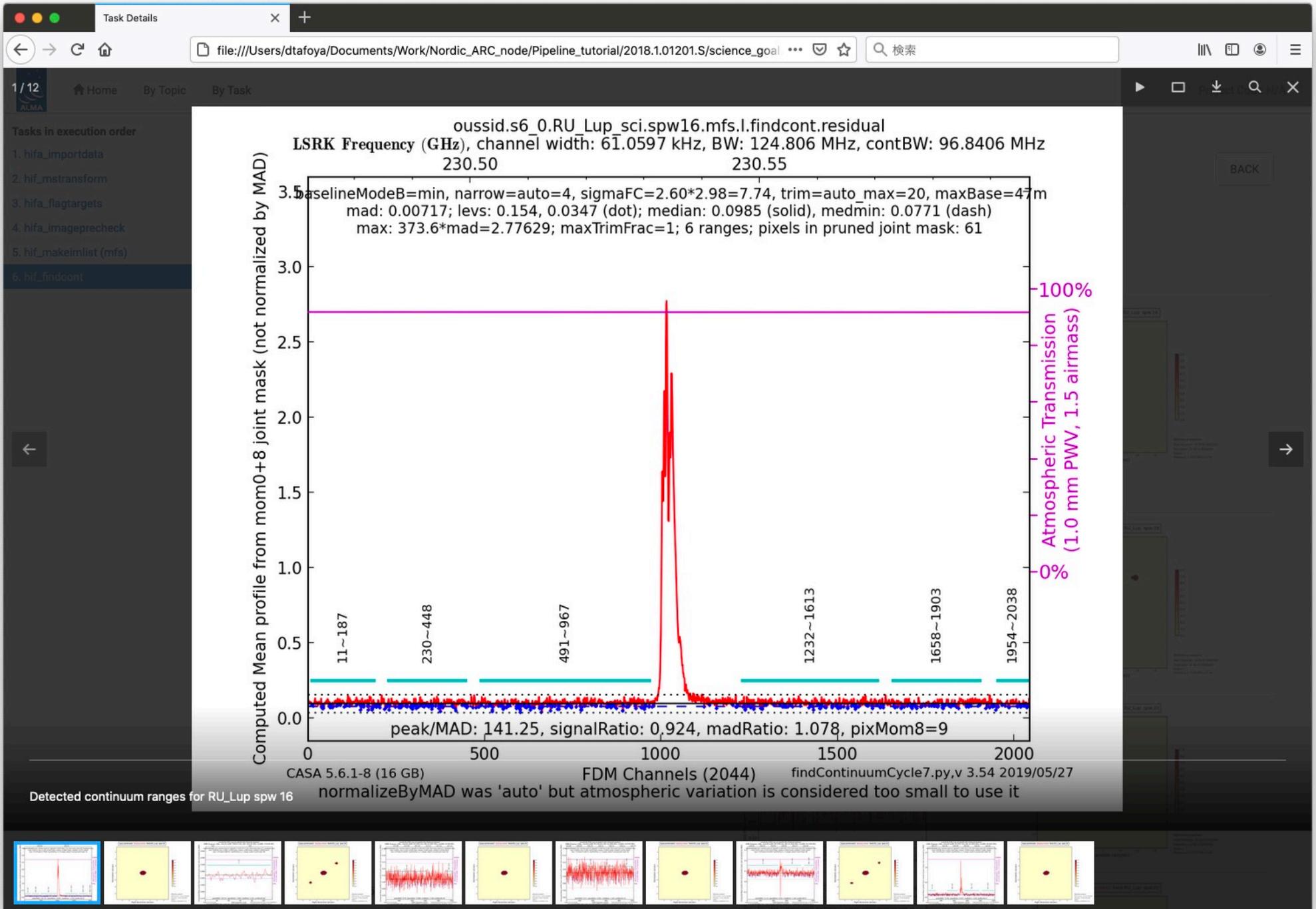
6. Find Continuum

BACK

| Field | Spw | Continuum Frequency Range | | Frame | Status | Average spectrum | Joint mask |
|--------|---------------|---------------------------|----------------|-------|--------|------------------|------------|
| | | Start | End | | | | |
| RU_Lup | 16 | 230.47255 GHz | 230.48336 GHz | LSRK | NEW | | |
| | | 230.48592 GHz | 230.49929 GHz | | | | |
| | | 230.50186 GHz | 230.53098 GHz | | | | |
| | | 230.54710 GHz | 230.57043 GHz | | | | |
| | | 230.57311 GHz | 230.58813 GHz | | | | |
| | | 230.59119 GHz | 230.59638 GHz | | | | |
| 18 | 231.83702 GHz | 233.55591 GHz | NEW, All cont. | | | | |
| 20 | NEW | 217.07165 GHz | 217.10848 GHz | | | | |
| | | 217.12565 GHz | 217.13261 GHz | | | | |
| 22 | 216.08631 GHz | 216.08961 GHz | | | | | |

file:///Users/dtafoya/Documents/Work/Nordic_ARC_node/Pipeline_tutorial/2018.1.01201.S/science_goal.uid__A001_X133d_X2c85/group.ui..._sci.spw16.mfs.1.findcont.residual.meanSpectrum.mom0mom8jointMask.min.min.2.6sigma.narrowauto=4.trimauto_max=20.overrideTrue.png

In the weblog (if `hif_findcont` was run without `cont.dat`):



Fitting the continuum:

```
dtafoya — -zsh — 143x53  
CASA <13>: hif_uvcontfit(automat
```

- This will take around 6 min.



```
1 hifa_importdata  
2 hif_mstransform  
3 hifa_flagtargets  
4 hifa_imageprecheck  
5 hif_makeimlist  
6 hif_findcont  
7 hif_uvcontfit  
8 hif_uvcontsub  
9 hif_makeimages  
10 hif_makeimlist  
11 hif_makeimages  
12 hif_makeimlist  
13 hif_makeimages  
14 hifa_exportdata
```

Subtracting the continuum:

```
CASA <14>: hif_uvcontsub(pipeline="automatic")
```

- The original continuum + line emission is contained in the DATA column of the MS, while the continuum subtracted data are written to the CORRECTED column.



```
1 hifa_importdata
2 hif_mstransform
3 hifa_flagtargets
4 hifa_imageprecheck
5 hif_makeimlist
6 hif_findcont
7 hif_uvcontfit
8 hif_uvcontsub
9 hif_makeimages
10 hif_makeimlist
11 hif_makeimages
12 hif_makeimlist
13 hif_makeimages
14 hifa_exportdata
```

Remember

- keep checking your weblog.
- refresh and see the steps appear.
- run `h_save('my_context.context')` regularly.



now we can actually make images:

`hif_makeimlist`

```
graph TD; A(hif_makeimlist) --> B(hif_makeimages);
```

This stage determines image parameters (image size, cell size, etc) to be used in the subsequent `hif_makeimages` stage.

`hif_makeimages`

Cleaned images are created for each spectral window using the parameters defined in `hif_makeimlist`.

First, make image of the aggregated continuum:

In step 5 `hif_makeimlist(specmode='mfs')` was run.

```
dtfoya --zsh-- 143x53  
CASA <15>: hif_makeimages(pipelinemode="automatic")
```

```
1 hifa_importdata  
2 hif_mstransform  
3 hifa_flagtargets  
4 hifa_imageprecheck  
5 hif_makeimlist  
6 hif_findcont  
7 hif_uvcontfit  
8 hif_uvcontsub  
9 hif_makeimages  
10 hif_makeimlist  
11 hif_makeimages  
12 hif_makeimlist  
13 hif_makeimages  
14 hifa_exportdata
```



- This will take around 6 min.
- This will create the images based on the previous `hif_makeimlist` (step 5).

In the weblog:

The screenshot shows a web browser window with the title 'Task Summaries'. The address bar shows a file path: `file:///Users/dtafoya/Documents/Work/Nordic_ARC_node/Pipeline_tutorial/2018.1.01201.S/science_goal`. The browser has a search bar with the text '検索' and a 'Project Code N/A' label in the top right.

The main content area is titled 'Task Summaries' and contains a table with the following columns: 'Task', 'QA Score', and 'Duration'. The table lists nine tasks, each with a QA score and a duration. The first task, '1. hifa_importdata: Register measurement sets with the pipeline', has a QA score of 0.50 and a duration of 0:21:59. The remaining eight tasks have a QA score of 1.00. A mouse cursor is pointing to the ninth task, '9. hif_makeimages: Make target per-spw continuum images', which has a QA score of 1.00 and a duration of 0:05:44.

Below the table is a section titled 'CASA logs and scripts' containing four links:

- [View, view in new tab or download casa-20201118-102913.log \(1.7 MB\)](#)
- [View, view in new tab or download casa_commands.log \(30.4 KB\)](#)
- [View, view in new tab or download casa_pipescript.py \(1.3 KB\)](#)
- [View, view in new tab or download \(896 bytes\)](#)

In the weblog:

Task Details

file:///Users/dtafoya/Documents/Work/Nordic_ARC_node/Pipeline_tutorial/2018.1.01201.S/science_goal ... 検索

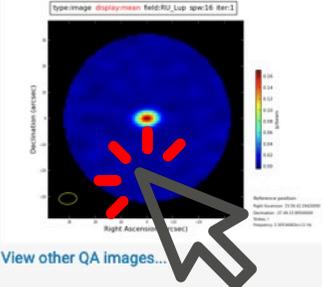
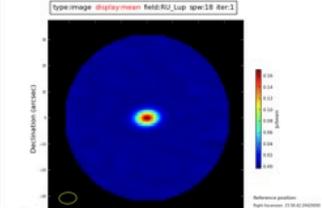
ALMA Home By Topic By Task Project Code N/A

9. Tclean/MakeImages

Make target per-spw continuum images

BACK

Image Details

| Field | Spw | Pol | Image details | Image result |
|-----------------|--|-----|--|---|
| RU_Lup (TARGET) | 16 / X1918034924#ALMA_RB_06#BB_1#SW-01 | I | <p>centre frequency of image 230.5345GHz (LSRK)</p> <p>beam 6.88 x 4.36 arcsec</p> <p>beam p.a. -86.4deg</p> <p>final theoretical sensitivity 0.0011 Jy/beam</p> <p>cleaning threshold 0.0078 Jy/beam Dirty DR: 1.5e+02 DR correction: 3.5</p> <p>clean residual peak / scaled MAD 4.79</p> <p>non-pbcor image RMS 0.0018 Jy/beam</p> <p>pbcor image max / min 0.171 / -0.0135 Jy/beam</p> <p>fractional bandwidth / nterms 0.054% / 1</p> <p>aggregate bandwidth 0.0968 GHz (LSRK)</p> <p>score 1.00</p> <p>image file oussid.s9_0.RU_Lup_sci.spw16.mfs.l.iter1.image</p> |  |
| RU_Lup (TARGET) | 18 / X1918034924#ALMA_RB_06#BB_2#SW-01 | I | <p>centre frequency of image 232.6965GHz (LSRK)</p> <p>beam 6.78 x 4.33 arcsec</p> <p>beam p.a. -86.9deg</p> <p>final theoretical sensitivity 0.00021 Jy/beam</p> |  |

In the weblog:

Task Details

file:///Users/dtafoya/Documents/Work/Nordic_ARC_node/Pipeline_tutorial/2018.1.01201.S/science_goal

1 / 6

Home By Topic By Task

Tasks in execution order

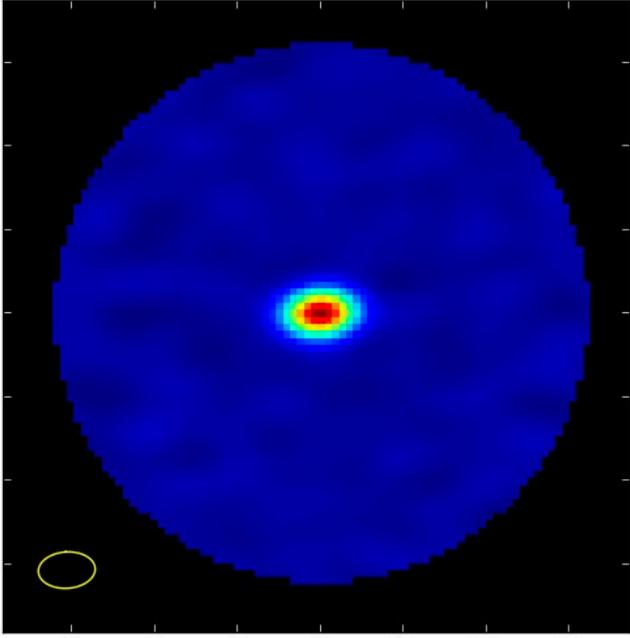
1. hifa_importdata
2. hif_mstransform
3. hifa_flagtargets
4. hifa_imageprecheck
5. hif_makeimlist (mfs)
6. hif_findcont
7. hif_uvcontfit
8. hif_uvcontsub
9. hif_makeimages (mfs)

9. Tclean/Makelimages

Make target per-spw continuum images

Image result

type:image display:mean field:RU_Lup spw:16 iter:1



Decination (arcsec)

Right Ascension (arcsec)

Jy/beam

Reference position:
Right Ascension: 15.56.42.29420000
Declination: -37.49.15.99500000
Stokes: I
Frequency: 2.30534462e+11 Hz

Iteration: 1
Spw: 16
Field: RU_Lup (TARGET)

| | |
|-------------------------------|--|
| image file | oussid_s9_0_RU_Lup_sci.spw16.mfa.i.iter1.image |
| centre frequency of image | 232.69655GHz (LSRK) |
| beam | 0.70 x 0.33 arcsec |
| beam p.a. | -86.9deg |
| final theoretical sensitivity | 0.00021 Jy/beam |

View other QA Images...

In the weblog:

Task Details

file:///Users/dtafoya/Documents/Work/Nordic_ARC_node/Pipeline_tutorial/2018.1.01201.S/science_goal ... 検索

ALMA Home By Topic By Task Project Code N/A

9. Tclean/MakeImages

Make target per-spw continuum images

BACK

Image Details

| Field | Spw | Pol | Image details | Image result |
|-----------------|--|-----|--|--------------------------------|
| RU_Lup (TARGET) | 16 / X1918034924#ALMA_RB_06#BB_1#SW-01 | I | <p>centre frequency of image 230.5345GHz (LSRK)</p> <p>beam 6.88 x 4.36 arcsec</p> <p>beam p.a. -86.4deg</p> <p>final theoretical sensitivity 0.0011 Jy/beam</p> <p>cleaning threshold 0.0078 Jy/beam Dirty DR: 1.5e+02 DR correction: 3.5</p> <p>clean residual peak / scaled MAD 4.79</p> <p>non-pbcor image RMS 0.0018 Jy/beam</p> <p>pbcor image max / min 0.171 / -0.0135 Jy/beam</p> <p>fractional bandwidth / nterms 0.054% / 1</p> <p>aggregate bandwidth 0.0968 GHz (LSRK)</p> <p>score 1.00</p> <p>image file oussid.s9_0.RU_Lup_sci.spw16.mfs.l.iter1.image</p> | <p>View other QA Images...</p> |
| RU_Lup (TARGET) | 18 / X1918034924#ALMA_RB_06#BB_2#SW-01 | I | <p>centre frequency of image 232.6965GHz (LSRK)</p> <p>beam 6.78 x 4.33 arcsec</p> <p>beam p.a. -86.9deg</p> <p>final theoretical sensitivity 0.00021 Jy/beam</p> | |

In the weblog: The cleaning masks (auto masking)

Task Details

file:///Users/dtafoya/Documents/Work/Onsala/Nordic_ARC_node/Pipeline_tutorial/2018.1.01201.S/sci ... 検索

ALMA Home By Topic By Task Project Code N/A

Tasks in execution order

1. hifa_importdata
2. hif_mstransform
3. hifa_flagtargets
4. hifa_imageprecheck
5. hif_makeimlist (mfs)
6. hif_findcont
7. hif_uvcontfit
8. hif_uvcontsub
9. hif_makeimages (mfs)
10. hif_makeimlist (cont)
11. hif_makeimages (cont)
12. hif_makeimlist (cube)
13. hif_makeimages (cube)
14. hifa_exportdata

Clean results for RU_Lup (TARGET) SpW 16

Navigation: ⏪ ⏩ BACK

| Iteration | Image | Residual | Clean Mask |
|-----------|-------|----------|------------|
| 1 | | | |
| 0 | | | |
| | | | |

Primary Beam PSE Final Model

The parameters of automasking

The values presented in this table are the standard values that the pipeline uses when it performs automasking and have been extensively tested.

| Array | <i>sidelobethreshold</i> | <i>noisethreshold</i> | <i>minbeamfrac</i> | <i>lownoisethreshold</i> | <i>negativethreshold</i> |
|--|--------------------------|-----------------------|--------------------|--------------------------|--|
| 12m (short) b75<300m | 2.0 | 4.25 | 0.3 | 1.5 | 0.0 (continuum) 15.0 (line) |
| 12m (long) b75>300m | 3.0 | 5.0 | 0.3 | 1.5 | 0.0 (continuum) 7.0 (line) |
| 7m (continuum/line) | 1.25 | 5.0 | 0.1 | 2.0 | 0.0 |
| 12m + 7m combined TENTATIVE | 2.0 | 4.25 | 0.3 | 1.5 | 0.0 |

check the CASA log to see the parameters used by the pipeline:

The screenshot shows a web browser window displaying the 'Task Details' page for an ALMA pipeline. The browser address bar shows the file path: `file:///Users/dtafoya/Documents/Work/Onsala/Nordic_ARC_node/Pipeline_tutorial/2018.1.01201.S/sci...`. The page has a navigation menu with 'Home', 'By Topic', and 'By Task' tabs. The 'By Task' tab is active, showing a list of tasks in execution order on the left. Task 9, 'hif_makeimages (mfs)', is selected and highlighted in blue. The main content area displays a table of parameters for this task, including 'image file', 'centre frequency of image', 'beam', 'beam p.a.', 'final theoretical sensitivity', 'cleaning threshold', 'clean residual peak / scaled MAD', 'non-pbcor image RMS', 'pbcor image max / min', 'fractional bandwidth / nterms', 'aggregate bandwidth', and 'score'. A 'score' of 1.00 is shown in a green box. To the right of the parameters is a circular image plot showing the distribution of the data. Below the parameters table are sections for 'Pipeline QA', 'Input Parameters', 'Tasks Execution Statistics', and 'CASA logs for stage 9'. In the 'CASA logs for stage 9' section, there is a link to 'view or download stage9/casapy.log (344.3 KB)'. A red starburst icon and a mouse cursor are pointing at this link.

Tasks in execution order

1. hifa_importdata
2. hif_mstransform
3. hifa_flagtargets
4. hifa_imageprecheck
5. hif_makeimlist (mfs)
6. hif_findcont
7. hif_uvcontfit
8. hif_uvcontsub
9. hif_makeimages (mfs)
10. hif_makeimlist (cont)
11. hif_makeimages (cont)
12. hif_makeimlist (cube)
13. hif_makeimages (cube)
14. hifa_exportdata

| | |
|----------------------------------|---|
| image file | oussid.s9_0.RU_Lup_sci.spw24.mfs.l.iter1.image |
| centre frequency of image | 220.3953GHz (LSRK) |
| beam | 7.18 x 4.50 arcsec |
| beam p.a. | -84.8deg |
| final theoretical sensitivity | 0.002 Jy/beam |
| cleaning threshold | 0.012 Jy/beam Dirty DR: 74 DR correction: 3 |
| clean residual peak / scaled MAD | 4.68 |
| non-pbcor image RMS | 0.0024 Jy/beam |
| pbcor image max / min | 0.146 / -0.0245 Jy/beam |
| fractional bandwidth / nterms | 0.027% / 1 |
| aggregate bandwidth | 0.0404 GHz (LSRK) |
| score | 1.00 |
| image file | oussid.s9_0.RU_Lup_sci.spw26.mfs.l.iter1.image |

View other QA images...

Pipeline QA

Input Parameters

Tasks Execution Statistics

CASA logs for stage 9

[view or download stage9/casapy.log \(344.3 KB\)](#)

scroll down to find the parameters used for tclean:

The screenshot shows a web browser window with the URL `file:///Users/dtafoya/Documents/Work/Onsala/Nordic_ARC_node/Pipeline_tutorial/2018.1.01201.S/sci...`. The page title is "Task Details". On the left, there is a sidebar with "Tasks in execution order" and a list of tasks. Task 9, "hif_makeimages (mfs)", is highlighted in blue. On the right, the main content area shows a log of task execution. The log includes the following text:

```
loading weight function
2020-12-03 07:46:52 INFO pipeline.infrastructure.casatools::imager::apparentSensitivity() (assuming that MS weights have correct scale and units)
2020-12-03 07:46:53 INFO pipeline.infrastructure.casatools::imager::apparentSensitivity() RMS Point source sensitivity : 0.00150252
2020-12-03 07:46:53 INFO pipeline.infrastructure.casatools::imager::apparentSensitivity() Relative to natural weighting : 1.04257
2020-12-03 07:46:53 INFO hif_makeimages::pipeline.infrastructure.casatools:: imager.apparentsens() CASA tool call took 0.347672s
2020-12-03 07:46:53 INFO hif_makeimages::pipeline.hif.heuristics.imageparams_base:: apparentsens result for EB uid__A002_Xd2b681_Xb99d_target Field 2 SPW 16 C
hanRange 0~2047: 0.00150252071427 Jy/beam
2020-12-03 07:46:53 INFO hif_makeimages::pipeline.hif.heuristics.imageparams_base:: Channel selection bandwidth heuristic (nbin or findcont; (spw BW / nchan_s
l BW) ** 0.5): Correcting sensitivity for EB uid__A002_Xd2b681_Xb99d_target Field RU_Lup SPW 16 by 1.13 from 0.0015 Jy/beam to 0.0017 Jy/beam
2020-12-03 07:46:53 INFO hif_makeimages::pipeline.hif.heuristics.imageparams_base:: Effective BW heuristic: Correcting sensitivity for EB uid__A002_Xd2b681_Xb
99d_target Field RU_Lup SPW 16 by 1.26 from 0.0017 Jy/beam to 0.00215 Jy/beam
2020-12-03 07:46:53 INFO hif_makeimages::pipeline.hif.heuristics.imageparams_base:: Final sensitivity estimate for Field RU_Lup, SPW 16 specmode mfs: 0.00111 J
y/beam
2020-12-03 07:46:53 INFO hif_makeimages::pipeline.hif.tasks.tclean.tclean:: Compute the dirty image
2020-12-03 07:46:53 INFO hif_makeimages::pipeline.hif.heuristics.imageparams_alma:: autobox heuristic: Representative baseline length is 30.3 meter
2020-12-03 07:46:53 INFO hif_makeimages::pipeline.infrastructure.jobrequest:: Executing tclean(vis=['uid__A002_Xd2b681_Xa1c2_target.ms', 'uid__A002_Xd2b681_Xb9
9d_target.ms'], field='RU_Lup', spw=['16:230.45896045~230.469885743GHz;230.472327149~230.485815919GHz;230.488257325~230.517493165GHz;230.533484376~230.559
485352~230.574622071GHz;230.577551758~230.582861817GHz', '16:230.458797266~230.469661524GHz;230.472163966~230.485652735GHz;230.488094141~230.517329981GHz;230.533382227~23
0.556758692GHz;230.559383204~230.574458887GHz;230.57744961~230.582698634GHz'], antenna=['0,1,2,3,4,5,6,7,8,9,10', '0,1,2,3,4,5,6,7,8,9,10,11'], scan=['6,8,11,13,16', '7,10
'], intent='OBSERVE_TARGET#ON_SOURCE', datacolumn='data', imagename='oussid.s9_0.RU_Lup_sci.spw16.mfs.I.iter0', imsize=[90, 90], cell=['0.85arcsec'], phasecenter='ICRS 15:
56:42.2942 -037.49.15.995', stokes='I', specmode='mfs', nchan=-1, outframe='LSRK', perchanweightdensity=False, grider='standard', chanchunks=-1, mosweight=False, usepoint
ing=False, pblimit=0.2, deconvolver='hogbom', restoration=False, restoringbeam='common', pbcor=False, weighting='briggs', robust=0.5, npixels=0, niter=0, threshold='0.0mJy
', nsigma=0.0, interactive=0, usemask='auto-multithresh', sidelobethreshold=1.25, noisethreshold=5.0, lownoisethreshold=2.0, negativethreshold=0.0, minbeamfrac=0.1, growit
erations=75, dogrowprune=True, minpercentchange=1.0, fastnoise=False, savemodel='none', parallel=False)
2020-12-03 07:46:53 INFO tclean:::
2020-12-03 07:46:53 INFO tclean::: + #####
2020-12-03 07:46:53 INFO tclean::: + ##### Begin Task: tclean #####
2020-12-03 07:46:53 INFO tclean::: tclean(vis=['/Users/dtafoya/Documents/Work/Onsala/Nordic_ARC_node/Pipeline_tutorial/2018.1.01201.S/science_goal.uid__A001_
X133d_X2c85/group.uid__A001_X133d_X2c86/member.uid__A001_X133d_X2c8b/calibrated/working/uid__A002_Xd2b681_Xa1c2_target.ms', '/Users/dtafoya/Documents/Work/Onsala/Nordic
_ARC_node/Pipeline_tutorial/2018.1.01201.S/science_goal.uid__A001_X133d_X2c85/group.uid__A001_X133d_X2c86/member.uid__A001_X133d_X2c8b/calibrated/working/uid__A002_Xd2
b681_Xb99d_target.ms'], selectdata=True, field="RU_Lup", spw=['16:230.45896045~230.469885743GHz;230.472327149~230.485815919GHz;230.488257325~230.517493165GHz;230.533484376~23
0.556921876GHz;230.559485352~230.574622071GHz;230.577551758~230.582861817GHz', '16:230.458797266~230.469661524GHz;230.472163966~230.485652735GHz;230.488094141~230.51732998
1GHz;230.533382227~230.556758692GHz;230.559383204~230.574458887GHz;230.57744961~230.582698634GHz'], timerange="",
2020-12-03 07:46:53 INFO tclean::: + uvrange="", antenna=['0,1,2,3,4,5,6,7,8,9,10', '0,1,2,3,4,5,6,7,8,9,10,11'], scan=['6,8,11,13,16', '7,10'], observatio
n="", intent="OBSERVE_TARGET#ON_SOURCE",
2020-12-03 07:46:53 INFO tclean::: + datacolumn="data", imagename="oussid.s9_0.RU_Lup_sci.spw16.mfs.I.iter0", imsize=[90, 90], cell=['0.85arcsec'], phasecen
ter="ICRS 15:56:42.2942 -037.49.15.995",
2020-12-03 07:46:53 INFO tclean::: + stokes="I", projection="SIN", startmodel="", specmode="mfs", reffreq="",
2020-12-03 07:46:53 INFO tclean::: + nchan=-1, start="", width="", outframe="LSRK", veltype="radio",
2020-12-03 07:46:53 INFO tclean::: + reftfreq=[], interpolation="linear", perchanweightdensity=False, grider="standard", facets=1,
2020-12-03 07:46:53 INFO tclean::: + psfphasecenter="", chanchunks=-1, wprojplanes=1, vptable="", mosweight=False,
2020-12-03 07:46:53 INFO tclean::: + aterm=True, psterm=False, bwawp=True, conjbeams=False, cfcache="",
2020-12-03 07:46:53 INFO tclean::: + usepointing=False, computepastep=360.0, rotatpastep=360.0, pointingoffsetsigdev=0.0, pblimit=0.2,
```

move on to continuum images:

```
CASA <16>: hif_makeimlist(specmode='cont')
```

- continuum using the channels found by `hif_findcont`.
- previous images are not deleted, each image is named representative of the stage number.



```
1 hifa_importdata
2 hif_mstransform
3 hifa_flagtargets
4 hifa_imageprecheck
5 hif_makeimlist
6 hif_findcont
7 hif_uvcontfit
8 hif_uvcontsub
9 hif_makeimages
10 hif_makeimlist
11 hif_makeimages
12 hif_makeimlist
13 hif_makeimages
14 hifa_exportdata
```

```
CASA <17>: hif_makeimages(pipelinemode="automatic")
```

now the spectral cubes:

```
CASA <18>: hif_makeimlist(specmode='cube')
```

- will image the spectral cubes for all the spws, but imaging can be controlled by defining input parameters (see next slides).



```
CASA <19>: hif_makeimages(pipelinemode="automatic")
```

```
1 hifa_importdata  
2 hif_mstransform  
3 hifa_flagtargets  
4 hifa_imageprecheck  
5 hif_makeimlist  
6 hif_findcont  
7 hif_uvcontfit  
8 hif_uvcontsub  
9 hif_makeimages  
10 hif_makeimlist  
11 hif_makeimages  
12 hif_makeimlist  
13 hif_makeimages  
14 hifa_exportdata
```

Change the imaging parameters:

```
dtafoya — -zsh — 143x53  
CASA <20>: hif_makeimlist(specmode='mfs',robust=-0.5,spw='24',hm_imsz=60)
```



- only image spw 24 with robust parameter set to -0.5 and image size = 60pixels.

```
dtafoya — -zsh — 143x53  
CASA <21>: hif_makeimages(pipeline="automatic")
```

Change the imaging parameters:

```
dtafoya — -zsh — 143x53  
CASA <22>: hif_makeimlist(specmode='cube', spw='24', nchan=200, start='219.553GHz')
```



- makes a cube of 200 channels of the spw 24, starting ar freq=219.553GHz.

```
dtafoya — -zsh — 143x53  
CASA <23>: hif_makeimages(pipeline="automatic")
```

Use `help(hif_makeimlist)` to see more options:

```
dtafoya -- zsh -- 143x53

CASA <24>: help(hif_makeimlist)

Help on instance of hif_makeimlist_cli_ in module hif_makeimlist_cli:

hif_makeimlist = class hif_makeimlist_cli_
|   Methods defined here:
|
|   __call__(self, vis=None, imagename=None, intent=None, field=None,
|   spw=None, contfile=None, linesfile=None, uvrange=None, specmode=None,
|   outframe=None, hm_imsz=None, hm_cell=None, calmaxpix=None,
|   phasecenter=None, nchan=None, start=None, width=None, nbins=None,
|   robust=None, uvtaper=None, clearlist=None, per_eb=None, calcsb=None,
|   parallel=None, pipelinemode=None, dryrun=None, acceptresults=None)
|       Compute list of clean images to be produced
|
|       Detailed Description:
|
|       Create a a list of images to be cleaned.
|
|       Arguments :
|           vis: List of input MeasurementSets
|           Default Value:
```

Finally, export new images to fits format, if desired:

```
dtafoya --zsh -- 143x53  
CASA <25>: hifa_exportdata(imaging_products_only=True)
```

```
1 hifa_importdata  
2 hif_mstransform  
3 hifa_flagtargets  
4 hifa_imageprecheck  
5 hif_makeimlist  
6 hif_findcont  
7 hif_uvcontfit  
8 hif_uvcontsub  
9 hif_makeimages  
10 hifa_imageprecheck  
11 hif_makeimages  
12 hif_makeimlist  
13 hif_makeimages  
14 hifa_exportdata
```

- Export the images to fits files.



A few final notes

- Once the continuum subtraction stages are done, its basically `hif_makeimlist` and `hif_makeimages` loops
- You should always have the weblog open to view in real-time
- If you made a mistake, don't worry, the stage just will fail and report an error but you can still continue
- If you want to change an image parameter before running `hif_makeimages`, run `hif_makeimlist` again, check the weblog to make sure of the parameters
- In imaging cubes, do each SPW separately, so you have full fine-tuning available
- Do `h_save()` regularly, if your CASA crashes the weblog and pipeline context are not saved correctly - you must restart
- For more examples, visit the ALMA Imaging Pipeline Reprocessing webpage:

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

End of tutorial