



HILADO @JIVE

Mark Kettenis, kettenis@jive.nl

Des Small, small@jive.nl

EVN: European VLBI Network

▶ Consortium of radio telescopes

- Ef, Mc, On, Jb, Cm, Nt, Tr, Wb, Sh, Ur, Hh, Ar, Mh, Ys, Sv, Bd, Zc, Ro, Ku, My, Wz, Ny, Sr
- Involving 14 different organizations around the world: Europe, China, Puerto Rico, South Africa, Russia
- And 12 more antennas for “Global observations” with NRAO

▶ Covering range of frequencies

- Workhorse frequencies 18cm, 6cm,
- Also available: SX, 5cm, 1.2cm
- And at specific stations 90cm, 21cm, UHF, 50cm, 2cm, 0.7mm

▶ Reaching mas resolutions

- From 15mas for 1.4 GHz EVN (can add MERLIN for brightness sensitivity)
- To 1 mas at 5GHz with Asian, African or American baselines

▶ Sensitivity of $5\mu\text{Jy}$ in 8hr at 1.4 GHz

- Combination of Big Antennas and 1 Gbps bandwidth (and soon 2 or 4Gbps)
- Big antennas also vital for spectroscopy (mJy sensitivity)

Joint Institute for VLBI in Europe

- Promote and advance the use of VLBI for astronomy
 - Central correlation
 - User services
 - Network support
 - Innovation
 - EC liaison/representation
- Founded in 1993
 - Base budget from partners in 8 countries:
 - China, France, Germany, Italy, Spain, Sweden, United Kingdom, South Africa, the Netherlands
 - hosted by ASTRON
- Large number of external projects
 - Many people with temp positions in R&D/Science
- 33 people, 11 nationalities



ParseITongue

- Python interface to “classic” AIPS
- Original goal: scripting
- Support for AIPS “TV” for interactive use

Executing Tasks

```
>>> mandl = AIPSTask('MANDL')
>>> mandl.imsiize = AIPSList([256]) # set the image size in cells
>>> mandl.outname = "MBROT"         # set the name of the output image
>>> mandl.outseq = 3                 # set output image sequence number
>>> mandl.go()
MANDL1: Task MANDL (release of 31DEC12) begins
MANDL1: Create MBROT .MANDL . 3 (MA) on disk 1 cno 1
MANDL1: Appears to have ended successfully
MANDL1: jop56 31DEC12 TST: Cpu= 0.0 Real= 1
>>>
```

Data Access

Create a new calibration table:

```
from AIPS import AIPS
from Wizardry.AIPSDATA import AIPSUVDATA

AIPS.userno = 667

data = AIPSUVDATA('N03L1', 'UVDATA', 1, 1)
oldcl = data.table('CL', 1)

newcl = data.attach_table('CL', 2, no_term=oldcl.keywords['NO_TERM'])
newcl.keywords['NO_ANT'] = oldcl.keywords['NO_ANT']

for row in oldcl:
    row.real1 = [2 * x for x in row.real1]
    row.real2 = [2 * x for x in row.real2]
    newcl.append(row)

newcl.close()
oldcl.close()
```

Parallelisation

Low-level stuff:

- **ParallelQueue()**

Queue tasks, and execute them in parallel:

```
queue = ParallelTask.ParallelQueue()  
  
mandl1 = AIPSTask.AIPSTask('MANDL')  
mandl2 = AIPSTask.AIPSTask('MANDL')  
  
# Add these tasks to the work queue  
queue.queue(mandl1)  
queue.queue(mandl2)  
  
# And we're off!  
queue.go()
```

- **rcopy(), rdiskappend()**

Facilitate copying data between AIPS installations

AIPS Lite

- Provides minimal AIPS installation
- Uses official binaries from NRAO
- Easy to setup from within Python:

```
import AIPSLite
```

```
AIPSLite.setup('/tmp/aips', '/jop31_0/scratch/DATA')
```


Software Correlation



Mark 4



SFXC

Multiple Phase Centers

