The ALMA keyword filler program

OPEN QUESTIONS

PADLIST

List of ALMA pad names contributing to data

- If mosaic, or different ms, now the list of all pads, without repetition is given

NANT

Number of antennas contributing to data

- If mosaic, or different ms, now the max of # antennas is given better the median?other?

fraction of flagging accepted (1 flagged, 0 unflagged):
now only if all the antenna is flagged is not considered

MINEL

Minimum elevation range achieved during observations of target data

EXPTIM

Integration time of ants

Consider only unflagged data, all data, a fraction?!

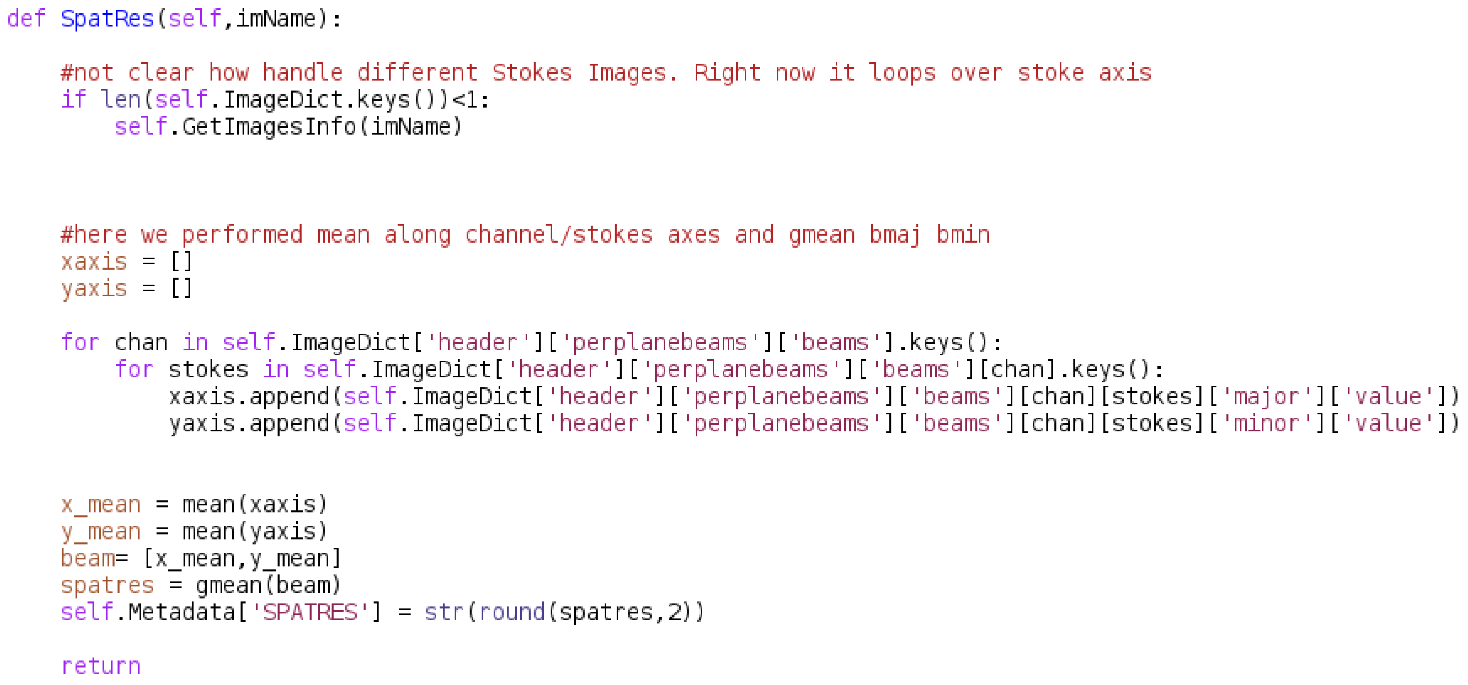
fractionflaggedperant = flaggedrant / (flaggedperant + unflaggedperant)
<= 0.9999 in the code

consequence on the run time of the code

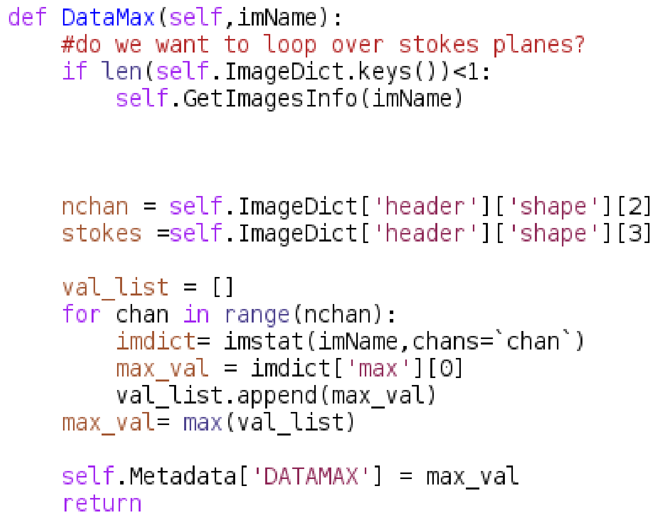
SPATRES

Geometric average of the max and min beam axes

- loop over stoke axis



```
def SpatRes(self,imName):

    #not clear how handle different Stokes Images. Right now it loops over stoke axis
    if len(self.ImageDict.keys())<1:
        self.GetImagesInfo(imName)


    #here we performed mean along channel/stokes axes and gmean bmaj bmin
    xaxis = []
    yaxis = []

    for chan in self.ImageDict['header']['perplanebeams']['beams'].keys():
        for stokes in self.ImageDict['header']['perplanebeams']['beams'][chan].keys():
            xaxis.append(self.ImageDict['header']['perplanebeams']['beams'][chan][stokes]['major']['value'])
            yaxis.append(self.ImageDict['header']['perplanebeams']['beams'][chan][stokes]['minor']['value'])


    x_mean = mean(xaxis)
    y_mean = mean(yaxis)
    beam= [x_mean,y_mean]
    spatres = gmean(beam)
    self.Metadata['SPATRES'] = str(round(spatres,2))

    return
```



```
def DataMax(self,imName):
    #do we want to loop over stokes planes?
    if len(self.ImageDict.keys())<1:
        self.GetImagesInfo(imName)


    nchan = self.ImageDict['header']['shape'][2]
    stokes =self.ImageDict['header']['shape'][3]

    val_list = []
    for chan in range(nchan):
        imdict= imstat(imName,chans=`chan`)
        max_val = imdict['max'][0]
        val_list.append(max_val)
    max_val= max(val_list)

    self.Metadata['DATAMAX'] = max_val
    return
```

DATAMAX

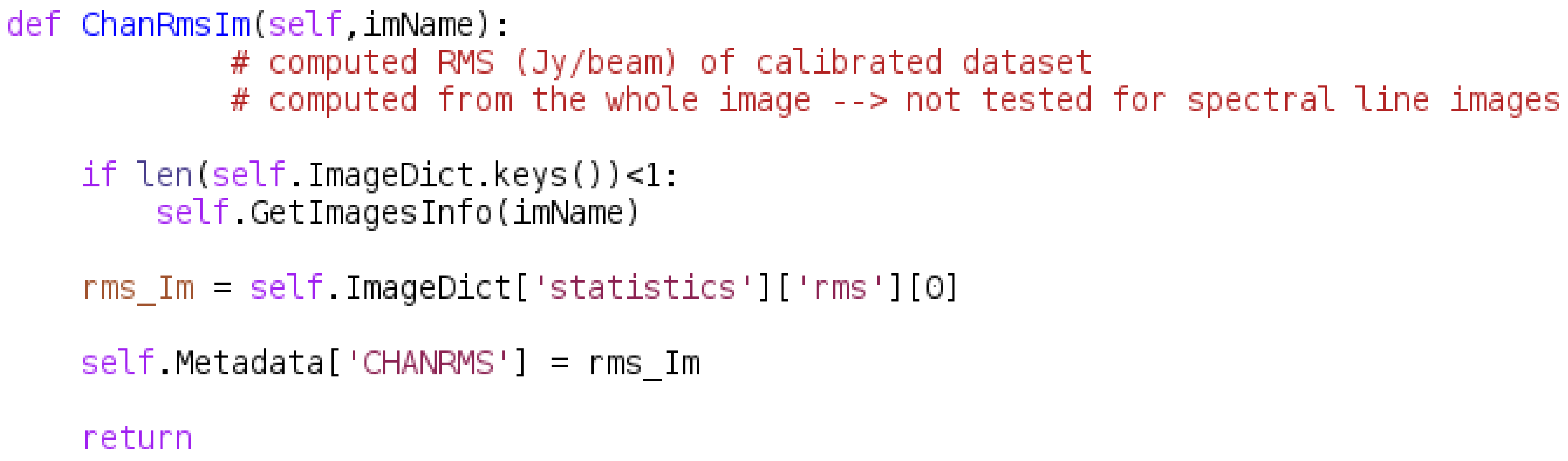
Maximum valid physical value represented by the array (Jy/beam)

- loop over stoke axis?!
- Same question for DATAMIN

CHANRMS

Computed RMS from calibrated dataset

- Now rms is calculated from image
- Source(s) not excluded   meaning of this rms??



```
def ChanRmsIm(self,imName):
            # computed RMS (Jy/beam) of calibrated dataset
            # computed from the whole image --> not tested for spectral line images

    if len(self.ImageDict.keys())<1:
        self.GetImagesInfo(imName)

    rms_Im = self.ImageDict['statistics']['rms'][0]

    self.Metadata['CHANRMS'] = rms_Im

    return
```

DYNRANGE

Estimation of dynamic range from interferometric data

- DATAMAX / CHANRMS
- Meaning of kw calculated like this?

UVNOISE:

RMS of all visibilities (12m, 7m, TP) combined

- Definition?!

RA_TARG and DEC_TARG

computed or specified from user and OT

- is this the position of the field in the MS?
- How should we behave in case of a mosaic? Would an average over the field positions be enough?

OBSGEO-X Y Z

coordinates in cartesian geocentric terrestrial reference frame, specifying the location the observation took place at time given in MJD-AVG

- need to add OBSGEO-X Y Z   how it is calculated? It needs a list of the pad-positions or antennas (X, Y, Z) in time, assuming that we take a fixed pad position / antenna to serve as reference point.

SCIREQ TICKET: SCIREQ-882

List of keywords that need to be passed to images from previous steps in image generation process

SIDLOB (requires the dirty beam or PSF information)
SPW ("Identification numbers of spectral window from ASDM")
OBSMODE ("observing modes contributing data to the image")
OBSTYPE ("intent of observation as described in the OT regardless of the role in the dataset") this is not the calibrator intent after the observations, or at least we cannot understand the difference between this keyword and CALIBR
CALIBR ("calibrator observing strategy") there is already something similar in the pipeline weblog, can someone point us to which files we should check to extract it?
ALMASW

***PI & Proposal information**

PROPCODE is not in the ms (the request is not for the project UID, but for the code in the usual 2015.x.xxxxx.S)
OBSERVER in the ms there is the ALMAID of the PI, not lastname and first name as requested by the keyword format. Where is this information?
COILIST
TITLE

***Pipeline Archive and Request Information**

OBJECT - listed as present in FITS product, but currently empty
PIPVER
PPRNAME
CASAVER
ASDMLIST
COMBLEVL
GROUP
MEMBER
SGOALNAME
SGOAL
SBNAMES
SBUIDS
LINTRN

They could provided by the pipeline or AQUA?

CODE FORM

Each method standalone?

-
  ```
  def KwiMS(self, ms, field)

  def KwsMS(self, ImName)
    loop for ms in mslist
       loop for fieldid in fieldidlist
  ```

OR

-
  ```
  def Kw1MS(self, imName)
   loop for ms in mslist
     loop for fieldid in fieldidlist

  ||||||||||||

  def KwNMS(self, imName)
   loop for ms in mslist
     loop for fieldid in fieldidlist
  ```

- Where the code will be run?!
  Form
  Kws to provide

Need of systematic telecons
with PL, ASAWG, AQUA groups