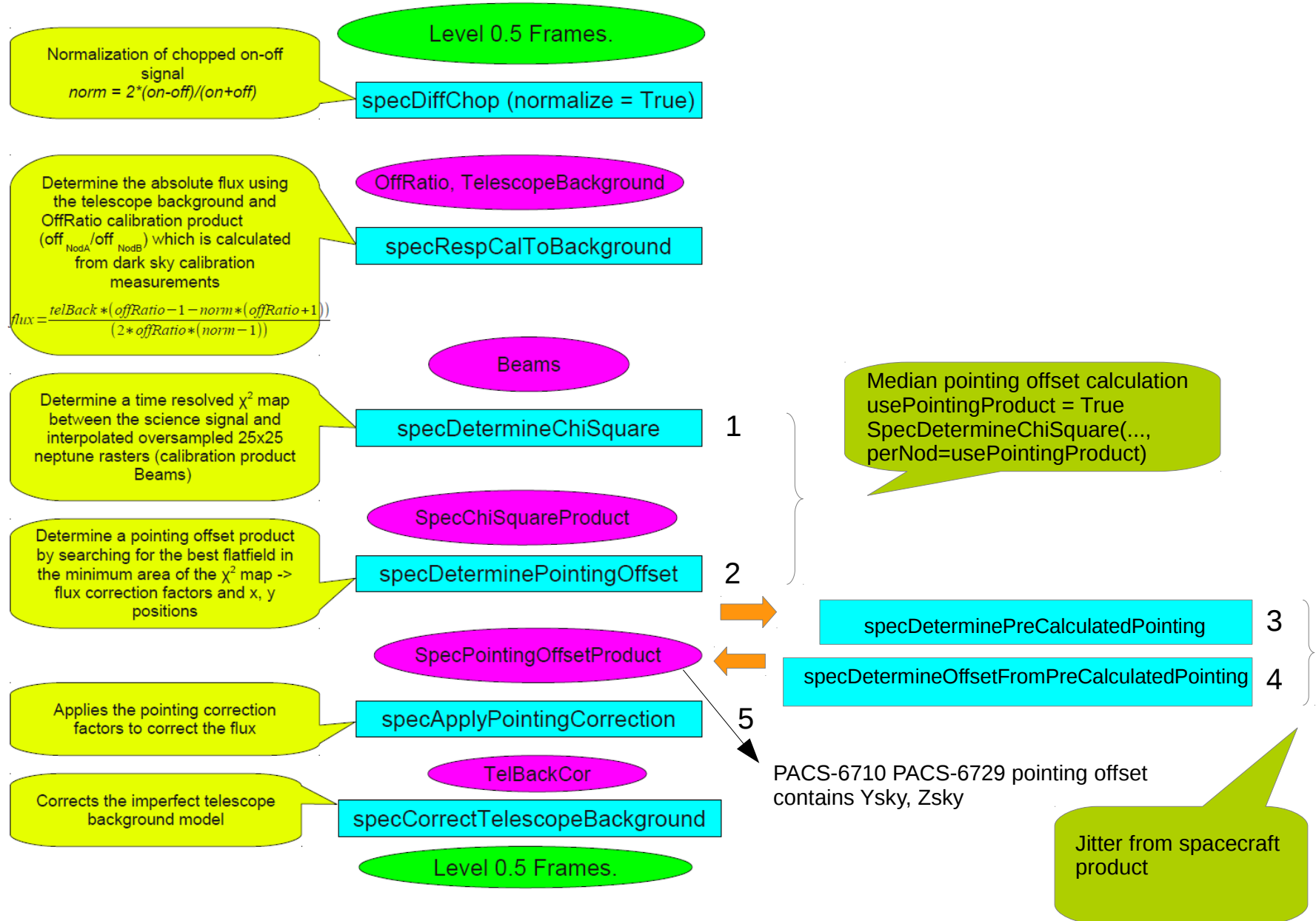


Point source calibrated data using the POC script

- General: POC does not include red or blue leaks
- Identify candidate obsids:
 - Observing Mode: ChopNod, single pointing
 - Conditions to apply POC:
 - Isolation: visual inspection to kick out very extended and crowded fields
 - Point-like nature: extractCentralSpectrum on rebinned cubes to extract the comparison to a perfectly point source $(s1/s9)_{observed}/(s1/s9)_{P.S.}$. Values of > 20% will serve to kick out semi-extended sources (despite the spatial extent-offset degeneracy)
 - Background free (no easy way to identify that)
- Tweak POC script in asymmetric TBN parameters (perNod=True) (Presentations from Jeroen Bouwman) trace JIRA tickets for last version changes in default parameters.
 - #The pointing error can also be seen as made of two components: an absolute offset, and a jitter around it.
 - # The absolute offset can only be determined thanks to the tasks described above,
 - # but one can determine an median offset per pointing, i.e. per nod. That allows
 - # to determine the absolute offset from coadded data, and thus have a better result for weaker sources.
 - # This is the path followed when usePointingProduct = True
 - # The jitter is then derived from the gyro-propagated pointing products that come with the observation.
 - # NB: these products have been improved by new calculations (gyro-propagation), so make sure
 - # your level0 products have been processed with SPG 11 or higher (see the PACS and Herschel wiki)
- Way to compare improvements in flux: comparison with SPG
- Two-stage approach: 1._ Store computed pointing offsets to establish science improvement criteria and compare to Sara's Pointing offset report (intended vs attained). 2._ Flux corrections to those cases for which we establish a reliable improvement (limiting brightness?, minimum offset?).
- Output: Spectrum1d and spectrum table
- Modelling?

Essential steps of the ChopNodPointingCorrection.py ipipe script for any brightness point sources



Red Leak reduction for lines

- 390 OBSIDS (210Line+23SmallRange+157SEDs)
- Modification of runPacsSpg task to toggle from current masking of inLeak ranges. Make standalone. ChopNod and Unchopped.
 - Use calTree v77 (parameter in task signature, although not in urm)
 - excludeLeaks=False
 - specFlatFieldLine(maxRange=[190.,217.])
 - specFlatFieldRange(selectedRange=[190.,217.])
 - For chopNod it means using the alternative calibration scheme calBlock+RSRF. Modification of L1_ChopNod.py and L2_ChopNod.py
 - Changing obsContext product naming convention to HPS3DL[R]...
- Output is spectrum1d (c9), does not include spectral tables as continuum is wrong.