**We have to align two sets of frames**:

1. For context, you can bring up the NIR filters at these links (at least know n=narrow, w=wide)
   1. <https://www.stsci.edu/hst/instrumentation/wfc3/performance/throughputs>,
   2. <https://wfc3.gsfc.nasa.gov/tech/filters-ir.html>,
   3. or <http://svo2.cab.inta-csic.es/svo/theory/fps3/index.php?id=HST/WFC3_IR.F160W>
2. Proper motions
   1. The f164n, f167n frames (“epoch 2”, recent observations) to the same WCS and parameters as the f160w observation (“epoch 1”, from the past)
3. Line fluxes and shock models
   1. The f126n, f128n, f130n, f164n, and f167n frames

**Here is likely what you should try (what I was planning next+what we discussed):**

Align a set of frames (1a) or mosaics (2a) per set (orbit). The mode for this HST project is standard, using guide stars that the telescope tracks as the sky moves. This allows the telescope to stare at one part of the sky, which helps with alignment and precision. So Sam needs to (skipping steps as needed) …

* Get Drizzlepac on your nice ssh-able machine. (see below)
* Download the two sets of images to your main directory and into two folders you can copy from. This acts as a backup so you don’t need to redownload things every time.
  1. For proper motions, only pick the f164n and f167n frame that overlaps the f160w frame. Then we can see whether it’s better to work with aligned mosaics or individual frames.
  2. If you want to check any info or that you’re grabbing the right regions, you can go directly to MAST (<https://mast.stsci.edu/portal/Mashup/Clients/Mast/Portal.html> or by Joel Green’s rec, <https://mast.stsci.edu/search/ui/#/hst> - both very useful)
* Next, copy the images into two separate directories (e.g. proper\_motions\_default and epoch2\_only\_default), use drizzlepac for default alignment (see align\_default.py) to check it’s all setup correctly
* Then setup another folder (proper\_motions\_dropsize0p5 or same with epoch2\_only) where you test the effect of oversampling via decreasing drop size
  1. (see <https://drizzlepac.readthedocs.io/en/latest/astrodrizzle.html> - mainly use driz\_sep\_pixrfrac but there is also final\_pixfrac).
  2. Tom said to try 0.75 and 0.5. I haven’t seen much of an impact, but it might be important when we combine it with the next step…
  3. For more on this concept, there’s the handbook: <https://hst-docs.stsci.edu/drizzpac/chapter-3-description-of-the-drizzle-algorithm/3-2-drizzle-concept>
* Finally, we will have to prioritize and discuss testing some parameters in TweakReg, namely…
  1. Using a template header=refimage
  2. Using a catalog for alignment=catfile, refcat, refxyunits, catalog , etc
  3. Getting all images rotated and cropped the same way. (?)
  4. For more (minobj, searchrad, tolerance, separation, fitgeometry), see: <https://drizzlepac.readthedocs.io/en/latest/tweakreg.html>

**Image processing packages in Python**:

* In general,
  + The main goal is to take a given frame from a database (e.g. MAST for HST) and align it with fixed reference points (stars). Many algorithms get confused by nebulosity, scattered/reflected light, etc.
  + Another goal is to mosaic or transform images so to overlap, blink, and compare
  + Yet another goal is to make this process as precise as possible and to eliminate any bright light sources that are irrelevant (e.g. cosmic rays, satellite trails).
* Drizzlepac (by STScI)
  + In theory, this should do everything we need.
  + Download drizzlepac
    - First, you’ll need to download drizzlepac. For that, I recommend adding a clean anaconda environment or downloaded miniconda environment. Set it up naming it “Astroconda” following: <https://astroconda.readthedocs.io/en/latest/> and/or <https://astroconda.readthedocs.io/en/latest/getting_started.html>
    - After you have astroconda, it should come with drizzlepac. Check this.
  + Setup files (see hst\_download.py)
    - Next, setup your directories. See the part of my script using astroquery. You should have astroquery and be able to query databases with the code.
    - Move files to main directory and copy them over as needed. This allows you to tweak and experiment.
  + From there, we have many options in Tweakreg to do any or all of:
    - Initial functions, such as footprints and RA/Dec uncertainty estimates.
    - We can align to a catalog (see align\_default.py)
    - Align to a sparse field (align\_sparse\_fields below, mostly Tweakreg parameters)
    - And align multiple visits together (generally I do this no matter what)
  + For more information, there are many examples, such as
    - <https://github.com/spacetelescope/notebooks/tree/master/notebooks/DrizzlePac> (see: align\_multiple\_visits, align\_sparse\_fields, and align\_to\_catalogs)
    - GAIA: <https://github.com/spacetelescope/gaia_alignment/blob/master/Gaia_alignment.ipynb>
* Montage (by IPAC)
  + It can be installed simply with pip install MontagePy
  + I use this to rotate, crop, and give all frames the same WCS. Only because I haven’t found a way to find the best template header/WCS AND implement that with drizzlepac.
  + <http://montage.ipac.caltech.edu/> -> <http://montage.ipac.caltech.edu/docs/algorithms.html> for understanding and <http://montage.ipac.caltech.edu/MontageNotebooks/> for implementation.
  + I use mHdr or mGetHdr and mProjExec. This uses a header template to reproject the image. You can invoke the drizzle algorithm and conserve flux as well.
* By hand (not recommended)
  + Many softwares to do it with (e.g. OpenCV, AstroImageJ, scikit-image, etc). Might help if we need 1 pixel offset or initial fixes, but not our primary method.