Things to do before the consortium meeting

SimCADO

1. Review/Work on all the SimCADO I/O stuff
    1. Saving/reading detector output
    2. Saving/reading objects: `UserCommands`, `Detector`, `OpticalTrain`, `Source`
2. Add own FFT/iFFT convolution algorithm - astropy is bloated
3. Add astropy units for within functions (to make sure functions are internally consistant)
4. Review/define units for the interfaces between functions
5. Add Logging
6. Add galaxy spectra, like the pickles spectra

Validation

1. Check the Photometry of objects
2. Check the background photon levels
3. Work out if the surface brightness of objects is realistic

Documentation

1. Update README.md

2. Write up many examples for generating `Source`-objects
    1. From arrays
    2. From images + spectra
        - From FITS image plus ASCII
        - From FITS image plus FITS table
        - From python generated image + blackbody
3. Examples for running Simulation

Quick simulations - `simcado.simulation.run()`

1. Creating a `Source` and then putting it though `.run()`
2. Adding kwargs to `.run()`
3. Creating and passing a `UserCommands`-object
4. Creating and passing custom `Detector` and `OpticalTrain`-objects

Full simulation - `apply_optical_train()` + `detector.read_out()`

Writing tests for the modules (in order of priority)

1. source.py
2. simultion.py
3. optics.py
4. detector.py
5. commands.py

---

6. spatial.py
7. psf.py
8. spectral.py

Generate some example `Source`-objects

Would be nice to include the code which creates these example `Source`-objects into `source.py`

Point source

1. Single star (FoV single chip, single spectrum)
2. Cluster (FoV 1 chip, many spectra)
3. Cluster (FoV 9 chips, many spectra)

Extended source

1. Nebula (FoV 1 chip, 1 spectrum)
2. Elliptical galaxy (FoV 1 chip, 1 spectrum)
3. Elliptical galaxy (FoV 9 chips, 1 spectrum)

4. Spiral Galaxy (Fov 1 chip, 2 or 3 spectra)
5. Spiral Galaxy (Fov 9 chips, 2 or 3 spectra)

Combined source

1. Star forming region with YSO point sources and surrounding nebula (FoV 1 chip, 2 spectra)
2. Spiral galaxy with bright stars, SF regions, old population (FoV 9 chips, 3 spectra)

Realistic full sky source

1. foreground stars, spiral galaxy with bright stars and star formation regions, globular clusters around galaxy, redshifted background galaxies, very bright stars, zodiacal light(?)

This document houses the list of things that need to be implemented or ideas which could be implemented in the future

Things to be done in the future

- `Detector`
  - rename the module `detector`
  - Detector.open move method into an unbound function, returning a `Detector` object
  - **BUG** - it calls `self.params` which is out-dated
  - **Potential BUG** - in `Chip._read_noise_frame()` we

need to make sure we are prepared for the case that `self.array.shape` > the shape of the noise FITS file

- `OpticalTrain`
    - rename the module `optical_train`
    - The `read` method should be renamed `read_optical_train` and should be turned into an unbound function in the OpticalTrain module
- Unify the names of the read and write methods for each class
- Change the structure of the `TransmissionCurve` and `EmissionCurve` to replace the `params` dict actual attributes and arguments in `__init__()` E.g. exp_time, pix_res, …
- Add a `FilterSet` class to `SpectralClass` that holds all the filters contained by the filter wheels
- add in the ability to define optical elements which call specific functions and thereby allow the user to define an optical train