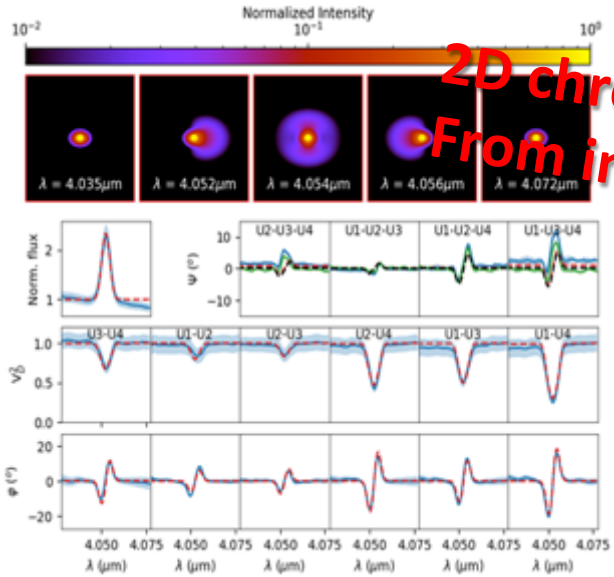
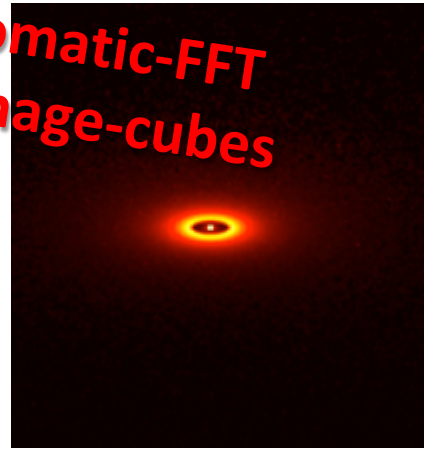
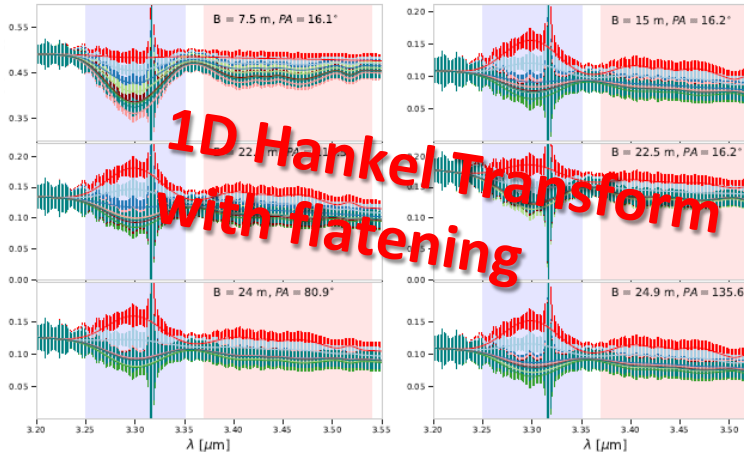
Modeling discussion in MATISSE consortium

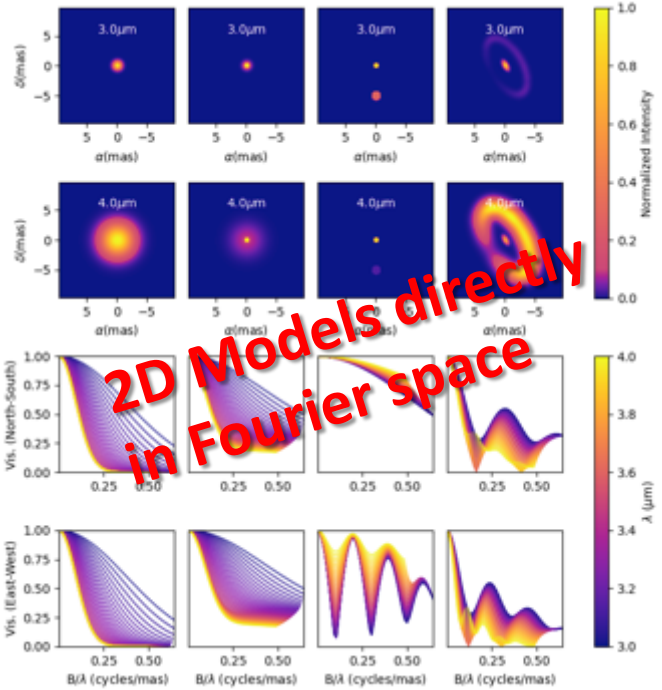


Rotating disk in Brα (Meilland+)



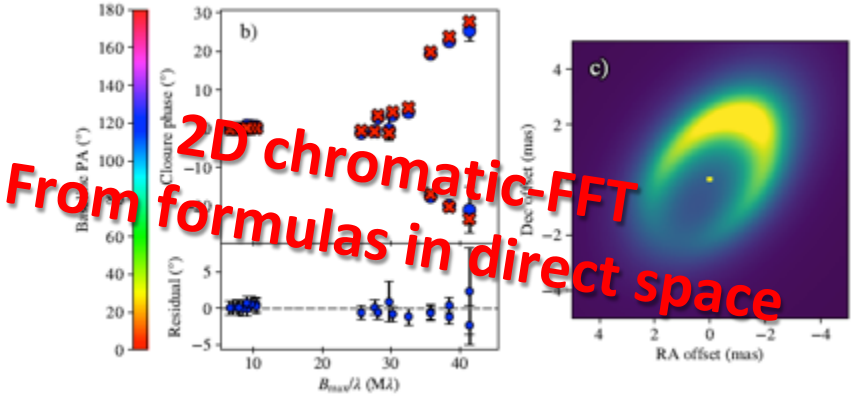
Radiative Transfer
(HDUST grid)



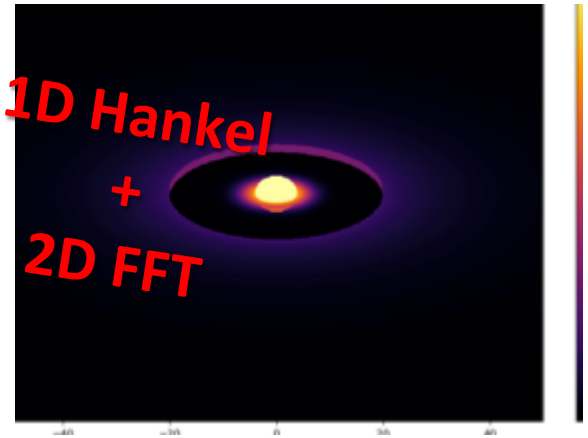


Temperature-Gradient disk + 3.3µm PAH emission template (Kokoulina+ 2021)

Simple chromatic analytic models
(Meilland+)



2D skewed Ring model + chromaticity (Vargas+ 2020)



Semi-physical B[e] model
DISCO+
(Domiciano de Souza+)

Can we create a single model-fitting framework for all of that?

- **Versatile**
- **Fast**
- **Expendable**
- **User friendly**
- **Scriptable**

Desired features :

•Python3 (the language we all use) & open-source (+ Git repository)

•provide a good set of simple models:
- Gaussian, Lorentzian, ring, uniform disk, point source, temperature gradient, azimuthal asymmetry

•Modularity (add up components)
- to build more complex geometries (e.g., star + disk + companion)

•Ability to input external codes/models/images
- e.g., load pre-computed grid of images (e.g. from RT models)
- Encapsulates fast external models (semi-physical, kinematics…)

•Expandability (put your own code at this X place!)
- object-oriented code

•Chromatic
- fit multi-band multi-wavelengths data-sets
- temperature-dependence (+ opacity)
- Kinematics in lines

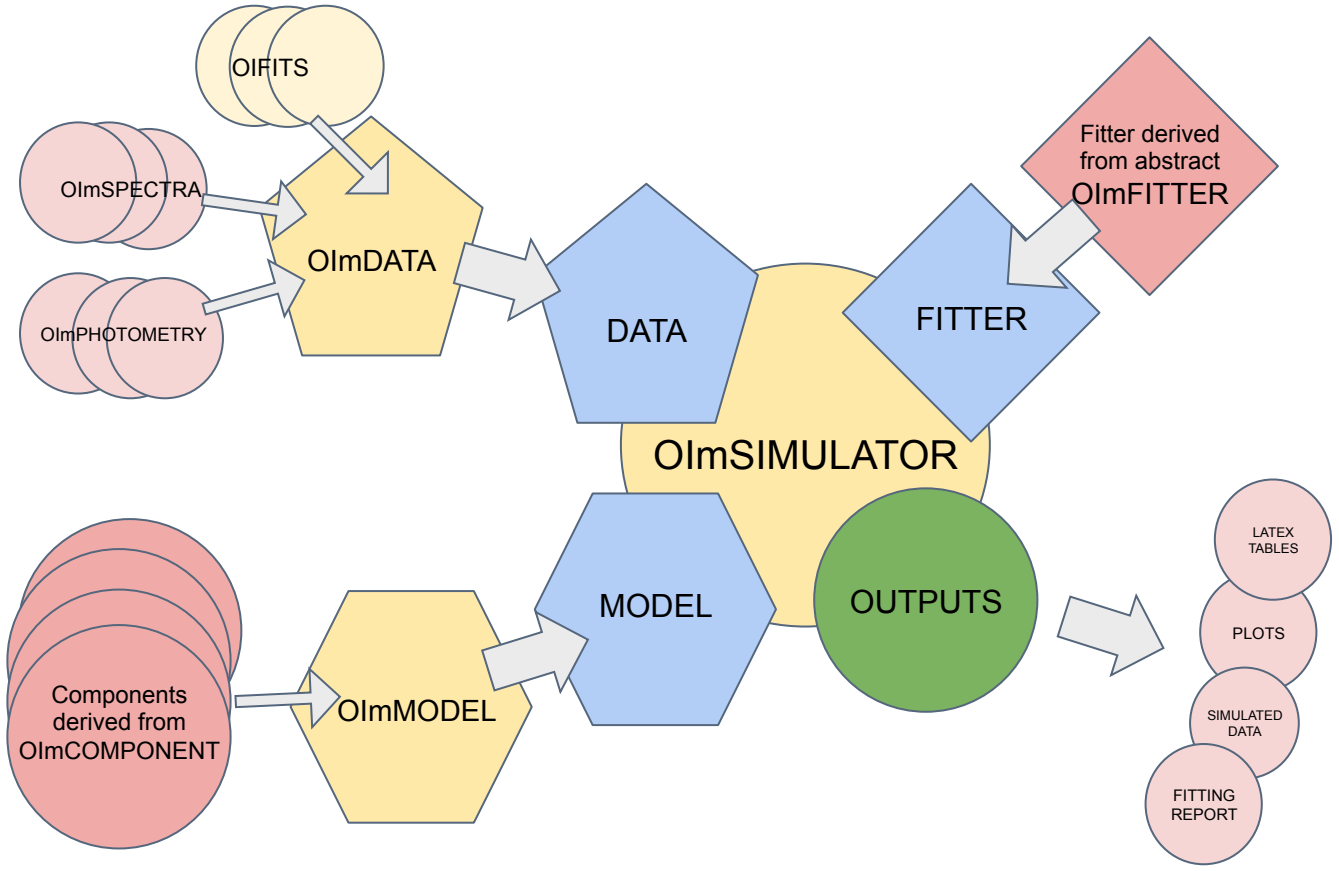
•Allow time-variable models
- e.g. (sub)stellar companion, orbiting disk clump

•Accept various inputs: OIFITS, photometry, spectra, etc.

•Allow different fitting algorithms
- including Markov Chain Monte Carlo (MCMC)
- people can implement their own fitters

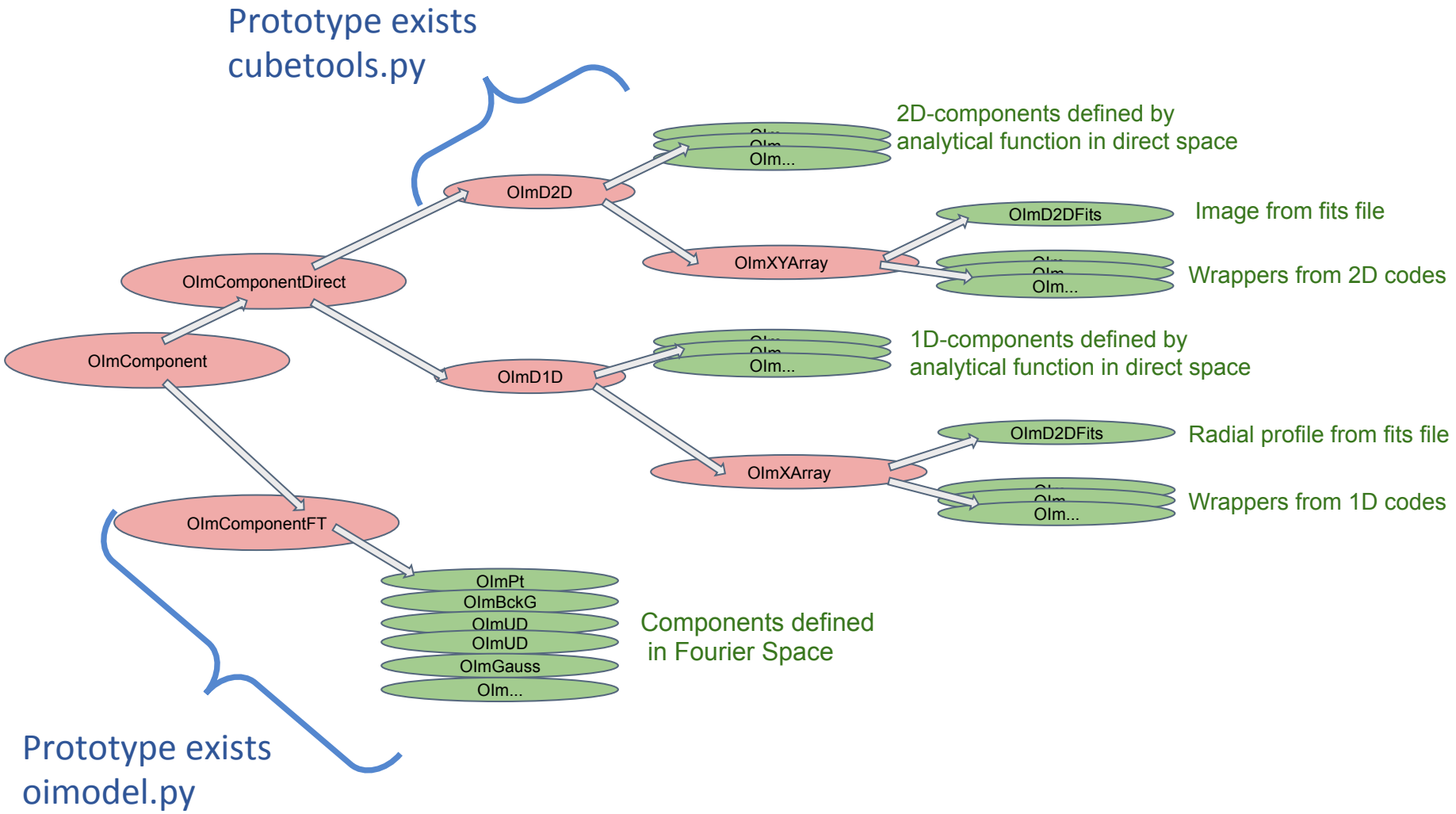
•Produce quality plots
- e.g., data with best-fit model, residuals, model image, MCMC posterior distributions

A modular object-based tool for versatility & expandability

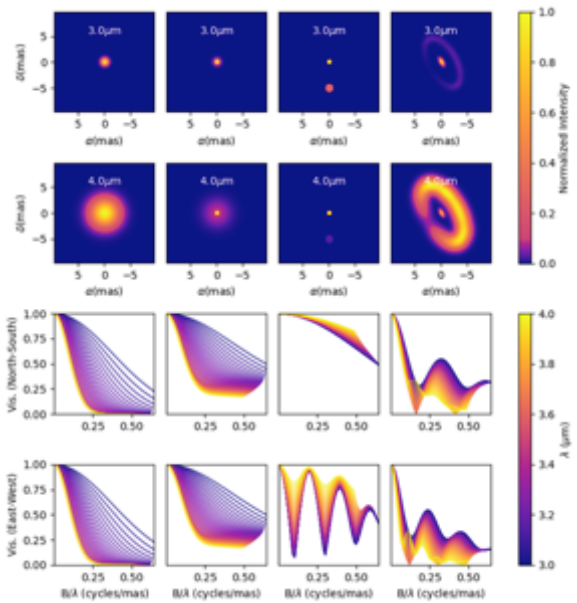


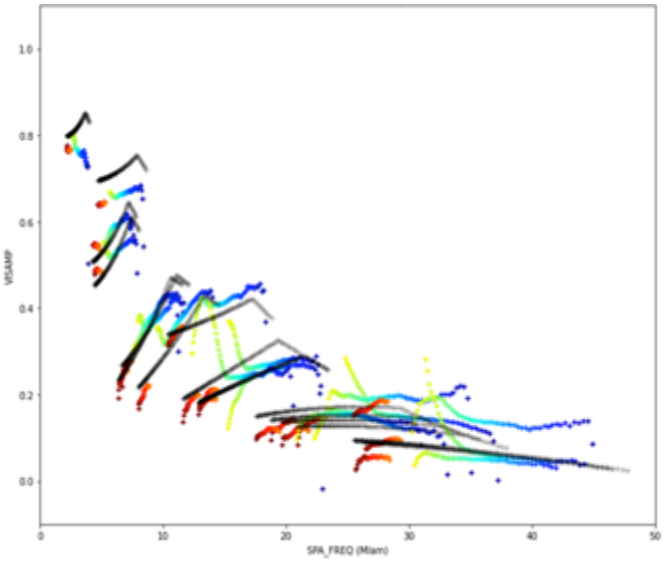
A modular object-based tool for versatility & expandability

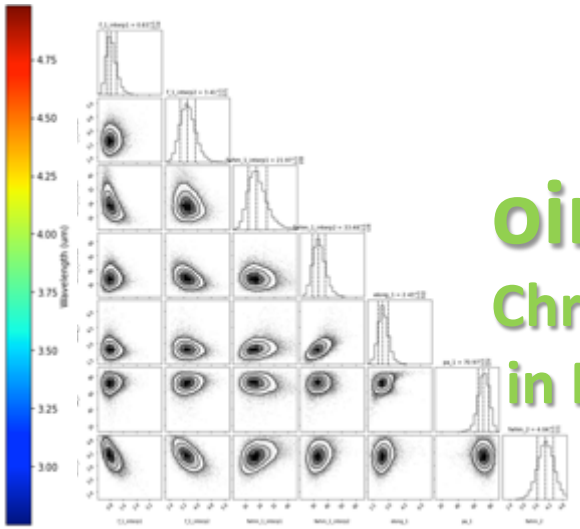
oimComponent inheritance



Merging two existing prototypes



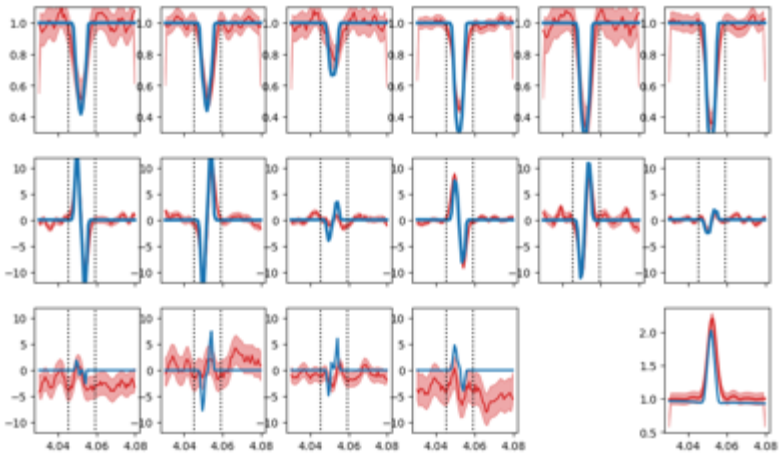




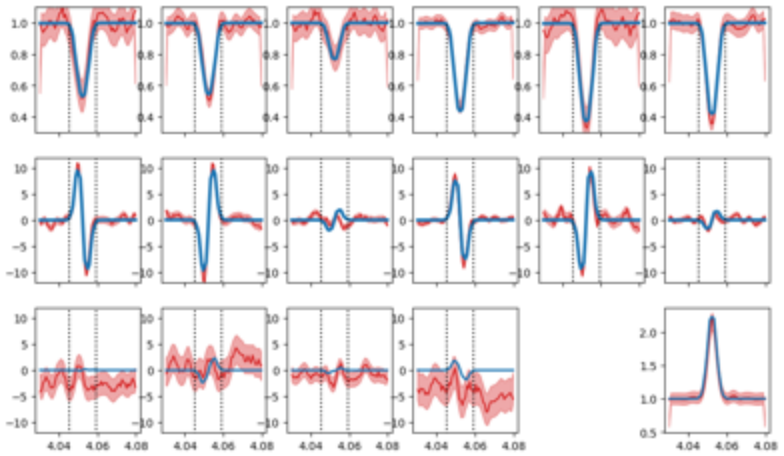
oimodel.py

Chromatic analytic models in Fourier space

grid of models (HDUST)

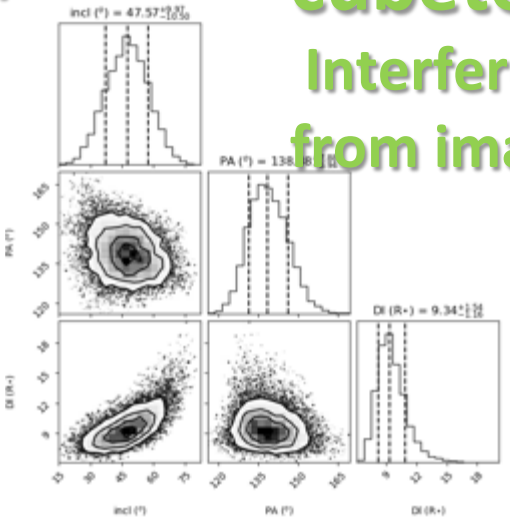


semi-physical models (BeToy)



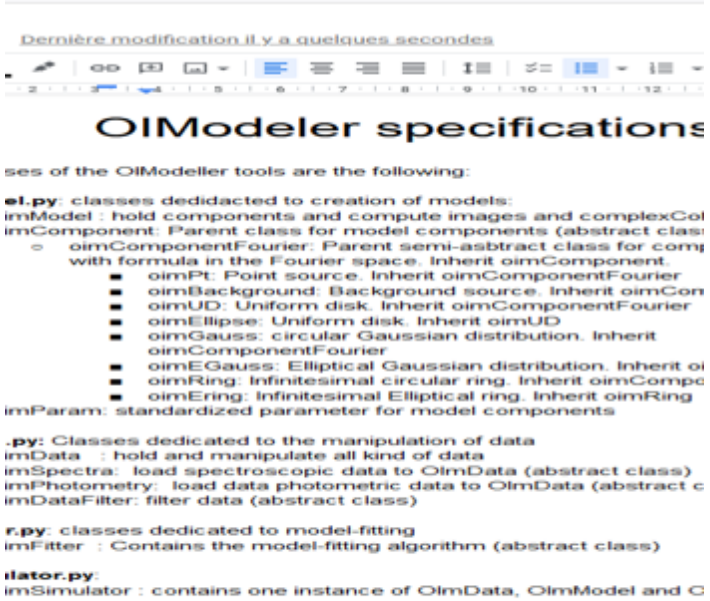
cubetools.py

Interferometric data from image cubes



OiModeler project

oiModeler Specification


Dernière modification il y a quelques secondes

OIModeler specifications

ses of the OIModeller tools are the following:

el.py: classes dedidacted to creation of models:
imModel : hold components and compute images and complexCol
imComponent: Parent class for model components (abstract clas
- oimComponentFourier: Parent semi-asbtract class for comp
  with formula in the Fourier space. Inherit oimComponent.
  - oimPt: Point source. Inherit oimComponentFourier
  - oimBackground: Background source. Inherit oimCon
  - oimUD: Uniform disk. Inherit oimComponentFourier
  - oimEllipse: Uniform disk. Inherit oimUD
  - oimGauss: circular Gaussian distribution. Inherit oimComponentFourier
  - oimEGauss: Elliptical Gaussian distribution. Inherit oi
  - oimRing: Infinitesimal circular ring. Inherit oimCompo
  - oimEring: Infinitesimal Elliptical ring. Inherit oimRing

imParam: standardized parameter for model components

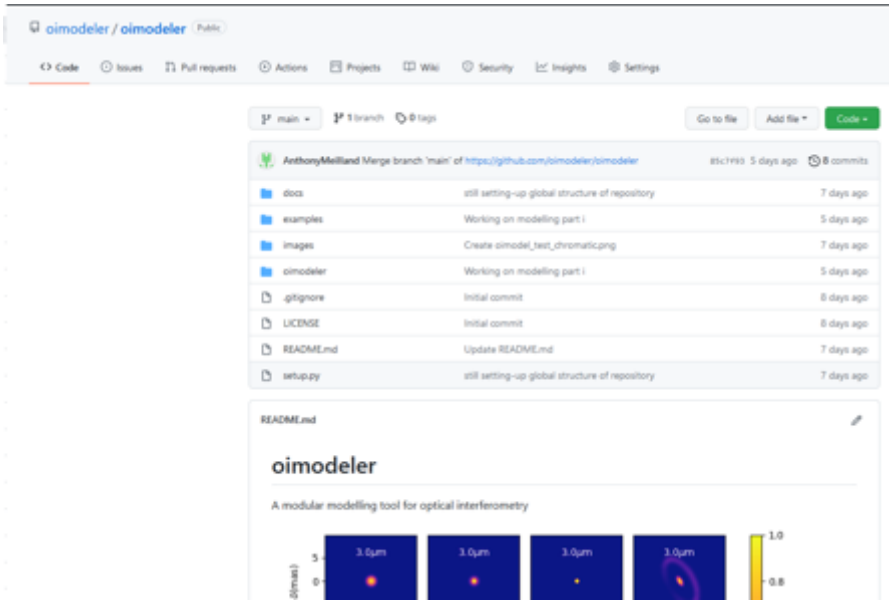
.py: Classes dedicated to the manipulation of data
imData : hold and manipulate all kind of data
imSpectra: load spectroscopic data to OImData (abstract class)
imPhotometry: load data photometric data to OImData (abstract c
imDataFilter: filter data (abstract class)

r.py: classes dedicated to model-fitting
imFitter : Contains the model-fitting algorithm (abstract class)

ilator.py:
imSimulator : contains one instance of OImData, OImModel and C

https://docs.google.com/document/d/1h0HPc_R4EE1xnGH_eaIDdsRkhUrytdFX9EdIncti4VA

oiModeler on Github



https://github.com/oimodeler/oimodeler

Open-source, easily expandable, OI modelling software
Working prototype Q1 2022
Interest outside MATISSE community? AMHRA?

TODO Before the end of the year

- Finish specification
- Make full code skeleton
- Implement :
  - analytical models
  - chromatic parameters
  - FT from images
  - HT from radial profile
  - Data class from Oifits
  - First look at fitter (mcmc)