SkyPy

Lucia F. de la Bella

And the SkyPy Collaboration

ICG Portsmouth

UNIVERSITY OF PORTSMOUTH

7. Configuration Files

1. Vision

6. Simulation pipelines

2. Members

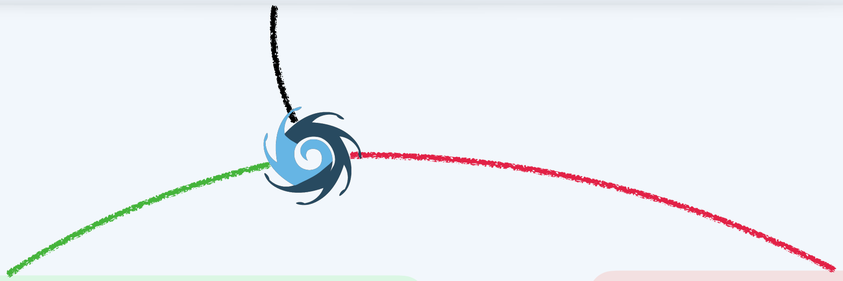
5. Research & Development

3. Community Package

4. The Library

1. The vision

- Observational cosmology limited by data access
- Open-data revolution in astronomy
- Challenge: access to sophisticated analysis **methods**.
- Emerging methods: forward modelling & machine learning.



- III generation of catalog production (**user-generated outputs**)
- **Open-source** off-project high-quality **Python** package
- **End-to-end simulations** of the astrophysical sky
- Interface with external software
- Enable **Forward Modelling** and **Machine Learning**

- Not a single pipeline simulation
- Do not replicate existing code

- **Reuse**
  - Astropy-affiliated packages
  - High-quality codes
- **Ecosystem** of compatible software

2. Members

Sarah Bridle

Juan Pablo Cordero

Ian Harrison

Laura Wolz

Richard Rollins

Nicolas Tessore

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

THE UNIVERSITY of EDINBURGH

UCL

MANCHESTER 1824

Adam Amara

Lucia F. de la Bella

Philipp Sudek

Ginevra Favole

Arthur Tolley

Coleman Ktawczyk

Ian Harry

Laura Nutall

Andrew Lundgren

Andrew Williamson

ICG Portsmouth

Brian Nord

Simon Birrer

Fermilab

Stanford University

Keiichi Umetsu

Sut-ieng Tam

https://skypyproject.org

3. Community Package

https://github.com/skypyproject/skypy.git

LIBRARY

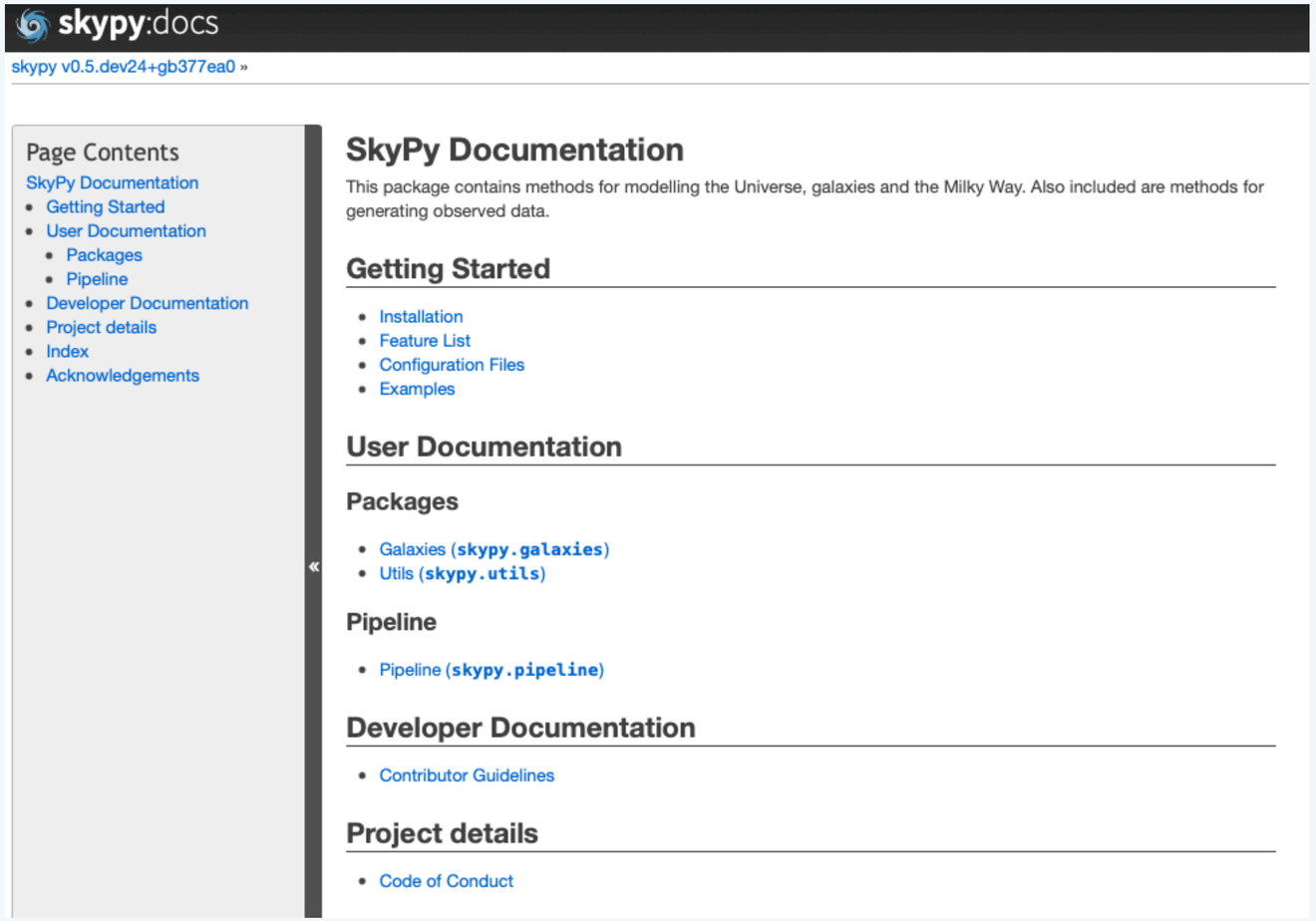
Astrophysical models
Research & Development

INFRASTRUCTURE

Functionality for
end-to-end sims
Enable methods for
open-access data

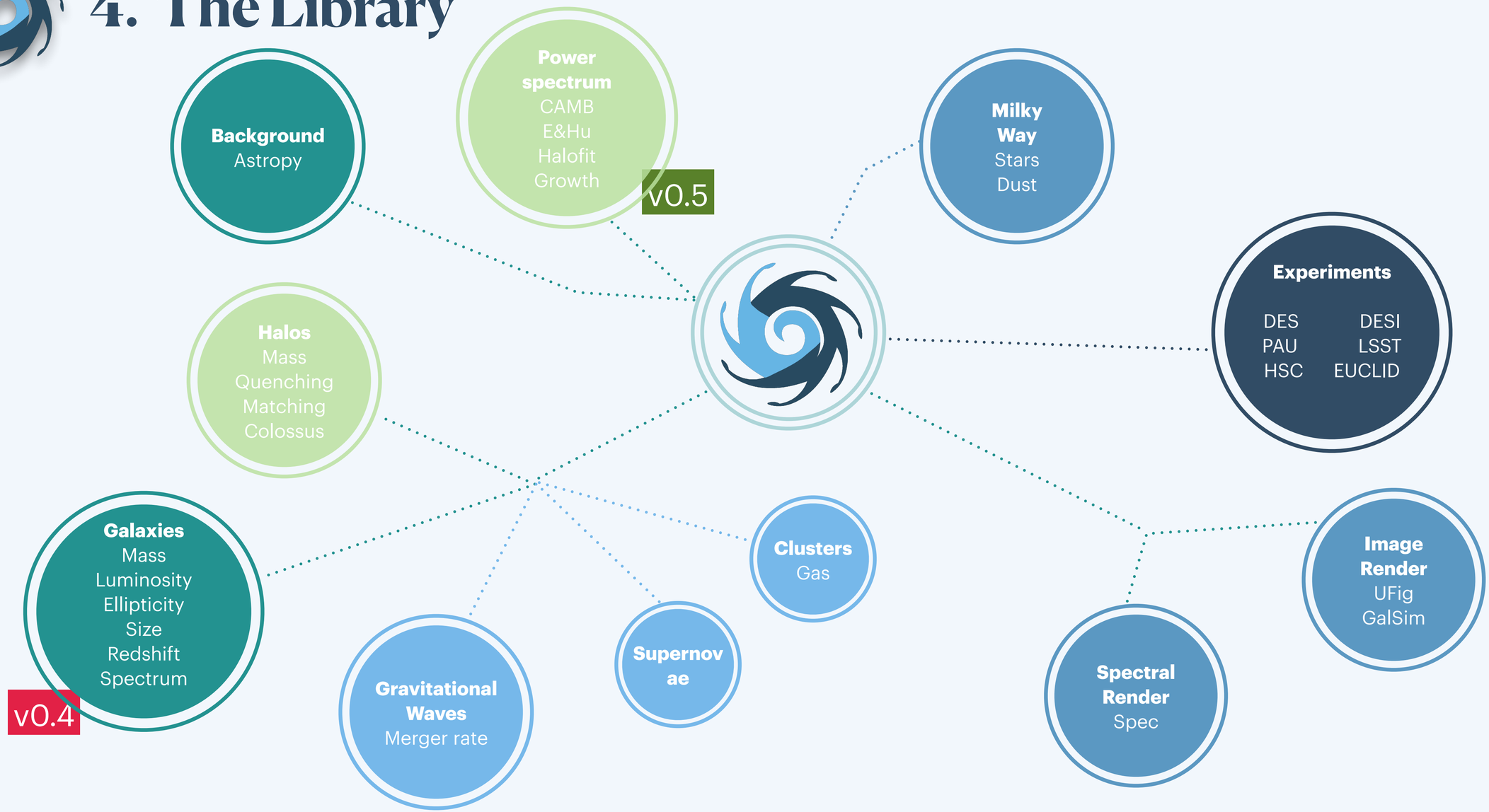
- **GitHub** organisation
- Unit tests & high-quality documentation
- Code review & Infrastructure team

```
my-pc: -$ pip install skypy or
my-pc: -$ conda install -c conda-forge skypy or
my-pc: -$ git clone https://github.com/skypyproject/skypy.git

my-pc: -$ ipython
…
[1]: import skypy
```



https://skypy.readthedocs.io/en/latest

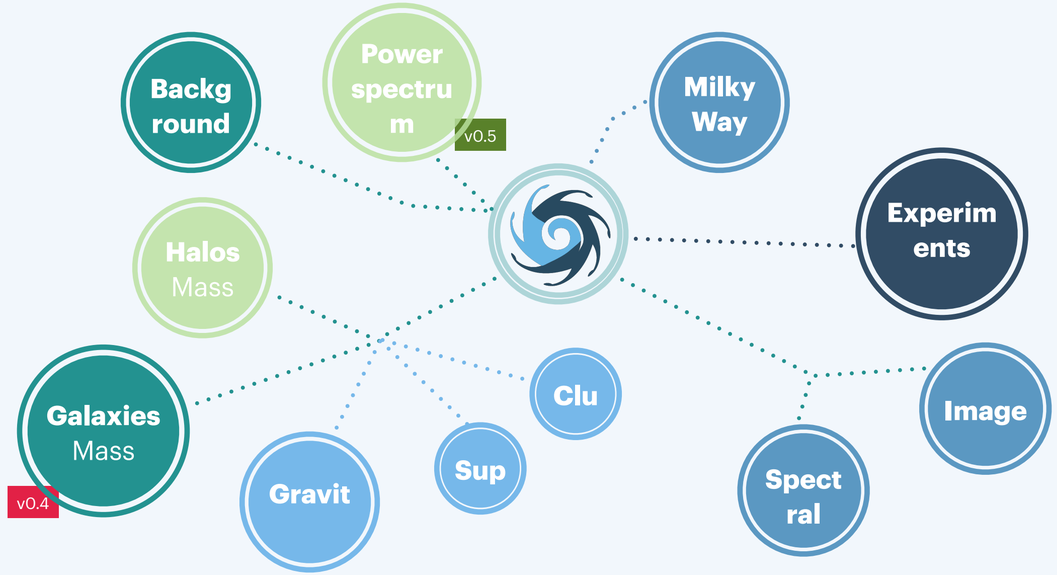
4. The Library



5. Research & Development

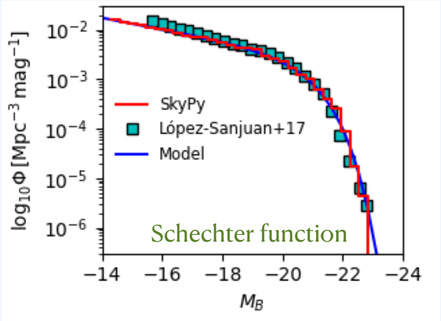
SkyPy **v0.4**

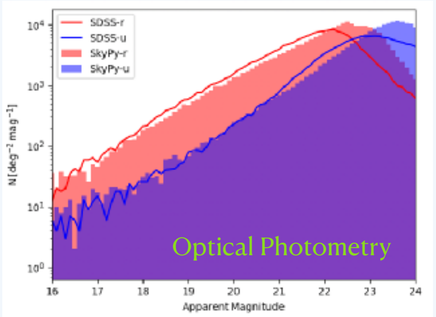
- **Luminosity Distributions** — Schechter Luminosity function
- **Morphological Distributions** — angular size, (early- and late-type) linear lognormal size distribution, beta ellipticity and Ryden 2004 ellipticity distributions.
- **Redshift Distributions** — redshifts from co-moving density, Schechter (luminosity and stellar mass) redshift distribution, Smail+94 redshift distribution.
- **Spectral Energy Distribution Modelling** — Dirichlet coefficients, Correct templates.
- **Stellar Mass Distribution** — Schechter stellar mass function.

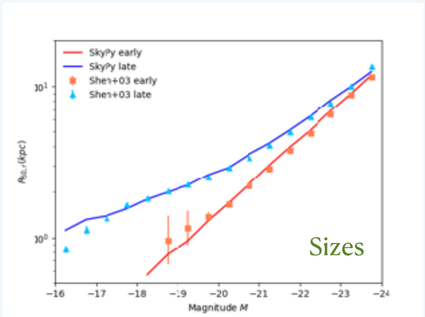


SkyPy **v0.5**

- **Power Spectrum** — CAMB, Halofit, CLASS, Eisenstein & Hu, growth functions
- **Dark Matter Halos** — Colossus, halo and sub-halo mass sampler, ellipsoidal and spherical collapse functions (Press-Schechter, Sheth-Tormen), abundance matching, quenching models







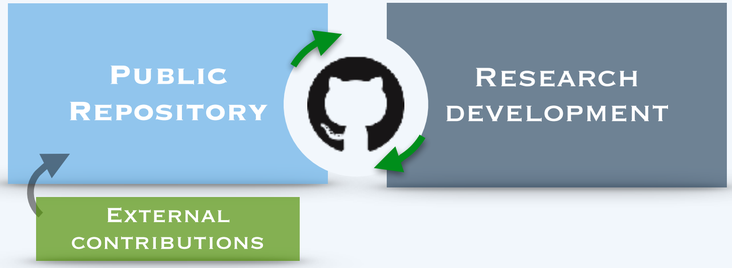
https://skypy.readthedocs.io/en/stable

https://github.com/skypyproject/skypy.git

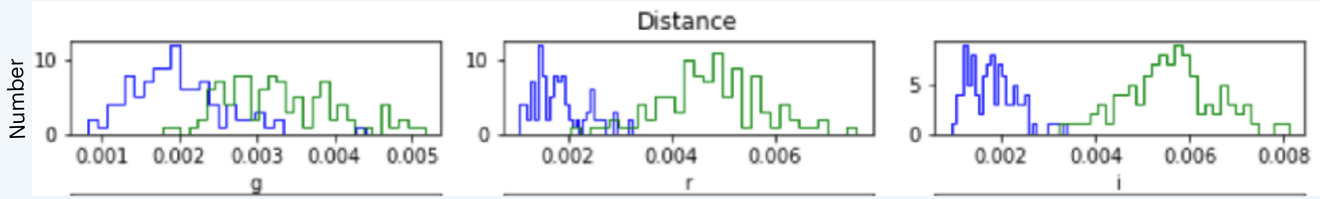
5. Research & Development

Key

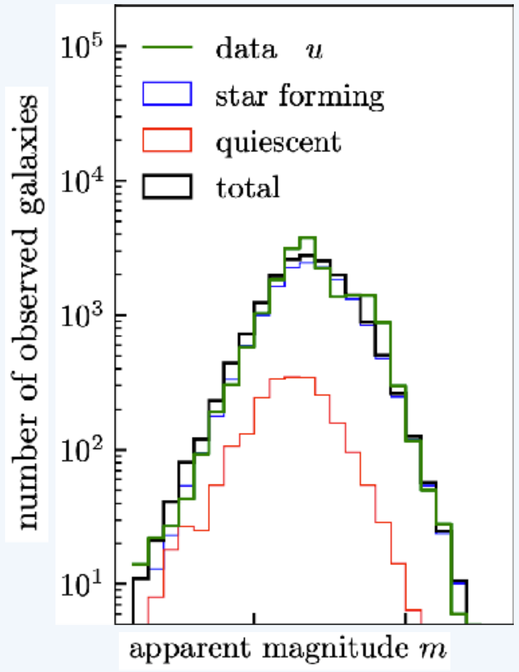
SkyPy is driven by science projects



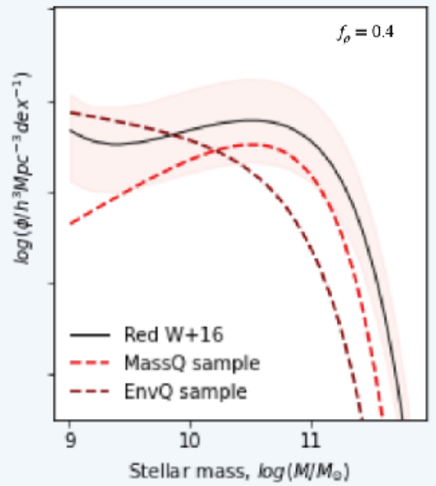
- ***Likelihood-Free Inference for Cluster Weak Lensing*** - Sut-Ieng Tam, ASIAA
- ***Projected Galaxy Ellipticities*** - Juan Pablo Cordero, Univ. of Manchester
- ***From Quenching to the Schechter Function*** - Lucia F. de la Bella, Univ. of Portsmouth
- ***Forecasting Optical Galaxy Surveys***- Philipp Sudek, Univ. of Portsmouth
- ***Galaxies*** - Nicolas Tessore, UCL
- ***Gravitational Wave Binary Merger Populations*** - Arthur Tolley, Univ. of Portsmouth



Investigation of Schechter parameter sensitivity of a DES-like survey (Sudek+ in prep).
Big difference of the green and blue histogram indicates high constraining power using the corresponding observable



Apparent magnitude distribution in the SDSS u filter simulated with SkyPy (blue, red, black) compared to SDSS data (Tessore+ in prep.)



Schechter function in the quenching model vs SDSS best fit (de la Bella + in prep.)

6. Simulation pipelines

Key

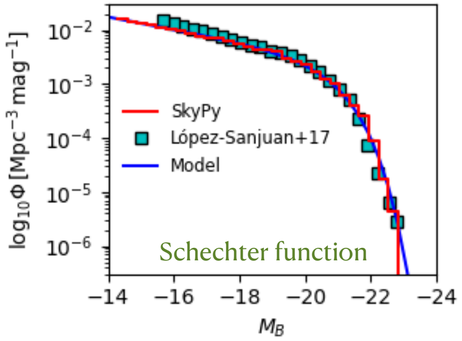
- YAML-based config files
- The **SkyPy Driver** runs end-to-end **pipelines**
- **Total flexibility**!

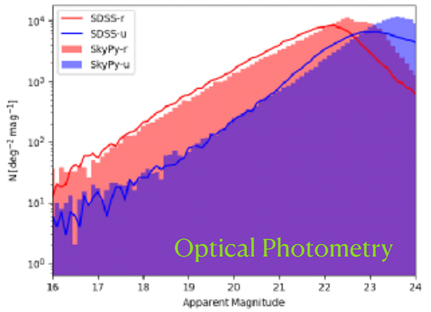
- SkyPy Pipeline
- **KEY: you** can write your own **pipelines!**

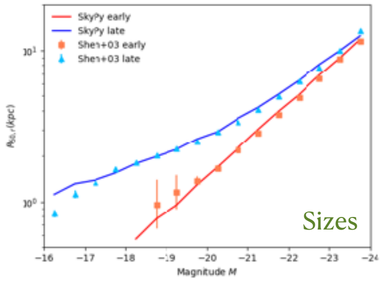
Combine SkyPy with your favourite software!

Simulated lensed galaxies using *SkyPy* and *lenstronomy* (Simon Birrer)

SkyPy PIPELINES







https://github.com/skypyproject/skypy.git

https://skypy.readthedocs.io/en/latest/examples/index.html

7. Configuration files

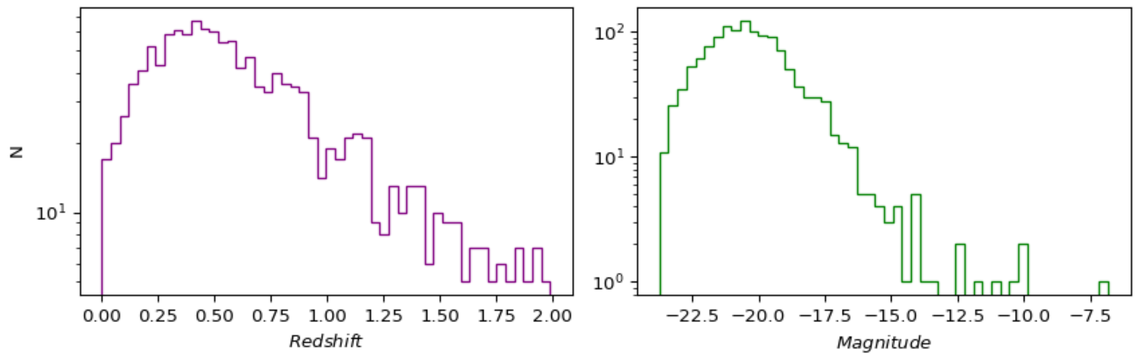
SkyPy Syntax

- **Variables** — Astropy quantities, import objects
- **Parameters** — variables modified at execution
- **Functions** — cosmology, job completion
- **Tables** — multicolumn assignment, table reference

Example: luminosity.yml

```yaml
cosmology: !astropy.cosmology.default_cosmology.get []
z_range: !numpy.linspace [0, 2, 21]
M_star: !astropy.modeling.models.Linear1D [-0.9, -20.4]
phi_star: !astropy.modeling.models.Exponential1D [3e-3, -9.7]
magnitude_limit: 23
sky_area: 0.1 deg2
tables:
  blue_galaxies:
    redshift, magnitude: !skypy.galaxies.schechter_lf
      redshift: $z_range
      M_star: $M_star
      phi_star: $phi_star
      alpha: -1.3
      m_lim: $magnitude_limit
      sky_area: $sky_area
```

```python
import matplotlib.pyplot as plt
from skypy.pipeline import Pipeline

# Execute SkyPy luminosity pipeline
pipeline = Pipeline.read("luminosity.yml")
pipeline.execute()

# Blue population
skypy_galaxies = pipeline['blue_galaxies']

# Plot histograms
fig, axs = plt.subplots(1, 2, figsize=(9, 3))

axs[0].hist(skypy_galaxies['redshift'], bins=50, histtype='step', color='purple')
axs[0].set_xlabel(r'$Redshift$')
axs[0].set_ylabel(r'$\mathrm{N}$')
axs[0].set_yscale('log')

axs[1].hist(skypy_galaxies['magnitude'], bins=50, histtype='step', color='green')
axs[1].set_xlabel(r'$Magnitude$')
axs[1].set_yscale('log')

plt.tight_layout()
plt.show()
```



You can also run the pipeline directly from the command line and write the outputs to a fits file:

```
$ skypy luminosity.yml luminosity.fits
```

https://skypy.readthedocs.io/en/latest/configuration_files.html

7. Configuration files

SkyPy Syntax

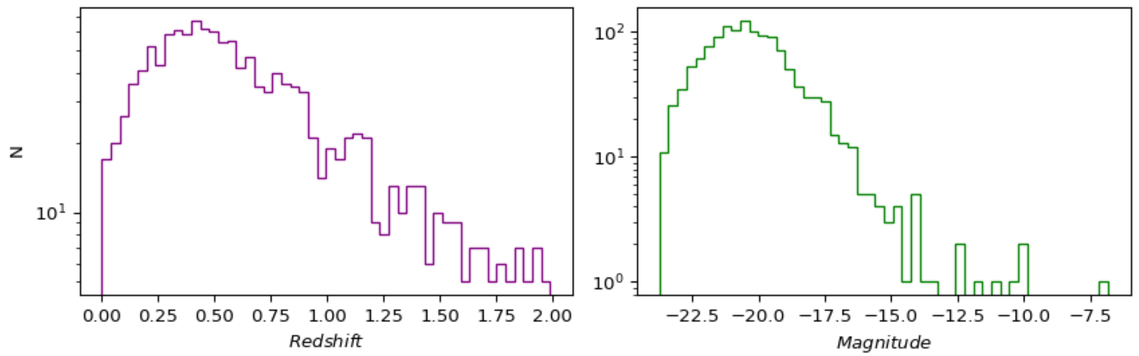
- **Variables** — Astropy quantities, import objects
- **Parameters** — variables modified at execution
- **Functions** — cosmology, job completion
- **Tables** — multicolumn assignment, table reference

Example: luminosity.yml

```yaml
cosmology: !astropy.cosmology.default_cosmology.get []
z_range: !numpy.linspace [0, 2, 21]
M_star: !astropy.modeling.models.Linear1D [-0.9, -20.4]
phi_star: !astropy.modeling.models.Exponential1D [3e-3, -9.7]
magnitude_limit: 23
sky_area: 0.1 deg2
tables:
  blue_galaxies:
    redshift, magnitude: !skypy.galaxies.schechter_lf
      redshift: $z_range
      M_star: $M_star
      phi_star: $phi_star
      alpha: -1.3
      m_lim: $magnitude_limit
      sky_area: $sky_area
```

https://skypy.readthedocs.io/en/latest/configuration_files.html

```python
import matplotlib.pyplot as plt
from skypy.pipeline import Pipeline

# Execute SkyPy luminosity pipeline
pipeline = Pipeline.read("luminosity.yml")
pipeline.execute()

# Blue population
skypy_galaxies = pipeline['blue_galaxies']

# Plot histograms
fig, axs = plt.subplots(1, 2, figsize=(9, 3))

axs[0].hist(skypy_galaxies['redshift'], bins=50, histtype='step', color='purple')
axs[0].set_xlabel(r'$Redshift$')
axs[0].set_ylabel(r'$\mathrm{N}$')
axs[0].set_yscale('log')

axs[1].hist(skypy_galaxies['magnitude'], bins=50, histtype='step', color='green')

plt.tight_layout()
plt.show()
```



You can also run the pipeline directly from the command line and write the outputs to a fits file:

```
$ skypy luminosity.yml luminosity.fits
```

Summary

COMMUNITY PACKAGE

- **Open-source** off-project
- High-quality **Python** package

LIBRARY

Astrophysical models
Research & Development

INFRASTRUCTURE

Functionality for end-to-end sims
Enable methods for open-access data

- **GitHub** organisation
- Unit tests & high-quality documentation
- Code review & Infrastructure team

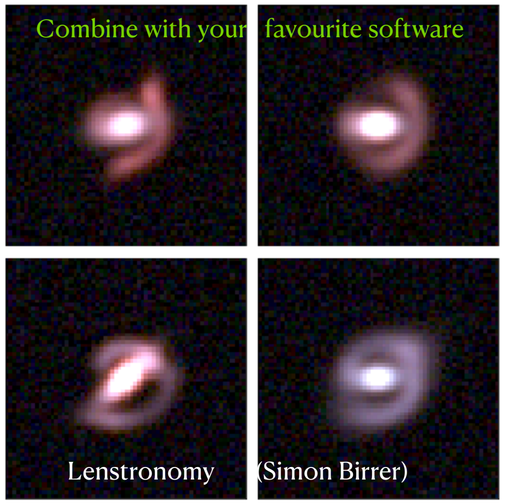
Next

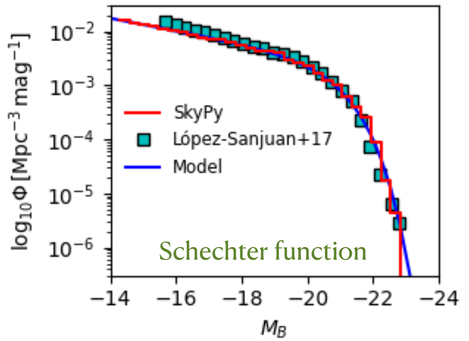
- **v0.5** release: halo & power spectrum.
- Journal of Open-Source Software
- Equality, Diversity and Inclusion projects

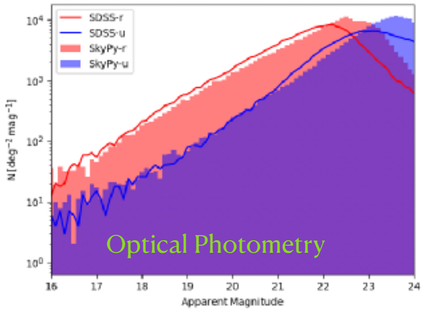
SIMULATION PIPELINES

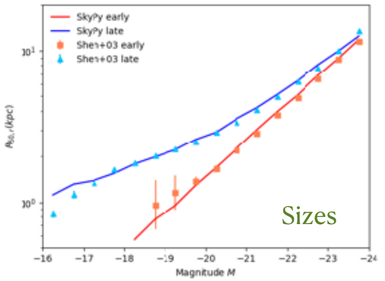
- YAML-based config files
- The **SkyPy Driver** runs end-to-end **pipelines**
- **Total flexibility**!

- SkyPy Pipeline
- **KEY: you** can write your own **pipelines!**









https://github.com/skypyproject/skypy.git

https://skypyproject.org

https://skypy.readthedocs.io/en/latest/examples/index.html

Open your terminal...

```
my-pc: -$ pip install skypy or
my-pc: -$ conda install -c conda-forge skypy or
my-pc: -$ git clone https://github.com/skypyproject/skypy.git

my-pc: -$ ipython
…
[1]: import skypy
```