COSMORATE

The user guide

download

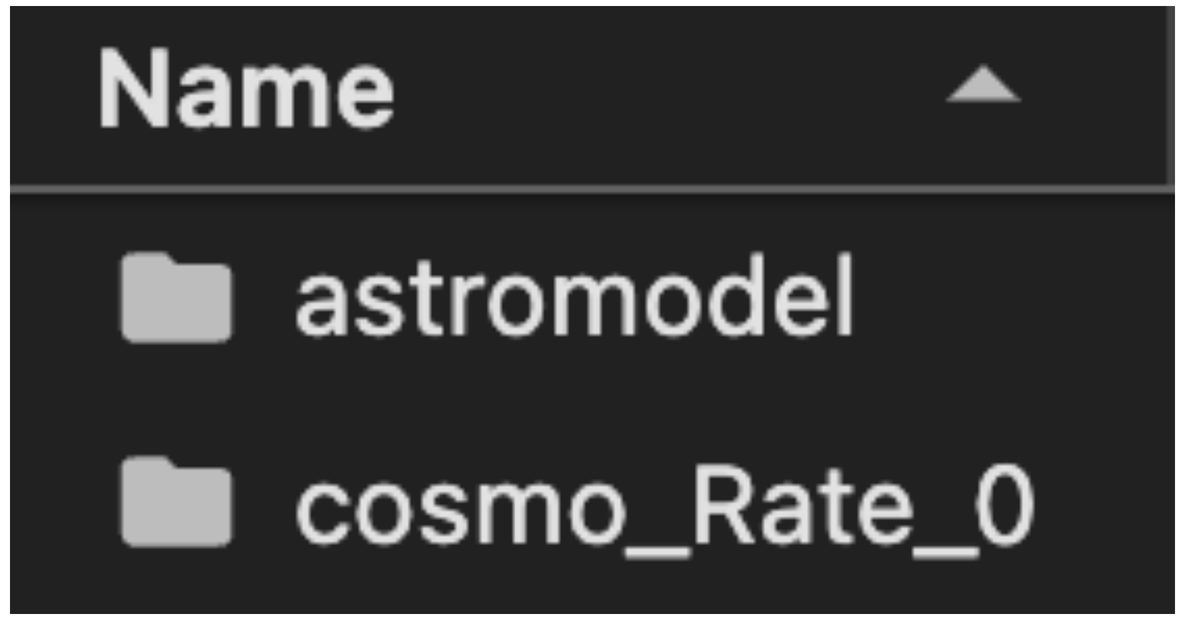
- `cosmoRate:`
  - https://gitlab.com/Filippo.santoliquido/cosmo_rate_public

Installation

- To be sure `cosmoRate` works on your laptop, we need to create a conda environment including all the necessary libraries:
  - `conda env create -f environment.yml`
  - `conda activate cosmo_env`
- The following command instals the conda environment on Jupyter lab as well:
  - `python -m ipykernel install --user --name=cosmo_env --display-name “cosmo_env”`

What you get with `cosmoRate`

This is the folder where we store the astrophysical models (e.g. from population-synthesis) and the outputs



This contains the Python scripts

A detailed explanation on input and output follows

- **cosmoRate** is made of three Python scripts

- Here we focus on learning how to use **cosmo_params.py**

Level 1: How to run `cosmoRate`

- Two ways:
    - directly from the jupyter notebook **`cosmo_rate_notebook.ipynb`**
    - or by launching **“`ipython cosmo_rate_notebook.py`”** from the shell

Level 2: How to modify `cosmo_params.py`

- Long list of parameters… most of them might not immediately clear to you
- That's why we need to have a look at **WHAT** `cosmoRate` does

COSMORATE

Santoliquido et al. 2020:
https://arxiv.org/pdf/2004.09533.pdf

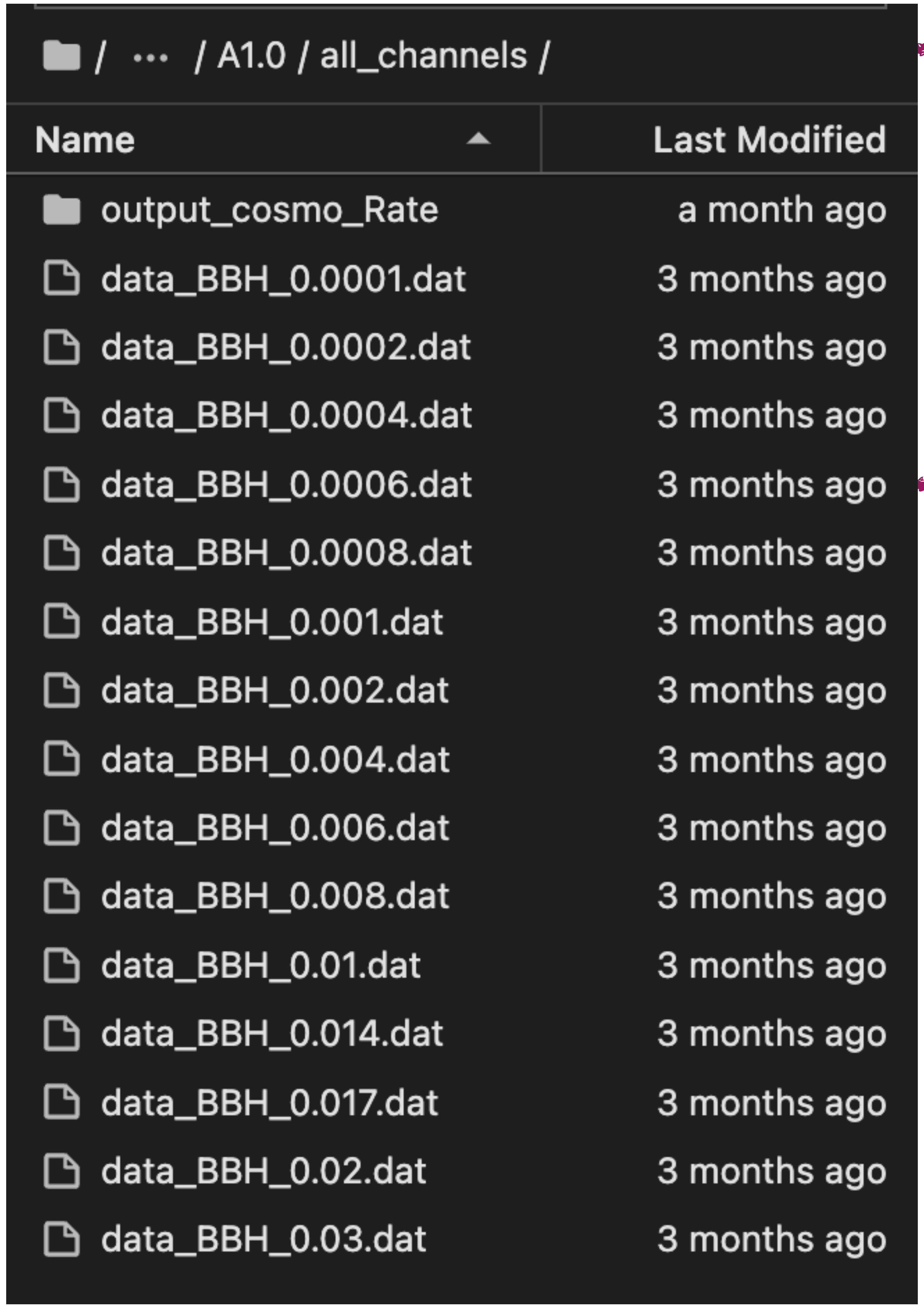
$$\mathcal{R}(z) = \int_{z_{\max}}^{z} \left[\int_{Z_{\min}}^{Z_{\max}} \boxed{\mathrm{SFRD}(z', Z)} \boxed{\mathcal{F}(z', z, Z)} \mathrm{d}Z \right] \frac{\mathrm{d}t(z')}{\mathrm{d}z'} \mathrm{d}z'$$

`cosmoRate` evaluates this equation, which is implemented in `cosmo_functions.py`

- `cosmoRate` evaluates the merger rate density by solving the above Equation. It basically counts how many compact objects merge at a given redshift
- You may ask yourself: how does the population of compact objects look like at a given redshift? `cosmoRate` also prints catalogs of merging compact objects
- These are the two main output of `cosmoRate`

$$\mathcal{R}(z) = \int_{z_{\max}}^{z} \left[\int_{Z_{\min}}^{Z_{\max}} \boxed{\mathrm{SFRD}(z', Z)} \boxed{\mathcal{F}(z', z, Z)} \mathrm{d}Z \right] \frac{\mathrm{d}t(z')}{\mathrm{d}z'} \mathrm{d}z'$$

This factor here is evaluated starting from the input catalogs you inject in `cosmoRate`.
These catalogs comes from population-synthesis (`MOBSE`, `SEVN`), dynamical formation channel, or others (e.g. `FASTCLUSTER`).



This is an example of how the input folder and the names of input files look like

COSMORATE

$$\mathcal{R}(z) = \int_{z_{\rm max}}^{z} \left[\int_{Z_{\rm min}}^{Z_{\rm max}} \boxed{\mathrm{SFRD}(z', Z)} \boxed{\mathcal{F}(z', z, Z)} \mathrm{d}Z \right] \frac{\mathrm{d}t(z')}{\mathrm{d}z'} \mathrm{d}z'$$

Santoliquido et al. 2020:
https://arxiv.org/pdf/2004.09533.pdf

- This factor here is evaluated starting from the input catalogs you inject in `cosmoRate`.
- These catalogs comes from population-synthesis (`MOBSE`, `SEVN`) or dynamical formation channel, or others (e.g. `FASTCLUSTER`).

First row, first column: Total simulated mass in Msun

Each following row is a compact object merger, defined by
Column 0: delay time in years [WARNING: this is mandatory!]
Column 1- n: any other intrinsic parameter (in the example here,
Column 1: primary mass m_1 in Msun
Column 2: secondary mass m_2 in Msun)

data_BBHs_0.0002.txt

```
215180664.01669434
2.759170000000000000e+07 2.555260000000000176e+01 3.258890000000000242e+01
1.028390000000000000e+07 1.681370000000000076e+01 2.140279999999999916e+01
2.521270000000000000e+07 8.191700000000000870e+00 8.944000000000000838e+00
9.975900000000000000e+06 1.678249999999999886e+01 2.309309999999999974e+01
3.283750000000000000e+07 7.543899999999999828e+00 8.209600000000000009e+00
1.252210000000000000e+07 1.860089999999999932e+01 2.935050000000000026e+01
3.405190000000000000e+07 7.788599999999999746e+00 1.349530000000000030e+01
2.809323000000000000e+08 7.694399999999999906e+00 6.939400000000000013e+00
1.328070000000000000e+07 2.066850000000000165e+01 2.687709999999999866e+01
```

$$\mathcal{R}(z) = \int_{z_{\max}}^{z} \left[\int_{Z_{\min}}^{Z_{\max}} \boxed{\mathrm{SFRD}(z', Z)} \boxed{\mathcal{F}(z', z, Z)} \mathrm{d}Z \right] \frac{\mathrm{d}t(z')}{\mathrm{d}z'} \mathrm{d}z'$$

```python
# Compact objects specificities (PopSynth / N-body)
# list of compact objects type
cb_class = ['BBH'] # it can be either 'BBHs', 'BHNS', 'BNSs' sys.argv[2] within the parentheses
# list of metallicity used from simulations, as float type, ascending order
# uncomment what you need!
#Z_simulated = [0.0002, 0.0004, 0.0008, 0.0012, 0.0016, 0.002, 0.004, 0.006, 0.008, 0.012, 0.016, 0.02]
Z_simulated = [0.0001, 0.0002, 0.0004, 0.0006, 0.0008, 0.001, 0.002,  0.004, 0.006, 0.008, 0.01, 0.014,
 0.017, 0.02, 0.03]
#Z_simulated = [1e-11]


formation_channel = 'iso'  # formation channel type (iso, dyn, popIII)

# this parameter is used to select your astrophysical model and
# in naming the output catalog files (i.e values of alpha, SN model type...)
#sim = 'popI_II/popII_I_for_comp/fiducial_Hrad_5M/A1.0'
#      'popI_II/popII_I_for_comp/fiducial_Hrad_5M_channels/A1.0/all_channels'
sim = sys.argv[1]

#it can be either sys.argv[1] or a string ( e.g. 'A5', 'popI_II/popII_I_for_comp/fiducial_pisnfarmer19/A1.0'',
'')


sim_name = 'MandF2017_met' # it can be equal to sim or selected by the user
              # WARNING: if sim is a path, sim_name must be not a path!
extension = '.dat' # also include the dot
```

- In the first part of `cosmo_params.py` you set all the parameters that refer to the input catalogs.
- For instance, the compact object type, the list of progenitor metallicity, the specific simulations

COSMORATE

$$\mathcal{R}(z) = \int_{z_{\max}}^{z} \left[\int_{Z_{\min}}^{Z_{\max}} \boxed{\mathrm{SFRD}(z', Z)} \boxed{\mathcal{F}(z', z, Z)} \mathrm{d}Z \right] \frac{\mathrm{d}t(z')}{\mathrm{d}z'} \mathrm{d}z'$$

```
# Compact objects specificities (PopSynth / N-body)
# list of compact objects type
cb_class = ['BBH'] # it can be either 'BBHs', 'BHNS', 'BNSs' sys.argv[2] within the parentheses
# list of metallicity used from simulations, as float type, ascending order
# uncomment what you need!
#Z_simulated = [0.0002, 0.0004, 0.0008, 0.0012, 0.0016, 0.002, 0.004, 0.006, 0.008, 0.012, 0.016, 0.02]
Z_simulated = [0.0001, 0.0002, 0.0004, 0.0006, 0.0008, 0.001, 0.002,  0.004, 0.006, 0.008, 0.01, 0.014,
 0.017, 0.02, 0.03]
#Z_simulated = [1e-11]


formation_channel = 'iso'  # formation channel type (iso, dyn, popIII)

# this parameter is used to select your astrophysical model and
# in naming the output catalog files (i.e values of alpha, SN model type...)
#sim = 'popI_II/popII_I_for_comp/fiducial_Hrad_5M/A1.0'
#      'popI_II/popII_I_for_comp/fiducial_Hrad_5M_channels/A1.0/all_channels'
sim = sys.argv[1]

#it can be either sys.argv[1] or a string ( e.g. 'A5', 'popI_II/popII_I_for_comp/fiducial_pisnfarmer19/A1.0'',
'')


sim_name = 'MandF2017_met' # it can be equal to sim or selected by the user
              # WARNING: if sim is a path, sim_name must be not a path!
extension = '.dat' # also include the dot
```

The `formation_channel` variable sets two quantities. 1) The fraction of star-formation rate happening in binary systems 2) the correction due to not sampled initial mass function (IMF). For example, channel `'iso'` sets the binary fraction to 50%, while channel `'popIII'` does not apply any correction for IMF. Details on this are provided in the main Jupyter notebook

COSMORATE

$$\mathcal{R}(z) = \int_{z_{\max}}^{z} \left[\int_{Z_{\min}}^{Z_{\max}} \boxed{\mathrm{SFRD}(z', Z)} \boxed{\mathcal{F}(z', z, Z)} \mathrm{d}Z \right] \frac{\mathrm{d}t(z')}{\mathrm{d}z'} \mathrm{d}z'$$

Santoliquido et al. 2020:
https://arxiv.org/pdf/2004.09533.pdf

Injected catalogs

$$\mathrm{SFRD}(z, Z) = \boxed{\psi(z)} \boxed{p(Z|z)}$$

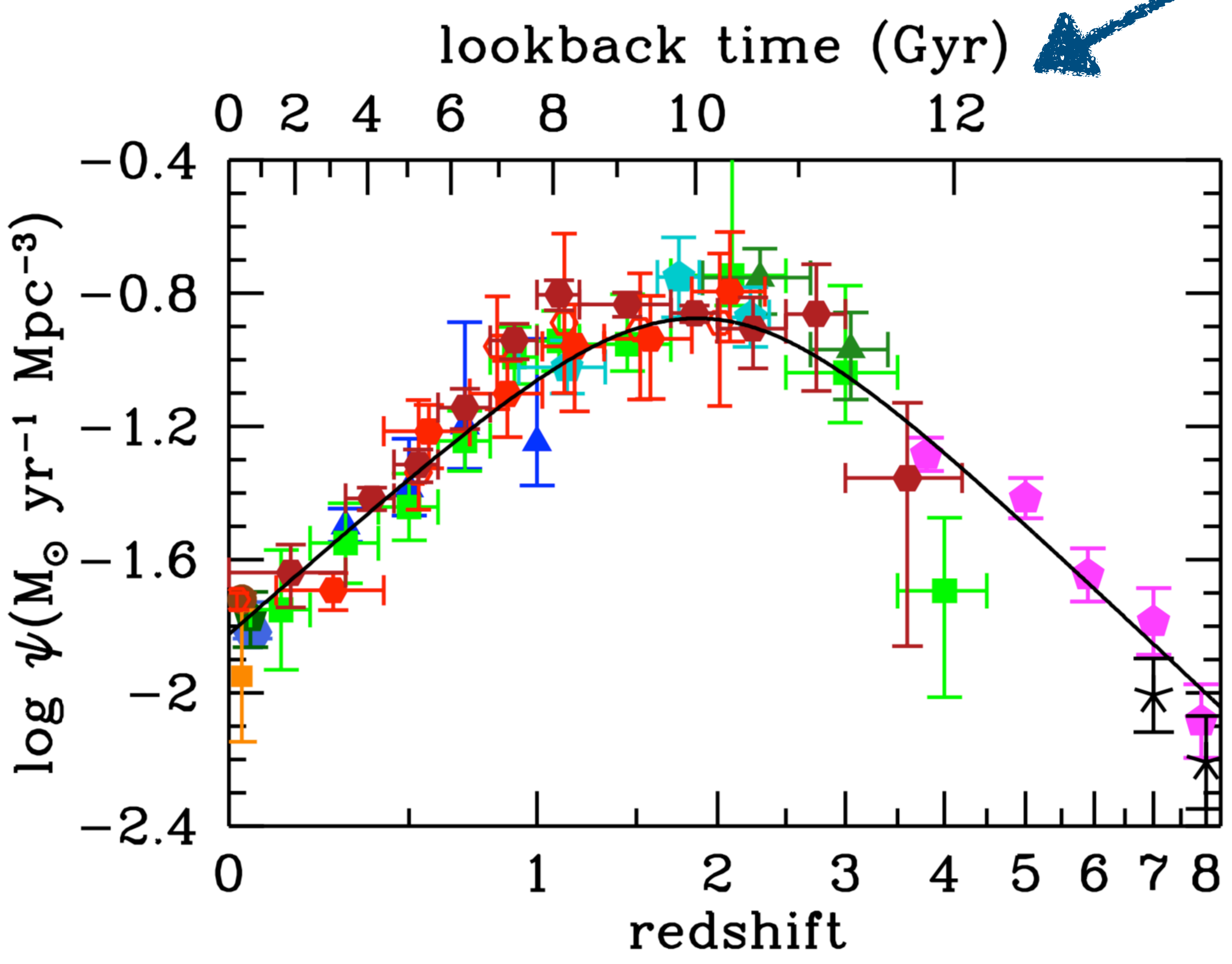
- `cosmoRate` couples these **injected catalogs with a distribution of star-formation rate density** (SFRD) as function of redshift and metallicity.
- The second part of `cosmo_params.py` refers to the parameters setting this distribution of SFRD

$$\mathcal{R}(z) = \int_{z_{\max}}^{z} \left[\int_{Z_{\min}}^{Z_{\max}} \boxed{\mathrm{SFRD}(z', Z)} \boxed{\mathcal{F}(z', z, Z)} \mathrm{d}Z \right] \frac{\mathrm{d}t(z')}{\mathrm{d}z'} \mathrm{d}z'$$

Santoliquido et al. 2020:
https://arxiv.org/pdf/2004.09533.pdf

Injected catalogs

$$\mathrm{SFRD}(z, Z) = \boxed{\psi(z)} \boxed{p(Z|z)}$$



Madau & Dickinson 2014, Madau & Fargos 2017

```
##########################################################
# select the population of stars

pop = 'I-II' # it can be 'III' or 'I-II'


# for popI-II star, 'MandF2017' is the only option available
# for popIII you can choose from many options:
# 1. 'A-sloth_smooth_off'
# 2. 'Hartwig16'
# 3. 'DeSouza_smooth_off'
# 4. 'Jaacks'
# 5. 'LiuBromm'
# for other options see cosmo_rate_notebook.ipynb
# or sys.argv[2]
SFRD_model = 'MandF2017'
```

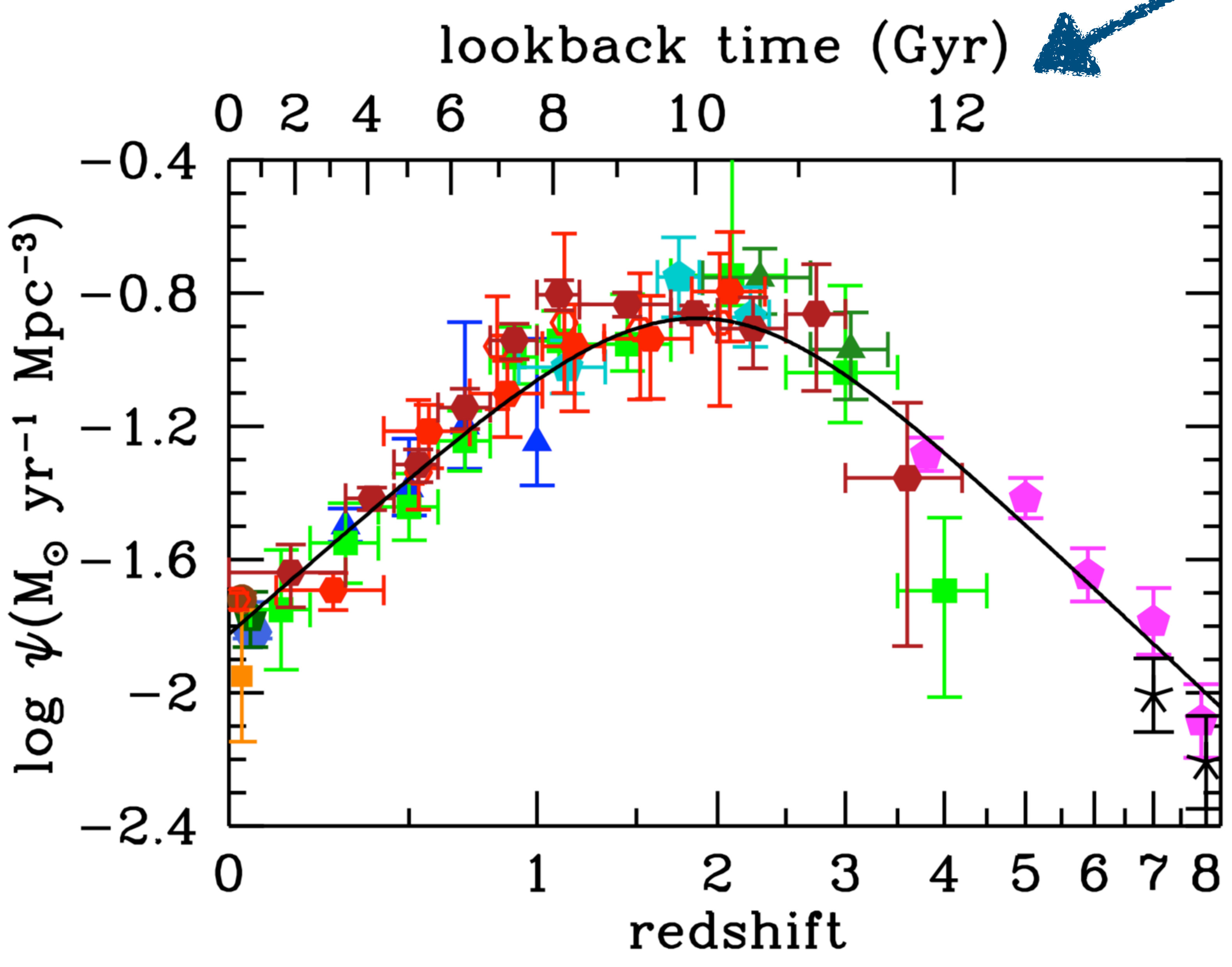
COSMORATE

Santoliquido et al. 2020:
https://arxiv.org/pdf/2004.09533.pdf

$$\mathcal{R}(z) = \int_{z_{\max}}^{z} \left[\int_{Z_{\min}}^{Z_{\max}} \boxed{\mathrm{SFRD}(z', Z)} \boxed{\mathcal{F}(z', z, Z)} \mathrm{d}Z \right] \frac{\mathrm{d}t(z')}{\mathrm{d}z'} \mathrm{d}z'$$

Injected catalogs

$$\mathrm{SFRD}(z, Z) = \boxed{\psi(z)} \boxed{p(Z|z)}$$



Madau & Dickinson 2014, Madau & Fargos 2017

$$p(Z|z) = \frac{1}{\sqrt{2\pi\,\sigma_Z^2}} \exp\left\{ -\frac{\left[\log\left(Z/\mathrm{Z}_{\odot}\right) - \mu(z)\right]^2}{2\,\sigma_Z^2} \right\}.$$

The **distribution of metallicities at fixed redshift** is by default a **gaussian** distribution. Here we can change the mean and the spread. The mean $\mu(z)$ is a function of redshift and for the model `'linear'` we also know the uncertainty

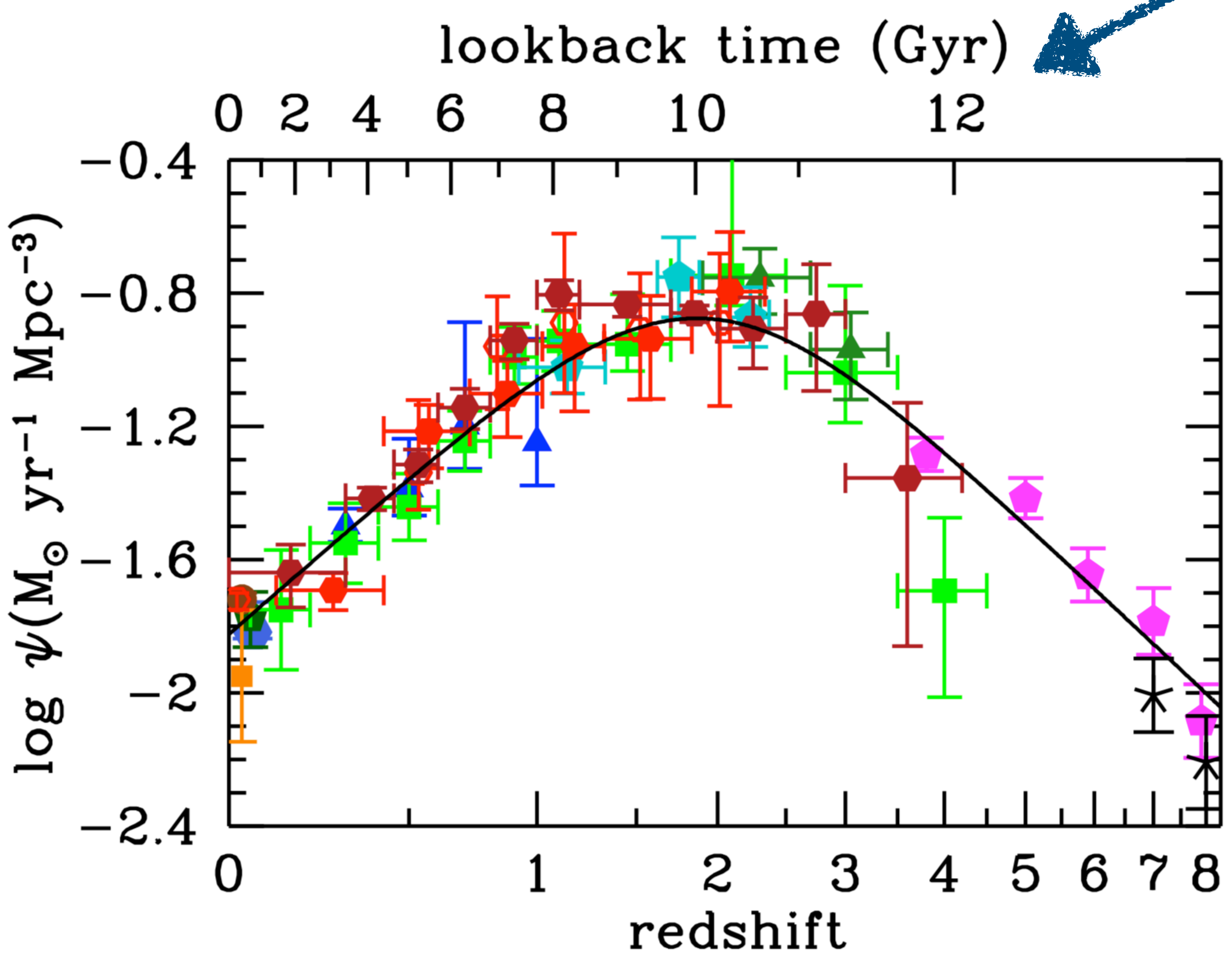
COSMORATE

Santoliquido et al. 2020:
https://arxiv.org/pdf/2004.09533.pdf

$$\mathcal{R}(z) = \int_{z_{\max}}^{z} \left[\int_{Z_{\min}}^{Z_{\max}} \boxed{\mathrm{SFRD}(z', Z)} \boxed{\mathcal{F}(z', z, Z)} \mathrm{d}Z \right] \frac{\mathrm{d}t(z')}{\mathrm{d}z'} \mathrm{d}z'$$

Injected catalogs

$$\mathrm{SFRD}(z, Z) = \boxed{\psi(z)} \boxed{p(Z|z)}$$



Madau & Dickinson 2014, Madau & Fragos 2017

$$p(Z|z) = \frac{1}{\sqrt{2\pi\, \sigma_Z^2}} \exp\left\{ - \frac{\left[\log\left(Z/\mathrm{Z}_\odot\right) - \boxed{\mu(z)}\right]^2}{2\, \sigma_Z^2} \right\}.$$

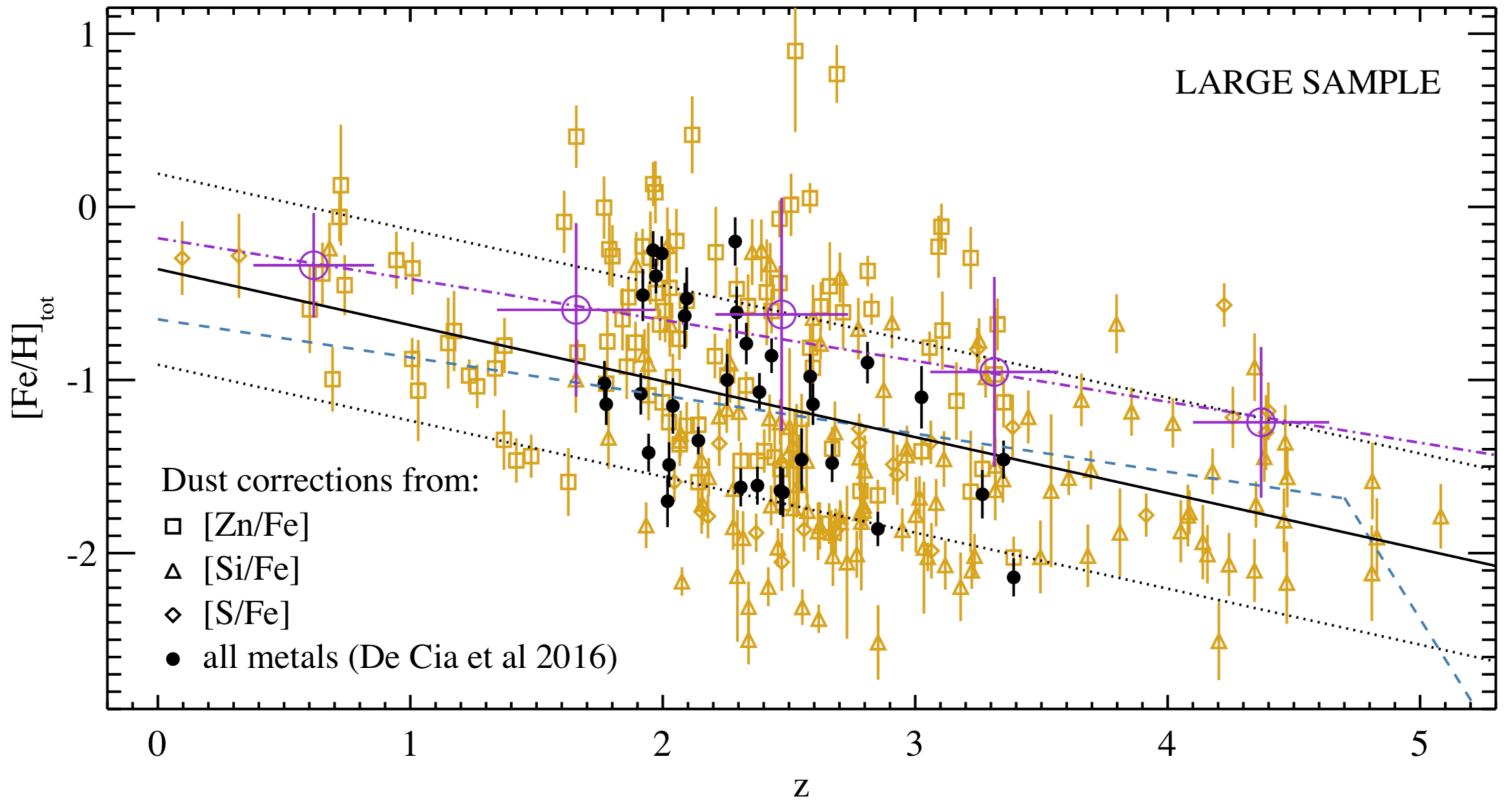
```
# Choose the model for average metallicity fit type
metmod_type = 'MandF2017'    # it can be 'linear' or 'MandF2017'
sigma_met = 0.20  # value of the metallicity spread in log-sun-scale
                  # it can be 0.10, 0.20, 0.30 or sys.argv[3]

# Definition of Solar metallicity
Z_sun = 0.019  # 0.0153 from Cauffau et al. 2011
               # 0.019 from Gallazzi et al. 2008
```

De Cia et al. 2018, Gallazzi et al. 2008, Madau & Fragos 2017

Level 2: How to modify `cosmo_params.py`

- `cosmoRate` has other useful features:
    - Propagate known sources of uncertainty on the merger rate density

For example, the `'linear'` model is extracted from the metallicity evolution of neutral hydrogen ([*De Cia et al. 2018*]). The model we implemented is the purple line in the figure, which is a linear fit. `cosmoRate` can propagate the uncertainty of slope and intercept on the merger rate density via a Monte Carlo approach. Each time it samples a value of slope and intercept forma a gaussian distribution. How to switch on this features is explained in the next slide.



Level 2: How to modify `cosmo_params.py`

- `cosmoRate` has other useful features:
  - Propagate known sources of uncertainty on the merger rate density

```
# Here you set if cosmo_Rate evaluates also the uncertainty through a Monte Carlo approach
N_iter = 1  # number of iteration to be done for evaluating the uncertainty. In any case, it must be at least
one!
# 1000 iterations are enough to perform a robust estimation of the uncertainties

# metallicity
met_uncertainty = 'No'   # if 'Yes' the uncertainty due to metallicity will be computed
met_slope_sprd = 0.014   # 0.14 for linear metallicity model or 0.014 from 'MandF2017'
met_inter_sprd = 0.14   # from Gallazzi et al. 2008

# star formation rate option
sfr_uncertainty = 'No'   # if 'Yes' the uncertainty due to sfr will be computed,
# spread over the normalisation value
sfr_sprd = 0.2  # from Madau & Dickinson 2017
```

Level 2: How to modify `cosmo_params.py`

- Creating catalogs of merging compact objects at some given redshift bins

```
catalogues_c = 'Yes'  # if 'Yes' merging compact object per redshift bin are produced
mrd_eval = 'Yes' # this option is to skip the evaluation of the merger rate density in case
                 # it has been already done in previous runs

# specify the redshift bins of which you want to have the catalogs printed
# WARNING: The code will search for the closest available redshift bin!
z_vect_cat_op =  False
#[30, 25, 20, 17, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0.1]
# some other options:
#[14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0.1]
# [20, 11, 5, 0.1]
# if this option is set to False the catalogs will be printed at every merger redshift
#[30, 25, 20, 17, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0.1]

# Specify what parameters cosmoRate will sample. The order of the parameter names must be exactly
# the same as the order of the corresponding column in input_catalogs/
# delay time is sampled by default.
# The number of parameters can be smaller than the total number of paramters given as input.
# Leave parameters = [] if in your catalogs you want just time-delay related parameters,
# i.e. merging redshift, formation redshift, delay time and progenitor star metallicity
parameters = ['ID', 'M1[Msun]', 'M2[Msun]']
```

Level 2: How to modify `cosmo_params.py`

- Creating catalogs of merging compact objects at some given redshift bins

```
catalogues_c = 'Yes'  # if 'Yes' merging compact object per redshift bin are produced
mrd_eval = 'Yes' # this option is to skip the evaluation of the merger rate density in case
                 # it has been already done in previous runs

# specify the redshift bins of which you want to have the catalogs printed
# WARNING: The code will search for the closest available redshift bin!
z_vect_cat_op =  False
#[30, 25, 20, 17, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0.1]
# some other options:
#[14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0.1]
# [20, 11, 5, 0.1]
# if this option is set to False the catalogs will be printed at every merger redshift
```

WARNING: the order of parameter names in the array `parameters` must be exactly the same as in your input catalogs!

```
# i.e. merging redshift, formation redshift, delay time and progenitor star metallicity
parameters = ['ID', 'M1[Msun]', 'M2[Msun]']
```

cosmoRate output

1. One folder per each compact object type
2. A plot of the adopted SFRD model

Catalogs of merging compact objects at a given redshift (e.g. 9.92, 9.77, 9.62)

This contains the merger rate density at a given redshift

cosmoRate output

MRD_spread_15Z_50_No_

```
#Redshift Merger_Rate_Density[Gpc^-3yr^-1]
7.5757575757575724620e-02 9.9999166146173941 17e+00
```

Merger rate density file

1. First column: middle of the redshift bin
2. Merger rate density in Gpc^-3 yr^-1

BBH_MandF2017_met_Mar

```
#ID M1[Msun] M2[Msun] z_merg z_form time_delay[yr] Z_progenitor
3304345.00000 30.32970 26.32721 0.07576 14.92424 13186997527.04119 0.00010
3566223.00000 31.37294 24.66961 0.07576 14.92424 13008058858.42008 0.00010
4949627.00000 22.47323 17.41708 0.07576 14.92424 13467649911.29968 0.00010
220280.00000 39.08434 32.70891 0.07576 14.92424 11612581593.33620 0.00010
4085742.00000 37.72191 27.38235 0.07576 14.92424 13304820193.02670 0.00010
2277682.00000 43.57659 40.35687 0.07576 14.92424 13455617795.50724 0.00010
3276481.00000 27.34489 21.45434 0.07576 14.92424 13237437606.31335 0.00010
3682468.00000 38.62225 30.58141 0.07576 14.92424 12471478414.62328 0.00010
```

Catalog of merging compact objects

WARNING: storage usage may be very high!