Grism SED Fitter (GSF)
version 1.0.0

Takahiro Morishita

February 14, 2020

1 Description

Grism SED Fitter, or GSF, is a python-based stellar energy distribution (SED) fitting code originally optimized for *HST* grism spectra plus broadband data sets. GSF fits spectra and broadband photometric data points simultaneously with model templates ("full-spectral fitting"). The latest version can handle various types of data set, e.g., without spectrum. The main conceptual feature is its use of non-functional forms for star formation histories—GSF rather determines the amount of stars of arbitrary ages, inferring the amount of star formation at different lookback times. GSF uses Markov chain Monte Carlo (MCMC) inference, to sample probability distributions of parameters, such as star formation, metallicity, and dust attenuation.

2 Python Package Pre-requirement

- astroconda (for general python packages)
- `emcee`(Foreman-Mackey+2013)
- `lmfit`(Newville+2017)
- Corner plot package (Foreman-Mackey+2013)
- cosmolopy
- fsps (Conroy+2009)
- python-fsps (Foreman-Mackey+2014)

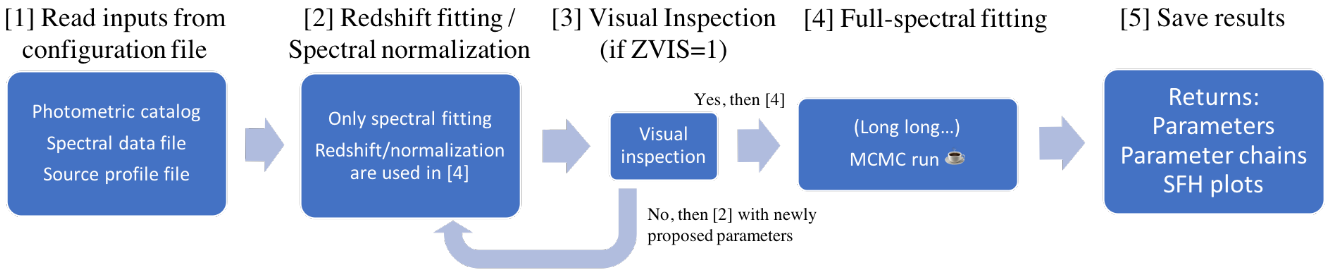

[1] Read inputs from configuration file
[2] Redshift fitting / Spectral normalization
[3] Visual Inspection (if ZVIS=1)
[4] Full-spectral fitting
[5] Save results
Photometric catalog
Spectral data file
Source profile file
Only spectral fitting
Redshift/normalization are used in [4]
Visual inspection
Yes, then [4]
No, then [2] with newly proposed parameters
(Long long...)
MCMC run ☕
Returns:
Parameters
Parameter chains
SFH plots

Figure 1: Flowchart of main steps in GSF.

3 Input files

GSF is run from a command line. For example;

```
python run_gsf.py [configuration file] [index]
```

where run_gsf.py is contained in the package source. All details of the fitting should be described in a configuration file (see below). [index] specifies a starting point of the process, where;

0: From the template generation at $z = 0$ (Section 4.1). Then, it goes to 1.

1: From the template generation at $z = z_{\rm inp}$ (Section 4.2). Then, it goes to 2.

2: From the fitting process (Section 5). Then, it goes to 3.

3: From the plotting process (Section 6.2)

Step 0 takes a lot of time, typically 30 min to few hours depending on how many templates you are generating. A useful tip is to avoid from running 0 as much as possible. For example, one needs to re-start from step 0 if any of **Critical Parameters** (see Section 3.1) is changed. Otherwise, users are recommended to start from step1 or later.

3.1 Configuration file

All parameters and setup are controlled from an input configuration file. Files should be in the following format;

```
#
# Params for MCMC
#
NMC   100000 # No. of iterations for the primary MCMC step.
NWALK  100 # No. of walkers for the primary MCMC step.
NMCZ  10 # No. of iterations for the pre−redshift fitting step.
NWALKZ  10 # No. of walkers for the pre−redshift fitting step.
NTEMP  0 # No. of parallel tempering.
NCPU  0 # No. of cpus used for multiprocessing.
FNELD  1 # Minimization method in lmfit. For initial values. 0=Powell, 1=Nelder (faster).
#
# Params for data
#
DIR_TEMP ./templates/ # Directory of model templates. If not exist, gsf creates one.
CAT_BB  ./m1149jd01_bb.cat # Broadband photometry catalog.
DIR_EXTR ./grism_extraction/ # Directory of spectroscopic data. If not, gsf will ignore and fit broadband data.
SPEC_FILE G102.cat,G141.cat # Comma separated list for spectral files. Placed in DIR_EXTR directory. G102 has \
to come first, then G141.
DIR_FILT  /Users/tmorishita/GitHub/Code/sedfitter/sedfitter/filter/ # Directory of filter response curve files.
FILTER  f105w,f110w,f125w,f140w,f160w,f435w,f475w,f555w,f606w,f625w,f775w,f814w,f850w,f225w,f275w,\
f336w,f390w,f230w,f360w,f450w # Filters of broadband photometry. Each string should match *.fil files in DIR_FILT.
#
# Critical Parameters for SED
# (If any of the following parameters is changed, you need to restart from Step0.)
#
AGE   0.01,0.03,0.1,0.3,0.5,0.7,1.0,1.5,2.0,3.0 # Set of age pixels, in Gyr. Lookback time.
TAU0  −1 # Fixed length of each age pixel, in Gyr (0.01 to 20Gyr). If 99, CSP is applied. If a negative value, SSP is applied.
ZMAX  0.4 # Maximum value for metalicity, in logZ.
ZMIN  −0.8 # Minimum value for metalicity, in logZ.
DELZ  0.2 # Resolution in metallicity, in logZ.
NIMF  0 # Choice of IMF. 0=Salpeter, 1=Chabrier, 2=Kroupa, 3=van Dokkum, 4=Dave, 5=tabulated, specified in imf.dat \
file located in the data directory.
ADD_LINES 1 #Emission lines in model templates. 0=no, 1=yes.
LOGU  −2.5 # Ionizing parameter U, in log.
#
# Other Parameters for SED
# (Changing the following parameters does not require step0 to be run.)
#
ZEVOL  0 # Variation in Metallicity at different age pixels. 0=no, 1=yes.
#AGEFIX  1.0 # If you wish to use a set of age from AGE for fitting, list them here. Could be more than one (e.g.,1.0,1.5).
#ZFIX  0.0 # If you want to fix logZ, comment out ZMAX,ZMIN,DELZ, and set this.
AVMIN  0.0 # Minimum Av (Dust Attenuation) in mag.
AVMAX  4.0 # Maximum Av (Dust Attenuation) in mag.
ZMC   1 # Redshift as a free parameter in the primary MCMC step. 0=no, 1=yes.
#
# Params for target object
#
ID  00491 # String.
PA  00 # String. Position angle for grism spectrum. If not, leave this arbitrary.
ZGAL 9.1 # Initial guess of source redshift.
CZ0   1.0 # Initial guess of spectral normalization for G102.
CZ1  1.0 # Initial guess of spectral normalization for G141.
LINE  0 # Wavelength of emission lines, in AA, if one wish to mask them out. 0=no.
ZVIS  1 # Visual inspection of redshift. 1=yes, 0=no. If you are not confident with ZGAL, CZ0/1, then one should leave \
this 1 for iteration.
MORP  moffat # Profile shape, if grism spectra. Used for convolution. moffat, gauss, or none.
MORP_FILE moffat.cat # Ascii file for morphology parameters.
```

```
VDISP 300.0 # Velocity dispersion in km/s. Will be used to convolve templates if MORP=none.
#
# Misc
#
TAU_COMP 0 # Shows comparison between different SFHs model. One has to have tau model. This is in preparation.
```

3.1.1 Broadband catalog file

`GSF` reads an ascii broadband catalog specified in CAT_BB in a configuration file. The catalog format is similar to EAzY and FAST (Brammer+2008; Kriek+2009), and should be;

```
# ID [flux of filter 1] [uncertainty in flux of filter 1]...
```

The flux unit has to be in f_ν, with a magnitude zero point $m_0 = 25$, i.e.

$$m = -2.5 \log_{10}(f_\nu) + m_0 \tag{1}$$

The order of flux & flux error pair must match those in FILTER array in the input configuration file. FILTER array must correspond to response curve files in DIR_FILT. For example, if one of FILTER keywords is “f105w”, then `GSF` will look into DIR_FILT directory to find a filter response file “f105w.fil”, where the format is in;

```
# Column_number Wavelength_in_AA Response_curve
```

Standard filter response curve files are contained in the package (copied from EAzY), while users can add their own filter files in the format explained above. `GSF` will find the column with ID that matches “ID” in the configuration file.

3.1.2 Spectral data file

`GSF` reads an ascii spectral file of for the target object, in DIR_EXTR in a configuration file. The file should be specified in [SPEC_FILE], whose formats are;

```
# Wavelength_in_AA Flux_nu Error_in_flux
```

The unit of flux and error has to be in f_ν with a magnitude zero point $m_0 = 25$.

For grism spectra, one is asked to provide morphological parameters of the target. In the current version, `GSF` convolves model templates either with a 1-dimensional Moffat function,

$$f(r; \alpha, \gamma) = A\Big[1 + \Big(\frac{r^2}{\gamma^2}\Big)\Big]^{-\alpha} \tag{2}$$

or Gaussian,

$$f(r; \gamma) = A\exp\Big(\frac{-r^2}{2\gamma^2}\Big) \tag{3}$$

The parameters should be stored in an ascii file, [MORP_FILE], in the following format;

```
# A gamma alpha
```

for both cases (i.e. put a random number for alpha if gaussian), where A is a normalization constant.

4 Fitting Templates

`GSF` generates spectral templates based on parameters listed in the configuration file. This step is two-folded—generating model templates at $z = 0$ via fsps (Conroy+2009), and shifting to an aimed redshift.

4.1 Generating Rest Frame Templates

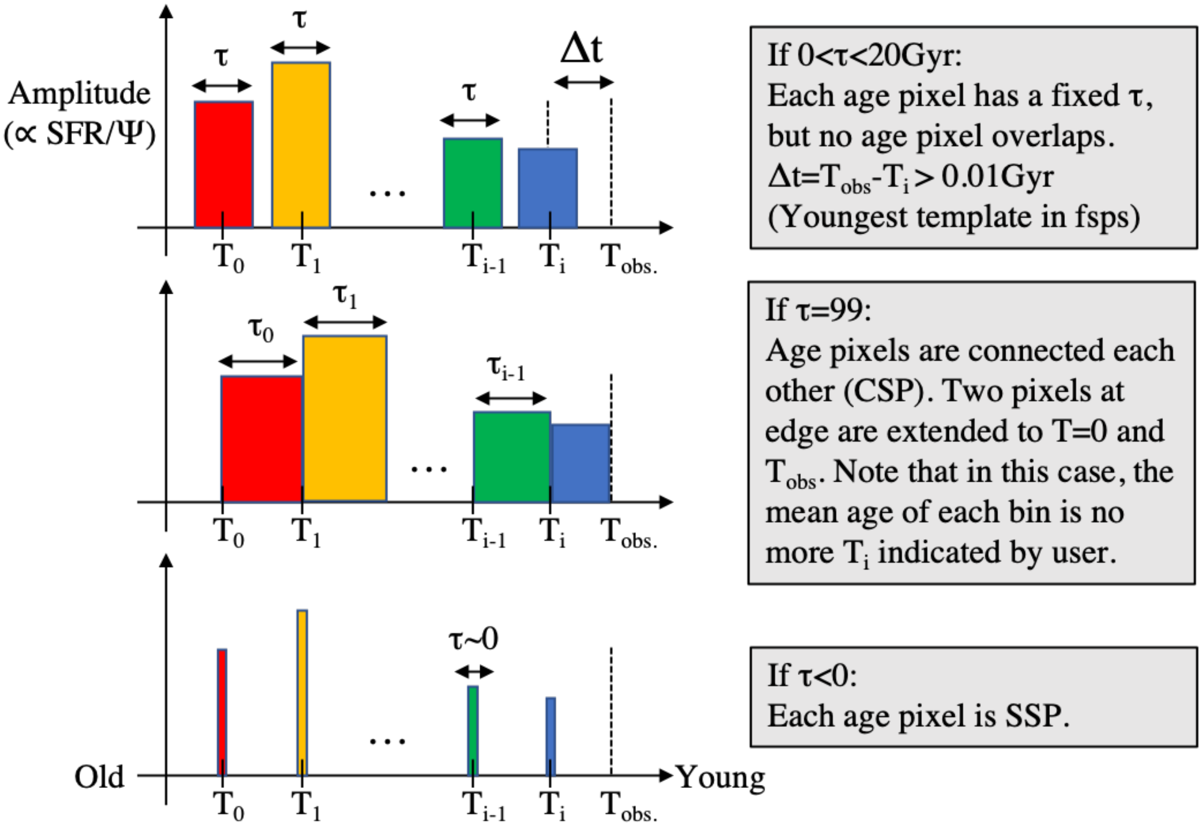
GSF generates model spectral templates via python-fsps (Foreman-Mackey+2014), based on the parameters in the configuration file. The parameters are limited from the original fsps (Conroy+2009), such as the range in metallicity, initial mass function (IMF), and ionizing parameter.

It is in this step that generates templates with different ages and metallicities. The templates are saved in a common file, “spec_all.fits” (irrespective to object IDs), for the aim of saving time. Redshifted templates (Section 4.2), which are actually used in the fitting process, are based on this file. Therefore, if one wish to change any of template parameters (IMF, age/metallicity range and step etc.), it is suggested to save the previous file to avoid overwrite, or move in another directory to avoid confusion.

Age pixels determine the number of templates at different lookback times. The length of star formation is specified by a parameter, TAU0, in the configuration file. The length, if specified to a value < 99, is uniform in all age pixels in the current version (Figure 2). If set to 99, then the CSP is applied so that age pixels are continuously connected each other. If set to a negative value, then SSP is applied.

4.2 Shifting Templates to a Specific Redshift

It is at this stage where GSF applies dust attenuation and also IGM attenuation for those $z > 6$.



If 0<τ<20Gyr:
Each age pixel has a fixed τ, but no age pixel overlaps.
$\Delta t = T_{obs} - T_i > 0.01$Gyr
(Youngest template in fsps)

If τ=99:
Age pixels are connected each other (CSP). Two pixels at edge are extended to T=0 and T_{obs}. Note that in this case, the mean age of each bin is no more T_i indicated by user.

If τ<0:
Each age pixel is SSP.

Figure 2: Age pixels and τ. Note that time scale is in log. Users can define a arbitrary value for the length of each age pixel, though no overlap is allowed between age pixels. If τ is set to negative value (or values smaller than the minimum length of fsps; 0.01 Gyr), SSP is used. See Section 4.1.

5 Fitting The Data

GSF consists of two fitting processes, one for redshift and spectrum normalization, and then (long) full-spectral fitting. While both steps utilize an MCMC scheme via emcee package, the former includes only spectral data to determine the redshift and normalization of spectral flux (see below) in an iterative visual inspection process.

At each MCMC step, fitting is proceeded based on the Metropolis–Hastings (M–H) method. While it is technically possible to implement arbitral forms of priors, it is beyond the current purpose, and flat priors are set within each parameter range.

5.1 Redshift Fitting and Flux Normalization

This step is preferred when spectral data exist, since the aim here is to determine the redshift via visual inspection, as well as the normalization of spectral data to the broadband flux. If ZVIS is set to 1 (i.e. yes), the fitted spectrum is displayed at each iteration for visual inspection of the propose redshift and normalization. The users can then determine if they want to use their original input values or newly proposed values. If the newly proposed values are selected, then the step is iterated but with a new set of templates (Section 4.2). This is because the templates used here were originally generated based on the initial redshift. After one more visual inspection, then user can proceed to the next step. This step returns the 16/50/84th percentiles of redshift and normalizations, which are used in the following step.

5.2 Full Spectral Fitting

This step processes fitting of all input data with model templates. The redshift and its range is set to the ones determined in the previous step (50th and 16th/84th percentiles). GSF shifts model templates within the range during the fit, rather than creates a new set of templates at redshift of each mcmc step. Redshift can also be fixed to the median value if ZMC is set to 1.

At each mcmc step, the fit is evaluated with likelihood calculated by

$$ln(likelihood) = -0.5 \sum_{i}^{n} \frac{(y_{i,model} - y_{i,data})^2}{scl_i^2} + ln(2\pi scl_i^2) \tag{4}$$

where scaled sigma is

$$scl_i^2 = \sigma_{i,data}^2 + f^2 y_{i,model}^2. \tag{5}$$

f is a parameter, if F_ERR is set to 1, to account for additional error underestimated in $\sigma_{i,data}$ only. If F_ERR is 0, then f is set to 0 and the term is vanished. The posterior is then calculated by

$$ln(posterior) \propto ln(likelihood) + ln(prior) \tag{6}$$

In the current version (1.0.0), the prior is set to uniform within the parameter range. At each step of mcmc ($\in$ NMC), the Metropolis-Hastings algorithm is used for each walker (NWALK) to evaluated the likelihood[1].

6 Saving Results and Plots

6.1 Result Files

At the end of the primary MCMC fitting step, GSF samples parameter chains. In default, the first half of the chain (NMC/2) is discarded (burn-in), so that values are not significantly affected by the initial values. In addition to summary of parameters (16/50/84th percentiles), GSF also saves Markov chain in a cpkl format, so that users can resample the chain and reproduce the result. The primary output files are as follows;

[1] See the instruction in http://dfm.io/emcee/current/

- Summary of parameters (summary_[ID]_PA[PA].fits). 16/50/84th percentiles.

- Markov Chain file (chain_[ID]_PA[PA]_corner.cpkl).

- Best fitting templates (./templates/gsf_spec_[ID].fits). 16/50/84th percentiles.

- Rest frame UVJ colors, recalculated from past to the present time ([ID]_PA[PA]_uvj.txt).

6.2 Plots

`GSF` uses the result to plot some physical properties of the target object. This includes star formation/metallicity/mass accumulation histories, SED plot, and color evolution. By default, `GSF` generates plots of the SED of target (input data and model), star formation history, and parameters' corner plot. The star formation rate of each age pixel represents the amount of stellar mass (corrected for mass-to-luminosity ratio) divided by the time length. Therefore, it can not distinguish contribution from external systems (i.e. merger).

These plots can be reproduced by using the saved files above.

7 References

Brammer, van Dokkum, & Coppi. 2008, ApJ, 686, 1503. Conroy, Gunn, & White. 2009, ApJ, 699, 486. Foreman-Mackey+. 2013, PASP, 125, 306. Foreman-Mackey, Sick, & Johnson. 2014. Kriek+. 2009, ApJ, 700, 221. Newville+. 2017.