CIGALE Manual (v1.0)

Yang, Guang (gyang206265@gmail.com)
Burgarella, Denis (denis.burgarella@lam.fr)

This manual (along with supplementary material) is available from:
https://cigale.lam.fr/documentation/

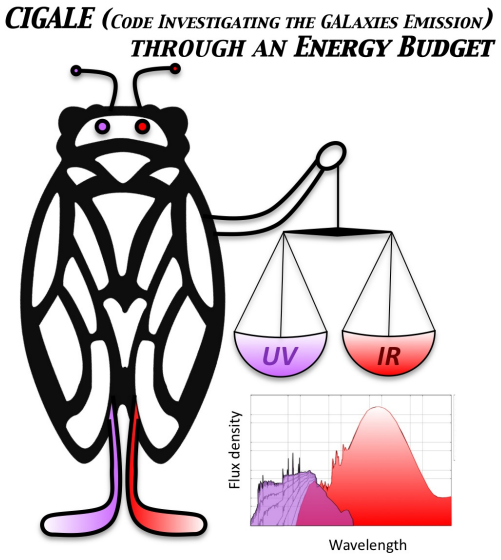

Table of Contents

List of Tables

List of Figures

1. Overview

Code Investigating GALaxy Emission (CIGALE) is a Python code for the fitting of spectral energy distribution (SED) of galaxies. It has been developed for more than a decade (e.g. Burgarella et al., 2005; Noll et al., 2009; Serra et al., 2011; Boquien et al., 2019). The detailed algorithm is described in Boquien et al. (2019). Yang et al. (2020) upgraded CIGALE to allow it fitting X-ray data, and this version is dubbed as "X-CIGALE". We further merged X-CIGALE into the main branch of CIGALE as well as implemented many improvements and functionalities (Yang et al., 2022). The new version is marked as v2022.0. This manual serves as a "quick and practical" reference for the user. Further questions can be asked in our discussion forum (https://github.com/mboquien/cigale/discussions). All materials of CIGALE (including this manual) can be found on https://cigale.lam.fr.

In §2, we describe the installation procedures. CIGALE has two working modes. One is fitting the observed galaxy SEDs, and the other is simulating model SEDs. These two modes are described in §3 and §4, respectively. Appendix A lists the main model parameters. Appendix B describes the supplementary files used in this manual.

2. Installation

The easiest way is to use pip installation. To do this, in the downloaded CIGALE directory, simply run
$ *pip install .*

However, the pip installation above only allows you to use the default downloaded code. If you want to modify the code to serve your own research interest, you can install from source. The instruction is detailed at https://github.com/mboquien/cigale/discussions/2

3. Run – Fitting Observed Data: the "pdf_analysis" mode

In this Section, we detail the procedures of fitting observed data.

3.1 Data Preparation

The input data for fitting should be a ASCII table with the format of the following:

# id	redshift	filter1	filter1_err	filter2	filter2_err	...
J175535.47+660959.0	1.22	1.091e-02	1.20e-03	1.533e-02	1.56e-03	...
19260817	3.12	9.325e-03	1.04e-03	4.107e-02	4.18e-03	...

The first column is the name for each source. The second column is the redshift information. The entry can be set to negative is you want to CIGALE to search for redshift, i.e. the "photometric redshift" mode. If the entry is set to 0, then the source is assumed to be at 10 pc. Note that an optional column, "distance" (in units of Mpc), can be inserted after the redshift column. If the distance column is provided, then it will be used in lieu of the distance computed from the redshift. The following columns are fluxes and 1σ uncertainties in units of mJy (photometry) or W m^{-2} (emission line). Note that "filter1", "filter2" ... should be the filter names in CIGALE database. You can run
$ *pcigale-filters list*
in the terminal to list all of the existing filters. You can also create your own filter in an ASCII file in the following format:

```
# filter1
# photon
# some comments
1340.62          0.0000
1350.49          0.1154
1370.21          0.1765
```

The first line is the filter name. The second line tells the filter type, and can be "energy" or "photon", which determines the way flux is calculated.[1] The third line presents some explanatory comments. The following lines are "wavelength" (in Å) and "transmission". To implement the filter in CIGALE database, you can add the filter ASCII file to CIGALE filter directory (pcigale/atabasebuilder/filters/) and re-build the code (see §2). Another way is to run the following command in the terminal:
$ *pcigale-filters add filter_file*

Aside from normal flux and error, CIGALE can also deal with upper limit. Fig. 1 summarizes the way how CIGALE deal with the input fluxes and errors.

[1] See http://svo2.cab.inta-csic.es/theory/fps/ for detailed formulas.

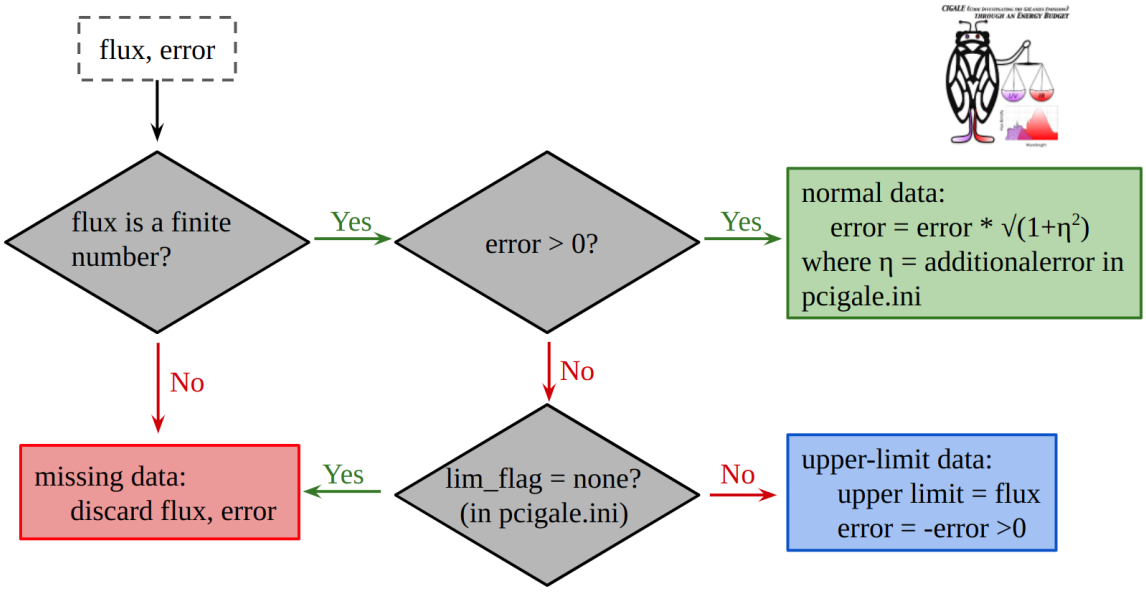

Fig. 1. This figure presents how CIGALE manages (flux, error) for each filter.

3.1.1 X-ray filters and fluxes

As explain in Yang et al. (2020), CIGALE requires that the input X-ray fluxes are intrinsic. This means that the energy-dependent instrumental response should have been corrected. Therefore, the X-ray filters should be flat, i.e. boxcar-shaped. Fig. 2 presents an example X-ray filter for 2–7 keV. CIGALE includes a few filters for typical X-ray bands. You can also create your own X-ray filters. For your convenience, we provide a Python code (code/xray_filter.py) to generate boxcar filters for a given X-ray band. For example, you can run it as

$ *python*
>>> *import xray_filter*
>>> *xray_filter.write_boxcar_filter("1to5.dat", "1to5", 1, 5)*
will write a filter named "1to5.dat" for 1–5 keV.

In X-ray catalogs, the fluxes are often given in the cgs units of erg s^{-1} cm^{-2}. CIGALE requires all inputs fluxes to be in units of mJy. Eq. 1 of Yang et al. (2020) gives the formula for the conversion. We also provide a Python code ("code/convert_Fx.py") to do this job. For example,

$ *python*
>>> *import convert_Fx*
>>> *Fnu, Fnu_err = convert_Fx.convt_Fx_to_Fnu([1e-16, 1e-15], [3e-17, 2e-16], 2, 7)*
will convert 2–7 keV fluxes of [1e-16, 1e-15] erg s^{-1} cm^{-2} and errors of [3e-17, 2e-16] erg s^{-1} cm^{-2} to mJy fluxes ("Fnu") and flux errors ("Fnu_err"). These outputs of "Fnu" and "Fnu_err" can then be written to CIGALE input data.

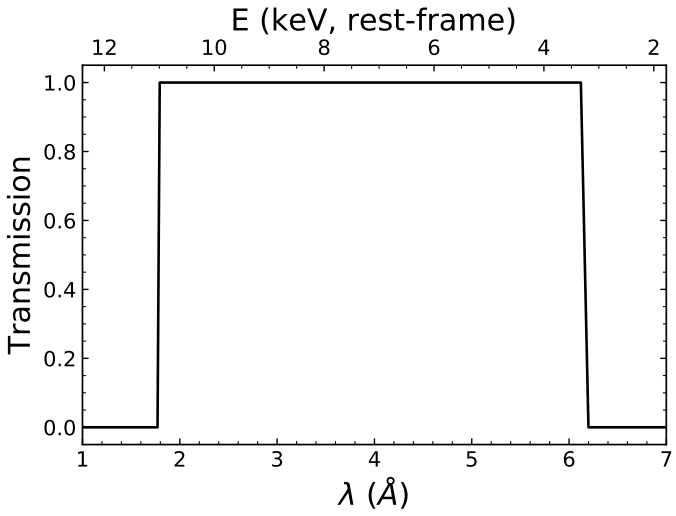

Fig. 2. An example X-ray boxcar filter for 2–7 keV.

3.2 Configuration and Run

Open terminal, cd to your working directory.
$ *pcigale init*
will initialize configuration files called "pcigale.ini" and "pcigale.ini.spec". You only need to edit "pcigale.ini". There are five parameters in this file (with many commentary words starting with #): "data_file", "parameters_file", "sed_modules", "analysis_method", "cores".

- "data_file" is the input data file (§3.1).
- "parameters_file" is the optional file when simulating data (see §4). It should be empty when fitting observed data.
- "sed_modules" lists the names of the SED modules that will be used in the run. The available modules are listed in the commentary parts of the "pcigale.ini" file. The module names should follow the order given in the commentary parts.
- "analysis_method" is CIGALE mode. Should be "pdf_analysis" for data-fitting purpose.
- "cores" is the number of CPU cores that will be used. Note that increasing the number of cores may not necessarily boost the speed.

We provide two example runs of AKARI-NEP AGNs and SDSS QSOs (Yang et al., 2020) along with this manual ("examples/akari_nep_xray_agn" and "examples/sdss_qso/"). In the test run, the configuration file reads:

data_file = sdss_qso.txt

parameters_file =

sed_modules = sfhdelayed, bc03, nebular, dustatt_calzleit, dale2014, skirtor2016, xray, redshifting

analysis_method = pdf_analysis

cores = 4

After setting the initial configuration file, run the following in terminal
$ *pcigale genconf*
which will generate the full configuration files "pcigale.ini" and "pcigale.ini.spec".

Open "pcigale.ini", and you will find more parameters have been added. Following "cores", there are two parameters "bands" and "properties". You can see that CIGALE already fills in the band and property names from the input data. But if you do not want to use some information, you can delete some entries.

The other new parameters fall into two categorises, [sed_modules_params] and [analysis_params]. [sed_modules_params] includes the configurations for each adopted SED module. These parameters should be self-explanatory, and we do not further explain them here. [sed_modules_params] determines the number of models that will be built. After finishing [sed_modules_params], you can check the number of models with
$ *pcigale check*
With this configuration cigale will compute 15966720 models.

[analysis_params] includes the configurations for the analysis, i.e.,

- "variables" is the list of the physical properties to estimate in the Bayesian-like style. The full list of properties can be found in Appendix A. Note that this parameter only affects Bayesian results. The best-fit (least-χ^2) values for all properties are calculated in the results anyway.

- "save_best_sed" can be "True" or "False". If "True", will save the best-fit SED and SFH models for each source.

- "save_chi2" can be "none", "fluxes", "properties", or "all". If "fluxes", will save the raw χ^2 for each photometric band for each source. If "properties", will save χ^2 for "variables" above for each source. If "all", will save χ^2 for both photometric bands and "variables". If "none", will not save χ^2. We provide a PYTHON script "code/read_chi.py" for reading the output χ^2 file (in .npy format).

- "lim_flag" can be "none" "full", or "noscaling". If "none", will discard all upper limits in the input data (see Fig. 1). If "full", will analyze upper limits using exact computation (slow speed). If "noscaling" (default), will use an approximate method to deal with upper limits, which is a good balance between efficiency and reliability.

- "mock_flag" can be "True" or "False". If "True", will create a mock catalog and analyze it. This is a quick way to check if the physical properties can be constrained in a self-consistent way (see §4.3 of Boquien et al. 2019).
- "redshift_decimals" is the number of decimals to round the observed redshifts. To disable rounding give a negative value.
- "blocks" is the number of blocks for the run. The default is 1, which is optimal for speed. But if your computer memory is not enough, you can set it to > 1.

After completing "pcigale.ini", you can run CIGALE with
$ *pcigale run*
Along with this manual, we provide an example configuration file, "examples/sdss_qso/pcigale.ini".

3.3 Results

After the run finishes, you can find the results in the "out/" directory. This directory contains:

- "results.txt" (ASCII format) and "results.fits" (FITS format), the source-property catalog from the fitting.
- "pcigale.ini" and "pcigale.ini.spec", the used configuration files.
- "observations.txt" and "observations.fits", the input observed data.
- "SOURCE ID_best_model.fits" (exist if "save_best_sed" is set to "True"), the best-fit SEDs for SOURCE ID, including the total and different components.
- "SOURCE ID_SFH.fits" (exist if "save_best_sed" is set to "True"), the best-fit SFH for SOURCE ID.
- "SOURCE ID_PROPERTY_chi2-block-BLOCK.npy" (exist if "save_chi2" is set to "True"), the raw χ^2 of PROPERTY for SOURCE ID in BLOCK. We provide a PYTHON script to read these files ("code/read_chi.py").
- "mock_observations.txt", "mock_observations.fits", "results_mock.txt", and "results_mock.fits" (exist if "mock_flag" is set to "True"), the mock catalog and fitting results.

You can visualize the results with *pcigale-plots* command. For example,
$ *pcigale-plots sed*
will generate the best-fit SED plot for each source in pdf format ("out/SOURCE ID_best_model.pdf"). Fig. 3 shows an example SED generated by the *pcigale-plots* command.

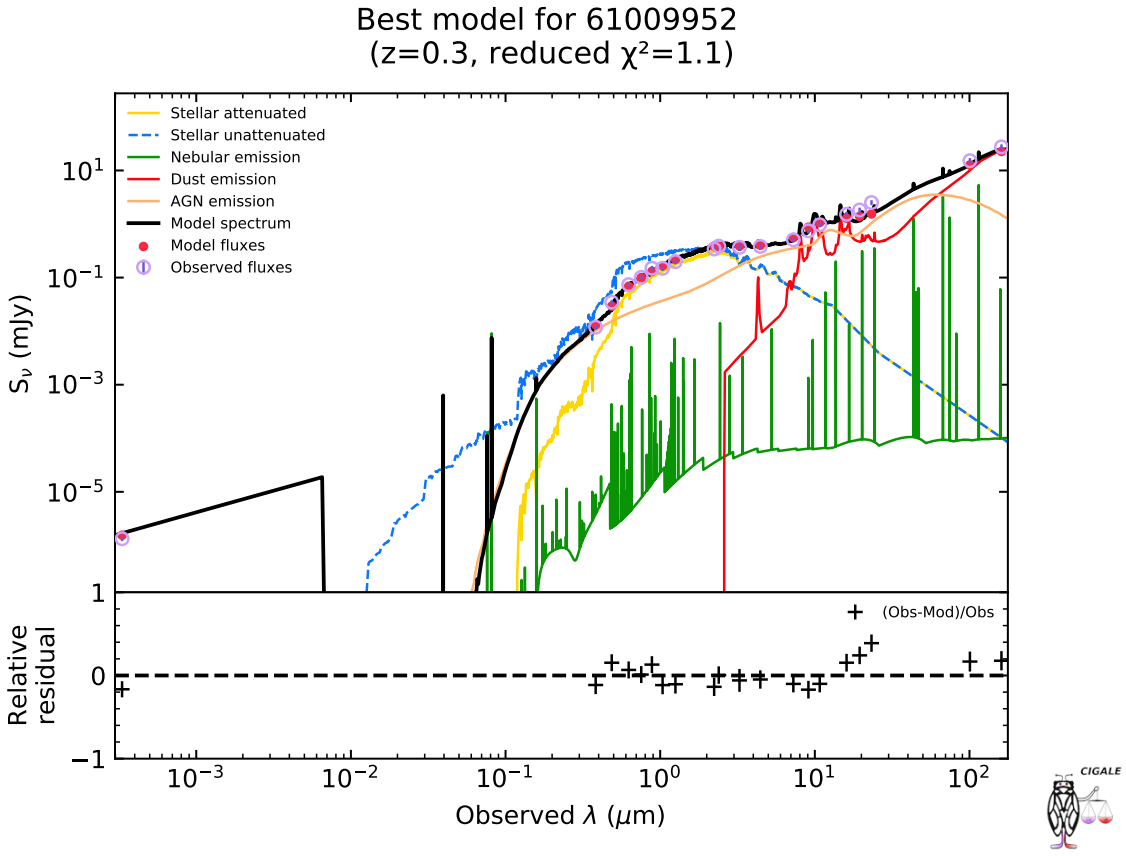

Fig. 3. An example SED generated by the *pcigale-plots* command.

4. Run – Simulating Data: the "savefluxes" mode

X-CIGALE can not only fit the observed data, but also simulate data from a user-defined model set. In this section, we detail the simulation procedures.

4.1 Configurations and Run

Similar as in §3.2, the first step is still the initialization of configuration files. In your working directory, run
$ *pcigale init*
resulting "pcigale.ini" and "pcigale.ini.spec". "pcigale.ini" has five parameters, i.e.

- "data_file" is the input data file, should leave empty when simulating data.
- "parameters_file" is the optional file containing the list of physical parameters. Each column must be in the form module_name.parameter_name, with each line being a different model. If this file is given, then CIGALE will neglect the parameters in [sed_modules_params].
- "sed_modules" lists the names of the SED modules that will be used in the run. The available modules are listed in the commentary parts of the "pcigale.ini" file. The module names should follow the order given in the commentary parts.
- "analysis_method" is CIGALE mode. Should be "savefluxes" for simulation purpose.
- "cores" is the number of CPU cores that will be used. Note that increasing the number of cores may not necessarily boost the speed.

Along with this manual, we provide an example simulation run ("examples/simulate_color"). In the test run, the configuration file reads:

data_file =

parameters_file =

sed_modules = sfhdelayed, bc03, nebular, dustatt_calzleit, dale2014, redshifting

analysis_method = savefluxes

cores = 4

After setting the initial configuration file, run
$ *pcigale genconf*
which will generate the full configuration files "pcigale.ini" and "pcigale.ini.spec".

As in §3.2, you will have to edit "pcigale.ini". Similar in the data-fitting mode §3, "pcigale.ini" has two new parameters, "bands" and "properties". You can key in your interested band and property names, which will appear in the result catalog after the run. Other new

parameters belong to [sed_modules_params] or [analysis_params]. [sed_modules_params] has the same parameters as in the pdf_analysis mode. Note that you must give "redshfit" values in the savefluxes mode, while you can leave it blank to use the redshift values in the input file in the pdf_analysis mode. [analysis_params] only has three parameters, i.e.,

- "variables" is the list of the model physical properties to appear in the results. The full list of properties can be found in Appendix A. You can leave it empty to include all available properties.
- "save_sed" can be "True" or "False". If "True", will save the best-fit SED and SFH models for each simulated source.
- "blocks" is the number of blocks for the run. The default is 1, which is optimal for speed. But if your computer memory is not enough, you can set it to an integer ≥ 2.

After finishing "pcigale.ini", you can run with
$ *pcigale run*
Along with this manual, we provide an example configuration file, "examples/simulate_bzk/pcigale.ini".

4.2 Results

The results are still in the "out/" directory. This directory contains:

- "models-block-0.txt" (ASCII format) and "models-block-0.fits" (FITS format), the simulated source-property catalog from the models.
- "pcigale.ini" and "pcigale.ini.spec", the used configuration files.
- "MODEL ID_best_model.fits" (exist if "save_sed" is set to "True"), the model SEDs for MODEL ID, including the total and different components.
- "MODEL ID_SFH.fits" (exist if "save_sed" is set to "True"), the model SFH for MODEL ID.

Note that the extensive properties (e.g., stellar mass and star formation rate) have not been properly normalized in the results. You might want to normalize by, e.g., stellar mass or flux, before using these quantities.

In our example ("examples/simulate_bzk/"), we simulate the *BzK* color-color diagram (Daddi et al., 2004) as displayed in Fig. 4.

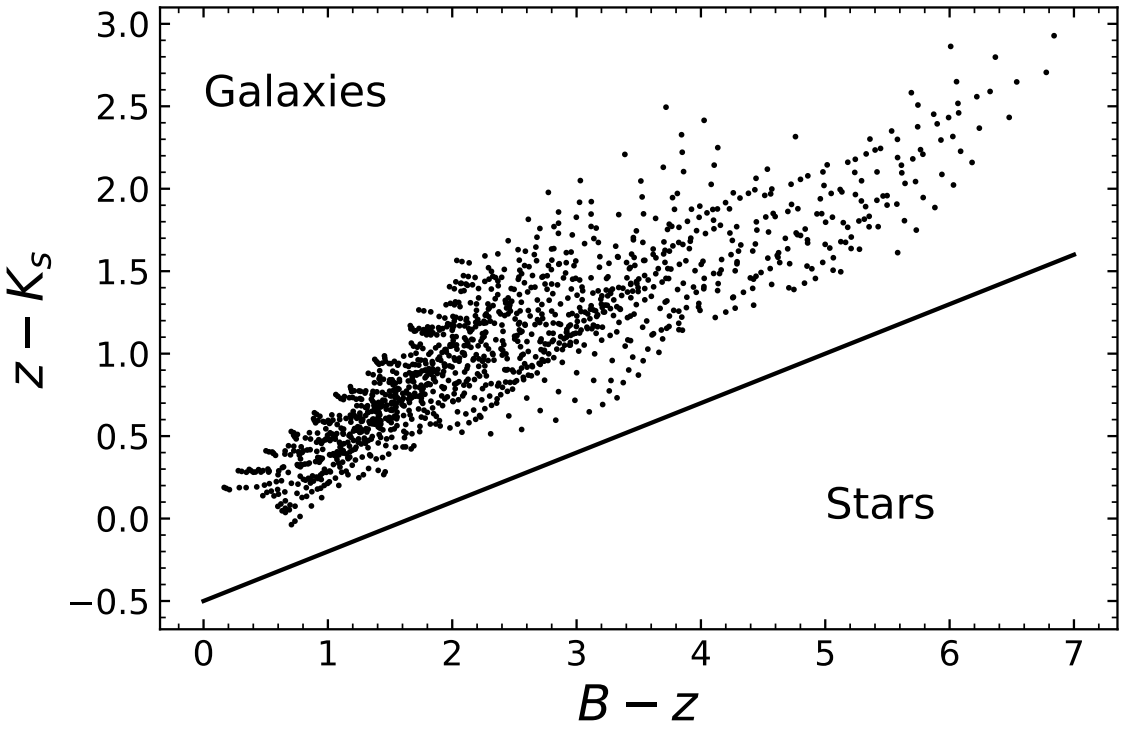

Fig. 4. A BzK diagram simulated in the savefluxes mode. The solid line indicates the empirical separation between galaxies and stars (Daddi et al., 2004).

References

Boquien, M., Burgarella, D., Roehlly, Y., et al. 2019, A&A, 622, A103
Burgarella, D., Buat, V., & Iglesias-Páramo, J. 2005, MNRAS, 360, 1413
Calzetti, D., Armus, L., Bohlin, R. C., et al. 2000, ApJ, 533, 682
Daddi, E., Cimatti, A., Renzini, A., et al. 2004, ApJ, 617, 746
Dale, D. A., Helou, G., Magdis, G. E., et al. 2014, ApJ, 784, 83
Draine, B. T., & Li, A. 2007, ApJ, 657, 810
Draine, B. T., Aniano, G., Krause, O., et al. 2014, ApJ, 780, 172
Fritz, J., Franceschini, A., & Hatziminaoglou, E. 2006, MNRAS, 366, 767
Gaskell, C. M., Goosmann, R. W., Antonucci, R. R. J., & Whysong, D. H. 2004, ApJ, 616, 147
Just, D. W., Brandt, W. N., Shemmer, O., et al. 2007, ApJ, 665, 1004
Noll, S., Burgarella, D., Giovannoli, E., et al. 2009, A&A, 507, 1793
Serra, P., Amblard, A., Temi, P., et al. 2011, ApJ, 740, 22
Yang, G., Boquien, M., Buat, V., et al. 2020, Monthly Notices of the Royal Astronomical Society, 491, 740
Yang, G., Boquien, M., Brandt, W. N., et al. 2022, ApJ, 927, 192

Appendix A: Model Parameters

The main parameters that can be analysed are listed below (see "pcigale.ini" and the output catalog for a full list of parameters) The free parameters that can be set directly in the "pcigale.ini" file are highlighted in blue.

If you wish to estimate the physical parameters in logarithmic, you only have to add "_log" at the end of the name of the parameter, e.g., sfh.burst_age will become sfh.burst_age_log and... le tour est joué!

Table 1. Physical parameters in CIGALE. Free parameters are highlighted in blue.

Module	Parameter	Description
sfh2exp	sfh.tau_main	e-folding [Myr] time of the main stellar population model
...	sfh.tau_burst	e-folding [Myr] time of the late starburst population model
...	sfh.f_burst	Mass fraction of the late burst population (0 to 1)
...	sfh.burst_age	Age [Myr] for the burst
...	sfh.age	Age [Myr] of the oldest stars in the galaxy
...	sfh.sfr	Instantaneous star formation rate
...	sfh.sfr10Myrs	Star formation rate averaged over 10 Myrs
...	sfh.sfr100Myrs	Star formation rate averaged over 100 Myrs
...	sfh.integrated	Star formation rate integrated from the star formation history
sfhdelayed	sfh.tau_main	e-folding [Myr] time of the main stellar population model
...	sfh.age	Age [Myr] of the oldest stars in the galaxy
...	sfh.sfr	Instantaneous star formation rate
...	sfh.sfr10Myrs	Star formation rate averaged over 10 Myrs
...	sfh.sfr100Myrs	Star formation rate averaged over 100 Myrs
...	sfh.integrated	Star formation rate integrated from the star formation history
sfhperiodic	sfh.delta_bursts	Elapsed time between the beginning of each burst in Myr.

...	sfh.tau_bursts	Duration (rectangle) or e-folding time of all short events in Myr.
...	sfh.integrated	Star formation rate integrated from the star formation history
sfhfromfile	sfh.id	id of the input SFH
...	sfh.sfr	Instantaneous star formation rate
...	sfh.sfr10Myrs	Star formation rate averaged over 10 Myrs
...	sfh.sfr100Myrs	Star formation rate averaged over 100 Myrs
...	sfh.integrated	Star formation rate integrated from the star formation history
m2005	stellar.imf	IMF of the stellar model
...	stellar.metallicity	Metallicity of the stellar model
...	stellar.old_young_separation_age	Age of the separation old/young stars
...	stellar.mass_total_old	Stellar mass of old stars
...	stellar.mass_alive_old	Stellar mass of old stars alive
...	stellar.mass_total_young	Stellar mass of young stars
...	stellar.mass_alive_young	Stellar mass of young stars alive
...	stellar.mass_total	Total stellar mass of stars
...	stellar.mass_alive	Total stellar mass alive
bc03	stellar.imf	IMF of the stellar model
...	stellar.metallicity	Metallicity of the stellar model
...	stellar.old_young_separation_age	Age of the separation old/young stars
...	stellar.m_star_young	Stellar mass of young stellar population
...	stellar.n_ly_young	Number of Ly continuum photons from young stellar population
...	stellar.m_star_old	Stellar mass of old stellar population
...	stellar.n_ly_old	Number of Ly continuum photons from old stellar population
...	stellar.m_star	Total mass of stars
nebular	nebular.f_esc	Fraction of Lyman continuum photons escaping the galaxy
...	nebular.f_dust	Fraction of Lyman continuum photons absorbed by dust
...	nebular.logU	Ionisation parameter

dustatt_calzleit	attenuation.uv_bump_amplitude	Amplitude of the UV bump. For the Milky Way: 3
...	attenuation.powerlaw_slope	Slope delta of the power law modifying the attenuation curve
...	attenuation.E_BVs.stellar.young	E(B-V) of the young stellar population. Note that E(B-V) is an internal parameter which does not correspond to E(B-V) except for the exact calzetti law (delta=0), E(B-V)= $A_B - A_V$ should be calculated by the user.
...	attenuation.ebvs_old_factor	Reduction factor of E(B-V) for the old population compared to the young one
...	attenuation.E_BVs.stellar.old	E(B-V) of the old stellar population. Note that E(B-V) is an internal parameter which does not correspond to E(B-V) except for the exact calzetti law (delta=0), E(B-V)= $A_B - A_V$ should be calculated by the user.
...	attenuation.(filter)	Attenuation in a given filter. This filter (e.g., FUV, B, V,...) must be provided to CIGALE.
dustatt_powerlaw	attenuation.uv_bump_amplitude	Amplitude of the UV bump. For the Milky Way: 3
...	attenuation.powerlaw_slope	Slope delta of the power law modifying the attenuation curve
...	attenuation.Av.stellar.young	V-band attenuation of the young population
...	attenuation.av_old_factor	Reduction factor of A_V for the old population compared to the young one
...	attenuation.Av.stellar.young	V-band attenuation of the old population
...	attenuation.(filter)	Attenuation in a given filter. This filter (e.g., FUV, B, V,...) must be provided to CIGALE
dl2014	dust.umin	Parameter U_min in Draine & Li (2007) templates

...	dust.alpha	Parameter alpha_max in Draine & Li (2007) templates
...	dust.gamma	Parameter gamma in Draine & Li (2007) templates
...	dust.qpah	Parameter q_{PAH} in Draine et al. (2014) updated templates
...	dust.luminosity	Estimated dust luminosity using an energy balance
dale2014	agn.fracAGN_dale2014	AGN fraction. Note that the AGN is a type 1
...	dust.alpha	Parameter $alpha_{max}$ in Dale et al. (2014) templates
...	dust.luminosity	Estimated dust luminosity using an energy balance
fritz2006	agn.gamma	Parameter gamma in Fritz et al. (2006)
...	agn.opening_angle	Full opening angle of the dust torus (Fig 1 of Fritz et al. (2006))
...	agn.psy	Angle between AGN axis and line of sight
...	agn.fracAGN	Fraction of AGN IR luminosity to total IR luminosity
...	agn.r_ratio	Ratio of the maximum to minimum radii of the dust torus
...	agn.tau	Torus optical depth at 9.7 microns
...	agn.beta	Parameter beta in Fritz et al. (2006)
...	agn.law	The extinction law of polar dust: 0 (SMC), 1 Calzetti et al. (2000), or 2 Gaskell et al. (2004)
...	agn.EBV	E(B-V) for extinction in polar direction
...	agn.temperature	Temperature of the polar dust in K
...	agn.emissivity	Emissivity index of the polar dust
...	agn.disk_luminosity	The AGN disc luminosity (might be extincted)
...	agn.therm_luminosity	The AGN dust reemitted luminosity
...	agn.scatt_luminosity	The AGN scattered luminosity
...	agn.luminosity	The sum of agn.disk_luminosity, agn.therm_luminosity, and agn.scatt_luminosity

...	agn.intrin_Lnu_2500A	The intrinsic AGN L_ν at 2500 Å
...	agn.accretion_power	The intrinsic AGN disk luminosity averaged over all directions
skirtor2016	agn.t	Average edge-on torus optical depth at 9.7 micron
...	agn.pl	Power-law exponent that sets radial gradient of dust density
...	agn.q	Index that sets dust density gradient with polar angle
...	agn.oa	Angle measured between the equatorial plan and edge of the torus
...	agn.R	Ratio of outer to inner radius, R_out/R_in
...	agn.i	Viewing angle. i=[0, 90°-oa): face-on, type 1 view; i=[90°-oa, 90°]: edge-on, type 2 view
...	agn.fracAGN	Fraction of AGN IR luminosity to total IR luminosity
...	agn.law	The extinction law of polar dust: 0 (SMC), 1 Calzetti et al. (2000), or 2 Gaskell et al. (2004)
...	agn.EBV	E(B-V) for extinction in polar direction
...	agn.temperature	Temperature of the polar dust in K
...	agn.emissivity	Emissivity index of the polar dust
...	agn.disk_luminosity	The observed AGN disc luminosity (might be extincted)
...	agn.dust_luminosity	The observed AGN dust reemitted luminosity
...	agn.luminosity	The sum of agn.disk_luminosity and agn.dust_luminosity
...	agn.intrin_Lnu_2500A	The intrinsic AGN L_ν at 2500 Å at viewing angle = 30°
...	agn.accretion_power	The intrinsic AGN disk luminosity averaged over all directions
xray	xray.gam	The photon index (Gamma) of AGN intrinsic X-ray spectrum
...	xray.max_dev_alpha_ox	Maximum deviation from the α_{ox}-$L_{2500Å}$ relation in Just et al. (2007)

...	xray.gam_lmxb	The photon index of AGN low-mass X-ray binaries
...	xray.gam_hmxb	The photon index of AGN high-mass X-ray binaries
...	xray.agn_Lnu_2keV	The AGN L_ν at 2 keV
...	xray.agn_Lx_2to10keV	The AGN 2–10 keV luminosity
...	xray.agn_Lx_total	The AGN total (0.25–1200 keV) X-ray luminosity
...	xray.alpha_ox	The AGN α_{ox}
...	xray.lmxb_Lx_2to10keV	The 2–10 keV LMXB luminosity
...	xray.hmxb_Lx_2to10keV	The 2–10 keV HMXB luminosity
...	xray.hotgas_Lx_0p5to2keV	The 0.5–2 keV hot-gas luminosity
radio	radio_qir	FIR/radio ratio
...	radio_alpha	slope of the power-law synchrotron emission
redshifting	universe.redshift	redshift
...	universe.luminosity_distance	Luminosity distance
...	universe.age	Age of the universe

Appendix B: Supplementary Files

The following PYTHON codes are available in the folder "code" along with this manual.

- "code/convert_Fx.py" contains a function that converts X-ray flux (erg s^{-1} cm^{-2}) to flux density (mJy), which can be used as CIGALE input (see §3.1.1).
- "code/read_chi.py" contains two functions that read the raw χ^2 .npy files in CIGALE outputs (see §3.3). One function, GET_CIGALE_PROB, read the χ^2 file for one parameter to plot the 1D probability density function (PDF). The other function, GET_CIGALE_PROB_2D, read the χ^2 files for two parameter to plot the 2D PDF.
- "code/xray_filter.py" contains a function that writes a boxcar-shaped X-ray filter that can be used by CIGALE (see §3.1.1).

The following CIGALE example runs are available in the folder "examples" along with this manual.

- "examples/akari_nep_xray_agn" contains the configuration and data files for the X-ray selected AGNs in the AKARI-NEP field (see §3.3 of Yang et al. 2020).
- "examples/sdss_qso" contains the X-ray detected AGNs in the SDSS DR14 QSO catalog (see §3.1 of Yang et al. 2020).
- "examples/simulate_bzk" contains the configuration files for the simulation of the *BzK* diagram (Fig. 4).