Using CIGALE to Fit SEDs and Spectra

Practical manual with a worked example. Generated on 2 August 2025

1. What is CIGALE?

CIGALE (Code Investigating GALaxy Emission[1]) is an open-source SED-fitting framework that builds physically motivated galaxy SEDs from X-rays to radio, compares them to observations (spectroscopic + photometric = spectrophotometric data), and derives Bayesian estimates of physical parameters (e.g., SFR, M⋆, attenuation). CIGALE can also just create modeled spectra from the various available science modules. CIGALE supports flexible star-formation histories, stellar population models, nebular emission, dust attenuation/emission, and more. While CIGALE's core workflow is designed for broadband photometry, spectra can now also be used by turning them into sets of narrow pseudo-filters ("narrow-band photometry") so they are handled consistently with the photometric likelihood. Fitting spectroscopy is straightforward and the building of the above pseudo-filters is transparent for the user.

Tip: CIGALE is distributed as the Python package "pcigale". In command-line examples below, the executable is named pcigale.

2. Installation

Recommended (conda-forge):

```
# New environment (recommended)
conda create -n cigale -c conda-forge python=3.11 cigale
conda activate cigale

# Verify installation & available commands
pcigale --version
pcigale-filters --help
pcigale-plots --help
```

3. The standard CIGALE workflow

CIGALE reads a flux table (catalog) and a configuration file (pcigale.ini), builds a model grid, then evaluates the likelihood in a Bayesian framework. Best-fit parameters are also provided in the output file.

a) Prepare your flux table (FITS or ASCII). It must contain one object per row with at least an identifier and redshift.

- Columns are the filter names and their uncertainties (one column per band and one column for its uncertainty).

[1] https://cigale.lam.fr/

- Flux densities should be consistent units across columns (mJy).
- Uncertainty columns typically use the same name with a suffix (e.g., “_err”).
- Example minimal header: id, redshift, sdss.u, sdss.u_err, sdss.g, sdss.g_err, …

b) Make sure the filter transmission curves referenced in your table are available (help: “pcigale-filters –h”).

- List known filters: pcigale-filters list
- Add or show a filter: pcigale-filters add <path_to_filter_file>
- pcigale-filters plot <name>
- Filter files are simple two-column ASCII (wavelength in Angstroms vs transmission).

c) Initialize a configuration in your working directory.

pcigale init

```
# This creates two files:
```

- `pcigale.ini and a "modules" section that you will need to edit.`
- `Pcigale.ini.spec that contains information on the type of parameters that you will never need to edit.`

d) Edit pcigale.ini: choose modules and parameter grids.

A common starting set: delayed SFH + SSP + nebular + dust attenuation + IR dust emission + redshift + restframe parameters.

Here is an example of pcigale.ini for spectrophotometric data (your input).

```
# File containing the input data. The columns are 'id' (name of the
# object), 'redshift' (if 0 the distance is assumed to be 10 pc),
# 'distance' (Mpc, optional, if present it will be used in lieu of the
# distance computed from the redshift), the filter (for instance FUV) or
# line names (prefixed with 'line.' for the latter, for instance
# line.H-alpha) for the fluxes, and the names of the physical properties
# as computed by the code (e.g. dust.luminosity). The uncertainties are
# given with the same column names with the '_err' suffix. The fluxes
# and the uncertainties must be in mJy for broadband data, in W/m² for
# emission lines, and the unit computed by the code for the physical
# properties. Fluxes can be positive or negative. Upper limits are
# indicated with a negative value for the uncertainty. In case some
# fluxes are missing for some entries, they can be replaced with NaN.
# This file is optional to generate the configuration file, in
# particular for the savefluxes module. If spectral data also used in
# the fit, the columns: 'spec_name', 'disperser', and 'norm_method'
# should be in the input datafile.
data_file = test.fits

# Is there any spectroscopic data to analyze? This spectroscopic
# datasetwill be used in conjonction with any photometric data,
# and/orand/or equivalents widths, line fluxes or other properties.The
# answer must be: True or False
use_spectro = True
# File containing the information on the spectral resolution of
# theinstrument used in spectroscopy. It should contain three columns: -
# Wavelength, in Angstroems, - Dlambda, in Angstroems per pixel, - R:
```

```
# spectral resolution.If no spectral data used in the fit (use_spectrois
# False), this file is not read.For instance:
# jwst_nirspec_prism_disp.fits or vlt_vimos_disp.fits
spectral_res_file = jwst_nirspec_prism_disp.fits

# 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. The columns must be in the order the modules will be
# called. The redshift column must be the last one. Finally, if this
# parameter is not empty, cigale will not interpret the configuration
# parameters given in pcigale.ini. They will be given only for
# information. Note that this module should only be used in conjonction
# with the savefluxes module. Using it with the pdf_analysis module will
# yield incorrect results.
parameters_file =

# Available modules to compute the models. The order must be kept.
# SFH:
# * sfh_stochasticity_physgaussproc (stochatic SFH with physically motivated
Gaussian processes )
# * sfh2exp (double exponential)
# * sfhdelayed (delayed SFH with optional exponential burst)
# * sfhdelayedbq (delayed SFH with optional constant burst/quench)
# * sfhfromfile (arbitrary SFH read from an input file)
# * sfhperiodic (periodic SFH, exponential, rectangle or delayed)
# SSP:
# * bc03 (Bruzual and Charlot 2003)
# * m2005 (Maraston 2005; note that it cannot be combined with the nebular
module)
# Nebular emission:
# * nebular (continuum and line nebular emission)
# Dust attenuation:
# * dustatt_modified_CF00 (modified Charlot & Fall 2000 attenuation law)
# * dustatt_modified_starburst (modified Calzetti 2000 attenuaton law)
# Dust emission:
# * casey2012 (Casey 2012 dust emission models)
# * dale2014 (Dale et al. 2014 dust emission templates)
# * dl2007 (Draine & Li 2007 dust emission models)
# * dl2014 (Draine et al. 2014 update of the previous models)
# * themis (Themis dust emission models from Jones et al. 2017)
# AGN:
# * skirtor2016 (AGN models from Stalevski et al. 2012, 2016)
# * fritz2006 (AGN models from Fritz et al. 2006)
# X-ray:
# * xray (from AGN and galaxies; skirtor2016/fritz2006 is needed for AGN)
# Radio:
# * radio (galaxy synchrotron emission and AGN; skirtor2016/fritz2006 is needed
for AGN)
# Restframe parameters:
# * restframe_parameters (UV slope (β), IRX, D4000, EW, etc.)
# Redshift+IGM:
# * redshifting (mandatory, also includes the IGM from Meiksin 2006)
sed_modules = sfhdelayed, bc03, nebular, dustatt_modified_starburst,
redshifting

# Method used for statistical analysis. Available methods: pdf_analysis,
# savefluxes.
analysis_method = pdf_analysis

# Number of CPU cores available. This computer has 24 cores.
cores = 24
```

e) Generate the full configuration templates for chosen modules, then set parameter grids.

pcigale genconf

```
# Edit the newly created parameter files under pcigale.ini (module sections)

# This phase can optionally normalize the observed spectra to the photometric
data. Three options are possible: none:  normalization, global: a chi2-
estimation single value and wave: the normalization is estimated by assuming
polynomials with deg = 0, 1, 2 and testing which one better fits the integrated
flux density of the observed spectrum to the photometric data by a BIC test.
Note: if flag_spec2phot is set to True in pcigale/utils/read_prism.py, you will
get files showing the spec2phot re-normalization, sometimes called slit loss.

This information on the spec2phot normalization will appear on the screen:
```

- Id: ./prism/example_z6p01.fits Chosen model: deg2 BIC: 80.10181932378609 N_bands: 11 Coeffs: [-0.18132695713037714, 0.17525921464438315, 7.03455628933782] lam0: 1921.0884787979167
- Id: ./prism/example_z8p62.fits Chosen model: deg0 BIC: nan N_bands: 0 Coeffs: [0.0] lam0: 26857.25512385658
- Id: ./prism/example_z4p59.fits Chosen model: deg2 BIC: 29.878847010002247 N_bands: 11 Coeffs: [-0.25058683439929985, 0.13292923118302927, 7.570865558660522] lam0: 1921.0884787979167
- Id: ./prism/example_z4p67.fits Chosen model: deg0 BIC: nan N_bands: 0 Coeffs: [0.0] lam0: 27367.35787430304
- Id: ./prism/example_z11p42.fits Chosen model: deg1 BIC: 46.43063074766499 N_bands: 7 Coeffs: [-0.3752301551628903, 6.995947358710403] lam0: 2605.0615727662234

Only 3 out of the 5 objects have photometry. For those without photometry (and with "none" in the input file, there is no normalization applied. We thus get 3 normalization files only:

- spec2phot_id_z6p01_20250825T160522Z.dat
- spec2phot_id_z4p59_20250825T160522Z.dat
- spec2phot_id_z11p42_20250825T160522Z.dat

```
# This phase creates a new input file test.dat from the initial test.fits. You
need to replace test.fits -> test.dat in pcigale.ini. The reason is that fits
files can only handle 999 columns while the new pseudo-filters create for
spectrophotometric data can need more columns. Note that this is a temporary
solution and we will update this to get a more transparent methos.
```

To the end of the above file (with a blue background), we now find (only what changed with respect to the initial pcigale.ini after **pcigale genconf** is shown in **blue**):

```
…

# Number of CPU cores available. This computer has 24 cores.
cores = 24

# Bands and emission lines to consider. To take uncertainties into
# account, the name of the band or line must be indicated with the _err
# suffix as in the input flux file. For instance: FUV, FUV_err. The list
# of allowed bands and lines can be obtained with pcigale-filters list.
# Line names must be prefixed with 'line.', for instance line.H-alpha,
# as in the input flux file.
bands = [long list of columns automatically filled: filter, filter_err including photometric filters
and a list of spectroscopic pseudo-filters, e.g., hst.wfc.F606W, hst.wfc.F606W_err,
hst.wfc.F814W, hst.wfc.F814W_err, HST-WFC3_IR.F105W, HST-
WFC3_IR.F105W_err, jwst.nircam.F115W, jwst.nircam.F115W_err,
hst.wfc3.F125W, hst.wfc3.F125W_err, hst.wfc3.F140W, hst.wfc3.F140W_err,
jwst.nircam.F150W, jwst.nircam.F150W_err, hst.wfc3.F160W,
hst.wfc3.F160W_err, jwst.nircam.F200W, jwst.nircam.F200W_err,
jwst.nircam.F277W, jwst.nircam.F277W_err, jwst.nircam.F356W,
jwst.nircam.F356W_err, jwst.nircam.F410M, jwst.nircam.F410M_err,
jwst.nircam.F444W, jwst.nircam.F444W_err, prism_Band_000,
prism_Band_000_err, prism_Band_001, prism_Band_001_err, …
prism_Band_510, prism_Band_510_err, prism_Band_511, prism_Band_511_err]

# Properties to be considered. All properties are to be given in the
# rest frame rather than the observed frame. This is the case for
# instance the equivalent widths and for luminosity densities. The names
# correspond to those used in the input file with the uncertainties
# indicated with the _err suffix.
properties =

# Relative error added in quadrature to the uncertainties of the fluxes
# and the extensive properties.
additionalerror = 0.1


# Configuration of the SED creation modules.
[sed_modules_params]

  [[sfhdelayed]]
  # e-folding time of the main stellar population model in Myr.
  tau_main = 500., 2000.0, 10000.0
  # Age of the main stellar population in the galaxy in Myr. The precision
  # is 1 Myr.
  age_main = 100, 500, 1000, 2500, 5000, 10000
  # e-folding time of the late starburst population model in Myr.
  tau_burst = 10000.0
  # Age of the late burst in Myr. The precision is 1 Myr.
  age_burst = 20
  # Mass fraction of the late burst population.
  f_burst = 0.0, 0.1, 0.2
  # Multiplicative factor controlling the SFR if normalise is False. For
  # instance without any burst: SFR(t)=sfr_A×t×exp(-t/τ)/τ²
  sfr_A = 1.0
  # Normalise the SFH to produce one solar mass.
  normalise = True

 [[bc03]]
  # Initial mass function: 0 (Salpeter) or 1 (Chabrier).
  imf = 1
  # Metalicity. Possible values are: 0.0001, 0.0004, 0.004, 0.008, 0.02,
  # 0.05.
  metallicity = 0.004, 0.02    # Age [Myr] of the separation between the young and the old star
  # populations. The default value in 10^7 years (10 Myr). Set to 0 not to
  # differentiate ages (only an old population).
  separation_age = 10

 [[nebular]]
  # Ionisation parameter. Possible values are: -4.0, -3.9, -3.8, -3.7,
  # -3.6, -3.5, -3.4, -3.3, -3.2, -3.1, -3.0, -2.9, -2.8, -2.7, -2.6,
  # -2.5, -2.4, -2.3, -2.2, -2.1, -2.0, -1.9, -1.8, -1.7, -1.6, -1.5,
  # -1.4, -1.3, -1.2, -1.1, -1.0.
  logU = -4.0, -2.0,
  # Gas metallicity. Possible values are: 0.0001, 0.0004, 0.001, 0.002,
  # 0.0025, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.011, 0.012,
  # 0.014, 0.016, 0.019, 0.020, 0.022, 0.025, 0.03, 0.033, 0.037, 0.041,
  # 0.046, 0.051.
  zgas = 0.004, 0.02
  # Electron density. Possible values are: 10, 100, 1000.
  ne = 100
  # Fraction of Lyman continuum photons escaping the galaxy. Possible
  # values between 0 and 1.
  f_esc = 0.0
  # Fraction of Lyman continuum photons absorbed by dust. Possible values
  # between 0 and 1.
  f_dust = 0.0
  # Line width in km/s.
  lines_width = 300.0
  # Include nebular emission.
  emission = True

 [[dustatt_modified_starburst]]
  # E(B-V)l, the colour excess of the nebular lines light for both the
  # young and old population.
  E_BV_lines =  1e-6, 1e-5, 1e-4, 0.001, 0.005, 0.010, 0.10, 0.25, 0.50
  # Reduction factor to apply on E_BV_lines to compute E(B-V)s the stellar
  # continuum attenuation. Both young and old population are attenuated
  # with E(B-V)s.
  E_BV_factor = 0.44
  # Central wavelength of the UV bump in nm.
  uv_bump_wavelength = 217.5
  # Width (FWHM) of the UV bump in nm.
  uv_bump_width = 35.0
  # Amplitude of the UV bump. For the Milky Way: 3.
  uv_bump_amplitude = 0.0
  # Slope delta of the power law modifying the attenuation curve.
  powerlaw_slope = -0.60, -0.30, 0.00, 0.30, 0.60
  # Extinction law to use for attenuating the emissio  n lines flux.
  # Possible values are: 1, 2, 3. 1: MW, 2: LMC, 3: SMC. MW is modelled
  # using CCM89, SMC and LMC using Pei92.
  Ext_law_emission_lines = 1
  # Ratio of total to selective extinction, A_V / E(B-V), for the
  # extinction curve applied to emission lines.Standard value is 3.1 for
  # MW using CCM89, but can be changed.For SMC and LMC using Pei92 the
```

```
  # value is automatically set to 2.93 and 3.16 respectively, no matter
  # the value you write.
  Rv = 3.1
  # Filters for which the attenuation will be computed and added to the
  # SED information dictionary. You can give several filter names
  # separated by a & (don't use commas).
  filters = B_B90 & V_B90 & FUV

 [[redshifting]]
  # Redshift of the objects. Leave empty to use the redshifts from the
  # input file.
  redshift =


# Configuration of the statistical analysis method.
[analysis_params]
 # List of the physical properties to estimate. Leave empty to analyse
 # all the physical properties (not recommended when there are many
 # models).
 variables = attenuation.FUV, attenuation.E_BV_lines, stellar.m_star_log, stellar.metallicity_log, nebular.zgas_log, nebular.logU, sfh.sfr, sfh.sfr100Myrs, sfh.sfr10Myrs, param.restframe_Lnu(FUV)_log
 # List of bands for which to estimate the fluxes. Note that this is
 # independent from the fluxes actually fitted to estimate the physical
 # properties.
 bands =
 # If true, save the best SED for each observation to a file.
 save_best_sed = True
 # Save the raw chi2. It occupies ~15 MB/million models/variable. Allowed
 # values are 'all', 'none', 'properties', and 'fluxes'.
 save_chi2 = none
 # Take into account upper limits. If 'full', the exact computation is
 # done. If 'noscaling', the scaling of the models will not be adjusted
 # but the χ^2 will include the upper limits adequately. Waiving the
 # adjustment makes the fitting much faster compared to the 'full' option
 # while generally not affecting the results in any substantial manner.
 # This is the recommended option as it achieves a good balance between
 # speed and reliability. Finally, 'none' simply discards bands with
 # upper limits.
 lim_flag = noscaling
 # If true, for each object we create a mock object and analyse them.
 mock_flag = True
 # When redshifts are not given explicitly in the redshifting module,
 # number of decimals to round the observed redshifts to compute the grid
 # of models. To disable rounding give a negative value. Do not round if
 # you use narrow-band filters.
 redshift_decimals = 2
 # Number of blocks to compute the models and analyse the observations.
 # If there is enough memory, we strongly recommend this to be set to 1.
 blocks = 1
```

f) Check, then run.

Note: the previous configuration will run woth only 97 200 models. A more typical run would use several millions to several hundred million of models, depending on your computer.

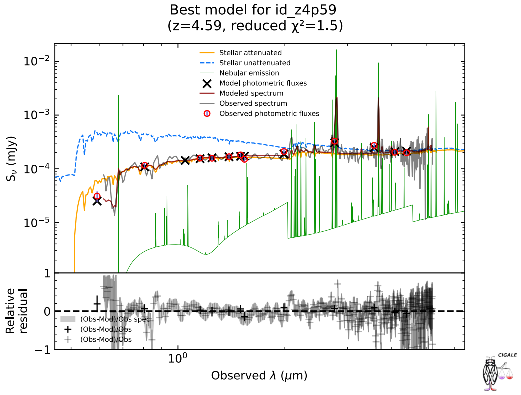
```
pcigale check    # validates filters, table and configuration
pcigale run      # builds grid, computes likelihoods & results
```

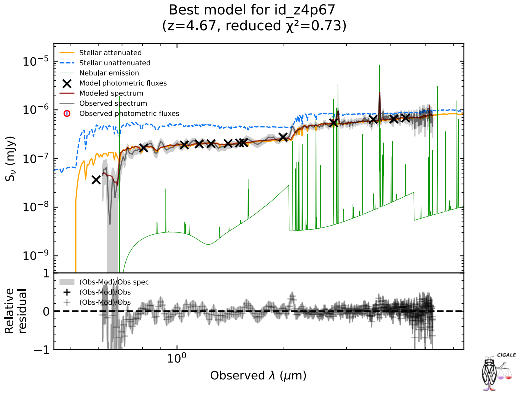
`*** Remember: replace test.fits -> test.dat in pcigale.ini`
`if you use spectroscopic data   ***`

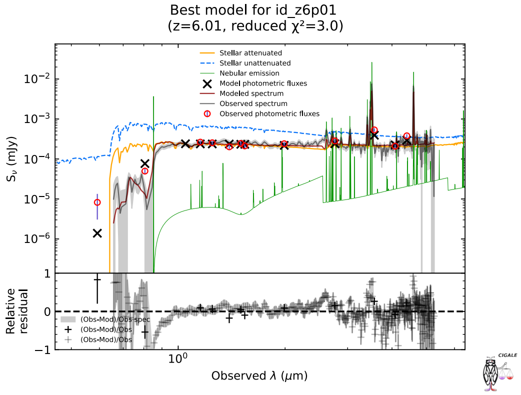
`data_file = test.dat`

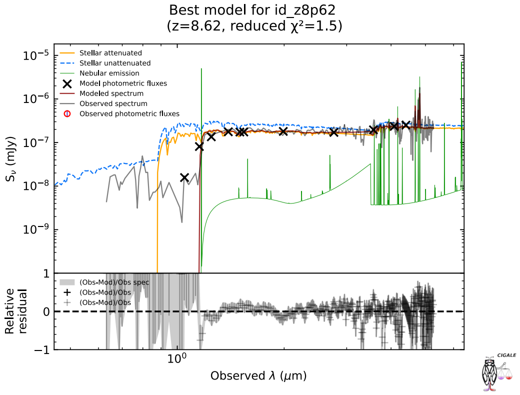
g) Inspect outputs.

- results.fits / results.txt: Bayesian estimates for each object.
- best_model: per-object best SED and component contributions (if enabled).
- Plots: **pcigale-plots sed** (creates SED figures for each object).
- Plots: **pcigale-plots mock** (creates plots that compare input and derived parameters for mock catalogs).









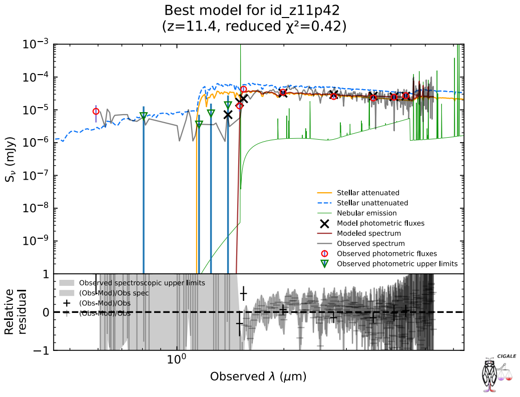


Figure 1: results of pcigale-plots sed

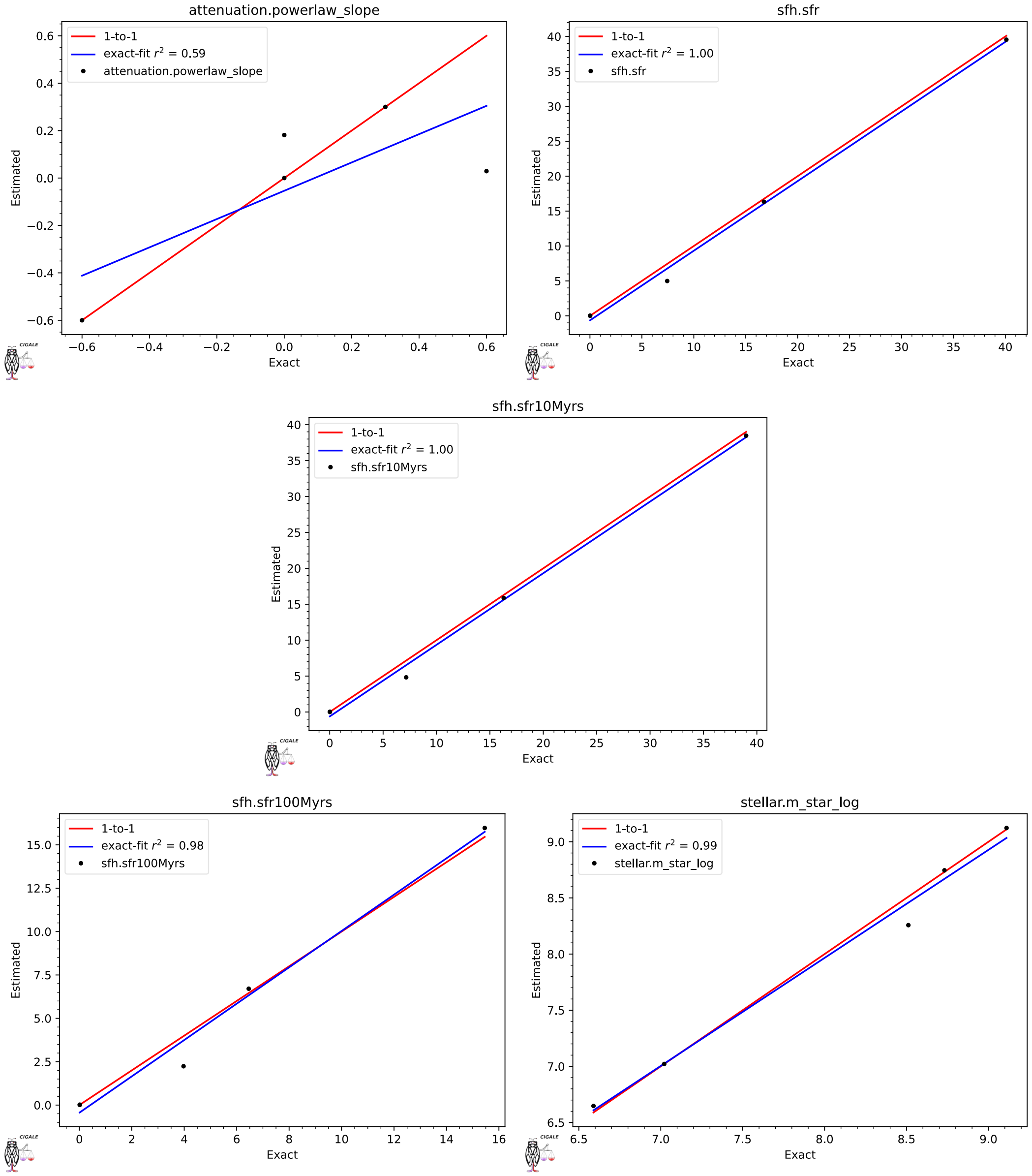


Figure 2: results of pcigale-plots mock

4. Minimal working example (toy)

You can find ll the files to reproduce this run on this link:
https://nuage.osupytheas.fr/s/AP42yYKprA2WZzs

Including an example of the dispersion file from JWST/NIRSpec-prism from here.

5. References & further reading

- Burgarella et al. 2005MNRAS.360.1413B : https://ui.adsabs.harvard.edu/abs/2005MNRAS.360.1413B/abstract
- Boquien, M., et al. 2019A&A...622A.103B : https://ui.adsabs.harvard.edu/abs/2019A%26A...622A.103B/abstract
- Burgarella D. et al. 2025, 2025A&A...699A.336B : https://ui.adsabs.harvard.edu/abs/2025A%26A...699A.336B/abstract
- Statistics on the use of CIGALE from ADS with the name CIGALE in the abstract since 2019