CIGALE How To – v0.9.0 22 Mar 2016

This is the log of a typical session that you might run using CIGALE (http://cigale.lam.fr) that illustrates SED fitting but also spectral modelling.

22 March 2016
D. Burgarella (denis.burgarella@lam.fr)

---

0. Download CIGALE v0.9.0 (or the latest version) from http://cigale.lam.fr. Some documentation with more details on the modules is available online : http://cigale.lam.fr/pcigale/pcigale_manual.html

1. Decompress CIGALE : cigale_v0.8.0_22Mar2016.tar.gz (or the latest version)

2. We assume you have Python 3.4 or higher working on your computer. We recommend using *ANACONDA* to install *Python 3*:

   <https://store.continuum.io/cshop/anaconda/>

3. Once *ANACONDA is downloaded*, you type successively the following command in a BASH terminal:

   - You have to ***install anaconda*** (this is machine-dependent and you need to read the recommendations on ANACONDA's website)
   - ***bash***
   - ***conda update conda***
   - ***conda update anaconda***
   - ***conda create -n py3 python=3.4 anaconda*** for Python 3.4
   - ***source activate py3*** : you should now see a new prompt « (py3)$ »
   - ***conda install astropy numpy scipy sqlite sqlalchemy matplotlib configobj***

4. ***cd pcigale*** (or whatever the name or the directory is) will take you to the CIGALE directory

5. If you want to add your own filters for the SED analysis or the creation of models in filters, it is the correct time to do so. A minimum list of filter transmissions is provided with the CIGALE distribution downloaded. However, we provide a list of filters in this CIGALE_How_To that you can add/replace into the directory: *./database_builder/filters/.*

6. ***(py3)$ Python setup.py build*** will build the database in ./pcigale/data/data.db. Note that if you already have this data.db file in the directory, you need to save it under another name (if you wish) and to delete it before re-building.

7. ***(py3)$ Python setup.py develop*** will now install CIGALE.

8. *(py3)$ cd Data*, now, you can move to any working directory (Data) where your data are in this example. **It is important to stress that you must not remove the above pcigale directory or CIGALE will no longer works.**

In Fig. 1, you can find some important information on how to manage the flux densities, uncertainties and upper limits.

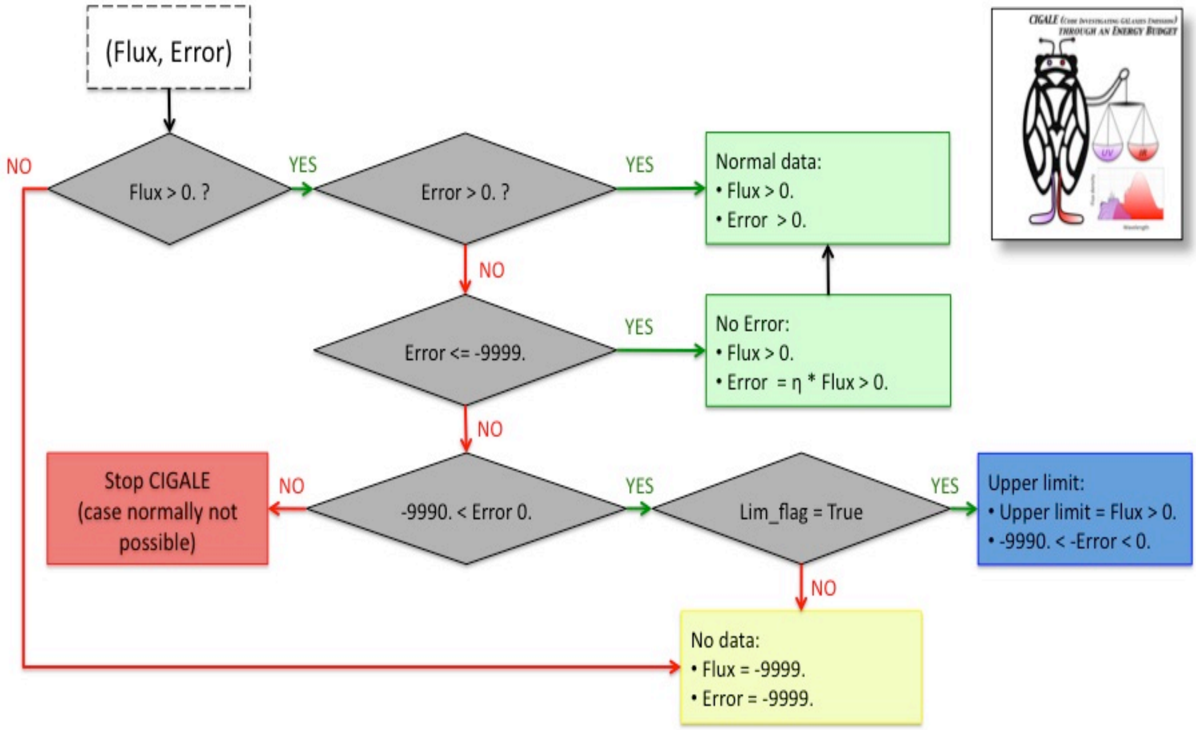


Figure 1 : This figure presents how CIGALE manages (Flux, Error) couples for each filter.

9. *(py3)$ pcigale init* will initialize a configuration file called pcigale.ini shown below. You do not need any input file. And you will have a message from CIGALE:

The initial configuration file was created. Please complete it with the data file name and the pcigale modules to use.

```
# File containing the input data. The columns are 'id' (name of the
# object), 'redshift' (if 0 the distance is assumed to be 10 pc), the
# filter names for the fluxes, and the filter names with the '_err'
# suffix for the uncertainties. The fluxes and the uncertainties must be
# in mJy. This file is optional to generate the configuration file, in
# particular for the savefluxes module.
data_file =

# Optional file containing the list of physical parameters. Each column
# must be in the form module_name.parameter_name, with each line behind
# 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
# parameters is not left empty, cigale will not interpret the
# configuration parameters given in pcigale.ini. They will be given only
# for information.
parameters_file =
```

```
# Order of the modules use for SED creation. Available modules:
# SFH: sfh2exp, sfhdelayed, sfhfromfile, sfhperiodic
# SSP: bc03, m2005
# Nebular emission: nebular
# Dust attenuation: dustatt_calzleit, dustatt_powerlaw
# Dust emission: casey2012, dale2014, dl2007, dl2014
# AGN: dale2014, fritz2006
# Radio: radio
# Redshift: redshifting (mandatory!)
sed_modules = ,

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

# Number of CPU cores available. This computer has 4 cores.
cores =
```

Using a text editor, you need now to ***define the configuration*** and save pcigale.ini. It should resemble (in this example) to the exemple below.

```
# File containing the input data. The columns are 'id' (name of the
# object), 'redshift' (if 0 the distance is assumed to be 10 pc), the
# filter names for the fluxes, and the filter names with the '_err'
# suffix for the uncertainties. The fluxes and the uncertainties must be
# in mJy. This file is optional to generate the configuration file, in
# particular for the savefluxes module.
data_file = Cigale_Test.mag

# Optional file containing the list of physical parameters. Each column
# must be in the form module_name.parameter_name, with each line behind
# 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
# parameters is not left empty, cigale will not interpret the
# configuration parameters given in pcigale.ini. They will be given only
# for information.
parameters_file =

# Order of the modules use for SED creation. Available modules:
# SFH: sfh2exp, sfhdelayed, sfhfromfile, sfhperiodic
# SSP: bc03, m2005
# Nebular emission: nebular
# Dust attenuation: dustatt_calzleit, dustatt_powerlaw
# Dust emission: casey2012, dale2014, dl2007, dl2014
# AGN: dale2014, fritz2006
# Radio: radio
# Redshift: redshifting (mandatory!)
sed_modules = sfhdelayed, bc03, nebular, dustatt_powerlaw, dale2014, redshifting
```

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

# Number of CPU cores available. This computer has 4 cores.
cores = 4
```

10. ***(py3)$ pcigale genconf*** will complete pcigale.ini by reading which modules you want to use and you need, now, to complete the file and save it, as follows. And you will have a message from CIGALE:

The configuration file has been updated. Please complete the various module parameters and the data file columns to use in the analysis.

```
# File containing the input data. The columns are 'id' (name of the
# object), 'redshift' (if 0 the distance is assumed to be 10 pc), the
# filter names for the fluxes, and the filter names with the '_err'
# suffix for the uncertainties. The fluxes and the uncertainties must be
# in mJy. This file is optional to generate the configuration file, in
# particular for the savefluxes module.
data_file = Cigale_Test.mag

# Optional file containing the list of physical parameters. Each column
# must be in the form module_name.parameter_name, with each line behind
# 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
# parameters is not left empty, cigale will not interpret the
# configuration parameters given in pcigale.ini. They will be given only
# for information.
parameters_file =

# Order of the modules use for SED creation. Available modules:
# SFH: sfh2exp, sfhdelayed, sfhfromfile, sfhperiodic
# SSP: bc03, m2005
# Nebular emission: nebular
# Dust attenuation: dustatt_calzleit, dustatt_powerlaw
# Dust emission: casey2012, dale2014, dl2007, dl2014
# AGN: dale2014, fritz2006
# Radio: radio
# Redshift: redshifting (mandatory!)
sed_modules = sfhdelayed, bc03, nebular, dustatt_powerlaw, dale2014, redshifting

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

# Number of CPU cores available. This computer has 4 cores.
cores = 4
```

```
# Bands to consider. To consider uncertainties too, the name of the band
# must be indicated with the _err suffix. For instance: FUV, FUV_err.
bands = FUV, FUV_err, NUV, NUV_err, UX_B90, UX_B90_err, BX_B90, BX_B90_err,
V_B90, V_B90_err, RC, RC_err, IC, IC_err, J, J_err, H, H_err, K, K_err, IRAC1,
IRAC1_err, IRAC2, IRAC2_err, IRAC3, IRAC3_err, IRAC4, IRAC4_err, IRAS1,
IRAS1_err, MIPS1, MIPS1_err, IRAS2, IRAS2_err, IRAS3, IRAS3_err, MIPS2,
MIPS2_err, IRAS4, IRAS4_err, MIPS3, MIPS3_err, SCUBA450, SCUBA450_err,
SCUBA850, SCUBA850_err


# Configuration of the SED creation modules.
[sed_modules_params]

 [[sfhdelayed]]
  # e-folding time of the main stellar population model in Myr.
  tau_main = 2000.0
  # Age of the oldest stars in the galaxy in Myr. The precision is 1 Myr.
  age = 5000
  # Multiplicative factor controlling the amplitude of SFR.
  sfr_A = 1.0
  # Normalise the SFH to produce one solar mass.
  normalise = True

 [[bc03]]
  # Initial mass function: 0 (Salpeter) or 1 (Chabrier).
  imf = 0
  # Metalicity. Possible values are: 0.0001, 0.0004, 0.004, 0.008, 0.02,
  # 0.05.
  metallicity = 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
  logU = -2.0
  # Fraction of Lyman continuum photons escaping the galaxy
  f_esc = 0.0
  # Fraction of Lyman continuum photons absorbed by dust
  f_dust = 0.0
  # Line width in km/s
  lines_width = 300.0
  # Include nebular emission.
  emission = True

 [[dustatt_powerlaw]]
  # V-band attenuation of the young population.
```

```
    Av_young = 1.0
    # Reduction factor for the V-band attenuation of the old population
    # compared to the young one (<1).
    Av_old_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 continuum.
    powerlaw_slope = -0.7
    # 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 = V_B90 & FUV

  [[dale2014]]
    # AGN fraction. It is not recommended to combine this AGN emission with
    # the of Fritz et al. (2006) models.
    fracAGN = 0.0
    # Alpha slope. Possible values are: 0.0625, 0.1250, 0.1875, 0.2500,
    # 0.3125, 0.3750, 0.4375, 0.5000, 0.5625, 0.6250, 0.6875, 0.7500,
    # 0.8125, 0.8750, 0.9375, 1.0000, 1.0625, 1.1250, 1.1875, 1.2500,
    # 1.3125, 1.3750, 1.4375, 1.5000, 1.5625, 1.6250, 1.6875, 1.7500,
    # 1.8125, 1.8750, 1.9375, 2.0000, 2.0625, 2.1250, 2.1875, 2.2500,
    # 2.3125, 2.3750, 2.4375, 2.5000, 2.5625, 2.6250, 2.6875, 2.7500,
    # 2.8125, 2.8750, 2.9375, 3.0000, 3.0625, 3.1250, 3.1875, 3.2500,
    # 3.3125, 3.3750, 3.4375, 3.5000, 3.5625, 3.6250, 3.6875, 3.7500,
    # 3.8125, 3.8750, 3.9375, 4.0000
    alpha = 2.0

  [[redshifting]]
    # Redshift to apply to the galaxy. 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 = sfh.sfr, sfh.sfr10Myrs, sfh.sfr100Myrs
  # If true, save the best SED for each observation to a file.
  save_best_sed = True
  # If true, for each observation and each analysed variable save the
  # reduced chi2.
  save_chi2 = False
```

 # If true, for each observation and each analysed variable save the
 # probability density function.
save_pdf = False
 # If true, for each object check whether upper limits are present and
 # analyse them.
lim_flag = False
 # If true, for each object we create a mock object and analyse them.
mock_flag = False

11. ***(py3)$ pcigale check*** will tell you how many models, per redshift, you will build.

With this configuration, pcigale must compute 4 SEDs.

12. ***(py3)$ pcigale run*** : if you are happy with the number of models, you can run CIGALE :

Initialising the analysis module...
Computing the models fluxes...
4/4 models computed in 8.6 seconds (0.5 models/s)

Analysing models...
8/8 objects analysed in 8.5 seconds (0.94 objects/s)

0.0% of the objects have chi^2_red~0 and 0.0% chi^2_red<0.5

Saving results...
Run completed!

(py3)$ find ./out lists the created files (note that if you would like to get tables, you have to set **create_table = True** in pcigale.ini:

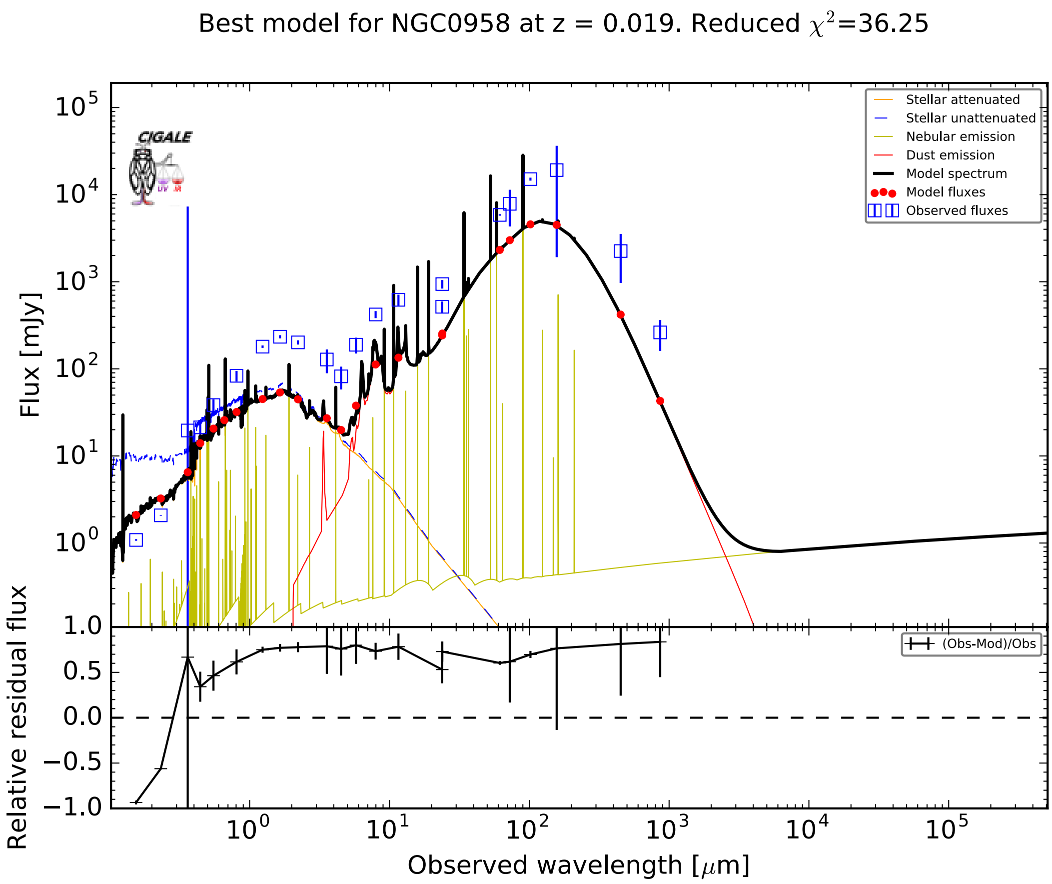
./out
./out/pcigale.ini => contains the parameters of the models
./out/pcigale.ini.spec
./out/results.fits => contains the output (best model and statistical) of the SED fitting (fits format)
./out/results.txt => contains the output (best model and statistical) of the SED fitting (fits format)
./out/CGCG011076_best_model.fits
./out/CGCG011076_SFH.fits
./out/CGCG247020_best_model.fits
./out/CGCG247020_SFH.fits
./out/IRASF105652448_best_model.fits
./out/IRASF105652448_SFH.fits
./out/IRASF152503608_best_model.fits
./out/IRASF152503608_SFH.fits
./out/NGC0958_best_model.fits
./out/NGC0958_SFH.fits
./out/NGC1614_best_model.fits

./out/NGC1614_SFH.fits
./out/UGC02369_best_model.fits
./out/UGC02369_SFH.fits
./out/VV705_best_model.fits
./out/VV705_SFH.fits

13. *(py3)$ pcigale-plots sed* : create the figures showing the best SED plotted over the observed data.

For each object analysed, you will find a PDF figure showing the best SED and the best (but bad fits, in this example, see Fig. 2) model plotted and the best model (all the components separately and the star formation history) in the *.fits files, e.g. NGC0958_best_model.fits for the plot above. To read the files, you can use any adapted xml reader, e.g., TOPCAT (http://www.star.bris.ac.uk/~mbt/topcat/)

CGCG011076_best_model.pdf
CGCG011076_best_model.fits
VV705_best_model.pdf
VV705_best_model.fits
...



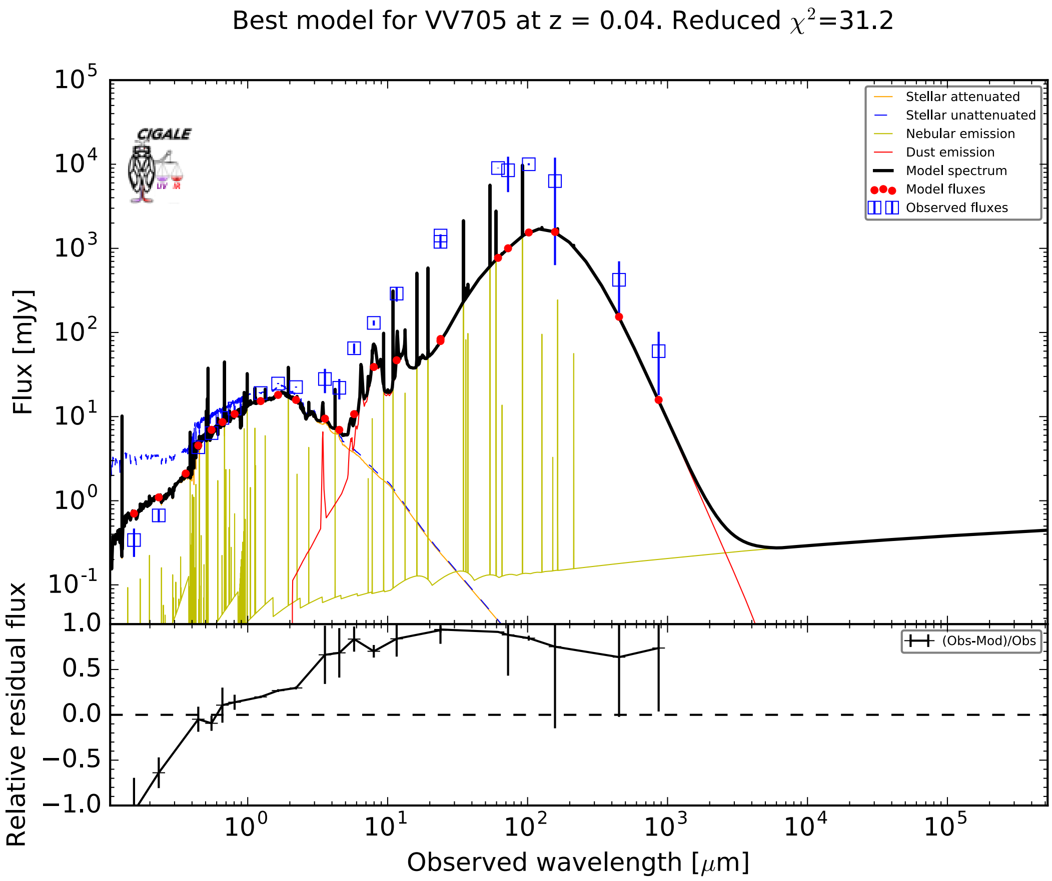


Fig. 2 : Using the default pcigale.ini provided in the distribution, fits can be performed. But, since there is only 4 models, the probability to get a good fit is very unlikely as illustrated here for this galaxy. Better fits are obtained below with a more developed pcigale.ini.

14. To improve the fit, we add many more models (a typical CIGALE run could build several hundreds thousands models or even more). This present run will provides better fits (Fig. 3) and the results can be found in the directory OUT-BETTER. You need to follow the same sequence as above after having edited the configuration file pcigale.ini. Building more models means that you can have several values given for a parameter, i.e. age, metallicity, etc.:

 # File containing the input data. The columns are 'id' (name of the
 # object), 'redshift' (if 0 the distance is assumed to be 10 pc), the
 # filter names for the fluxes, and the filter names with the '_err'
 # suffix for the uncertainties. The fluxes and the uncertainties must be
 # in mJy. This file is optional to generate the configuration file, in
 # particular for the savefluxes module.
data_file = Cigale_Test.mag

 # Order of the modules use for SED creation. Available modules:
 # SFH: sfh2exp, sfhdelayed, sfhfromfile, sfhperiodic
 # SSP: bc03, m2005
 # Nebular emission: nebular
 # Dust attenuation: dustatt_calzleit, dustatt_powerlaw
 # Dust emission: casey2012, dale2014, dl2007, dl2014

```
# AGN: dale2014, fritz2006
# Radio: radio
# Redshift: redshifting (mandatory!)
creation_modules = sfhdelayed, bc03, nebular, dustatt_powerlaw, dale2014,
redshifting

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

# Number of CPU cores available. This computer has 4 cores.
cores = 4

# List of the columns in the observation data file to use for the
# fitting.
column_list = FUV, FUV_err, NUV, NUV_err, UX_B90, UX_B90_err, BX_B90, BX_B90_err,
V_B90, V_B90_err, RC, RC_err, IC, IC_err, J, J_err, H, H_err, K, K_err, IRAC1, IRAC1_err,
IRAC2, IRAC2_err, IRAC3, IRAC3_err, IRAC4, IRAC4_err, IRAS1, IRAS1_err, MIPS1,
MIPS1_err, IRAS2, IRAS2_err, IRAS3, IRAS3_err, MIPS2, MIPS2_err, IRAS4, IRAS4_err,
MIPS3, MIPS3_err, SCUBA450, SCUBA450_err, SCUBA850, SCUBA850_err


# Configuration of the SED creation modules.
[sed_creation_modules]

 [[sfhdelayed]]
   # e-folding time of the main stellar population model in Myr.
   tau_main = 250, 500, 1000, 2000, 4000, 6000, 8000
   # Age of the oldest stars in the galaxy in Myr. The precision is 1 Myr.
   age = 250, 500, 1000, 2000, 4000, 8000, 10000, 12000
   # Multiplicative factor controlling the amplitude of SFR.
   sfr_A = 1.0
   # Normalise the SFH to produce one solar mass.
   normalise = True

 [[bc03]]
   # Initial mass function: 0 (Salpeter) or 1 (Chabrier).
   imf = 0
   # Metalicity. Possible values are: 0.0001, 0.0004, 0.004, 0.008, 0.02,
   # 0.05.
   metallicity = 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
   logU = -2.0
```

```
  # Fraction of Lyman continuum photons escaping the galaxy
  f_esc = 0.0
  # Fraction of Lyman continuum photons absorbed by dust
  f_dust = 0.0
  # Line width in km/s
  lines_width = 300.0
  # Include nebular emission.
  emission = True

[[dustatt_powerlaw]]
  # V-band attenuation of the young population.
  Av_young = 0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0
  # Reduction factor for the V-band attenuation of the old population
  # compared to the young one (<1).
  Av_old_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, 1.0, 2.0, 3.0
  # Slope delta of the power law continuum.
  powerlaw_slope = -0.7
  # 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 = V_B90 & FUV

[[dale2014]]
  # AGN fraction. It is not recommended to combine this AGN emission with
  # the of Fritz et al. (2006) models.
  fracAGN = 0.0
  # Alpha slope. Possible values are: 0.0625, 0.1250, 0.1875, 0.2500,
  # 0.3125, 0.3750, 0.4375, 0.5000, 0.5625, 0.6250, 0.6875, 0.7500,
  # 0.8125, 0.8750, 0.9375, 1.0000, 1.0625, 1.1250, 1.1875, 1.2500,
  # 1.3125, 1.3750, 1.4375, 1.5000, 1.5625, 1.6250, 1.6875, 1.7500,
  # 1.8125, 1.8750, 1.9375, 2.0000, 2.0625, 2.1250, 2.1875, 2.2500,
  # 2.3125, 2.3750, 2.4375, 2.5000, 2.5625, 2.6250, 2.6875, 2.7500,
  # 2.8125, 2.8750, 2.9375, 3.0000, 3.0625, 3.1250, 3.1875, 3.2500,
  # 3.3125, 3.3750, 3.4375, 3.5000, 3.5625, 3.6250, 3.6875, 3.7500,
  # 3.8125, 3.8750, 3.9375, 4.0000
  alpha = 1.0, 1.5, 2.0, 2.5

[[redshifting]]
  # Redshift to apply to the galaxy. Leave empty to use the redshifts from
  # the input file.
  redshift =
```

```
# Configuration of the statistical analysis method.
[analysis_configuration]
  # If true, for each observation and each analysed variable save the
  # probability density function.
  save_pdf = False
  # If true, save the best SED for each observation to a file.
  save_best_sed = True
  # List of the variables (in the SEDs info dictionaries) for which the
  # statistical analysis will be done.
  analysed_variables = sfh.sfr, sfh.sfr10Myrs, sfh.sfr100Myrs, stellar.m_star,
attenuation.FUV, attenuation.V_B90, dust.luminosity, sfh.tau_main, sfh.age
  # If true, for each observation and each analysed variable save the
  # reduced chi2.
  save_chi2 = False
  # If true, for each object we create a mock object and analyse them.
  mock_flag = True
  # If true, for each object check whether upper limits are present and
  # analyse them.
  lim_flag = False
```

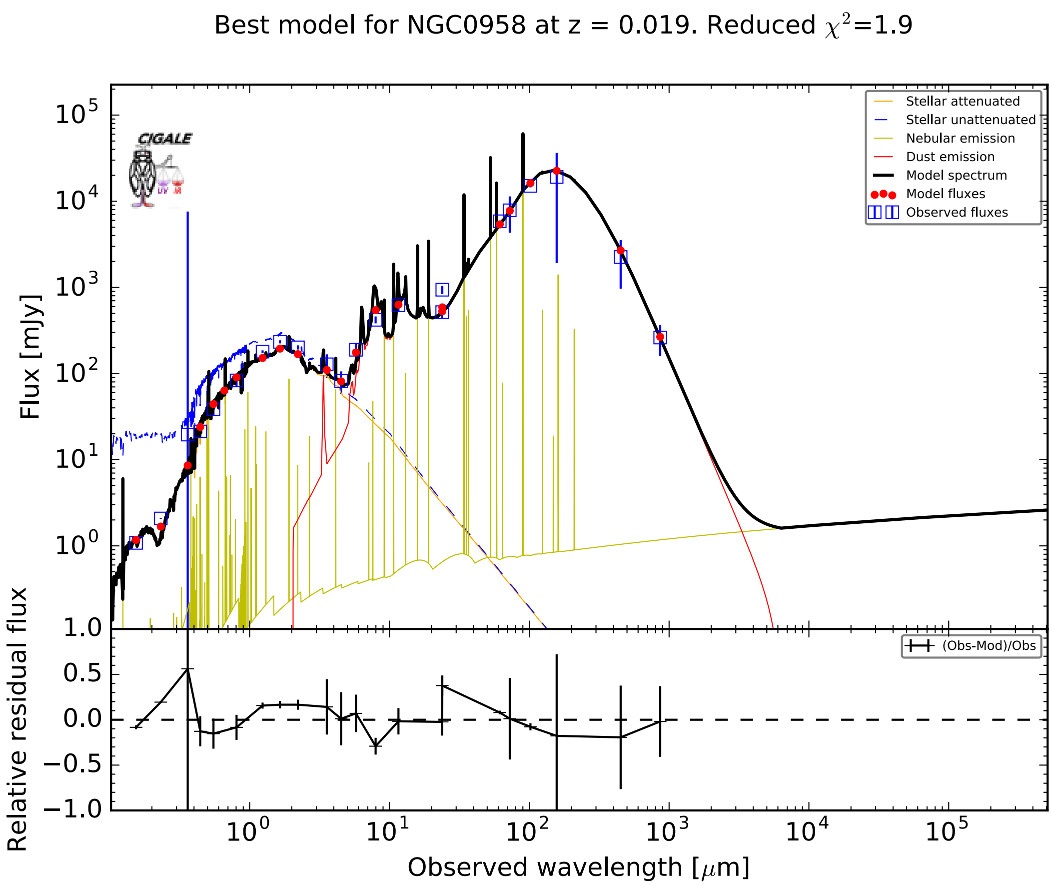
15. *(py3)$ pcigale check*
With this configuration, pcigale must compute 25088 SEDs.

16. *(py3)$ pcigale run*

Note that if you already have a folder ./out, it will be renamed to YYYYMMDDHHMM_out:

```
The existing out/ directory was renamed to 201603221603_out/

Initialising the analysis module...
The existing out/ directory was renamed to 201603221603_out/
Computing the models fluxes...
25088/25088 models computed in 49.1 seconds (510.5 models/s)

Analysing models...
8/8 objects analysed in 9.0 seconds (0.88 objects/s)

0.0% of the objects have chi^2_red~0 and 0.0% chi^2_red<0.5

Saving results...
Run completed!
```

17. *(py3)$ pcigale-plots sed* creates the figures (Fig. 3) with the observed SEDs and the best model.


Best model for NGC0958 at z = 0.019. Reduced χ^2=1.9
CIGALE
Stellar attenuated
Stellar unattenuated
Nebular emission
Dust emission
Model spectrum
Model fluxes
Observed fluxes
(Obs-Mod)/Obs
Flux [mJy]
Relative residual flux
Observed wavelength [μm]

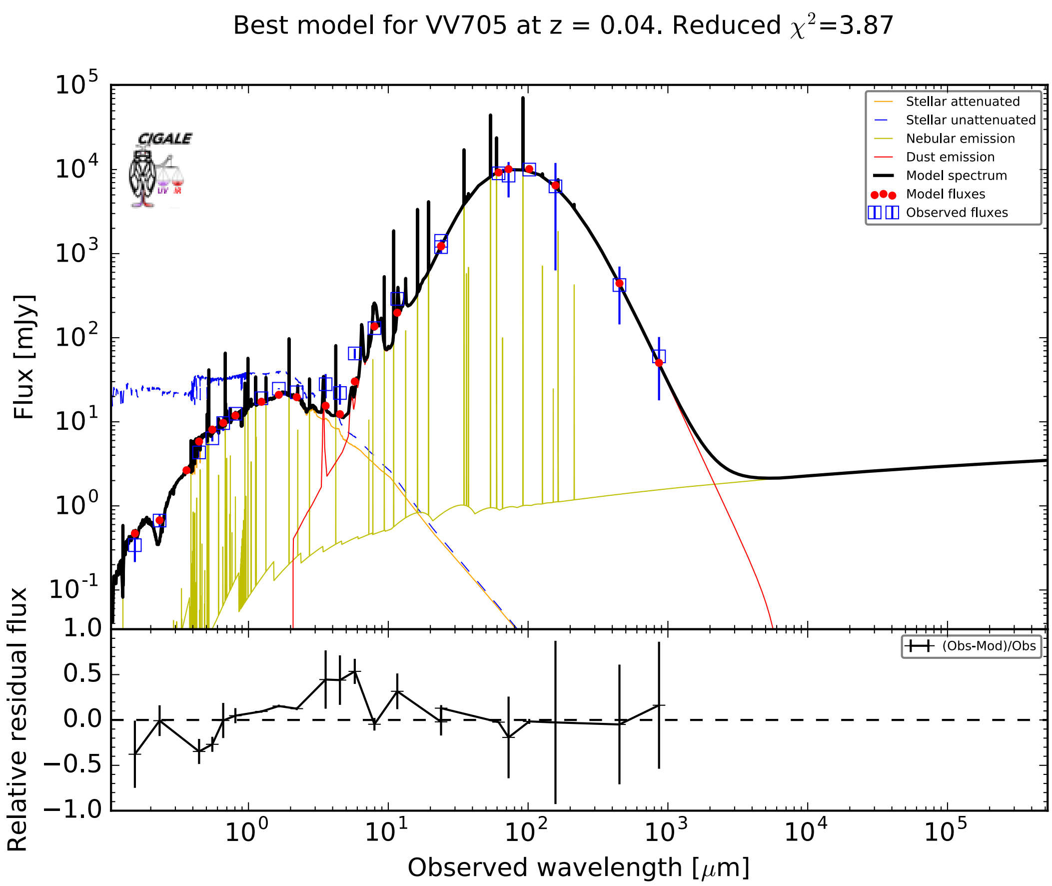
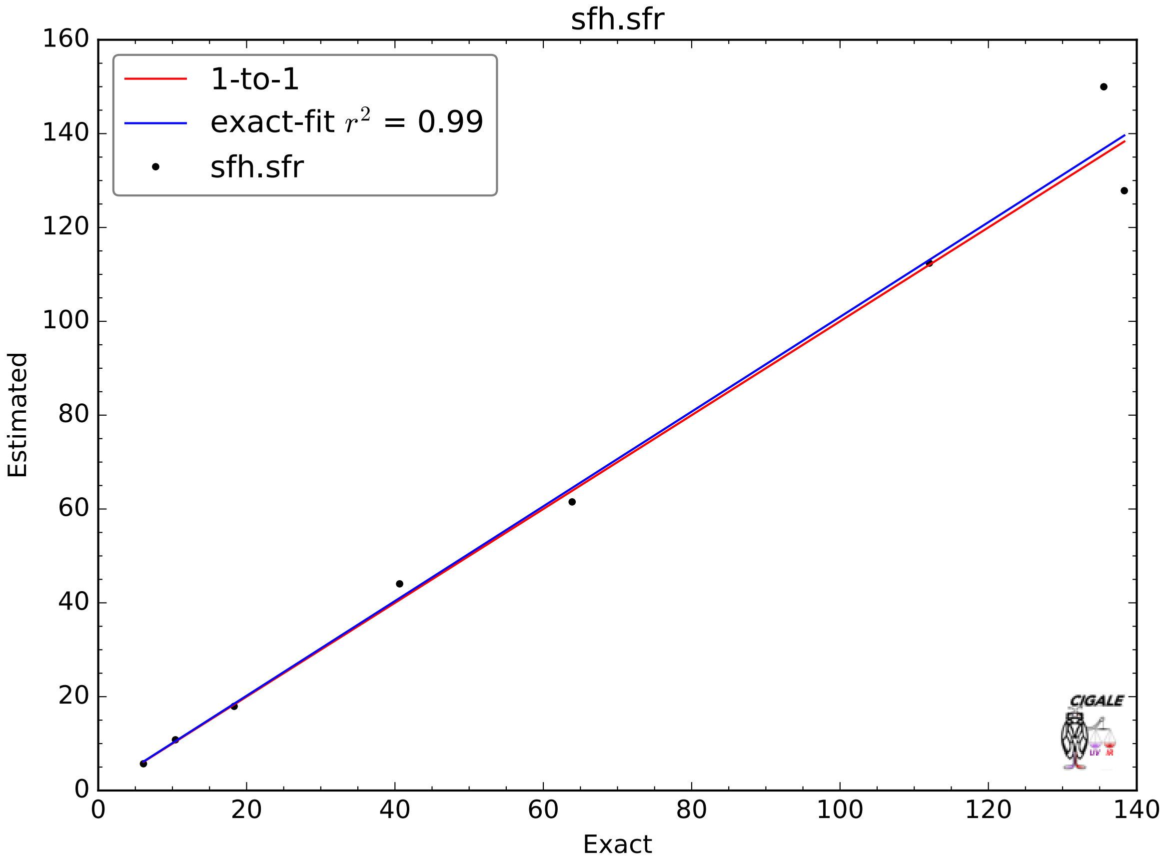

Best model for VV705 at z = 0.04. Reduced χ^2=3.87
CIGALE
Stellar attenuated
Stellar unattenuated
Nebular emission
Dust emission
Model spectrum
Model fluxes
Observed fluxes
(Obs-Mod)/Obs
Flux [mJy]
Relative residual flux
Observed wavelength [μm]

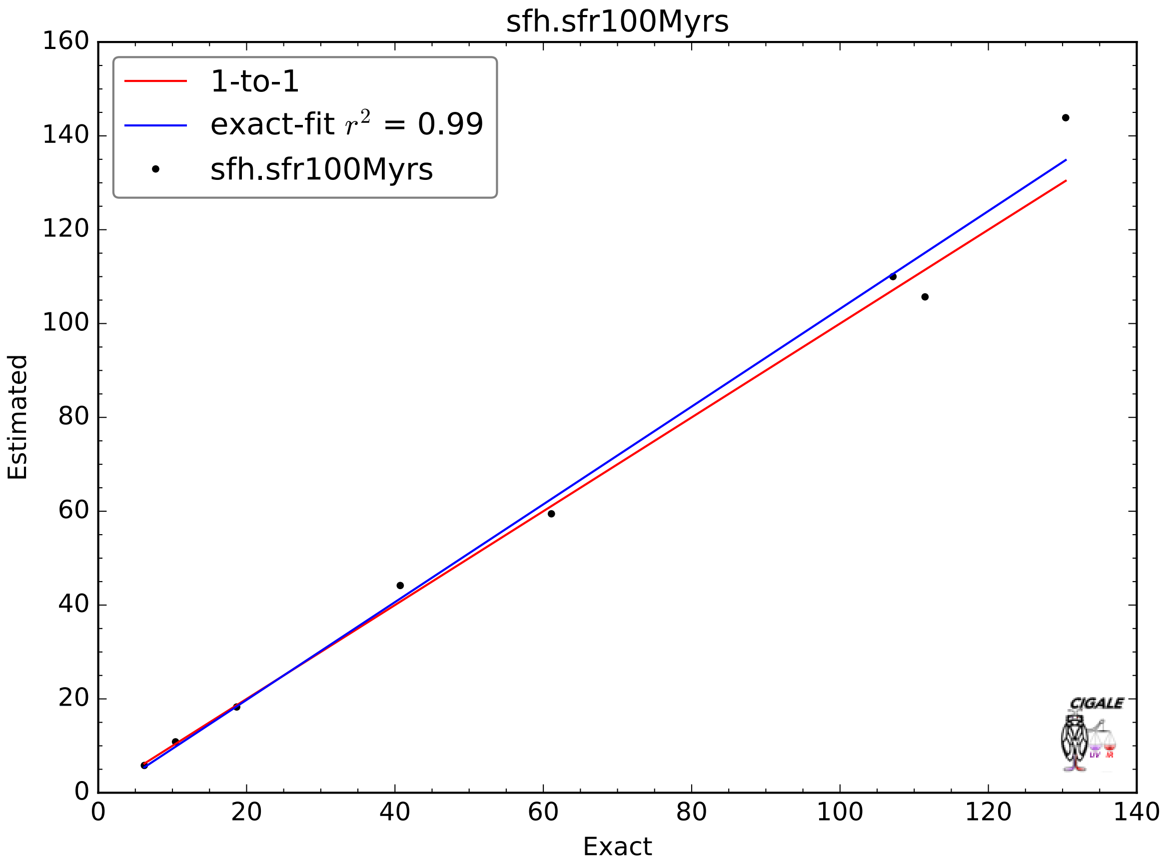
Fig. 3 : Using the improved pcigale.ini provided in the distribution, fits can be performed. Now, more models are built and a better fit is found.

18. ***(py3)$ pcigale-plots mock*** creates a mock catalogue. If you modify the last section of the file pcigale.ini as follows by switching mock_flag = True, CIGALE creates the figures (Fig. 4) with the comparison of input and output parameters for the mock catalogue built from the observed ones.

 # Configuration of the statistical analysis method.
[analysis_configuration]
 # If true, for each observation and each analysed variable save the
 # probability density function.
save_pdf = False
 # If true, save the best SED for each observation to a file.
save_best_sed = True
 # List of the variables (in the SEDs info dictionaries) for which the
 # statistical analysis will be done.
analysed_variables = sfh.sfr, sfh.sfr10Myrs, sfh.sfr100Myrs, stellar.m_star, attenuation.FUV, attenuation.V_B90, dust.luminosity, sfh.tau_main, sfh.age
 # If true, for each observation and each analysed variable save the
 # reduced chi2.
save_chi2 = False
 # If true, for each object we create a mock object and analyse them.
mock_flag = True
 # If true, for each object check whether upper limits are present and
 # analyse them.
lim_flag = False



sfh.sfr
1-to-1
exact-fit r^2 = 0.99
sfh.sfr
Estimated
Exact
CIGALE



sfh.sfr100Myrs
1-to-1
exact-fit r^2 = 0.99
sfh.sfr100Myrs
Estimated
Exact
CIGALE

Fig. 4 : From the best-fit model, a mock catalogue is drawn, using the observed uncertainties) for each analysed object. Then, CIGALE analyses this mock catalogue and a comparison between ths input (X-axis) and estimated (Y-axis) parameters is performed for each of the analysed parameters.

19. Finally, you can run CIGALE as a model creator, without any data. The following pcigale.ini gives you an example to do so.

20. ***(py3)$ pcigale init*** that you need to edit and complete as below :

 # File containing the input data. The columns are 'id' (name of the
 # object), 'redshift' (if 0 the distance is assumed to be 10 pc), the
 # filter names for the fluxes, and the filter names with the '_err'
 # suffix for the uncertainties. The fluxes and the uncertainties must be
 # in mJy. This file is optional to generate the configuration file, in
 # particular for the savefluxes module.
data_file = Cigale_Test.mag

 # Optional file containing the list of physical parameters. Each column
 # must be in the form module_name.parameter_name, with each line behind
 # 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
 # parameters is not left empty, cigale will not interpret the
 # configuration parameters given in pcigale.ini. They will be given only
 # for information.
parameters_file =

 # Order of the modules use for SED creation. Available modules:
 # SFH: sfh2exp, sfhdelayed, sfhfromfile, sfhperiodic
 # SSP: bc03, m2005
 # Nebular emission: nebular
 # Dust attenuation: dustatt_calzleit, dustatt_powerlaw
 # Dust emission: casey2012, dale2014, dl2007, dl2014
 # AGN: dale2014, fritz2006
 # Radio: radio
 # Redshift: redshifting (mandatory!)
sed_modules = sfhdelayed, bc03, nebular, dustatt_powerlaw, dale2014, redshifting

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

 # Number of CPU cores available. This computer has 4 cores.
cores = 4

21. ***(py3)$ pcigale genconf*** . You can have some warnings, depending on the modules you select.

WARNING! Choosing the nebular module is recommended. Without it the Lyman

continuum is left untouched. Options are: nebular.
No radio module found. Options are: radio.
The configuration file has been updated. Please complete the various module parameters and the data file columns to use in the analysis.

22. You edit and save the file. Note that if you do not need to provide data for the creation od models, you absolutely need redshifts that are read in the pcigale.fini file in the [[redshifting]] section. You can have some warnings, depending on the modules you select.

```
# File containing the input data. The columns are 'id' (name of the
# object), 'redshift' (if 0 the distance is assumed to be 10 pc), the
# filter names for the fluxes, and the filter names with the '_err'
# suffix for the uncertainties. The fluxes and the uncertainties must be
# in mJy. This file is optional to generate the configuration file, in
# particular for the savefluxes module.
data_file = Cigale_Test.mag

# Optional file containing the list of physical parameters. Each column
# must be in the form module_name.parameter_name, with each line behind
# 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
# parameters is not left empty, cigale will not interpret the
# configuration parameters given in pcigale.ini. They will be given only
# for information.
parameters_file =

# Order of the modules use for SED creation. Available modules:
# SFH: sfh2exp, sfhdelayed, sfhfromfile, sfhperiodic
# SSP: bc03, m2005
# Nebular emission: nebular
# Dust attenuation: dustatt_calzleit, dustatt_powerlaw
# Dust emission: casey2012, dale2014, dl2007, dl2014
# AGN: dale2014, fritz2006
# Radio: radio
# Redshift: redshifting (mandatory!)
sed_modules = sfhdelayed, bc03, nebular, dustatt_powerlaw, dale2014, redshifting

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

# Number of CPU cores available. This computer has 4 cores.
cores = 4

# Bands to consider. To consider uncertainties too, the name of the band
# must be indicated with the _err suffix. For instance: FUV, FUV_err.
bands = FUV, FUV_err, NUV, NUV_err, UX_B90, UX_B90_err, BX_B90, BX_B90_err,
V_B90, V_B90_err, RC, RC_err, IC, IC_err, J, J_err, H, H_err, K, K_err, IRAC1, IRAC1_err,
```

```
IRAC2, IRAC2_err, IRAC3, IRAC3_err, IRAC4, IRAC4_err, IRAS1, IRAS1_err, MIPS1,
MIPS1_err, IRAS2, IRAS2_err, IRAS3, IRAS3_err, MIPS2, MIPS2_err, IRAS4, IRAS4_err,
MIPS3, MIPS3_err, SCUBA450, SCUBA450_err, SCUBA850, SCUBA850_err


# Configuration of the SED creation modules.
[sed_modules_params]

 [[sfhdelayed]]
  # e-folding time of the main stellar population model in Myr.
  tau_main = 500.0, 10000., 2000.0
  # Age of the oldest stars in the galaxy in Myr. The precision is 1 Myr.
  age = 100, 500, 1000, 2000
  # Multiplicative factor controlling the amplitude of SFR.
  sfr_A = 1.0
  # Normalise the SFH to produce one solar mass.
  normalise = True

 [[bc03]]
  # Initial mass function: 0 (Salpeter) or 1 (Chabrier).
  imf = 0
  # Metalicity. Possible values are: 0.0001, 0.0004, 0.004, 0.008, 0.02,
  # 0.05.
  metallicity = 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
  logU = -2.0
  # Fraction of Lyman continuum photons escaping the galaxy
  f_esc = 0.0
  # Fraction of Lyman continuum photons absorbed by dust
  f_dust = 0.0
  # Line width in km/s
  lines_width = 300.0
  # Include nebular emission.
  emission = True

 [[dustatt_powerlaw]]
  # V-band attenuation of the young population.
  Av_young = 0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0
  # Reduction factor for the V-band attenuation of the old population
  # compared to the young one (<1).
  Av_old_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, 1.0, 2.0, 3.0
  # Slope delta of the power law continuum.
  powerlaw_slope = -0.7
  # 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 = V_B90 & FUV

 [[dale2014]]
  # AGN fraction. It is not recommended to combine this AGN emission with
  # the of Fritz et al. (2006) models.
  fracAGN = 0.0
  # Alpha slope. Possible values are: 0.0625, 0.1250, 0.1875, 0.2500,
  # 0.3125, 0.3750, 0.4375, 0.5000, 0.5625, 0.6250, 0.6875, 0.7500,
  # 0.8125, 0.8750, 0.9375, 1.0000, 1.0625, 1.1250, 1.1875, 1.2500,
  # 1.3125, 1.3750, 1.4375, 1.5000, 1.5625, 1.6250, 1.6875, 1.7500,
  # 1.8125, 1.8750, 1.9375, 2.0000, 2.0625, 2.1250, 2.1875, 2.2500,
  # 2.3125, 2.3750, 2.4375, 2.5000, 2.5625, 2.6250, 2.6875, 2.7500,
  # 2.8125, 2.8750, 2.9375, 3.0000, 3.0625, 3.1250, 3.1875, 3.2500,
  # 3.3125, 3.3750, 3.4375, 3.5000, 3.5625, 3.6250, 3.6875, 3.7500,
  # 3.8125, 3.8750, 3.9375, 4.0000
  alpha = 2.0

 [[redshifting]]
  # Redshift to apply to the galaxy. Leave empty to use the redshifts from
  # the input file.
  redshift = 0., 0.5, 1., 1.5, 2., 2.5, 3., 3.5, 4., 4.5, 5., 5.5, 6.


# Configuration of the statistical analysis method.
[analysis_params]
 # List of the physical properties to save. Leave empty to save all the
 # physical properties (not recommended when there are many models).
 variables =
 # Name of the output file that contains the parameters of the model(s)
 # and the flux densities in the bands
 output_file = computed_fluxes.txt
 # If True, save the generated spectrum for each model.
 save_sed = False
 # Format of the output file. Any format supported by astropy.table e.g.
 # votable or ascii.
 output_format = ascii
```

CIGALE is run exactly in the same way as above. But, this time, the files found in the output are different and the main one is ***computed_fluxes.txt*** that contains the models built.

(py3)$ pcigale check
With this configuration, pcigale must compute 4368 SEDs.
(py3)$ pcigale run
4368/4368 models computed in 5.4 seconds (809.2 models/s)

From this type of models, you can create, e.g. color-color diagrams (Fig. 5).

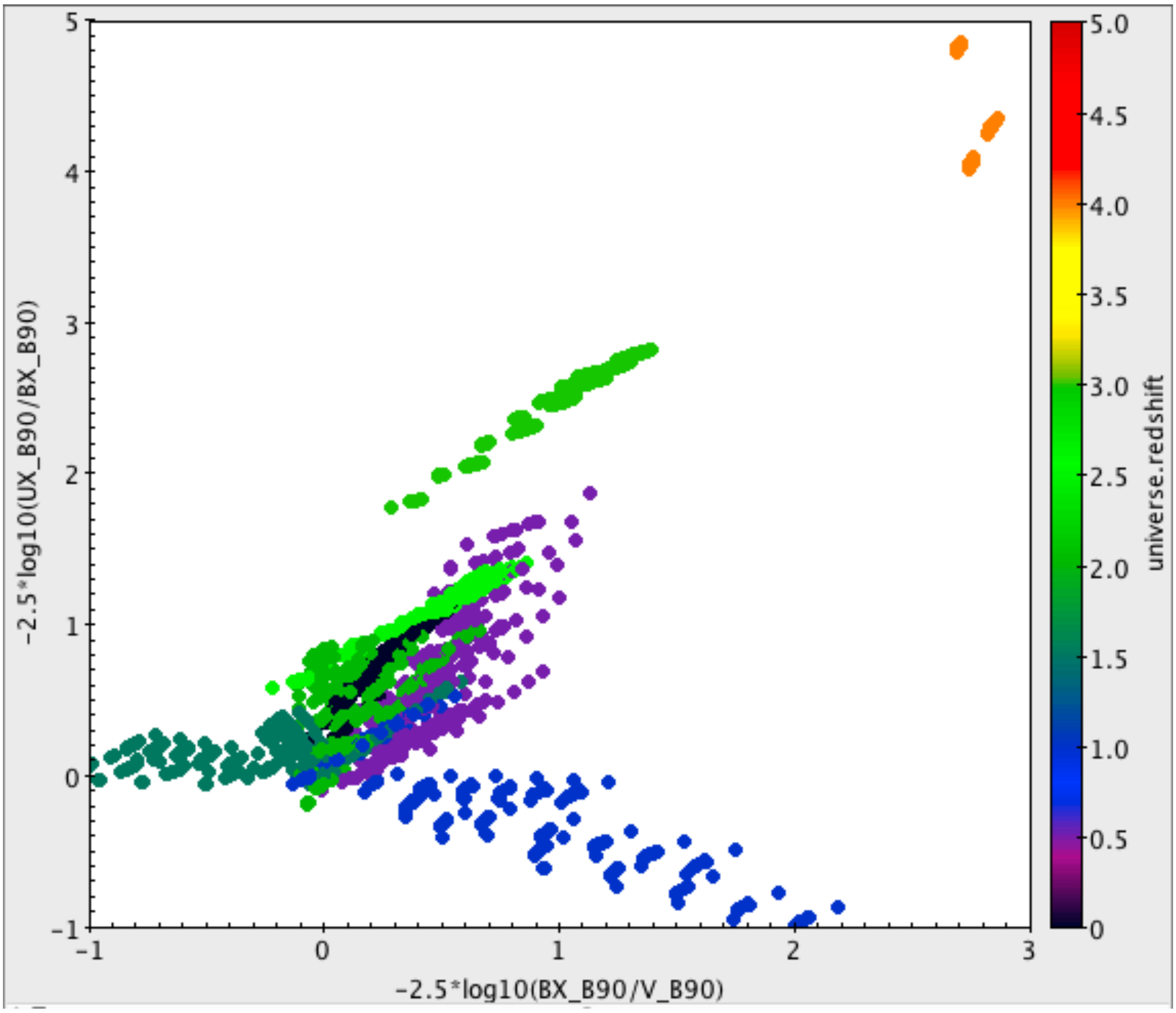


Figure 5 : color-color diagram as those used to select Lyman break galaxies.

Annex

The main parameters that can be analysed are listed below. Note that some of them are not free, see notes after the table. To get an exhaustive list, you have to look into the modules themselves in the directory pcigale/create_module. This list is also available in the documentation online: http://cigale.lam.fr/pcigale/pcigale_manual.html

Note: if you wish to estimate the physical parameters in *Log*, 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!'

Which parameters to analyse in CIGALE (add « _log » at the end of the name of the parameter to get a logarithmic scale analysis)		
Treat	**Quantity**	**Description**
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

Which parameters to analyse in CIGALE (add « _log » at the end of the name of the parameter to get a logarithmic scale analysis)		
Treat	**Quantity**	**Description**
		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 sepation 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
‘ ‘	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 sepation 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 Lyman cont photons from old stellar population
‘ ‘	stellar.m_star	Total mass of stars

Which parameters to analyse in CIGALE (add « _log » at the end of the name of the parameter to get a logarithmic scale analysis)		
Treat	**Quantity**	**Description**
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.old	E(B-V) of the old stellar population. Note that E(B-V) is an internal parameter which does not correspond to A_B-A_V except for the exact calzetti law (delta=0), E(B-V)= A_B-A_V should be calculated by the user
‘ ‘	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 A_B-A_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.(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.stellar.old	V-band attenuation of the old population
‘ ‘	attenuation.av_old_factor	Reduction factor of A_V for the old population compared to the young one
‘ ‘	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 &

Which parameters to analyse in CIGALE (add « _log » at the end of the name of the parameter to get a logarithmic scale analysis)		
Treat	**Quantity**	**Description**
		Li (2007) templates
‘ ‘	dust.alpha	Parameter alpha_max in Draine & Li (2007) templates
‘ ‘	dust.gamma	Parameter gamma in Draine & Li (2007) templates
‘ ‘	dust.luminosity	Estimated dust luminosity using an energy balance
‘ ‘	dust.qpah	Parameter q_{PAH} in Draine & Li (2014) updated templates
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 (2007)
‘ ‘	agn.opening_angle	Full opening angle of the dust torus
‘ ‘	agn.psy	Angle between AGN axis and line of sight
‘ ‘	agn.fracAGN	AGN fraction [0.0, 1.0[in Fritz (2007)
‘ ‘	agn.r_ratio	Parameter r_ratio in Fritz (2007)
‘ ‘	agn_tau	Parameter tau in Fritz (2007)
‘ ‘	agn_tau	Parameter beta in Fritz (2007)
‘ ‘	agn.luminosity	Estimated total luminosity of the AGN
‘ ‘	agn.therm_luminosity	Estimated thermal luminosity of the AGN
‘ ‘	agn.scatt_luminosity	Estimated scattered luminosity of the AGN
‘ ‘	agn.agn_luminosity	Estimated luminosity of the central AGN only
nebular	nebular.f_esc	Fraction of Lyman continuum photons escaping the galaxy
‘ ‘	nebular.f_dust	Fraction of Lyman continuum photons absorbed by dust
radio	radio_qir	FIR/radio ratio
‘ ‘	radio_alpha	slope of the power-law synchrotron emission

Which parameters to analyse in CIGALE (add « _log » at the end of the name of the parameter to get a logarithmic scale analysis)		
Treat	**Quantity**	**Description**
redshifting	universe.redshift	redshift
‘ ‘	universe.luminosity_distance	Luminosity distance
‘ ‘	universe.age	Age of the universe