pydirtygrid Documentation

Release 0.1.dev26

Karl Gordon, Jeremy Chastenet

April 22, 2017

Contents

This is an affiliated package for the AstroPy package. The documentation for this package is here:

CHAPTER 1

PyDirtyGrid

Code to access and work on the DirtyGrid set of stellar population/dust RT models.

Uses the astropy affiliated package template

CHAPTER 2

Repository

Github: <https://github.com/karllark/pydirtygrid>

CHAPTER 3

DirtyGrid Data

The DirtyGrid is provide as a large hd5f file. This file provides photometry in a number of bands.

Location: <https://stsci.box.com/v/dirtygrid>

CHAPTER 4

Reference API

pydirtygrid.PhotDG Module

Classes

PhotDG([datafile])	Read in the DirtyGrid cube from a HDF5 file

PhotDG

class pydirtygrid.PhotDG.**PhotDG**(*datafile='data/dirtygrid_29mar17.hdf5'*)

Read in the DirtyGrid cube from a HDF5 file

Returns

(self.)seds: a list that grows with the SED you choose to extract

(self.)dgrid: the HDF5 file: with attributes corresponding to the parameters values; with each dataset corresponding to each band

Methods Summary

findIndexFromParams(grain, geom, sf_type, ...)	Give a set of parameters and find where they will be stored in the
photAddNew(wave0, band_name, newcube[, datafile])	Create the new dataset and add it to the file
photGet(grain, geom, sf_type, metal, age, ...)	Save photometry points given a set of parameters.
photPlot([ind])	Plot photometry points, either giving a specific set of bands, or all
print_parameters()	

Methods Documentation

findIndexFromParams(*grain*, *geom*, *sf_type*, *metal*, *age*, *sfr*, *tau*)

Give a set of parameters and find where they will be stored in the DirtyGrid cube

Parameters

grain: string

type of grain

geom: string

geometry

sf_type: string

star formation type

metal: float

metallicity

age: float

age of the stellar population

sfr: float

star formation rate

tau: float

optical depth

Returns

indgt, indgm, indst, indmt, indsa, indsr, indta: integers

the corresponding indices

`photAddNew`(*wave0*, *band_name*, *newcube*, *datafile='data/dirtygrid_29mar17.hdf5'*)

Create the new dataset and add it to the file

Parameters

wave0: float

central wavelength of the new filter

band_name: string

name of the new filter

newcube: array(3, 6, 2, 5, 50, 29, 25) (float)

new photometry to be added to the file as a new dataset

`photGet`(*grain*, *geom*, *sf_type*, *metal*, *age*, *sfr*, *tau*, *bands=-1*)

Save photometry points given a set of parameters. The function allows for:

- values not in the DGrid parameter sampling, and finds the closest point
- specific bands only

Parameters

grain: string

type of grain

geom: string

geometry

sf_type: string

star formation type

metal: float

metallicity

age: float

age of the stellar population

sfr: float

star formation rate

tau: float

optical depth

Returns

seds: save the corresponding SED

`photPlot`(*ind=-1*)

Plot photometry points, either giving a specific set of bands, or all
of them either giving a specific index of which if more that one saved, or all of them

Parameters

ind: integer(s) (Optional)

indices of the SED you wish to plot

`print_parameters`()

Class Inheritance Diagram

pydirtygrid.SpecDG Module

Classes

SpecDG()	Read in the mapping file for all spectra

SpecDG

class pydirtygrid.SpecDG.**SpecDG**

Read in the mapping file for all spectra

Returns

(self.)plan: Table

table containing the identification of each spectrum, as well as its parameters

(self.)seds: list
empty list to save the spectra if you want to

Methods Summary

findFile(file_id)	Returns the full filename for a given GID from the spectrum mapping.
findGidFromParam(grain, geom, sf_type, ...)	Returns the GID given a set of parameters
findParamsFromGid(thisgid)	Returns the parameter values for a given GID
spec2Phot(trans_curve, trans_waves, wave0[, ...])	Compute the new photometry
specGet(filename)	Save a spectrum from the filename
specPlot([ind])	Plot SEDs, either giving a specific index of which if more that

Methods Documentation

findFile(*file_id*)
Returns the full filename for a given GID from the spectrum mapping.

Returns
filepath+filename: string

the absolute path and name of the .fits file

findGidFromParam(*grain*, *geom*, *sf_type*, *metal*, *age*, *sfr*, *tau*)
Returns the GID given a set of parameters (If you want to plot or something)

Parameters
grain: string

type of grain

geom: string

geometry

sf_type: string

star formation type

metal: float

metallicity

age: float

age of the stellar population

sfr: float

star formation rate

tau: float

optical depth

Returns
file_gid: string

identification number of the spectrum with the given set

of parameters

`findParamsFromGid`(*thisgid*)
Returns the parameter values for a given GID (Used later to update the cube)

Parameters
thisgid: string

identification number of the spectrum

Returns
thisgt, thisgm, thissf, thismt, thissa, thissr, thista: integers

indices of the parameter values for the given spectrum,

in the multidimensional photometry cube

`spec2Phot`(*trans_curve*, *trans_waves*, *wave0*, *energy=1*)
Compute the new photometry

Parameters
trans_curve: array

transmission curve of the new filter /!Alread read, as an array

trans_waves: array

wavelengths of the new filter

wave0: float

center/effective wavelength of the new filter

energy: integer (Optional)

specification for integration with energy(=1, default) or photons

Returns
newcube: array(3, 6, 2, 5, 50, 29, 25) (float)

new photometry cube

`specGet`(*filename*)
Save a spectrum from the filename

Parameters
filename: string

absolute path to the spectrum you want to read

Returns
(self.)seds: list

update the list to add the SED

(self.)waves: array

the wavelengths to the SED

`specPlot`(*ind=-1*)
Plot SEDs, either giving a specific index of which if more that one saved, or all of them

Parameters
ind: integer(s) (Optional)

indices of the SED you wish to plot

Class Inheritance Diagram

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

p

F

P

S