metallicity-stack-commons

Release 1.3.1

Chun Ly, Reagen Leimbach, and Caroline McCormick

Jun 22, 2021

CONTENTS:

1 Metallicity_Stack_Commons **1**

2 API Documentation **3**

2.1 Metallicity_Stack_Commons package 3

3 Indices and tables **17**

Python Module Index **19**

Index **21**

CHAPTER

ONE

METALLICITY_STACK_COMMONS

Set of common codes used in metallicity studies that use the stacking techniques

CHAPTER

TWO

API DOCUMENTATION

2.1 Metallicity_Stack_Commons package

2.1.1 Subpackages

analysis subpackage

Submodules

Metallicity_Stack_Commons.analysis.attenuation module

`Metallicity_Stack_Commons.analysis.attenuation.`**`Hb_SFR`**(*log_LHb*, *EBV*, *verbose=False*, *log=<Logger stdout_logger (INFO)>*)

Determine dust-corrected SFR using the H-beta luminosity and a measurement for nebular attenuation

Equation below is based on Eq. 2 in Ly et al. (2015), ApJ, 805, 45 DOI: https://doi.org/10.1088/0004-637X/805/1/45

Parameters

- **log_LHb** (Union[float, ndarray]) – Logarithm of H-beta luminosity in units of erg/s
- **EBV** (Union[float, ndarray]) – E(B-V) value(s)
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Return type Union[float, ndarray]

Returns SFRs in logarithmic units of M_sun/yr

`Metallicity_Stack_Commons.analysis.attenuation.`**`compute_A`**(*EBV*, *verbose=False*, *log=<Logger stdout_logger (INFO)>*)

Compute A(Lambda) for all possible emission lines

Parameters

- **EBV** (float) – E(B-V) value Has not been configured to handle a large array. Some array handling would be needed
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Return type dict

Returns A(lambda) with keys identical to `k_dict`

`Metallicity_Stack_Commons.analysis.attenuation.`**`compute_EBV`**(*ratio*, *source='HgHb'*, *zero_neg=True*, *verbose=False*, *log=<Logger stdout_logger (INFO)>*)

Determines E(B-V) from Hg/Hb or Hd/Hb flux ratios using Case B assumptions

Parameters

- **`ratio`** (`Union[float, ndarray]`) – Float or array containing Hg/Hb or Hd/Hb values
- **`source`** (`str`) – Indicate ratio type. Either 'HgHb' or 'HdHb'. Default: 'HgHb'
- **`zero_neg`** (`bool`) – Indicate whether to zero out negative reddening. Default: True
- **`verbose`** (`bool`) – Write verbose message to stdout. Default: file only
- **`log`** (`Logger`) – logging.Logger object

Return type `Union[float, ndarray, Tuple[ndarray, ndarray]]`

Returns E(B-V) values, E(B-V) peak values

Note: When only E(B-V) values is returned, correction does not account for negative reddening

`Metallicity_Stack_Commons.analysis.attenuation.`**`line_ratio_atten`**(*ratio*, *EBV*, *wave_top*, *wave_bottom*, *verbose=False*, *log=<Logger stdout_logger (INFO)>*)

Determine dust-corrected emission-line ratios

Parameters

- **`ratio`** (`Union[float, ndarray]`) – Float or array of observed flux ratios
- **`EBV`** (`Union[float, ndarray]`) – E(B-V) value(s)
- **`wave_top`** (`str`) – Emission-line name for flux ratio numerator
- **`wave_bottom`** (`str`) – Emission-line name for flux ratio denominator
- **`verbose`** (`bool`) – Write verbose message to stdout. Default: file only
- **`log`** (`Logger`) – logging.Logger object

Return type `Union[float, ndarray]`

Returns Float or array of dust-corrected flux ratios

Metallicity_Stack_Commons.analysis.composite_indv_detect module

`Metallicity_Stack_Commons.analysis.composite_indv_detect.`**`main`**(*fitspath*, *dataset*, *revised=False*, *det3=True*, *verbose=False*, *log=<Logger stdout_logger (INFO)>*)

Reads in composite table(s) containing bin information to determine temperature-based metallicity from composite average T_e and individual line ratios ([OII]/H-beta, [OIII]/H-beta)

Parameters

- **`fitspath`** (`str`) – Folder full path
- **`dataset`** (`str`) – Sub-folder path (specific to stacking approach)

- **revised** (bool) – Indicates whether to use revised bin properties (e.g., revised.tbl files). Default: False
- **det3** (bool) – Indicates whether individual galaxy files is limited to those satisfying emission-line det3 requirement Default: True
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Files identified by default:

composite_file: Filename of composite data e.g., '[dataset]/bin_derived_properties.tbl', '[dataset]/bin_derived_properties.revised.tbl'

indv_em_line_file: Filename that contains emission-line information for each galaxy e.g., 'individual_properties.tbl'

indv_bin_file: Filename that contains bin information for each galaxy e.g., '[dataset]/individual_bin_info.tbl'

outfile: Filename of output file e.g., '[dataset]/individual_derived_properties.tbl'

Metallicity_Stack_Commons.analysis.error_prop module

`Metallicity_Stack_Commons.analysis.error_prop.`**`fluxes_derived_prop`**(*path*, *raw=False*, *binned_data=True*, *apply_dust=False*, *revised=True*, *verbose=False*, *log=<Logger stdout_logger (INFO)>*)

Use measurements and their uncertainties to perform a randomization approach. The randomization is performed on individual emission lines. It carries that information to derived flux ratios and then determines electron temperature and metallicity

Parameters

- **path** (str) – Full path
- **raw** (bool) – Do a simple calculation, no randomization. Default: False
- **binned_data** (bool) – Whether to analyze binned data. Default: True
- **apply_dust** (bool) – Whether to apply dust attenuation. Default: False
- **revised** (bool) – Indicate if revised validation table is used. Default: True
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

`Metallicity_Stack_Commons.analysis.error_prop.`**`replace_oiii4363`**(*flux_file*, *detection*, *flux_dict*, *log=<Logger stdout_logger (INFO)>*)

`Metallicity_Stack_Commons.analysis.error_prop.`**`write_npz`**(*path*, *npz_files*, *dict_list*, *verbose=False*, *log=<Logger stdout_logger (INFO)>*)

Write numpy files with provided dictionaries

Parameters

- **path** (str) – Prefix for filename output

- **npz_files** (list) – NPZ file names
- **dict_list** (list) – List of dict for each corresponding npz file
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Metallicity_Stack_Commons.analysis.fitting module

`Metallicity_Stack_Commons.analysis.fitting.OIII4363_flux_limit(combine_flux_file, verbose=False, log=<Logger stdout_logger (INFO)>)`

Determine 3-sigma limit on [OIII]4363 based on H-gamma measurements

Parameters

- **combine_flux_file** (str) – Filename of ASCII file containing emission-line flux measurements
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Return type Optional[ndarray]

Returns Array containing 3-sigma flux limit

`Metallicity_Stack_Commons.analysis.fitting.double_gauss(x, xbar, s1, a1, c, s2, a2)`

Function providing double Gaussian profile (emission and absorption) for curve_fit

Parameters

- **x** (ndarray) – Wavelength array
- **xbar** (float) – Central wavelength of Gaussian fit
- **s1** (float) – Sigma (width) of first Gaussian fit (positive)
- **a1** (float) – Amplitude of first Gaussian fit
- **c** (float) – Continuum constant for Gaussian fit
- **s2** (float) – Sigma (width) of second Gaussian fit (absorption)
- **a2** (float) – Amplitude of second Gaussian fit

Return type array

Returns Double Gaussian fit

`Metallicity_Stack_Commons.analysis.fitting.gauss(x, xbar, s, a, c)`

Function providing Gaussian profile for curve_fit

Parameters

- **x** (ndarray) – Wavelength array
- **xbar** (float) – Central wavelength of Gaussian fit
- **s** (float) – Sigma (width) of Gaussian fit
- **a** (float) – Amplitude of Gaussian fit
- **c** (float) – Continuum constant for Gaussian fit

Return type ndarray

Returns Gaussian fit

`Metallicity_Stack_Commons.analysis.fitting.`**`movingaverage_box1D`**(*values*, *width*, *boundary='fill'*, *fill_value=0.0*)

Applies as boxcar kernel to smooth spectra

Parameters

- **values** (ndarray) – Array containing the spectrum
- **width** (float) – Width for smoothing
- **boundary** (str) – Handling of boundary values. Options are: 'None', 'fill', 'wrap', and 'extend' See astropy.convolution.convolve for more information
- **fill_value** (float) – Indicate fill value for default boundary='fill'

Return type ndarray

Returns Array contained the smoothed/convolved spectrum

`Metallicity_Stack_Commons.analysis.fitting.`**`oxy2_gauss`**(*x*, *xbar*, *s1*, *a1*, *c*, *s2*, *a2*)

Function providing [OII] doublet Gaussian profile for curve_fit

Parameters

- **x** (ndarray) – Wavelength array
- **xbar** (float) – Central wavelength of [OII]3726 Gaussian fit
- **s1** (float) – Sigma (width) of [OII]3726 Gaussian fit
- **a1** (float) – Amplitude of [OII]3726 Gaussian fit
- **c** (float) – Continuum constant for Gaussian fit
- **s2** (float) – Sigma (width) of [OII]3728 Gaussian fit
- **a2** (float) – Amplitude of [OII]3728 Gaussian fit

Return type ndarray

Returns [OII] doublet Gaussian fit

`Metallicity_Stack_Commons.analysis.fitting.`**`rms_func`**(*wave*, *dispersion*, *lambda_in*, *y0*, *sigma_array*, *mask_flag*, *verbose=False*, *log=<Logger stdout_logger (INFO)>*)

Compute rms in the spectra

Parameters

- **wave** (ndarray) – Array of rest wavelengths
- **dispersion** (float) – Spectral dispersion in AA/pix
- **lambda_in** (float) – Central wavelength of fit
- **y0** (ndarray) – Array of fluxes in units of erg/s/cm2/AA
- **sigma_array** (float) – Gaussian sigma (AA)
- **mask_flag** (ndarray) – Indicates spectra are masked for OH skyline contamination
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Returns

Metallicity_Stack_Commons.analysis.ratios module

`Metallicity_Stack_Commons.analysis.ratios.flux_ratios(flux_dict, binned_data=False, get_R=True, verbose=False, log=<Logger stdout_logger (INFO)>)`

Primary code to determine a variety of line ratios based on a dict containing emission-line fluxes

Parameters

- **flux_dict** (dict) – Contains emission-line fluxes
- **get_R** (bool) – Indicates populating OIII4363/OIII5007 flux ratio
- **binned_data** (bool) – Whether to analyze binned data. Default: False
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Return type dict

Returns Emission-line flux ratios

Metallicity_Stack_Commons.analysis.temp_metallicity_calc module

`Metallicity_Stack_Commons.analysis.temp_metallicity_calc.R_calculation(OIII4363, OIII5007, verbose=False, log=<Logger stdout_logger (INFO)>)`

Computes the excitation flux ratio of [OIII]4363 to [OIII]5007. Adopts a 3.1-to-1 ratio for 5007/4959

Parameters

- **OIII4363** (Union[float, ndarray]) – OIII4363 fluxes
- **OIII5007** (Union[float, ndarray]) – OIII5007 fluxes
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Return type Union[float, ndarray]

Returns O++ excitation flux ratio

`Metallicity_Stack_Commons.analysis.temp_metallicity_calc.metallicity_calculation(T_e, TWO_BETA, THREE_BETA, EBV=None, det3=None, verbose=False, log=<Logger stdout_logger (INFO)>)`

Determines 12+log(O/H) from electron temperature and [OII]/Hb and [OIII]/Hb flux ratio

Parameters

- **T_e** (ndarray) – Array of electron temperatures (see temp_calculation)

- **TWO_BETA** (ndarray) – Array of [OII]/Hb flux ratios
- **THREE_BETA** (ndarray) – Array of [OIII]/Hb flux ratios
- **EBV** (Optional[ndarray]) – Optional array containing EBV distribution
- **det3** (Optional[ndarray]) – Optional array to pass in to identify those satisfying det3 requirements. Default: None means full array is considered

  Note: for Monte Carlo inputs, a 1-D np.array index satisfying det3 requirements will suffice
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Return type dict

Returns Contains 12+log(O/H), O+/H, O++/H, log(O+/H), log(O++/H)

`Metallicity_Stack_Commons.analysis.temp_metallicity_calc.`**`temp_calculation`**(*R*, *EBV=None*, *verbose=False*, *log=<Logger stdout_logger (INFO)>*)

Computes electron temperature (T_e) from O++ excitation flux ratio

Formula is: T_e = a(-log(R)-b)^(-c) where a = 13025, b=0.92506, and c=0.98062 (Nicholls et al. 2014)

Parameters

- **R** (ndarray) – Array of O++ excitation flux ratio (see R_calculation)
- **EBV** (Optional[ndarray]) – Array of E(B-V). Set to zero if not applying attenuation
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Return type ndarray

Returns Array of T_e (Kelvins)

Module contents

others subpackage

Submodules

Metallicity_Stack_Commons.others.extract_deep2_data module

`Metallicity_Stack_Commons.others.extract_deep2_data.`**`main`**(*infile*, *log=<Logger stdout_logger (INFO)>*)

Import previous DEEP2 Ly et al. (2015) dataset and export an astropy Table called bin_fit.tbl

Parameters

- **infile** (str) – Input filename
- **log** (Logger) – logging.Logger object

Module contents

`plotting` subpackage

Submodules

Metallicity_Stack_Commons.plotting.balmer module

balmer

Generates plots illustrating Balmer recombination lines

This code was created from: https://github.com/astrochun/Zcalbase_gal/blob/master/Analysis/DEEP2_R23_O32/balmer_plots.py

`Metallicity_Stack_Commons.plotting.balmer.HbHgHd_fits(fitspath, out_pdf_prefix='HbHgHd_fits', use_revised=False, verbose=False, log=<Logger stdout_logger (INFO)>)`

Generate PDF plots that illustrate H-delta, H-gamma, and H-beta line profiles and best fit

Parameters

- **fitspath** (str) – Full path
- **out_pdf_prefix** (str) – Prefix for output PDF file
- **use_revised** (bool) – Indicate whether to use regular or revised tables
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

`Metallicity_Stack_Commons.plotting.balmer.extract_fit(astropy_table, line_name, balmer=False, verbose=False, log=<Logger stdout_logger (INFO)>)`

Extract best fit from table and fluxes, return a list of fitting parameters and fluxes

Parameters

- **astropy_table** (Table) – Astropy table containing fitting result
- **line_name** (str) – Name of Line to extract fit results
- **balmer** (bool) – Indicate whether line is a Balmer line
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Return type Optional[dict]

Returns Fitting results

`Metallicity_Stack_Commons.plotting.balmer.fitting_result(wave, y_norm, lambda_cen, line_fit, line_fit_neg, flux_gauss, flux_spec, use_revised=False)`

Returns fitting results based on inputs of best fit

Parameters

- **wave** (ndarray) – Array of rest-frame wavelengths
- **y_norm** (ndarray) – Normalize 1-D spectra in units of 10^-17 erg/s/cm2/AA

- **lambda_cen** (float) – Central wavelength in Angstroms
- **line_fit** (list) – List containing Balmer emission fits
- **line_fit_neg** (list) – List containing the absorption ("stellar") Balmer fit
- **flux_gauss** (float) – Flux from Gaussian model
- **flux_spec** (float) – Flux from spectrum (above median)
- **use_revised** (bool) – Indicate whether fluxes have been revised. Default: False

Return type dict

Returns Dictionary of fitting results

Module contents

2.1.2 Submodules

Metallicity_Stack_Commons.column_names module

`Metallicity_Stack_Commons.column_names.`**`indv_M_LHb`**`()`

Use remove_from_list() to provide simplified list that contains ID, logM and logLHb

Return type list

Returns List containing just ID, logM, logLHb

`Metallicity_Stack_Commons.column_names.`**`indv_R23_O32`**`()`

Use remove_from_list() to provide simplified list that contains ID, logR23 and logO32

Return type list

Returns List containing just ID, logR23, logO32

`Metallicity_Stack_Commons.column_names.`**`line_fit_suffix_add`**`(`*line_name0*, *line_type0*`)`

Simple list comprehension combining emission line fit suffixes with the emission line. This works for individual lines

Parameters

- **line_name0** (str) – Line name
- **line_type0** (str) – Emission-line type (e.g., 'Balmer')

Return type list

Returns List of strings formatted as [LINE]_[SUFFIX]

`Metallicity_Stack_Commons.column_names.`**`merge_column_names`**`(`**args*`)`

Merges multiple lists containing column names.

Usage: column_names = merge_column_names(bin_names0, indv_names0)

Parameters **args** (list) – An undefined number of lists

Return type list

Returns Merged list

`Metallicity_Stack_Commons.column_names.`**`remove_from_list`**`(`*list0*, *remove_entries*`)`

Purpose: Remove entries from list of column names

Parameters

- **list0** (list) – List of column names
- **remove_entries** (list) – List of column names to remove

Return type list

Returns List of column names after removal

Metallicity_Stack_Commons.logging module

class Metallicity_Stack_Commons.logging.**LogClass**(*log_dir*, *logfile*)

Bases: object

Main class to log information to stdout and ASCII logfile

Note: This code is identical to the one used in ReQUIAM: https://github.com/ualibraries/ReQUIAM

To use: log = LogClass(log_dir, logfile).get_logger()

Parameters

- **log_dir** (str) – Relative path for exported logfile directory
- **logfile** (str) – Filename for exported log file

get_logger()

Return type Logger

Metallicity_Stack_Commons.logging.**get_user_hostname**()

Retrieve user, hostname, IP, and OS configuration

Return type dict

Returns Dictionary with 'user' 'hostname' and 'ip' keys

Metallicity_Stack_Commons.logging.**log_stdout**()

Returns stdout logging object

Return type Logger

Metallicity_Stack_Commons.logging.**log_verbose**(*log*, *message*, *verbose=False*)

Log message depending on verbosity

Parameters

- **log** (Logger) – logging.Logger object
- **message** (str) – Message
- **verbose** (bool) – Write verbose message to stdout. Default: file only

Metallicity_Stack_Commons.update_det4363_info module

`Metallicity_Stack_Commons.update_det4363_info.get_index(det4363_table, input_table, column_name, verbose=False, log=<Logger stdout_logger (INFO)>)`

Uses either OBJNO or AP/SLIT info to get index for an existing table

Parameters

- **det4363_table** (Table) – Astropy table containing DEEP2 [OIII]4363-detected sample
- **input_table** (Table) – Astropy table containing the entire sample to be updated
- **column_name** (str) – Column name for cross-matching
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Return type Tuple[ndarray, ndarray]

Returns Index arrays for det4363_table, input_table

Metallicity_Stack_Commons.valid_table module

`Metallicity_Stack_Commons.valid_table.compare_to_by_eye(fitspath, dataset)`

This function takes the automated validation table and checks it against inputted measurement that are determined by eye. These inputted measurements are in the np.where statements. It outputs a revised validation table based on the inputted measurements.

Usage: valid_table.make_validation_table(fitspath, dataset)

Parameters

- **fitspath** (str) – Full file path where the input file is and where the output file will be placed.
- **dataset** (str) – Determine which eye measurements to use

Outputs:

fitspath + 'bin_validation_revised.tbl' and '.csv' Validation table containing bin IDs; number of galaxies in each bin; and column indicating OIII4363 detection/non-detection, OIII4363_Flux_Observed, OIII4363_S/N, Notes

`Metallicity_Stack_Commons.valid_table.make_validation_table(fitspath, vmin_4363SN=3, vmin_5007SN=100, vmax_4363sig=1.6, rlmin_4363SN=3, rlmax_4363sig=1.6, rlmin_5007SN=100)`

This function creates a validation table for a given binning set. The validation table contains a OIII4363 detection column where 1.0 means detection, 0.5 means non-detection with reliable OIII5007, and 0.0 means unreliable non-detection. This function will be run every time the analysis is completed and will create a validation table for every analysis.

Usage: valid_table.make_validation_table(fitspath, bin_type_str)

Parameters

- **fitspath** (str) – Full file path where the input file is and where the output file will be placed.
- **vmin_4363SN** – int. minimum OIII4363 S/N for valid detection
- **vmin_5007SN** – int. minimum OIII5007 S/N for valid detection
- **vmax_4363sig** – int. maximum OIII4363 sigma for valid detection
- **rlmin_4363SN** – int. minimum OIII4363 S/N for robust limit
- **rlmax_4363sig** – int. maximum OIII4363 sigma for robust limit
- **rlmin_5007SN** – int. minimum OIII5007 S/N for robust limit

Outputs:

fitspath + 'bin_validation.tbl' Validation table containing bin IDs; number of galaxies in each bin; and column indicating OIII4363 detection/non-detection, OIII4363_Flux_Observed, OIII4363_S/N

2.1.3 Common functions

`Metallicity_Stack_Commons.dir_date(folder_name, path_init='', year=False, verbose=False, log=<Logger stdout_logger (INFO)>)`

This function finds and returns the path to a directory named after the current date (MMDDYYYY). If the directory doesn't exist yet, it creates a new directory named after the current date in the provided `folder_name` directory.

Originally from https://github.com/rafia37/Evolution-of-Galaxies/blob/master/general.py

Usage: fitspath = dir_date(folder_name, year=True)

Parameters

- **folder_name** (str) – Directory for date subdirectory will be in
- **path_init** (str) – root path. Default: empty string
- **year** (bool) – Indicate whether to include year in date folder. Default: False
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Return type str

Returns Full path to the date directory

`Metallicity_Stack_Commons.exclude_outliers(objno, verbose=False, log=<Logger stdout_logger (INFO)>)`

Exclude spectra that are identified as outliers.

Generally this is because the spectra have very high S/N on the continuum.

Parameters

- **objno** (Union[list, ndarray]) – Array of eight-digit identifier
- **verbose** (bool) – Write verbose message to stdout. Default: file only
- **log** (Logger) – logging.Logger object

Return type ndarray

Returns Array of zeros (not flagged) and ones (flagged

`Metallicity_Stack_Commons.get_user(username=None, verbose=False, log=<Logger stdout_logger (INFO)>)`

Get the corresponding path for a given `username`

Parameters

- **username** (`Optional[str]`) – Optional input for username
- **verbose** (`bool`) – Write verbose message to stdout. Default: file only
- **log** (`Logger`) – logging.Logger object

Return type `str`

Returns Full path to the date directory

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

m

Metallicity_Stack_Commons, 14
Metallicity_Stack_Commons.analysis, 9
Metallicity_Stack_Commons.analysis.attenuation, 3
Metallicity_Stack_Commons.analysis.composite_indv_detect, 4
Metallicity_Stack_Commons.analysis.error_prop, 5
Metallicity_Stack_Commons.analysis.fitting, 6
Metallicity_Stack_Commons.analysis.ratios, 8
Metallicity_Stack_Commons.analysis.temp_metallicity_calc, 8
Metallicity_Stack_Commons.column_names, 11
Metallicity_Stack_Commons.logging, 12
Metallicity_Stack_Commons.others, 10
Metallicity_Stack_Commons.others.extract_deep2_data, 9
Metallicity_Stack_Commons.plotting, 11
Metallicity_Stack_Commons.plotting.balmer, 10
Metallicity_Stack_Commons.update_det4363_info, 13
Metallicity_Stack_Commons.valid_table, 13

INDEX

C

compare_to_by_eye() (*in module Metallicity_Stack_Commons.valid_table*), 13
compute_A() (*in module Metallicity_Stack_Commons.analysis.attenuation*), 3
compute_EBV() (*in module Metallicity_Stack_Commons.analysis.attenuation*), 4

D

dir_date() (*in module Metallicity_Stack_Commons*), 14
double_gauss() (*in module Metallicity_Stack_Commons.analysis.fitting*), 6

E

exclude_outliers() (*in module Metallicity_Stack_Commons*), 14
extract_fit() (*in module Metallicity_Stack_Commons.plotting.balmer*), 10

F

fitting_result() (*in module Metallicity_Stack_Commons.plotting.balmer*), 10
flux_ratios() (*in module Metallicity_Stack_Commons.analysis.ratios*), 8
fluxes_derived_prop() (*in module Metallicity_Stack_Commons.analysis.error_prop*), 5

G

gauss() (*in module Metallicity_Stack_Commons.analysis.fitting*), 6
get_index() (*in module Metallicity_Stack_Commons.update_det4363_info*), 13
get_logger() (*Metallicity_Stack_Commons.logging.LogClass method*), 12
get_user() (*in module Metallicity_Stack_Commons*), 15
get_user_hostname() (*in module Metallicity_Stack_Commons.logging*), 12

H

Hb_SFR() (*in module Metallicity_Stack_Commons.analysis.attenuation*), 3
HbHgHd_fits() (*in module Metallicity_Stack_Commons.plotting.balmer*), 10

I

indv_M_LHb() (*in module Metallicity_Stack_Commons.column_names*), 11
indv_R23_O32() (*in module Metallicity_Stack_Commons.column_names*), 11

L

line_fit_suffix_add() (*in module Metallicity_Stack_Commons.column_names*), 11
line_ratio_atten() (*in module Metallicity_Stack_Commons.analysis.attenuation*), 4
log_stdout() (*in module Metallicity_Stack_Commons.logging*), 12
log_verbose() (*in module Metallicity_Stack_Commons.logging*), 12
LogClass (*class in Metallicity_Stack_Commons.logging*), 12

M

main() (*in module Metallicity_Stack_Commons.analysis.composite_indv_detect*), 4
main() (*in module Metallicity_Stack_Commons.others.extract_deep2_data*), 9
make_validation_table() (*in module Metallicity_Stack_Commons.valid_table*), 13
merge_column_names() (*in module Metallicity_Stack_Commons.column_names*), 11

metallicity_calculation() (*in module Metallicity_Stack_Commons.analysis.temp_metallicity_calc*), 8
Metallicity_Stack_Commons
module, 14
Metallicity_Stack_Commons.analysis
module, 9
Metallicity_Stack_Commons.analysis.attenuation
module, 3
Metallicity_Stack_Commons.analysis.composite_indv_detect
module, 4
Metallicity_Stack_Commons.analysis.error_prop
module, 5
Metallicity_Stack_Commons.analysis.fitting
module, 6
Metallicity_Stack_Commons.analysis.ratios
module, 8
Metallicity_Stack_Commons.analysis.temp_metallicity_calc
module, 8
Metallicity_Stack_Commons.column_names
module, 11
Metallicity_Stack_Commons.logging
module, 12
Metallicity_Stack_Commons.others
module, 10
Metallicity_Stack_Commons.others.extract_deep2_data
module, 9
Metallicity_Stack_Commons.plotting
module, 11
Metallicity_Stack_Commons.plotting.balmer
module, 10
Metallicity_Stack_Commons.update_det4363_info
module, 13
Metallicity_Stack_Commons.valid_table
module, 13
module
Metallicity_Stack_Commons, 14
Metallicity_Stack_Commons.analysis, 9
Metallicity_Stack_Commons.analysis.attenuation, 3
Metallicity_Stack_Commons.analysis.composite_indv_detect, 4
Metallicity_Stack_Commons.analysis.error_prop, 5
Metallicity_Stack_Commons.analysis.fitting, 6
Metallicity_Stack_Commons.analysis.ratios, 8
Metallicity_Stack_Commons.analysis.temp_metallicity_calc, 8
Metallicity_Stack_Commons.column_names, 11
Metallicity_Stack_Commons.logging, 12
Metallicity_Stack_Commons.others, 10
Metallicity_Stack_Commons.others.extract_deep2_data, 9
Metallicity_Stack_Commons.plotting, 11
Metallicity_Stack_Commons.plotting.balmer, 10
Metallicity_Stack_Commons.update_det4363_info, 13
Metallicity_Stack_Commons.valid_table, 13
movingaverage_box1D() (*in module Metallicity_Stack_Commons.analysis.fitting*), 7

O

OIII4363_flux_limit() (*in module Metallicity_Stack_Commons.analysis.fitting*), 6
oxy2_gauss() (*in module Metallicity_Stack_Commons.analysis.fitting*), 7

R

R_calculation() (*in module Metallicity_Stack_Commons.analysis.temp_metallicity_calc*), 8
remove_from_list() (*in module Metallicity_Stack_Commons.column_names*), 11
replace_oiii4363() (*in module Metallicity_Stack_Commons.analysis.error_prop*), 5
rms_func() (*in module Metallicity_Stack_Commons.analysis.fitting*), 7

T

temp_calculation() (*in module Metallicity_Stack_Commons.analysis.temp_metallicity_calc*), 9

W

write_npz() (*in module Metallicity_Stack_Commons.analysis.error_prop*), 5